Language Without Borders: A Step-by-Step Guide to Analyzing Webcam Eye-Tracking Data for L2 Research

Jason Geller[1], Yanina Prystauka[2], Sarah E. Colby[3], and Julia R. Drouin[4]
[1]Department of Psychology and Neuroscience, Boston College
[2]Department of Linguistic, Literary and Aesthetic Studies, University of Bergen
[3]Department of Linguistics, University of Ottawa
[4]Division of Speech and Hearing Sciences, University of North Carolina at Chapel Hill

Abstract

Eye-tracking has become a valuable tool for studying cognitive processes in second language acquisition and bilingualism (Godfroid et al., 2024). While research-grade infrared eye-trackers are commonly used, several factors limit their widespread adoption. Recently, consumer-based webcam eye-tracking has emerged as an attractive alternative, requiring only a personal webcam and internet access. However, webcam-based eye-tracking introduces unique design and preprocessing challenges that must be addressed to ensure valid results. To help researchers navigate these challenges, we developed a comprehensive tutorial focused on visual world webcam eye-tracking for second language research. This guide covers key preprocessing steps—from reading in raw data to visualization and analysis—highlighting the open-source R package webgazeR, freely available at: https://github.com/jgeller112/webgazer. To demonstrate these steps, we analyze data collected via the Gorilla platform (Anwyl-Irvine et al., 2020) using a single-word Spanish visual world paradigm (VWP), showcasing evidence of competition both within and between Spanish and English. This tutorial aims to empower researchers by providing a step-by-step guide to successfully conduct webcam-based visual world eye-tracking studies. To follow along, please download the complete manuscript, code, and data from: https://github.com/jgeller112/L2_VWP_Webcam.

Keywords: VWP, Tutorial, Webcam eye-tracking, R, Gorilla, Spoken word recognition, L2 processing

Eye-tracking technology, which has a history spanning over a century, has seen remarkable advancements. In the early days, eye-tracking often required the use of contact lenses fitted with search coils—sometimes necessitating anesthesia—or the attachment of suction cups to the sclera of the eyes (Płużyczka, 2018). These methods were not only cumbersome for researchers, but also uncomfortable and invasive for participants. Over time, such approaches have been replaced by non-invasive, lightweight, and user-friendly systems. Today, modern eye-tracking technology is widely accessible in laboratories worldwide, enabling researchers to tackle critical questions about cognitive processes. This evolution has had a profound impact on fields such as psycholinguistics and bilingualism, opening up new possibilities for understanding how language is processed in real time (Godfroid et al., 2024).

In the last decade, there has been a gradual shift towards conducting more behavioral experiments online (Anderson et al., 2019; Rodd, 2024). This "onlineification" of behavioral research has driven the development of remote eye-tracking methods that do not rely on traditional laboratory settings. Allowing participants to use their own equipment from anywhere in the world opens the door to recruiting more diverse and historically underrepresented populations [Gosling et al. (2010)]. Behavioral research has long struggled with a lack of diverse and representative samples, relying heavily on participants who are predominantly Western, Educated, Industrialized, Rich, and Democratic (WEIRD) (Henrich et al., 2010). Additionally, we propose adding able-bodied to this acronym (WEIRD-A) (Peterson, 2021), to highlight the exclusion of individuals with disabilities who may face barriers to accessing research facilities. In language research, this issue is especially pronounced, as studies often focus on "modal" listeners and speakers—typically young, monolingual, and neurotypical (Blasi et al., 2022; Bylund et al., 2024; McMurray et al., 2010).

In this paper, we contribute to the growing body of research suggesting that webcam-based eye-tracking, which is administered remotely and requires access to only a computer webcam, can increase inclusivity and representation of the participant samples we include in research studies. Namely, by minimizing the requirements for participants to travel to a lab, use specialized equipment, or meet strict scheduling demands, webcam-based approaches can facilitate participation from individuals in rural or geographically isolated areas and people with disabilities that make getting to a lab difficult. This approach also promotes inclusion of broader sociodemographic groups that have been historically underrepresented in cognitive and developmental research. We illustrate this by replicating a visual world eye-tracking study with bilingual English-Spanish speaking participants (Sarrett et al., 2022) using online methods (i.e., recruitment via Prolific.co and webcam-based eye-tracking). To facilitate broader adoption of this approach, we also introduce our R package, webgazeR, and present a step-by-step tutorial for analyzing webcam-based VWP data.

This paper is divided into three parts. First, we introduce automated webcam-based eye-tracking.

Jason Geller https://orcid.org/0000-0002-7459-4505
Yanina Prystauka https://orcid.org/0000-0001-8258-2339
Sarah E. Colby https://orcid.org/0000-0002-2956-3072
Julia R. Drouin https://orcid.org/0000-0003-0798-3268

This study was not preregistered. The data and code for this manuscript can be found at https://github.com/jgeller112/L2_VWP_Webcam. The authors have no conflicts of interest to disclose. This work was supported by research start-up funds to JRD. Author roles were classified using the Contributor Role Taxonomy (CRediT; https://credit.niso.org/) as follows: Jason Geller: conceptualization, writing, data curation, editing, software, formal analysis; Yanina Prystauka: methodology, editing, formal analysis; Sarah E. Colby: methodology, editing; Julia R. Drouin: methodology, conceptualization, editing, funding acquisition

Correspondence concerning this article should be addressed to Jason Geller, Department of Psychology and Neuroscience, Boston College, Mcguinn Hall 405, Chestnut Hill, MA 02467-9991, USA, drjasongeller@gmail.com: jason.geller@bc.edu

Second, we review the viability of conducting VWP studies using online eye-tracking methods. Third, we present a detailed tutorial for analyzing webcam-based VWP data with the webgazeR package, using our replication experiment to highlight the steps needed for preprocessing.

Webcam eye-tracking with WebGazer.js

There are two popular methods for online eye-tracking. One method, manual eye-tracking (Trueswell, 2008), involves using video recordings of participants, which can be collected through online teleconferencing platforms such as Zoom (www.zoom.com). Here eye gaze (direction) is manually analyzed post-hoc frame by frame from these recordings. However, this method raises ethical and privacy concerns, as not all participants may be comfortable having their videos recorded and stored for analysis.

Another method, which is the focus of this paper, is automated eye-tracking or webcam eye-tracking. Webcam eye-tracking generally has three requirements for the participant: (1) a personal computer, tablet, or smartphone (see Chen-Sankey et al., 2023), (2) an internet connection, and (3) a built-in or external camera. Gaze data is collected directly through a web browser without requiring any additional software installation, making it highly accessible.

A popular tool for enabling webcam-based eye-tracking is WebGazer.js (Papoutsaki et al., 2016) [1], an open-source, freely available, and actively maintained JavaScript library. WebGazer.js has already been integrated into several popular experimental platforms, including Gorilla, jsPsych, PsychoPy, Labvanced, and PCIbex (Anwyl-Irvine et al., 2020; Kaduk et al., 2024; Leeuw, 2015; Peirce et al., 2019; Zehr & Schwarz, 2018). Because WebGazer.js runs locally on the participant's machine, it does not store webcam video recordings, helping alleviate ethical and privacy concerns associated with online eye-tracking.

Under the hood, WebGazer.js uses machine learning to estimate gaze position in real time by fitting a facial mesh to the participant and detecting the location of the eyes. At each sampling point—determined by the participant's device and webcam capabilities—x and y gaze coordinates are recorded. To improve accuracy, participants complete calibration and validation routines in which they fixate on targets in specific locations on the screen (in some cases a manual approach is used where users click on targets).

Eye-tracking in the lab vs. online

Several studies in psychology and psycholinguistics have evaluated the viability of WebGazer.js for online research. Generally, lab-based effects can be successfully replicated in online environments using WebGazer.js (Bogdan et al., 2024; Bramlett & Wiener, 2024, 2025; Özsoy et al., 2023; Prystauka et al., 2024; Slim et al., 2024; Slim & Hartsuiker, 2023; Van der Cruyssen et al., 2024; Vos et al., 2022). However, a critical finding across online replication studies is that effect sizes are often smaller and more variable than those observed in laboratory settings (Bogdan et al., 2024; Slim et al., 2024; Slim & Hartsuiker, 2023; Van der Cruyssen et al., 2024).

[1] **It is important to note that WebGazer.js is not the only method available. Other methods have been implemented by companies like Tobii (www.tobii.com) and Labvanced (Kaduk et al., 2024) . However, because these methods are proprietary, they are less accessible and difficult to reproduce.**

These attenuated effects likely stem from several technical limitations inherent to webcam-based eye-tracking. Unlike research-grade trackers that use infrared illumination and pupil–corneal reflection techniques—and can sample at rates up to 2,000 Hz with sub-degree spatial precision (0.1° to 0.35°) (Carter & Luke, 2020; Hooge et al., 2024)—WebGazer.js typically operates at lower frame rates, around 30 Hz (Bramlett & Wiener, 2024; Prystauka et al., 2024). Moreover, the performance of the algorithm is highly dependent on ambient lighting conditions, making it more susceptible to variability introduced by differences in head position, screen brightness, and background contrast.

There are also notable issues with the spatial and temporal accuracy of webcam-based eye-tracking using WebGazer.js. Spatial precision is often lower, with average errors frequently exceeding 1° of visual angle (Papoutsaki et al., 2016). Temporal delays are also substantially larger, ranging from 200 ms to over 1000 ms (Semmelmann & Weigelt, 2018; Slim et al., 2024; Slim & Hartsuiker, 2023). Additionally, recent work by Bogdan et al. (2024) has documented a systematic bias in gaze estimates favoring centrally located stimuli.

Bringing the visual world paradigm (VWP) online

Despite these technical challenges, webcam-based eye-tracking has proven particularly well-suited for adapting the visual world paradigm (VWP) (Tanenhaus et al., 1995; cf. Cooper, 1974) to online environments.

In the field of language research, few methods have had as enduring an impact as the VWP. Over the past 25 years, the VWP has enabled researchers to address a broad range of topics, including sentence processing (Altmann & Kamide, 1999; Huettig et al., 2011; Kamide et al., 2003), spoken word recognition (Allopenna et al., 1998; Dahan et al., 2001; Huettig & McQueen, 2007; McMurray et al., 2002), bilingual language processing (Hopp, 2013; Ito et al., 2018; Rossi et al., 2019), the effects of brain damage on language (Mirman & Graziano, 2012; Yee et al., 2008), and the impact of hearing loss on lexical access (McMurray et al., 2017).

What makes the widespread use of the VWP particularly remarkable is the simplicity of the task. In a typical VWP experiment, participants view a display of several objects, each represented by a picture, while their eye movements are recorded in real time as they listen to a spoken word or phrase. Although variations of the task exist—and implementations may differ depending on specific research goals or design choices—the core finding remains consistent: listeners reliably shift their gaze to the image corresponding to the spoken word, often before the word is fully articulated. This robust effect provides compelling evidence for anticipatory or predictive processing during language comprehension.

While eye movements are often time-locked to linguistic input, the relationship between eye movements and lexical processing is not one-to-one. Lexical activation interacts with non-lexical factors such as selective attention, visual salience, task demands, working memory, and prior expectations—all of which can shape where and when participants look (Bramlett & Wiener, 2025; Eberhard et al., 1995; Huettig et al., 2011; Kamide et al., 2003). Nonetheless, the VWP remains a powerful and flexible tool for studying online language processing, offering fine-grained insights into how linguistic and cognitive processes unfold moment by moment.

Several attempts have been made to conduct these experiments online using webcam-based eye-tracking. Most online VWP replications have focused on sentence-based language processing. These studies have looked at effects of set size and determiners (Degen et al., 2021), verb semantic constraint (Prystauka et al., 2024; Slim & Hartsuiker, 2023), grammatical aspect and event comprehension (Vos et al., 2022), and lexical interference (Prystauka et al., 2024).

More relevant to the current tutorial are findings from single-word VWP studies conducted online. Recent research examined single-word speech perception online using a phonemic cohort task (Bramlett & Wiener, 2025; Slim et al., 2024). In the cohort task, pictures were displayed randomly in one of four quadrants, and participants were instructed to fixate on the target based on the auditory cue. On each trial, one of the pictures was phonemically similar to the target in onset (e.g., *MILK – MITTEN*). Slim et al. (2024) were able to observe significant fixations to the cohort compared to the control condition, replicating lab-based single word VWP experiments with research grade eye-trackers (e.g., Allopenna et al., 1998). However, time course differences were observed in the webcam-based setting such that competition effects occurred later in processing compared to traditional, lab-based eye-tracking.

Several factors have been proposed to explain the poor temporal performance in the VWP. These include reduced spatial precision, computational demands introduced by the WebGazer.js algorithm, slower internet connections, smaller areas of interest (AOIs), and calibration quality (Boxtel et al., 2024; Degen et al., 2021; Slim et al., 2024).

Importantly, temporal issues are not observed in every case. Work has begun to address many of these challenges by leveraging updated versions of WebGazer.js and adopting different experimental platforms. For instance, Vos et al. (2022) reported a substantial reduction in temporal delays—approximately 50 ms—when using a newer version of WebGazer.js embedded within the jsPsych framework (Leeuw, 2015). Similarly, studies by Prystauka et al. (2024) and Bramlett and Wiener (2024), which utilized the Gorilla Experiment Builder in combination with the improved WebGazer algorithm, found timing and competition effects closely aligned with those observed in traditional lab-based VWP studies.

While these temporal delays do present a challenge, and are at present an open issue, the general findings that WebGazer.js can approximate looks to areas on the screen and replicate lab-based findings underscore the potential of adapting the VWP to online environments using webcam-based eye-tracking. Importantly, recent studies demonstrate that this approach can successfully capture key psycholinguistic effects—such as lexical competition during single-word speech recognition—in a manner comparable to traditional lab-based methods (Slim et al., 2024).

Bilingual competition: A visual world webcam eye-tracking replication

A goal of the present study was to conceptually replicate a study by Sarrett et al. (2022) wherein they examined the competitive dynamics of second-language (L2) learners of Spanish, whose first language (L1) is English, during spoken word recognition. Specifically, we investigated both within-language and cross-language (L2/L1) competition using webcam-based eye-tracking.

It is well established that lexical competition plays a central role in language processing (Magnuson et al., 2007). During spoken word recognition, as the auditory signal unfolds over time, multiple lexical candidates—or competitors—can become partially activated. Successful recognition depends on resolving this competition by inhibiting or suppressing mismatching candidates. For example, upon hearing the initial segments of the word *wizard*, phonologically similar words such as *whistle* (cohort competitor) may be briefly activated. As the word continues to unfold, additional competitors like *blizzard* (a rhyme competitor) might also become active. For *wizard* to be accurately recognized, activation of competitors such as *whistle* and *blizzard* must ultimately be suppressed.

One important area of exploration concerns lexical competition across languages. There is growing evidence that lexical competition can occur cross-linguistically (see Ju & Luce, 2004; Spivey & Marian, 1999). In a recent study, Sarrett et al. (2022) investigated whether cross-linguistic competition arises in unbalanced L2 Spanish speakers—that is, individuals who acquired Spanish later in life. They used carefully controlled stimuli to examine both within-language and cross-language competition in adult L2 Spanish learners. Using a Spanish-language visual world paradigm, their study included two critical conditions:

1. Spanish-Spanish (within) condition: A Spanish competitor was presented alongside the target word. For example, if the target word spoken was *cielo* (sky), the Spanish competitor was *ciencia* (science).

2. Spanish-English (cross-ligustic) condition: An English competitor was presented for the Spanish target word. For example, if the target word spoken was *botas* (boots), the English competitor was *border*.

Sarrett et al. (2022) also included a no competition condition where the Spanish-English pairs were not cross-linguistic competitors (e.g., *frontera* as the target word and *botas* - *boots* as an unrelated item in the pair). They observed competition effects in both of the critical conditions: within (e.g., *cielo* - *ciencia*) and between (e.g., *botas* - *border*). Herein, we collected data to conceptually replicate their pattern of findings using a webcam approach.

There are two key differences between our dataset and the original study by Sarrett et al. (2022) worth noting. First, Sarrett et al. (2022) focused on adult unbalanced L2 Spanish speakers and posed more fine-grained questions about the time course of competition and resolution and its relationship with L2 language acquisition. Second, unlike Sarrett et al. (2022) , who measured Spanish proficiency objectively using LexTALE-esp (Izura et al., 2014)) and ran this study using participants from a Spanish college course, we relied on participant filtering on Prolific (www.prolific.co) to recruit L2 Spanish speakers.

To conduct our online webcam replication, we used the experimental platform Gorilla (Anwyl-Irvine et al., 2020), which integrates WebGazer.js for gaze tracking. We selected Gorilla because it offers robust WebGazer.js integration and seems to address several temporal accuracy concerns identified in other platforms (Slim et al., 2024; Slim & Hartsuiker, 2023).

Tutorial Overview

This paper has two aims. First, we aim to provide evidence for lexical competition within and across languages in L2 Spanish speakers, using webcam-based eye-tracking with WebGazer.js. While there is growing interest in using VWP using webcam-based methods, lexical competition in single-word L2 processing has not yet been investigated using the online version of the VWP, making this a novel application. We hope that this work encourages researchers to explore more detailed questions about L2 processing using webcam-based eye-tracking.

Second, we offer a tutorial that outlines key preprocessing steps for analyzing webcam-based eye-tracking data. Building on recommendations proposed by (Bramlett & Wiener, 2024), our contribution focuses on data preprocessing— transforming raw gaze data into a format suitable for visualization and analysis. Here we introduce a new R package—`webgazeR`(Geller & Prystauka, 2024)—designed to streamline and standardize preprocessing for webcam-based eye-tracking studies. We believe that offering multiple, complementary resources enhances methodological transparency and supports broader adoption of webcam-based eye-tracking methods. For in-depth guidance on experimental design considerations, we refer readers to Bramlett and Wiener (2024).

Although Bramlett and Wiener (2024)'s tutorial provides a lot of useful code, the experiment-specific nature of the code may pose challenges for newcomers. In contrast, the `webgazeR` package offers a modular, generalizable approach. It includes functions for importing raw data, filtering and visualizing sampling rates, extracting and assigning areas of interest (AOIs), downsampling and upsampling gaze data, interpolating and smoothing time series, and performing non-AOI-based analyses such as intersubject correlation (ISC), a method increasingly used to explore gaze synchrony in naturalistic paradigms (i.e., online learning) with webcam-based eye-tracking (Madsen et al., 2021).

We first begin by outlining the general methods used to conduct our webcam-based visual world experiment. Second, we detail the data preprocessing steps needed to prepare the data for analysis using `webgazeR`. Third, we demonstrate a statistical approach for analyzing the preprocessed data, highlighting its application and implications.

To promote transparency and reproducibility, all analyses were conducted in R (R Core Team, 2024) using Quarto (Allaire et al., 2024), an open-source publishing system that enables dynamic and reproducible documents. Figures, tables, and text are generated programmatically and embedded directly in the manuscript, ensuring seamless integration of results. To further enhance computational reproducibility, we employed the `rix` package (Rodrigues & Baumann, 2025), which leverages the Nix ecosystem (Dolstra & contributors, 2023). This approach captures not only the R package versions but also system dependencies at runtime. Researchers can reproduce the exact computational environment by installing the Nix package manager and using the provided `default.nix` file. Detailed setup instructions are included in the README file of the accompanying GitHub repository. A video tutorial is also provided.

Method

All tasks herein can be previewed here (https://app.gorilla.sc/openmaterials/953693). The manuscript, data, and R code can be found on Github (https://github.com/jgeller112/webcam_gazeR_VWP).

Participants

Participants were recruited through Prolific (www.prolific.co, 2024), an online participant recruitment platform. Our goal was to approximately double the sample size of Sarrett et al. (2022) to enhance statistical power and ensure greater generalizability of the findings. However, due to practical constraints and the challenges associated with online webcam eye-tracking (e.g., calibration failures) and also the limited pool of bilingual Spanish speakers, we were unable to achieve the targeted usable sample size. Therefore, we report the final sample based on all participants who met our predefined inclusion criteria.

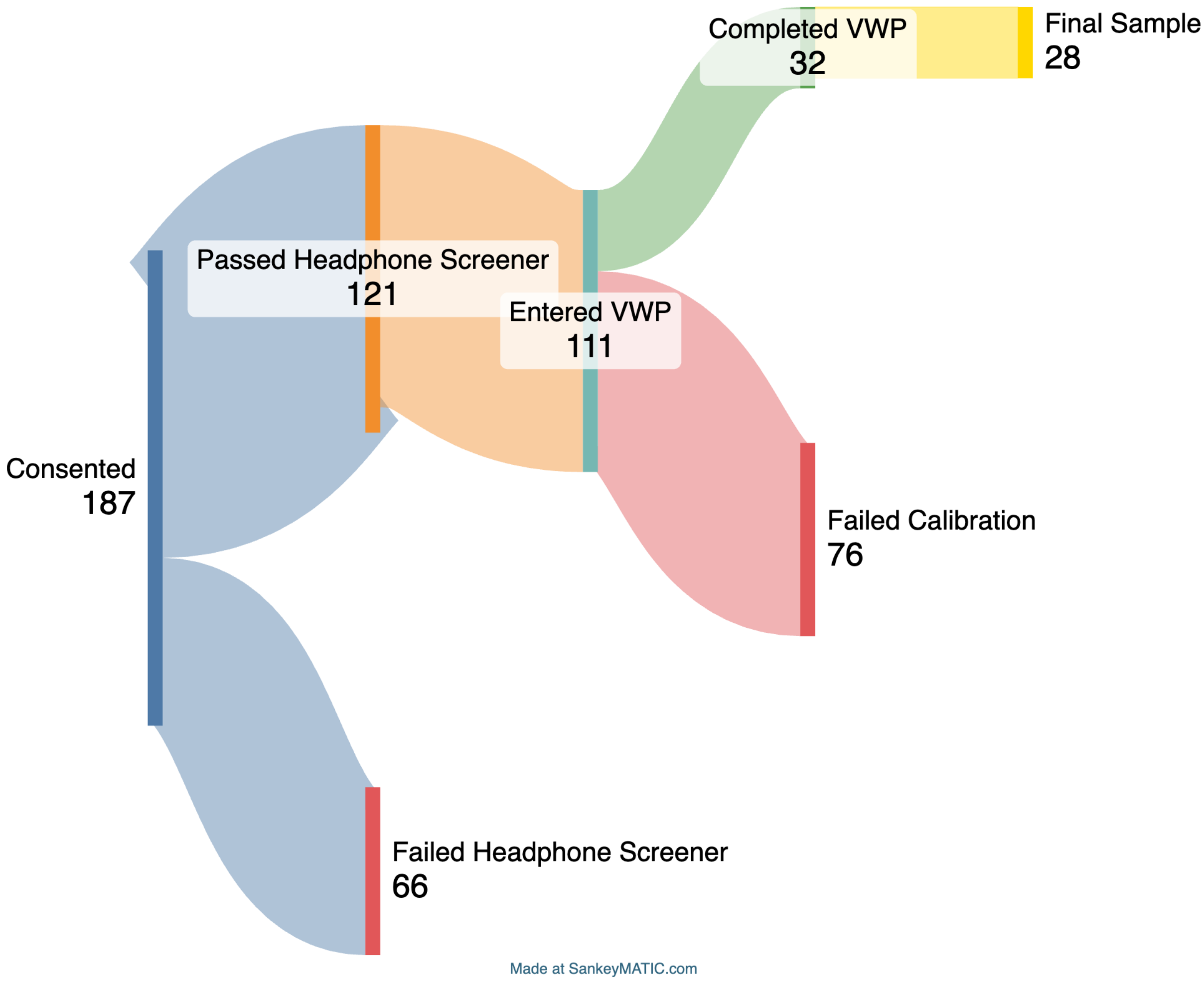
Inclusion criteria required participants to: (1) be between 18 and 36 years old, (2) be native English speakers, (3) also be fluent in Spanish, and (4) reside in the United States. Criterion 1 was based on findings from Colby and McMurray (2023), which suggest that age-related changes in spoken word recognition begin to emerge in individuals in their 40s; thus, we limited our sample to participants younger than 36. Criteria 2 and 3 ensured that we were recruiting native English speakers and those fluent in Spanish to test L1 and L2 interactions. Criterion 4 matched the population of the original study, which was conducted with university students in Iowa, and therefore we restricted recruitment to U.S. residents.

After agreeing to participate, individuals were redirected to the Gorilla experiment platform (www.gorilla.sc; (Anwyl-Irvine et al., 2020)). A flow diagram of participant progression through the experiment is shown in Figure 1. In total, 187 participants assessed the experimental platform and consented to be in the study. Of these, 121 passed the headphone screener checkpoint, and 111 proceeded to the VWP task. Out of the 111 participants who entered the VWP, 91 completed the final surveys at the end of the experiment. Among these, 32 participants successfully completed the VWP task with at least 100 trials, while 79 participants did not provide adequate data for inclusion, primarily due to failed calibration attempts. After applying additional exclusion criteria—namely, overall VWP task accuracy below 80%, excessive missing eye-tracking data (>30%), and sampling rate < 5hz —the final analytic sample consisted of 28 participants with usable eye-tracking data. Descriptive demographic information for the full sample that made it to the final survey is provided in Table 1.

```
#| echo: false

knitr::include_graphics(here::here("_manuscript", "Figures",
↪  "snakey_experiment.png"))
```

Figure 1

*This sankey plot illustrates the flow of participants from initial consent (*N *= 187) through each stage of the study to the final analyzed sample (N = 28). The width of each stream is proportional to the number of participants. Detours indicate points of attrition, including failures in the headphone screener (*N *= 66) and calibration (*N *= 76). Only participants who passed all screening and calibration stages, and completed the Visual World Paradigm (VWP), were included in the final sample.*



Materials

VWP..

Items. We adapted materials from Sarrett et al. (2022). In their cross-linguistic VWP, participants were presented with four pictures and a spoken Spanish word and had to select the image that matched the spoken word by clicking on it. The word stimuli for the experiment were chosen from textbooks used by students in their first and second year college Spanish courses.

The item sets consisted of two types of phonologically-related word pairs: one pair of Spanish-Spanish words and another of Spanish-English words. The Spanish-Spanish pairs were unrelated to the

Table 1

Participant demographic variables

Characteristic	**N = 91**[1]
Age	(20.0, 35.0), 28.2(4.4)
Gender	
Female	42 / 91 (46%)
Male	49 / 91 (54%)
Spoken dialect	
Do not know	11 / 91 (12%)
Midwestern	19 / 91 (21%)
New England	11 / 91 (12%)
Other (please specify)	7 / 91 (7.7%)
Pacific northwest	7 / 91 (7.7%)
Pacific southwest	7 / 91 (7.7%)
Southern	21 / 91 (23%)
Southwestern	8 / 91 (8.8%)
Ethnicity	
Decline to state	1 / 91 (1.1%)
Hispanic or Latino	38 / 91 (42%)
Not Hispanic or Latino	52 / 91 (57%)
Race	
American Indian/Alaska Native	2 / 91 (2.2%)
Asian	13 / 91 (14%)
Black or African American	10 / 91 (11%)
Decline to state	7 / 91 (7.7%)
More than one race	4 / 91 (4.4%)
White	55 / 91 (60%)
Browser	
Chrome	77 / 91 (85%)
Edge	3 / 91 (3.3%)
Firefox	7 / 91 (7.7%)
Safari	4 / 91 (4.4%)
Years Speaking Spanish	(0, 35), 15(10)
% Experience Using Spanish Daily Life	25(23)

[1](Min, Max), Mean(SD); n / N (%); Mean(SD)

Spanish-English pairs. All the word pairs were carefully controlled on a number of dimensions (see Sarrett et al., 2022). There were three experimental conditions: (1) the Spanish-Spanish (within) condition, where one of the Spanish words was the target and the other was the competitor; (2) the Spanish-English (cross-linguistic) condition, where a Spanish word was the target and its English phonological cohort served as the competitor; and (3) the No Competitor condition, where the Spanish word did not overlap with any other word in the set. The Spanish-Spanish condition had twice as many trials as the other conditions due to the interchangeable nature of the target and competitor words in that pair.

Each item within a set appeared four times as the target word, resulting in a total of 240 trials (15 sets × 4 items per set × 4 repetitions). Each set included one Spanish–Spanish cohort pair and one Spanish–English cohort pair. In the Spanish–Spanish condition, both words in the pair served as mutual competitors—for example, *cielo* activated *ciencia*, and vice versa. This bidirectional relationship yielded 120 trials for the Spanish–Spanish condition.

In contrast, the Spanish–English pairs had an asymmetrical relationship: only one item in each pair functioned as a competitor (e.g., *botas* could activate *frontera*, but *frontera* did not have a corresponding competitor). As a result, there were 60 trials each for the Spanish–English and No Competitor conditions. Across all trials, target items were equally distributed among the four screen quadrants to ensure balanced visual presentation

Stimuli. In Sarrett et al. (2022) all auditory stimuli were recorded by a female bilingual speaker whose native language was Mexican Spanish and also spoke English. Stimuli were recorded in a sound-attenuated room sampled at 44.1 kHz. Auditory tokens were edited to reduce noise and remove clicks. The auditory tokens were then amplitude normalized to 70 dB SPL. For each target word, there were four separate recordings so each instance was unique.

Visual stimuli were images from a commercial clipart database that were selected by a consensus method involving a small group of students. All .wav files were converted to .mp3 for online data collection. All stimuli can be found here: https://osf.io/mgkd2/.

Headphone screener. Headphones were required for all participants. To ensure compliance, we administered a six-trial headphone screening task adapted from Milne et al. (2021), which is available for implementation on the Gorilla platform. On each trial, three tones of the same frequency and duration were presented sequentially. One tone had a lower amplitude than the other two tones. Tones were presented in stereo, but the tones in the left and right channels were 180 out of phase across stereo channels—in free field, these sounds should cancel out or create distortion, whereas they will be perfectly clear over headphones. The listener picked which of the three tones was the quietest. Performance is generally at the ceiling when wearing headphones but poor when listening in the free field (due to phase cancellation).

Participant background and experiment conditions questionnaire. We had participants complete a demographic questionnaire as part of the study. The questions covered basic demographic information, including age, gender, spoken dialect, ethnicity, and race. To gauge L2 experience, we asked participants when they started speaking Spanish, how many years of Spanish speaking experience they had, and to provide, on a scale between 0-100, how often they use Spanish in their daily lives.

To further probe into data quality issues and get a better sense of why participants could not make it through the experiment, participants answered a series of questions at the end of the experiment related to their personal health and environmental conditions during the experiment. These questions addressed any history of vision problems (e.g., corrected vision, eye disease, or drooping eyelids) and whether they were currently taking medications that might impair judgment. Participants also indicated if they were wearing eyeglasses, contacts, makeup, false eyelashes, or hats.

The questionnaire asked about natural light in the room, if they were using a built-in camera or an external one (with an option to specify the brand), and their estimated distance from the camera. Participants were asked to estimate how many times they looked at their phone or got up during the experiment and whether their environment was distraction-free.

Additional questions assessed the clarity of calibration instructions, allowing participants to suggest improvements, and asked if they were wearing a mask during the session. These questions aimed to gather insights into personal and environmental factors that could impact data quality and participant comfort during the experiment.

Procedure

All tasks and questionnaires were developed using the Gorilla Experiment Builder's graphical user interface (GUI) and integrated coding tools (Anwyl-Irvine et al., 2020). Each participant completed the study in a single session lasting approximately 45 minutes. Tasks were presented in a fixed order: informed consent, headphone screening, the spoken word Visual World Paradigm (VWP) task, and a set of questionnaire items. These are available to view here: https://app.gorilla.sc/openmaterials/953693.

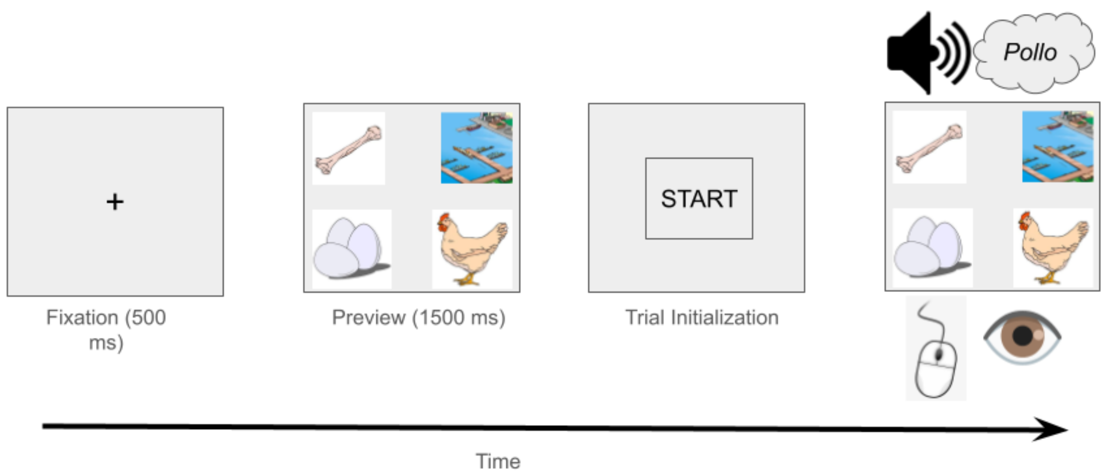
Only personal computers were permitted for participation. Upon entering the study from Prolific, participants were presented with a consent form. Once consent was given, participants completed a headphone screening test. They had three attempts to pass this test. If unsuccessful by the third attempt, participants were directed to an early exit screen, followed by the questionnaire. They had three attempts to pass this test. If unsuccessful by the third attempt, participants were directed to an early exit screen, followed by the questionnaire.

If the headphone screener was passed, participants were next introduced to the VWP task. This began with instructional videos providing specific guidance on the ideal experiment setup for eye-tracking and calibration procedures. You can view the videos here: https://osf.io/mgkd2/. Participants were then required to enter full-screen mode before calibration. A 9-point calibration procedure was used. Calibration occurred every 60 trials for a total of 3 calibrations. Participants had three attempts to successfully complete each calibration phase. If calibration was unsuccessful, participants were directed to an early exit screen, followed by the questionnaire.

In the main VWP task, each trial began with a 500 ms fixation cross at the center of the screen. This was followed by a preview screen displaying four images, each positioned in a corner of the screen. After 1500 ms, a start button appeared in the center. Participants clicked the button to confirm they were focused on the center before the audio played. Once clicked, the audio was played, and the images remained visible. Participants were instructed to click the image that best matched the spoken target word, while their eye movements were recorded. Eye movements were only recorded on that screen. Figure 2 displays the VWP trial sequence.

Figure 2

VWP trial schematic



After completing the main VWP task, participants proceeded to the final questionnaire, which included questions about the eye-tracking task and basic demographic information. Participants were then thanked for their participation.

Preprocessing data

After the data is collected you can begin preprocessing your data. Below we highlight the steps needed to preprocess your webcam eye-tracking data and get it ready for analysis. For some of this preprocessing we will use the newly created `webgazeR` package (v. 0.7.2).

For preprocessing visual world webcam eye data, we follow seven general steps (see Figure 3):

1. Reading in data
2. Data exclusion
3. Combining trial- and eye-level data

4. Assigning areas of interest (AOIs)
5. Time binning
    1. Downsampling
    2. Upsampling (optional)
6. Aggregating (optional)
7. Visualization (optional)

For each of these steps, we will display R code chunks demonstrating how to perform each step with helper functions (if applicable) from the `webgazeR` (Geller & Prystauka, 2024) package in R.

Load packages

Package Installation and Setup. Before proceeding, make sure to load the required packages by running the code below. If you already have these packages installed and loaded, feel free to skip this step. The code in this tutorial will not run correctly if any of the necessary packages are missing or not properly loaded.

webgazeR installation. The `webgazeR` package is installed from the Github repository using the `remotes` (Csárdi et al., 2024) package.

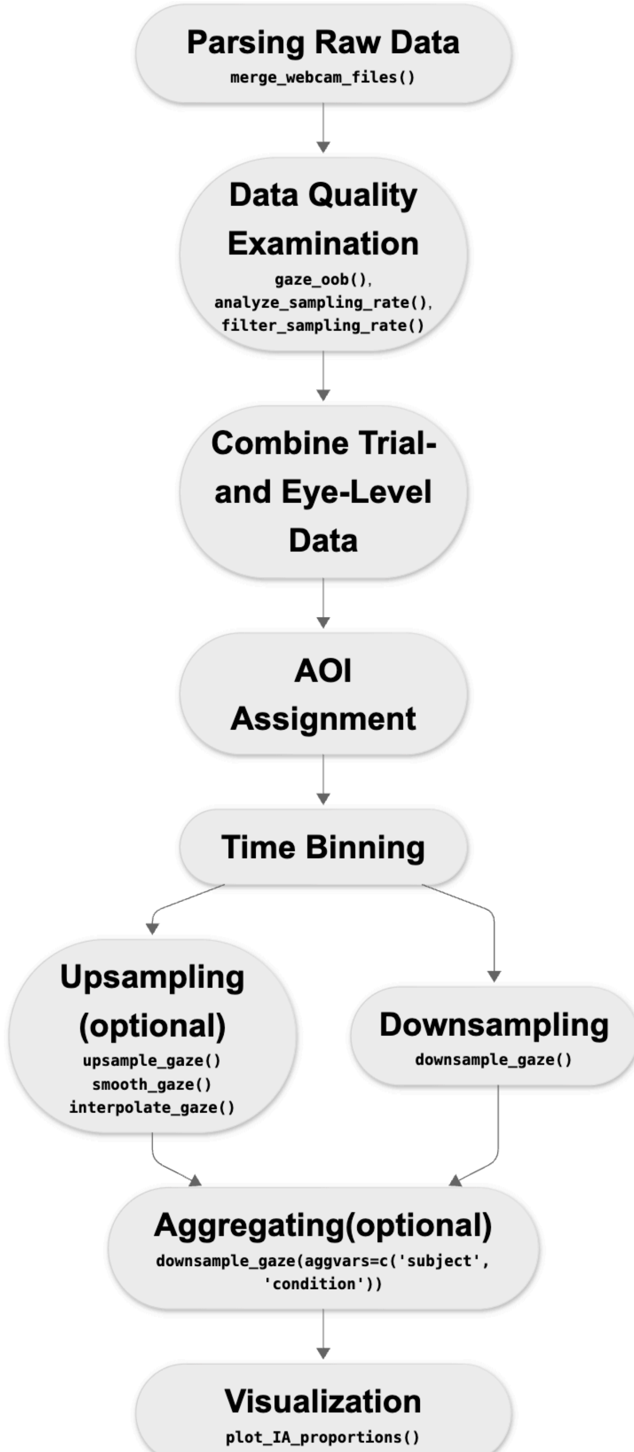
```
library(remotes) # install github repo

remotes::install_github("jgeller112/webgazeR")
```

Once this is installed, `webgazeR` can be loaded along with additional useful packages. The following code will load the required packages or install them if you do not have them on your system.

```
# List of required packages
required_packages <- c(
  "tidyverse",      # data wrangling
  "here",           # relative paths instead of absolute aids in reproducibility
  "tinytable",      # nice tables
  "janitor",        # functions for cleaning up your column names
  "webgazeR",       # has webcam functions
  "readxl",         # read in Excel files
  "ggokabeito",     # color-blind friendly palettes
  "flextable",      # Word tables
  "permuco",        # permutation analysis
```

Figure 3

Preprocessing steps for webcam eye-tracking data using webgazeR functions



```
  "foreach",         # permutation analysis
  "geomtextpath",    # for plotting labels on lines of ggplot figures
  "cowplot"          # combine ggplot figures
)
```

Once `webgazeR` and other helper packages have been installed and loaded the user is ready to start cleaning your data.

Reading in data

Behavioral, trial-level, data. To process eye-tracking data you will need to make sure you have both the behavioral data and the eye-tracking data files. We have all the data needed in the repository by navigating to the L2 subfolder from the main project directory (~/data/L2). For the behavioral data, Gorilla produces a `.csv` file that includes trial-level information (here contained in the object `L2_data`). The files needed are called `data_exp_196386-v5_task-scf6.csv`. and `data_exp_196386-v6_task-scf6.csv`. We have two files because we ran a modified version of the experiment.

The .csv files contain meta-data for each each trial, such as what picture were presented on each trial, which object was the target, reaction times, audio presentation times, what object was clicked on, etc. To load our data files into our R environment, we use the `here` (Müller, 2020) package to set a relative rather than an absolute path to our files. We read in the data files from the repository for both versions of the task and merge the files together. `L2_data` merges both `data_exp_196386-v5_task-scf6.csv` and `data_exp_196386-v6_task-scf6.csv` into one object.

```
# load in trial level data
# combine data from version 5 and 6 of the task
L2_1 <- read_csv(here("data", "L2", "data_exp_196386-v5_task-scf6.csv"))
L2_2 <- read_csv(here("data", "L2", "data_exp_196386-v6_task-scf6.csv"))

L2_data <- rbind(L2_1, L2_2) # bind the two objects together
```

Eye-tracking data. Gorilla currently saves each participant's eye-tracking data on a per-trial basis. The `raw` subfolder in the project repository contains the eye-tracking files by participant for each trial individually (~/data/L2/raw). Contained in those files, we have information pertaining to each trial such as participant id, time since trial started, x and y coordinates of looks, convergence (the model's confidence in finding a face (and accurately predicting eye movements), face confidence (represents the support vector machine (SVM) classifier score for the face model fit), and information pertaining to the the AOI screen coordinates (standardized and user-specific). The `vwp_files_L2` object below contains a list of all the files contained in the folder. Because `vwp_files_L2` contains trial data as well as calibration data, we remove the calibration trials and save the non-calibration to to `vwp_paths_filtered_L2`.

```r
# Get the list of all files in the folder

# thank you to Reviewer 1 for suggesting this code
vwp_files_L2 <- list.files(here::here("data", "L2", "raw"), full.names = TRUE,
↪  pattern = "\\.(csv|xlsx)$") %>%
# remove calibration trials
discard(~ grepl("calibration", .x))
```

When data is generated from Gorilla, each trial in your experiment is saved as a separate file. To analyze the data, these individual files need to be combined into a single dataset. The `merge_webcam_files()` function from webgazeR is designed for this purpose. It reads all trial-level files from a specified folder—regardless of file format (.csv, .tsv, or .xlsx)—and merges them into one cohesive tibble or data frame.

Before using `merge_webcam_files()`, ensure your working directory is set to the location where the raw files are stored. The function automatically standardizes column names using clean_names(), binds the files together, and filters the data to retain only the relevant rows. Specifically, it keeps rows where the type column equals "prediction", which are the rows that contain actual eye-tracking predictions. It also filters based on the screen_index argument: if you collected gaze data across multiple screens, you can specify one or several indices (e.g., screen_index = c(1, 4, 5)).

In addition to merging and filtering, `merge_webcam_files()` requires the user to explicitly map critical columns—subject, trial, time, and x/y gaze coordinates. This makes the function highly flexible and robust across different experimental platforms. For instance, the function automatically renames the spreadsheet_row column to trial, and converts subject and trial into factors for compatibility with downstream analyses.

Currently, the kind argument supports "gorilla" data, but future extensions will add support for other platforms like Labvanced (Kaduk et al., 2024), PsychoPy (Peirce et al., 2019), and PCIbex (Zehr & Schwarz, 2018). By explicitly allowing platform specification and flexible column mapping, merge_webcam_files() ensures a consistent and streamlined pipeline for preparing webcam eye-tracking data for analysis.

As a general note, all steps should be followed in order due to the renaming of column names. If you encounter an error it might be because column names have not been changed.

```r
setwd(here::here("data", "L2", "raw")) # set working directory to raw data folder

edat_L2 <- merge_webcam_files(vwp_files_L2, screen_index=4, col_map =
↪  list(subject = "participant_id", trial="spreadsheet_row",
↪  time="time_elapsed", x="x_pred_normalised", y="y_pred_normalised"),
↪  kind="gorilla")
```

To ensure high-quality data, we applied a set of behavioral and eye-tracking exclusion criteria prior to merging datasets. Participants were excluded if they met any of the following conditions: (1) failure to successfully calibrate throughout the experiment (fewer than 100 completed trials), (2) low behavioral accuracy (below 80%), (3) low sampling rate (below 5 Hz), or (4) a high proportion of gaze samples falling outside the display area (greater than 30%).

Successful calibration is critical for reliable eye-tracking measurements, as poor calibration directly compromises the spatial accuracy of gaze data (Blascheck et al., 2017). Requiring a sufficient number of completed trials is crucial for ensuring adequate statistical power and stable individual-level parameter estimates, particularly in tasks with high trial-to-trial variability (Brysbaert & Stevens, 2018). We choose 100 trials as this meant participants passed at least two calibration attempts during the study. Behavioral accuracy (>= 80%) was used as an additional screening measure because low task performance may indicate a lack of attention, misunderstanding of the task, or random responding, all of which could undermine both the behavioral and eye-movement data quality (Bianco et al., 2021). Filtering based on sampling rate ensures that datasets with too few gaze samples (due to technical or environmental issues) are removed, as low sampling rates significantly degrade temporal precision and bias gaze metrics (Semmelmann & Weigelt, 2018). Finally, we excluded participants with excessive off-screen data (>30%) because this indicates poor gaze tracking, likely caused by head movement, poor lighting, or loss of face detection. At this time, there is no set guide on what constitutes acceptable data loss for webcam-based studies. We felt 30% was a reasonable cut-off. At the trial-level, we also removed incorrect trials and trials where sampling rate was < 5 Hz.

What we will do first is create a cleaned up version of our behavioral, trial-level data `L2_data` by creating an object named `eye_behav_L2` that selects useful columns from that file and renames stimuli to make them more intuitive. Because most of this will be user-specific, no function is called here. Below we describe the preprocessing done on the behavioral data file. The below code processes and transforms the `L2_data` dataset into a cleaned and structured format for further analysis. First, the code renames several columns for easier access using `janitor::clean_names()`(Firke, 2023) function. We then select only the columns we need and filter the dataset to include only rows where `screen_name` is "VWP" and `zone_type` is called "response_button_image", representing the picture selected for that trial. Afterward, the function renames additional columns (`tlpic` to `TL`, `trpic` to `TR`, etc.). We also renamed `participant_private_id` to `subject`, `spreadsheet_row` to `trial`, and `reaction_time` to `RT`. This makes our columns consistent with the `edat_L2` above for merging later on. Lastly, `reaction time` (RT) is converted to a numeric format for further numerical analysis.

It is important to note here that what the behavioral spreadsheet denotes as trial is not in fact the trial number used in the eye-tracking files. Thus it is imperative you use `spreadsheet row` as trial number to merge the two files successfully.

```
eye_behav_L2 <- L2_data %>%

  janitor::clean_names() %>%

  # Select specific columns to keep in the dataset
```

```
  dplyr::select(participant_private_id, correct, tlpic, trpic, blpic, brpic,
↪  condition,
                eng_targetword, targetword, typetl, typetr, typebl, typebr,
↪  zone_name,
                zone_type, reaction_time, spreadsheet_row, response, screen_name)
↪  %>%

  # Filter the rows where 'Zone.Type' equals "response_button_image"
  # participants clicked on preview screen so now need to filter based on screen.
  ↪
  dplyr::filter(screen_name == "VWP", zone_type == "response_button_image") %>%

  # Rename columns for easier use and readability
  dplyr::rename(
    TL = tlpic,             # Rename 'tlpic' to 'TL'
    TR = trpic,             # Rename 'trpic' to 'TR'
    BL = blpic,             # Rename 'blpic' to 'BL'
    BR = brpic,             # Rename 'brpic' to 'BR'
    targ_loc = zone_name,   # Rename 'zone_name' to 'targ_loc'
    subject = participant_private_id,  # Rename 'participant_private_id' to
    ↪  'subject'
    trial = spreadsheet_row,  # Rename 'spreadsheet_row' to 'trial'
    acc = correct,          # Rename 'correct' to 'acc' (accuracy)
    RT = reaction_time      # Rename 'reaction_time' to 'RT'
  ) %>%

  # Convert the 'RT' (Reaction Time) column to numeric type
  dplyr::mutate(RT = as.numeric(RT),
                subject = as.factor(subject),
                trial = as.factor(trial))
```

Audio onset. Because we are playing audio on each trial and running this experiment from the browser, audio onset is never going to be consistent across participants. In Gorilla there is an option to collect advanced audio features (you must make sure you select this when designing the study) such as when the audio play was requested, played, and ended. We will want to incorporate this timing information into our analysis pipeline. Gorilla records the onset of the audio which varies by participant. We are extracting that in the `audio_rt_L2` object by filtering `zone_type` to `content_web_audio` and a response equal to "AUDIO PLAY EVENT FIRED". This will tell us when the audio was triggered in the experiment. We are creating a column called (`RT_audio`) which we will use later on to correct for audio delays. Please note that on some trials the audio may not play. This is a function of the browser a participant is using and the experimenter has no control over this (see https://support.gorilla.sc/support/troubleshooting-and-technical/technical-checklist#autoplayingsoundandvideo). When running your experiment on a different platform, make sure you try and request this information, or at the very least acknowledge audio delay.

```
audio_rt_L2 <- L2_data %>%

  janitor::clean_names()%>%

select(participant_private_id,zone_type, spreadsheet_row, reaction_time,
↪  response) %>%

  filter(zone_type=="content_web_audio", response=="AUDIO PLAY EVENT FIRED")%>%
  distinct() %>%
dplyr::rename("subject" = "participant_private_id",
       "trial" ="spreadsheet_row",
       "RT_audio" = "reaction_time",
       "Fired" = "response") %>%
select(-zone_type) %>%
mutate(RT_audio=as.numeric(RT_audio))
```

We then merge this information with `eye_behav_L2`.

```
# merge the audio Rt data to the trial level object
trial_data_rt_L2 <- merge(eye_behav_L2, audio_rt_L2, by=c("subject", "trial"))
```

Trial removal. As stated above, participants who did not successfully calibrate 3 times or less were rejected from the experiment. Deciding to remove trials is ultimately up to the researcher. In our case, we removed participants with less than 100 trials. Let's take a look at how many participants meet this criterion by probing the `trial_data_rt_L2` object.In Table 2 we can see several participants failed some of the calibration attempts and do not have an adequate number of trials. Again we make no strong recommendations here. If you decide to use a criterion such as this, we recommend pre-registering your choice.

```
# find out how many trials each participant had
edatntrials_L2 <-trial_data_rt_L2 %>%
 dplyr::group_by(subject)%>%
 dplyr::summarise(ntrials=length(unique(trial)))
```

Let's remove participants with less than 100 trials from the analysis using the below code.

Table 2

Participants with less than 100 trials

subject	ntrials
12102265	2
12110638	55
12110829	59
12110878	59
12110897	60
12111234	57
12111244	58
12111363	58
12111663	57
12111703	58
12111869	60
12111960	46
12112152	59
12212113	56
12213826	99
12213965	59

```
trial_data_rt_L2 <- trial_data_rt_L2 %>%
  filter(subject %in% edatntrials_bad_L2$subject)
```

Low accuracy. In our experiment, we want to make sure accuracy is high (> 80%). Again, we want participants that are fully attentive in the experiment. In the below code, we keep participants with accuracy equal to or above 80% and only include correct trials and assign it to `trial_data_acc_clean_L2`.

```
# Step 1: Calculate mean accuracy per subject and filter out subjects with mean
↪  accuracy < 0.8
subject_mean_acc_L2 <- trial_data_rt_L2 %>%
  group_by(subject) %>%
  dplyr::summarise(mean_acc = mean(acc, na.rm = TRUE)) %>%
  filter(mean_acc > 0.8)

# Step 2: Join the mean accuracy back to the main dataset and exclude trials with
↪  accuracy < 0.8
trial_data_acc_clean_L2 <- trial_data_rt_L2 %>%
```

```
inner_join(subject_mean_acc_L2, by = "subject") %>%
filter(acc==1) # only use accurate responses for fixation analysis
```

RTs. There is much debate on how to handle reaction time (RT) data (see Miller, 2023). Because of this. we leave it up to the reader and researcher to decide what to do with RTs. In this tutorial we leave RTs untouched.

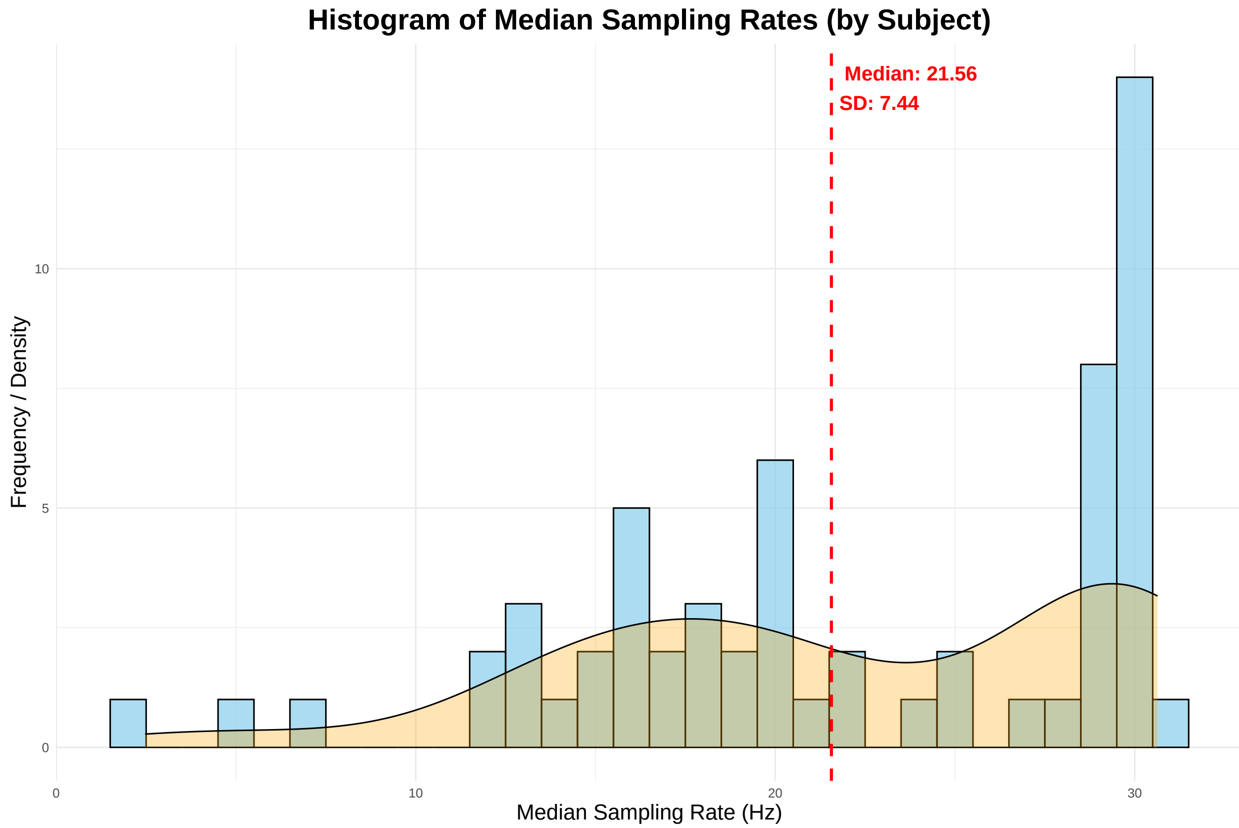
Sampling rate. While most commercial eye-trackers sample at a constant rate, data captured by webcams are widely inconsistent. Below is some code to calculate the sampling rate of each participant. Ideally, you should not have a sampling rate less than 5 Hz. It has been recommended you drop those values (Bramlett & Wiener, 2024) The below function `analyze_sample_rate()` calculates the sampling rate for each subject and each trial in our eye-tracking dataset (`edat_L2`). The `analyze_sample_rate()` function provides overall statistics, including the option to report mean or median (Bramlett & Wiener, 2024) sampling rate and standard deviation of sampling rates in your experiment. Sampling rate calculations followed standard procedures (e.g., Bramlett & Wiener, 2024; Prystauka et al., 2024). The function also generates a histogram of sampling rates by-subject. Looking at Figure 4, the sampling rate ranges from 5 to 35 Hz with a median sampling rate of 21.56. This corresponds to previous webcam eye-tracking work (e.g., Bramlett & Wiener, 2024; Prystauka et al., 2024)

```
samp_rate_L2 <- analyze_sampling_rate(edat_L2, summary_stat="Median")
```

```
Overall Median Sampling Rate (Hz): 21.56
Overall SD of Sampling Rate (Hz): 7.44
```

Figure 4

Participant sampling-rate for L2 experiment. A histogram and overlayed density plot shows median sampling rate by participant. The overall median and SD is highlighted in red.



When using the above function, separate data frames are produced by-participants and by-trial. These can be added to the behavioral data frame using the below code.

```
trial_data_L2 <- merge(trial_data_acc_clean_L2, samp_rate_L2, by=c("subject",
↪  "trial"))
```

Now we can use this information to filter out data with poor sampling rates. Users can use the `filter_sampling_rate()` function. The `filter_sampling_rate()` function is designed to process a dataset containing participant-level and trial-level sampling rates. It allows the user to either filter out data that falls below a certain sampling rate threshold or simply label it as "bad". The function gives flexibility by allowing the threshold to be applied at the participant-level, trial-level, or both. It also lets the user decide whether to remove the data or flag it as below the threshold without removing it. If `action` = remove, the function will output how many subjects and trials were removed using the threshold. We leave it up to the user to decide what to do with low sampling rates and make no specific recommendations. Here we use the `filter_sampling_rate()` function to remove trials and participants from the`trial_data_L2` object.

```
filter_edat_L2 <- filter_sampling_rate(trial_data_L2,threshold = 5,
                                          action = "remove",
                                          by = "both")
```

Out-of-bounds (outside of screen). It is essential to exclude gaze points that fall outside the screen, as these indicate unreliable estimates of gaze location. The `gaze_oob()` function quantifies how many data points fall outside these bounds, using the eye-tracking dataset (e.g., edat_L2) and the standardized screen dimensions—here set to (1, 1) because Gorilla recommends using standardized coordinates. If the `remove` argument is set to TRUE, the function applies an outer-edge filtering method to eliminate these out-of-bounds points (see Bramlett & Wiener, 2024). The outer-edge approach appears to be a less biased approach based on demonstrations from Bramlett and Wiener (2024), where they showed minimal data loss compared to other approaches (e.g., inner-edge approach).

The function returns a summary table showing the total number and percentage of gaze points that fall outside the bounds, broken down by axis (X, Y), as well as the combined total (see Table 3). It also returns three additional tibbles: (1) missingness by-subject, (2) missingness by-trial, and (3) a cleaned dataset with all the data merged, and the problematic rows removed if specified. These outputs can be referenced in a final report or manuscript. As shown in Figure 5, no fixation points fall outside the standardized coordinate range.

```
oob_data_L2 <- gaze_oob(data=edat_L2, subject_col = "subject",
                       trial_col = "trial",
                       x_col = "x",
                       y_col = "y",
                      screen_size = c(1, 1), # standardized coordinates have
                       ↪ screen size 1,1
                       remove = TRUE)
```

```
#| echo: false

oob_data_L2$subject_results %>%
  mutate(across(where(is.numeric), ~round(.x, 2))) %>%
  rename_with(~ gsub("_", "\n", .x)) %>%         # Replace underscores with line
  ↪ breaks
  rename_with(~ gsub("percentage", "%", .x, ignore.case = TRUE)) %>%  # Replace
  ↪ 'percent' with '%'
  head() %>%
        flextable() %>%
    fontsize(size = 12) %>%  # Reduce font size
```

Table 3

Out of bounds gaze statistics by-participant (for 6 participants)

subject	totaltrials	totalpoints	outsidecount	subjectmissing%	xoutsidecount	youtsidecount	xoutside%	youtside%
12102265	60.00	6,192.00	1,132.00	18.28	202.00	947.00	3.26	15.29
12102286	240.00	11,765.00	354.00	3.01	267.00	181.00	2.27	1.54
12102530	240.00	9,011.00	385.00	4.27	244.00	147.00	2.71	1.63
12110559	240.00	11,887.00	415.00	3.49	194.00	221.00	1.63	1.86
12110579	178.00	5,798.00	1,061.00	18.30	696.00	435.00	12.00	7.50
12110585	240.00	13,974.00	776.00	5.55	83.00	694.00	0.59	4.97

```
  padding(padding = 1) %>%
  font(fontname = "Times New Roman", part = "all") %>%
  set_table_properties(layout="autofit") %>% # Reduce padding inside cells
  autofit() %>%
  theme_apa()
```

We can use the `data_clean` tibble returned by the `gaze_oob()` function to filter out trials and subjects with more than 30% missing data. The value of 30% is just a suggestion and should not be used as a rule of thumb for all studies nor are we endorsing this value.

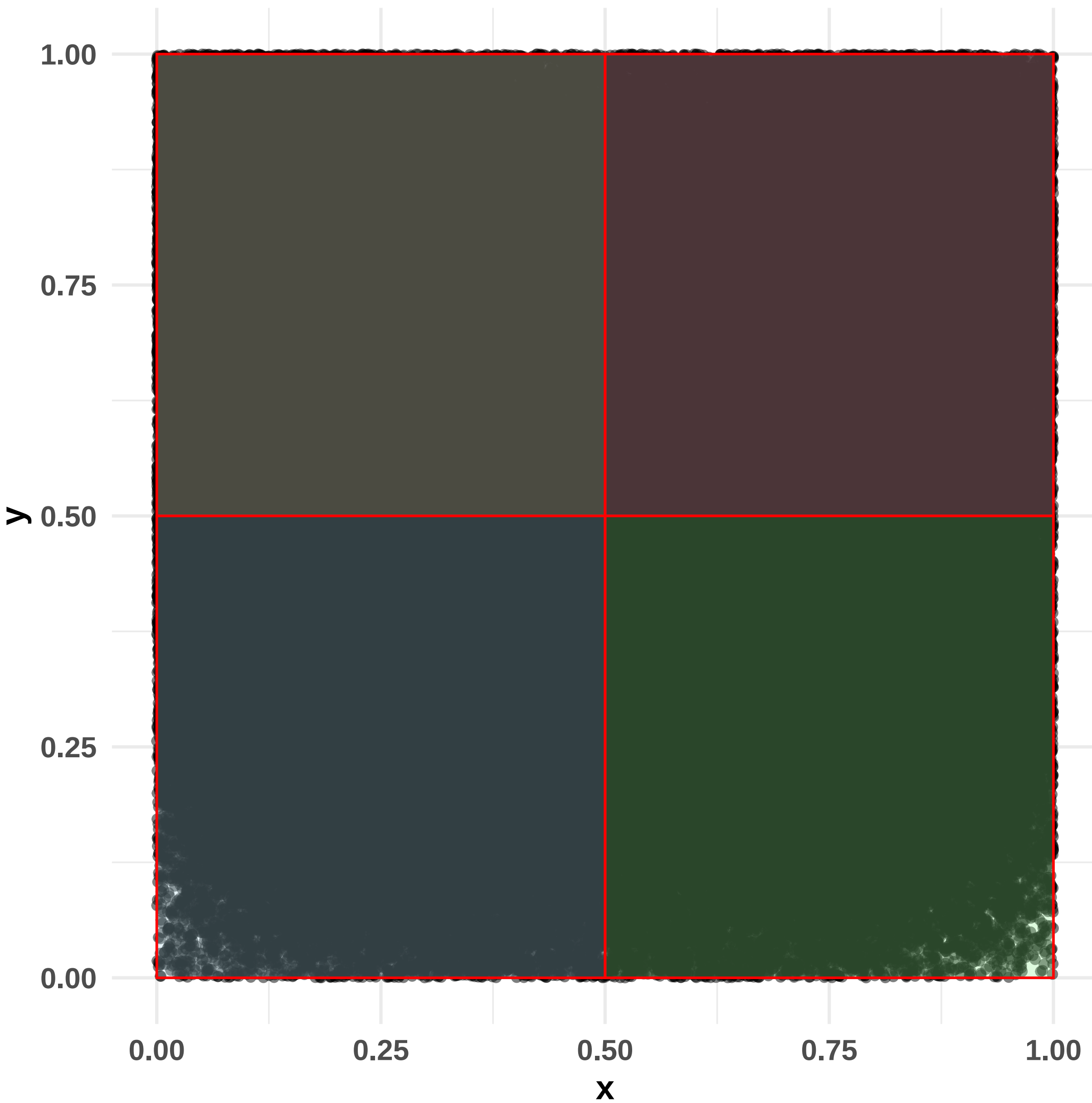
```
# remove participants with more than 30% missing data and trials with more than
↪  30% missing data
filter_oob <- oob_data_L2$data_clean %>%
  filter(trial_missing_percentage <= 30 | subject_missing_percentage <= 30)
```

Eye-tracking data

Convergence and confidence. To ensure data quality, we removed rows with poor convergence and low face confidence from our eye-tracking dataset. As described in Prystauka et al. (2024), the Gorilla eye-tracking output includes two key columns for this purpose: `convergence` and `face_conf` (similar variables may be available in other platforms as well). The `convergence` column contains values between 0 and 1, with lower values indicating better convergence—that is, greater model confidence in predicting gaze location and finding a face. Values below 0.5 typically reflect adequate convergence. The `face_conf` column reflects how confidently the algorithm detected a face in the frame, also ranging from 0 to 1. Here, values above 0.5 indicate a good model fit.

Figure 5

Looks to each quadrant of the screen



Accordingly, we filtered the `edat_L2`dataset to include only rows where convergence < 0.5 and face_conf > 0.5, and saved the cleaned dataset as edat_1_L2.

```
edat_1_L2 <- filter_oob %>%
 dplyr::filter(convergence <= .5, face_conf >= .5) # remove poor convergnce and
↪  face confidence
```

Combining eye and trial-level data. Next, we will combine the eye-tracking data and behavioral data. In this case, we'll use merge to add the behavioral data to the eye-tracking data. This ensures that all rows from the eye-tracking data are preserved, even if there isn't a matching entry in the behavioral data (missing values will be filled with NA). The resulting object is called dat_L2.

```
dat_L2 <- merge(edat_1_L2, filter_edat_L2)
```

Areas of Interest

Zone coordinates

In the lab, we can control many aspects of the experiment that cannot be controlled online. Participants will be completing the experiment under a variety of conditions including, different computers, with very different screen dimensions. To control for this, Gorilla outputs standardized zone coordinates (labeled as `x_pred_normalised` and `y_pred_normalised` in the eye-tracking file) . As discussed in the Gorilla documentation, the Gorilla lays everything out in a 4:3 frame and makes that frame as big as possible. The normalized coordinates are then expressed relative to this frame; for example, the coordinate 0.5, 0.5 will always be the center of the screen, regardless of the size of the participant's screen. We used the normalized coordinates in our analysis (in general, you should always use normalized coordinates). However, there are a few different ways to specify the four coordinates of the screen, which are worth highlighting here.

Quadrant approach. One way is to make the AOIs as big as possible, dividing the screen into four quadrants. This approach has been used in several studies [e.g., (Bramlett & Wiener, 2024; Prystauka et al., 2024). Table 4 lists coordinates for the quadrant approach and Figure 6 shows how each quadrant looks in standardized space.

Matching conditions with screen locations. The goal of the below code is to assign condition codes (e.g., Target, Unrelated, Unrelated2, and Cohort) to each image in the dataset based on the screen location where the image is displayed (e.g., TL, TR, BL, BR).

For each trial, the images are dynamically placed at different screen locations, and the code maps each image to its corresponding condition based on these locations.

Figure 6

AOI coordinates in standardized space using the quadrant approach

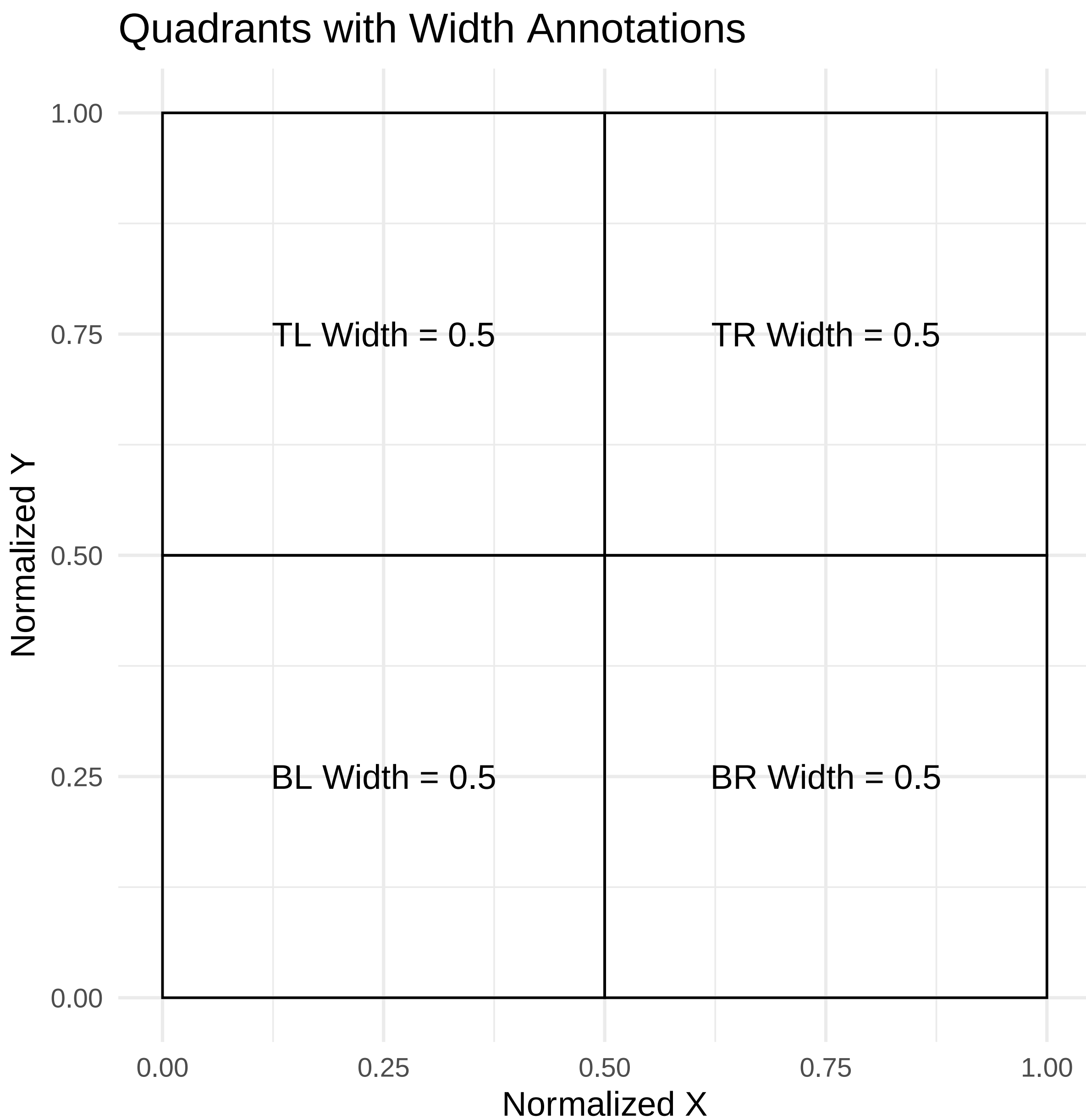


Table 4

Quandrant coordinates in standardized space

loc	x_normalized	y_normalized	width_normalized	height_normalized	xmin	ymin	xmax	ymax
TL	0.00	0.50	0.50	0.50	0.00	0.50	0.50	1.00
TR	0.50	0.50	0.50	0.50	0.50	0.50	1.00	1.00
BL	0.00	0.00	0.50	0.50	0.00	0.00	0.50	0.50
BR	0.50	0.00	0.50	0.50	0.50	0.00	1.00	0.50

```
# Assuming your data is in a data frame called dat_L2
dat_L2 <- dat_L2 %>%
  mutate(
    Target = case_when(
      typetl == "target" ~ TL,
      typetr == "target" ~ TR,
      typebl == "target" ~ BL,
      typebr == "target" ~ BR,
      TRUE ~ NA_character_  # Default to NA if no match
    ),
    Unrelated = case_when(
      typetl == "unrelated1" ~ TL,
      typetr == "unrelated1" ~ TR,
      typebl == "unrelated1" ~ BL,
      typebr == "unrelated1" ~ BR,
      TRUE ~ NA_character_
    ),
    Unrelated2 = case_when(
      typetl == "unrelated2" ~ TL,
      typetr == "unrelated2" ~ TR,
      typebl == "unrelated2" ~ BL,
      typebr == "unrelated2" ~ BR,
      TRUE ~ NA_character_
    ),
    Cohort = case_when(
      typetl == "cohort" ~ TL,
      typetr == "cohort" ~ TR,
      typebl == "cohort" ~ BL,
```

```
      typebr == "cohort" ~ BR,
      TRUE ~ NA_character_
    )
  )
```

In addition to tracking the condition of each image during randomized trials, a custom function, `find_location()`, determines the specific screen location of each image by comparing it against the list of possible locations. This function ensures that the appropriate location is identified or returns `NA` if no match exists. Specifically, `find_location()` first checks if the image is `NA` (missing). If the image is `NA`, the function returns `NA`, meaning that there's no location to find for this image. If the image is not `NA`, the function creates a vector called `loc_names` that lists the names of the possible locations. It then attempts to match the given image with the locations. If a match is found, it returns the name of the location (e.g., TL, TR, BL, or BR) of the image.

```
# Apply the function to each of the targ, cohort, rhyme, and unrelated columns

dat_colnames_L2 <- dat_L2 %>%
  rowwise() %>%
  mutate(
    targ_loc = find_location(c(TL = TL, TR = TR, BL = BL, BR = BR), Target),
    cohort_loc = find_location(c(TL = TL, TR = TR, BL = BL, BR = BR), Cohort),
    unrelated_loc = find_location(c(TL = TL, TR = TR, BL = BL, BR = BR),
    ↪  Unrelated),
    unrelated2_loc = find_location(c(TL = TL, TR = TR, BL = BL, BR = BR),
    ↪  Unrelated2)
  ) %>%
  ungroup()
```

Once we do this we can use the `assign_aoi()` function to loop through our object called `dat_colnames_L2` and assign locations (i.e., TR, TL, BL, BR) to where participants looked at on the screen. This requires the `x` and `y` coordinates and the location of our aois `aoi_loc`. Here we are using the quadrant approach. This function will label non-looks and off screen coordinates with NA. To make it easier to read we change the numerals assigned by the function to actual screen locations (e.g., TL, TR, BL, BR).

```
assign_L2 <- webgazeR::assign_aoi(dat_colnames_L2,X="x", Y="y",aoi_loc = aoi_loc)



AOI_L2 <- assign_L2 %>%
  mutate(loc1 = case_when(
```

```
    AOI==1 ~ "TL",
    AOI==2 ~ "TR",
    AOI==3 ~ "BL",
    AOI==4 ~ "BR"
  ))
```

In `AOI_L2` we label looks to Targets, Unrelated, and Cohort items with 1 (looked) and 0 (no look) using the `case_when` function from the `tidyverse` (Wickham, 2017)

```
AOI_L2 <- AOI_L2 %>%
  mutate(
    target = case_when(loc1 == targ_loc ~ 1, TRUE ~ 0),
    unrelated = case_when(loc1 == unrelated_loc ~ 1, TRUE ~ 0),
    unrelated2 = case_when(loc1 == unrelated2_loc ~ 1, TRUE ~ 0),
    cohort = case_when(loc1 == cohort_loc ~ 1, TRUE ~ 0)
  )
```

The locations of looks need to be pivoted into long format—that is, converted from separate columns into a single column. This transformation makes the data easier to visualize and analyze. We use the `pivot_longer()` function from the `tidyverse` to combine the columns (Target, Unrelated, Unrelated2, and Cohort) into a single column called `condition1`. Additionally, we create another column called `Looks`, which contains the values from the original columns (e.g., 0 or 1 for whether the area was looked at).

```
dat_long_aoi_me_L2 <- AOI_L2 %>%
  select(subject, trial, condition, target, cohort, unrelated, unrelated2, time,
  ↪  x, y, RT_audio) %>%
  pivot_longer(
    cols = c(target, unrelated, unrelated2, cohort),
    names_to = "condition1",
    values_to = "Looks"
  )
```

We further clean up the object by first cleaning up the condition codes. They have a numeral appended to them and that should be removed. We then adjust the timing in the `gaze_sub_L2_comp` object by aligning time to the actual audio onset. To achieve this, we subtract `RT_audio` from time for each trial. In addition, we subtract 300 ms from this to account for the 100 ms of silence at the beginning of each audio clip and 200 ms to account for the oculomotor delay when planning an eye movement (Viviani, 1990). Additionally, we set our interest period between 0 ms (audio onset) and 2000 ms. This was chosen based on the time course figures in Sarrett et al. (2022) . It is important that you choose your interest area carefully and preferably you preregister it. The interest period you choose can bias your findings (Peelle & Van Engen, 2021). We also filter out gaze coordinates that fall outside the standardized window, ensuring only valid data points are retained. The resulting object `gaze_sub_long_L2` provides the corrected time column spanning from -200 ms to 2000 ms relative to stimulus onset with looks outside the screen removed.

```
# repalce the numbers appended to conditions that somehow got added
dat_long_aoi_me_comp <- dat_long_aoi_me_L2 %>%
  mutate(condition = str_replace(condition, "TCUU-SPENG\\d*", "TCUU-SPENG")) %>%
  mutate(condition = str_replace(condition, "TCUU-SPSP\\d*", "TCUU-SPSP"))%>%
  na.omit()
```

```
# dat_long_aoi_me_comp has condition corrected

gaze_sub_L2_long <-dat_long_aoi_me_comp%>%
group_by(subject, trial, condition) %>%
  mutate(time = (time-RT_audio)-300) %>% # subtract audio rt onset and account
   ↪  for occ motor planning and silence in audio
 filter(time >= -200, time < 2000)
```

Samples to bins

Downsampling

Downsampling into larger time bins is a common practice in gaze data analysis, as it helps create a more manageable dataset and reduces noise. When using research grade eye-trackers, downsampling is an optional step in the preprocessing pipeline. However, with consumer-based webcam eye-tracking it is recommended you downsample your data so participants have consistent bin sizes (e.g., (Slim et al., 2024; Slim & Hartsuiker, 2023)). In `webgazeR` we included the `downsample_gaze()` function to assist with this process. We apply this function to the `gaze_sub_L2_long` object,and set the `bin.length` argument to 100, which groups the data into 100-millisecond intervals. This adjustment means that each bin now represents a 100 ms passage of time. We specify time as the variable to base these bins on, allowing us to focus on broader patterns over time rather than individual millisecond fluctuations. There is no agreed upon downsampling value, but with webcam data larger bins are preferred (see Slim & Hartsuiker, 2023).

In addition, the `downsample_gaze()` allows you to aggregate across other variables, such as `condition`, `condition1`, and use the newly created `time_bins` variable, which represents the time intervals over which we aggregate data. The resulting downsampled dataset, output as Table 5, provides a simplified and more concise view of gaze patterns, making it easier to analyze and interpret broader trends.

Table 5

Aggregated proportion looks for each condition in each 100 ms time bin

condition	condition1	time_bin	Fix
TCUU-ENGSP	cohort	-200.00	0.26
TCUU-ENGSP	cohort	-100.00	0.26
TCUU-ENGSP	cohort	0.00	0.25
TCUU-ENGSP	cohort	100.00	0.25
TCUU-ENGSP	cohort	200.00	0.23
TCUU-ENGSP	cohort	300.00	0.23

```
gaze_sub_L2 <- webgazeR::downsample_gaze(gaze_sub_L2_long, bin.length=100, timevar="time", aggvars=c("condition", "condition1", "time_bin"))
```

To simplify the analysis, we combine the two unrelated conditions and average them (this is for the proportional plots).

```
# Average Fix for unrelated and unrelated2, then combine with the rest
gaze_sub_L2_avg <- gaze_sub_L2 %>%
    group_by(condition, time_bin) %>%
    summarise(
        Fix = mean(Fix[condition1 %in% c("unrelated", "unrelated2")], na.rm = TRUE),
        condition1 = "unrelated",  # Assign the combined label
        .groups = "drop"
    ) %>%
    # Combine with rows that do not include unrelated or unrelated2
    bind_rows(gaze_sub_L2 %>% filter(!condition1 %in% c("unrelated", "unrelated2")))
```

The above will not include the subject variable. If you want to keep participant-level data we need to add `subject` to the `aggvars` argument.

```
# add subject-level data
gaze_sub_L2_id <- webgazeR::downsample_gaze(gaze_sub_L2_long, bin.length=100, timevar="time", aggvars=c("subject", "condition", "condition1", "time_bin"))
```

Upsampling

Users may wish to upsample their data rather than downsample it. This is standard in some preprocessing pipelines in pupillometry (Kret & Sjak-Shie, 2018) and has recently been applied to webcam-based eye-tracking data (Madsen et al., 2021). Like downsampling, upsampling standardizes the time intervals between samples; however, it also increases the sampling rate, which can produce smoother, less noisy data. This is useful if you want to align webcam eye-tracking with other measures (e.g., EEG).

Our webgazeR package provides several functions to assist with this process. The `upsample_gaze()` function allows users to upsample their gaze data to a higher sampling rate (e.g., 250 Hz or even 1000 Hz). After upsampling, users can apply the `smooth_gaze()` function to reduce noise (`webgazeR` uses a n-point moving average) followed by the `interpolate_gaze()` function to fill in missing values using linear interpolation. Below we show you how to use the function, but do not apply to the data.

```
AOI_upsample <- AOI %>%
  group_by(subject, trial) %>%
  upsample_gaze(
    gaze_cols = c("x", "y"),
    upsample_pupil = FALSE,
    target_hz = 250)
```

```
AOI_smooth=smooth_gaze(AOI_upsample, n = 5, x_col = "x", y_col = "y",
                        trial_col = "trial", subject_col = "subject")
```

```
aoi_interp <- interpolate_gaze(deduplicated_data,x_col = "x_pred_normalised",
↪  y_col = "y_pred_normalised",
                        trial_col = "trial", subject_col = "subject",
                        ↪  time_col="time" )
```

Aggregation

Aggregation is an optional step. If you do not plan to analyze proportion data, and instead what time binned data with binary outcomes preserved please set the `aggvars` argument to “none.” This will return a time binned column, but will not aggregate over other variables.

```
# get back trial level data with no aggregation
gaze_sub_id <- downsample_gaze(gaze_sub_L2_long, bin.length=100, timevar="time",
↪  aggvars="none")
```

We need to make sure we only have one unrelated value.

```
# make only one unrelated condition
gaze_sub_id <- gaze_sub_id %>%
  mutate(condition1 = ifelse(condition1=="unrelated2", "unrelated", condition1))
```

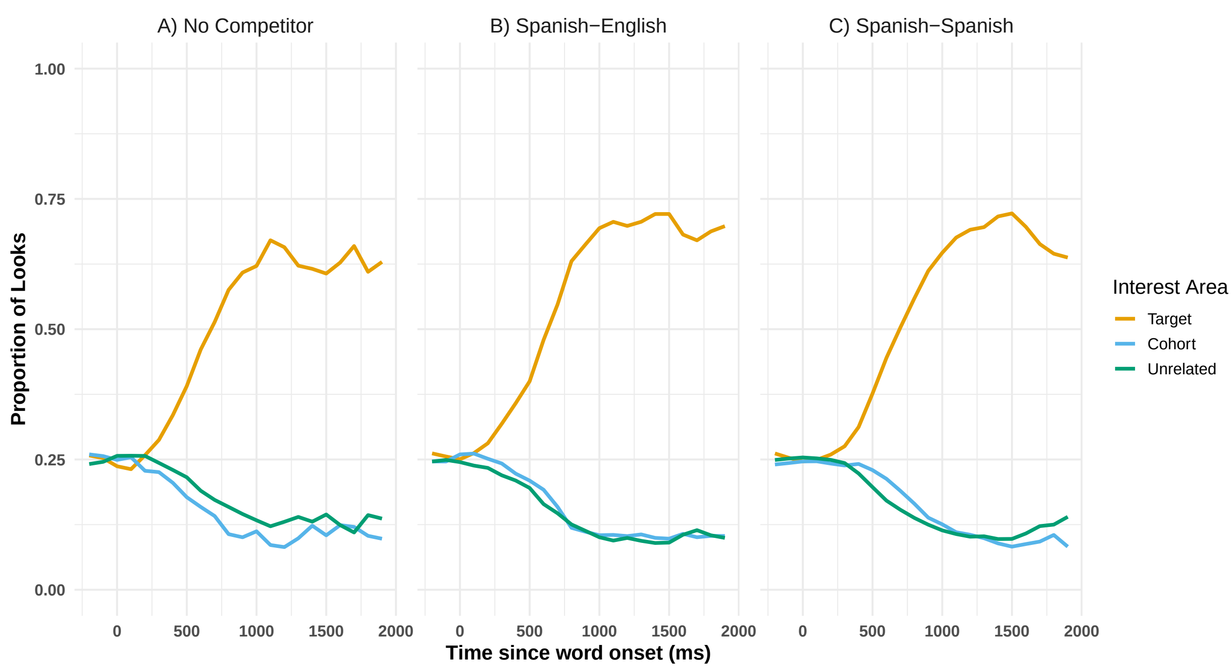
Visualizing time course data

To simplify plotting your time-course data, we have created the `plot_IA_proportions()` function. This function takes several arguments. The `ia_column` argument specifies the column containing your AOI labels. The `time_column` argument requires the name of your time bin column, and the `proportion_column` argument specifies the column containing fixation or look proportions. Additional arguments allow you to specify custom names for each IA in the `ia_mapping` argument, enabling you to label them as desired. In order to use this function, you must use the `downsample_gaze()` function.

Below, we have plotted the time-course data for each condition in Figure 7. By default, the graphs utilize a color-blind-friendly palette from the `ggokabeito` package (Barrett, 2021). However, you can set the argument `use_color` = FALSE to generate a non-colored version of the figure, where different line types and shapes differentiate conditions. Additionally, since these are ggplot objects, you can further customize them as needed to suit your analysis or presentation preferences.

Figure 7

Comparison of L2 competition effect in the No Competitor (a), Spanish-English (b), the Spanish-Spanish (c) conditions



Gorilla provided coordinates

Thus far, we have used the coordinates representing the four quadrants of the screen. However, Gorilla provides their own quadrants representing image location on the screen. To the authors' knowledge, these quadrants have not been looked at in any studies reporting eye-tracking results. Let's examine how reasonable our results are with the Gorilla provided coordinates.

We will use the function `extract_aois()` to get the standardized coordinates for each quadrant on screen. You can use the `zone_names` argument to get the zones you want to use. In our example, we want the `TL`, `BR`, `BL` `TR` coordinates. We input the object from above `vwp_paths_filtered_L2` that contains all our eye-tracking files and extract the coordinates we want. These are labeled in Table 6. In Figure 8 we can see that the AOIs are a bit smaller than then when using the quadrant approach. We can take these coordinates and use them in our analysis.Looking at Figure 9, we see the data is a bit noisier than the quadrant approach, but the curves are reasonable.

```
# apply the extract_aois fucntion
aois_L2 <- extract_aois(vwp_paths_filtered_L2, zone_names =  c("TL", "BR", "TR",
↪  "BL"))
```

```
#| echo: false

aois_L2 %>%
  flextable() %>%
     fontsize(size = 12) %>%  # Reduce font size
  padding(padding = 0) %>%
  font(fontname = "Times New Roman", part = "all") %>%
  set_table_properties(layout="autofit") %>% # Reduce padding inside cells
  autofit() %>%
  theme_apa()
```

```
assign_L2_gor <- webgazeR::assign_aoi(dat_colnames_L2,X="x", Y="y",aoi_loc =
↪  aois_L2)
```

Modeling data

Once the data have been preprocessed, the next step is analysis. A variety of analytic approaches are available for VWP data, including growth curve analysis (GCA), cluster permutation analysis (CPA), generalized additive mixed models (GAMMs), logistic multilevel models, and divergent point analysis (DPA). Fortunately, there is a wealth of excellent resources and tutorials demonstrating how to apply these methods

Figure 8

Gorilla provided standardized coordinates for the four quadrants on the screen

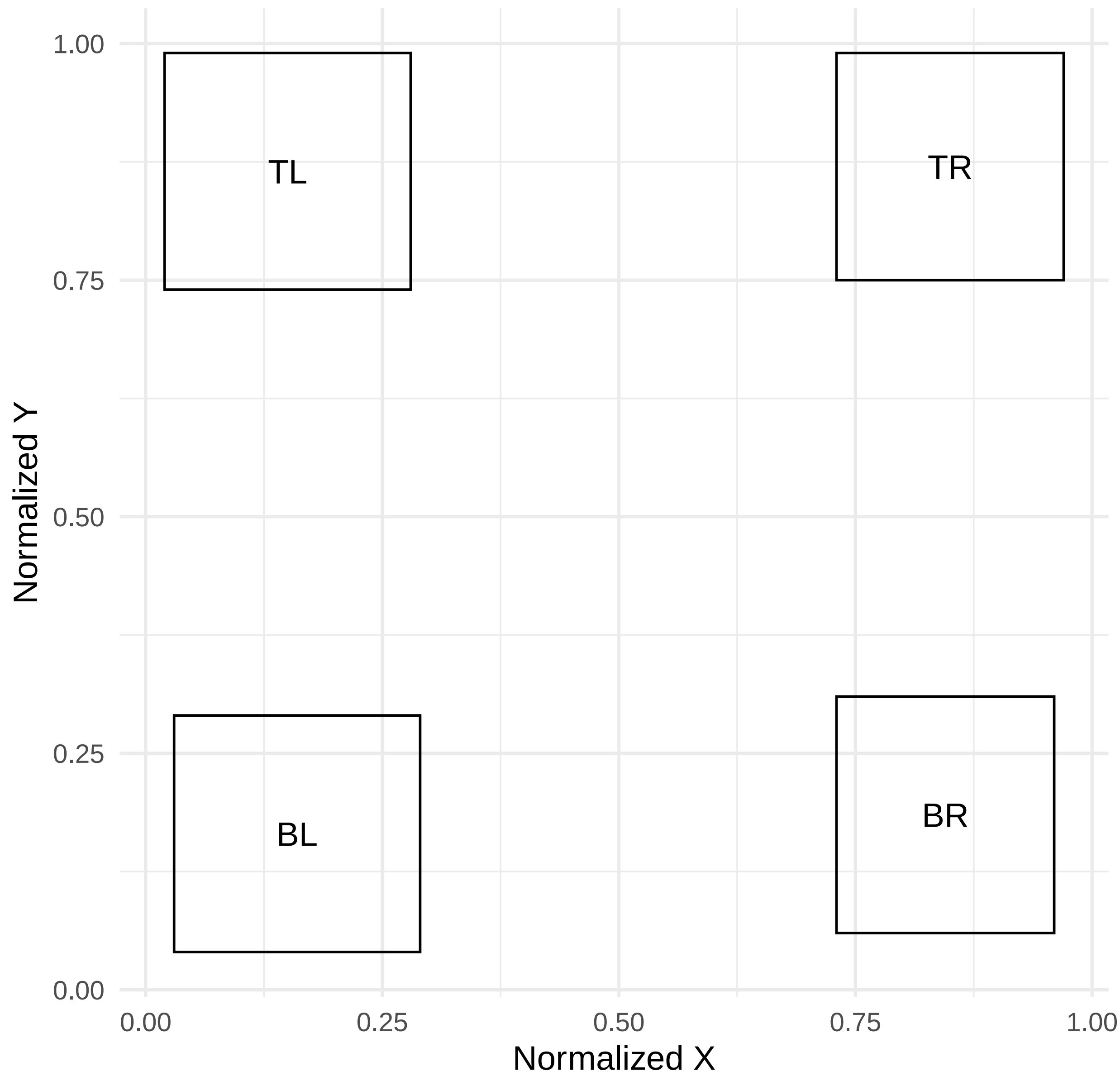


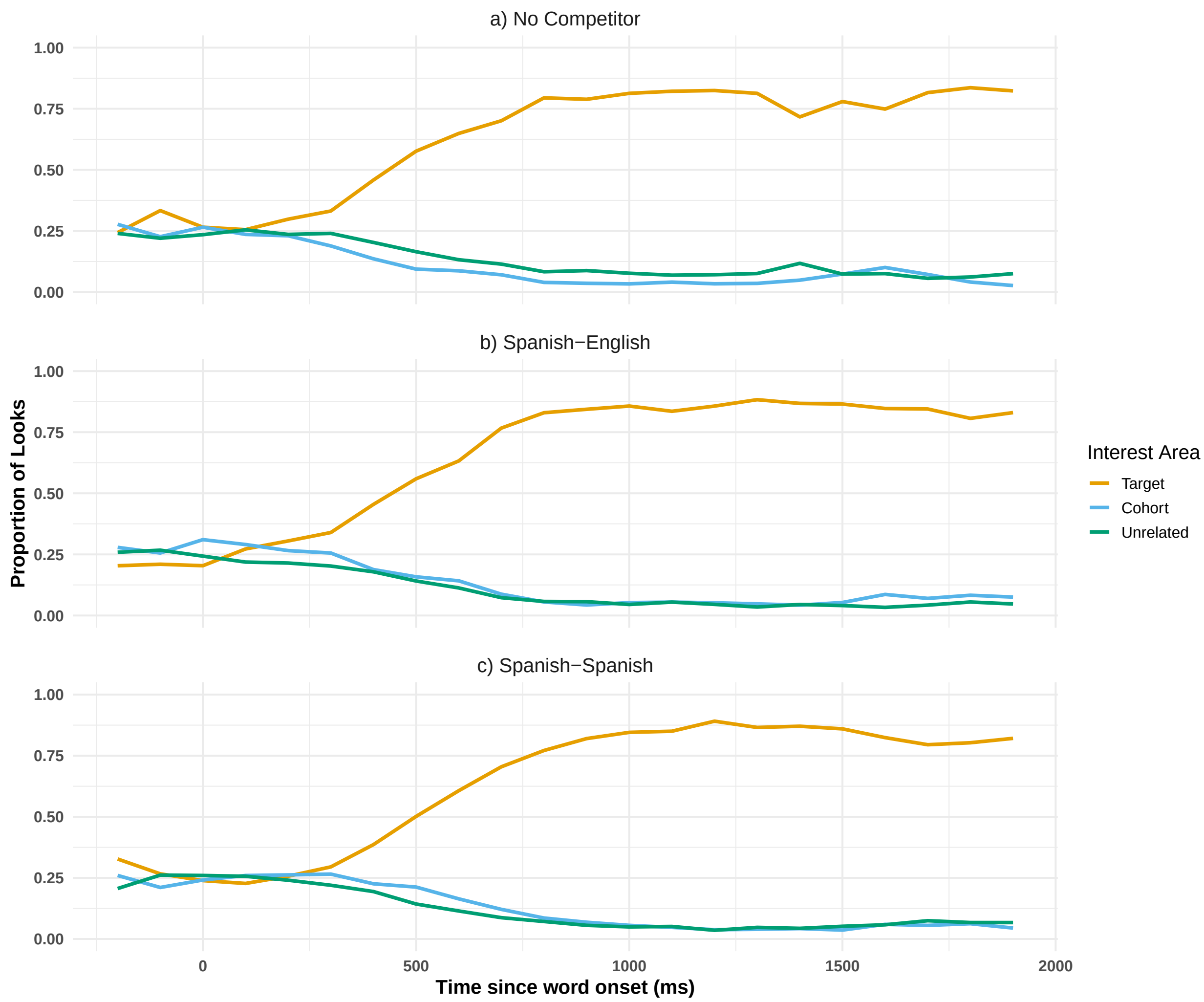
Table 6

Gorilla provided standardized gaze coordinates

loc	x_normalized	y_normalized	width_normalized	height_normalized	xmin	ymin	xmax	ymax
BL	0.03	0.04	0.26	0.25	0.03	0.04	0.29	0.29
TL	0.02	0.74	0.26	0.25	0.02	0.74	0.28	0.99
TR	0.73	0.75	0.24	0.24	0.73	0.75	0.97	0.99
BR	0.73	0.06	0.23	0.25	0.73	0.06	0.96	0.31

Figure 9

Comparison of competition effects with Gorilla standardized coordinates



to both lab-based (Coretta & Casillas, 2024; see Ito & Knoeferle, 2023; Mirman & CRC Press., n.d.; Seedorff et al., 2018; Stone et al., 2021) and online (see Bramlett & Wiener, 2024) visual world eye-tracking data.

This paper's goal, however, is to not evaluate different analytic approaches and tell readers what they should use. All methods have their strengths and weaknesses (see Ito & Knoeferle, 2023). Nevertheless, statistical modeling should be guided by the questions researchers have and thus serious thought needs to be given to the proper analysis. In the VWP, there are two general questions one might be interested in: (1) Are there any overall difference in fixations between conditions and (2) Are there any time course differences in fixations between conditions (and/or groups).

With our data, one question we might want to answer is if there are any fixation differences between the cohort and unrelated conditions across the time course. One statistical approach we chose to highlight to answer this question is a cluster permutation analysis (CPA). The CPA is suitable for testing differences between two conditions or groups over an interest period while controlling for multiple comparisons and autocorrelation. Given the time latency issues common in webcam-basted studies, Slim et al. (2024) recommended using an approach like CPA.

CPA

CPA is a technique that has become increasingly popular, particularly in the field of cognitive neuropsychology, for analyzing MEG and EEG data (Maris & Oostenveld, 2007). While its adoption in VWP studies has been relatively slow, it is now beginning to appear more frequently (see Huang & Snedeker, 2020; Ito & Knoeferle, 2023). Notably, its use is growing in online eye-tracking studies (see Slim et al., 2024; Slim & Hartsuiker, 2023; Vos et al., 2022).

Before we show you how to apply this method to the current dataset, we want to briefly explain what CPA is. The CPA is a data-driven approach that increases statistical power while controlling for Type I errors across multiple comparisons—exactly what we need when analyzing fixations across the time course.

The clustering procedure involves three main steps:

1. Cluster Formation: With our data, a multilevel logistic model is conducted for every data point (condition by time). Please note that any statistical test can be run here. Adjacent data points that surpass the mass univariate significance threshold (e.g., $p < .05$) are combined into clusters. The cluster-level statistic, typically the sum of the t-values (or F-values) within the cluster, is computed labeled as SumStatitic is output below). By clustering adjacent significant data points, this step accounts for autocorrelation by considering temporal dependencies rather than treating each data point as independent.

2. Null Distribution Creation: Next, the same analysis is run as in step 1. However, the analysis is based on randomly permuting or shuffling the conditions within subjects. This principle of exchangeability is important here, as it suggests that the condition labels can be exchanged without altering the underlying data structure. This randomization is repeated n times (e.g., 1000 shuffles), and for each permutation,

Table 7

Clustermass statistics for the Spanish-Spanish condition

cluster	cluster_mass	p.cluster_mass	bin_start	bin_end	t	sign	time_start	time_end
1	236.34	0	7	13	5.48	1	500	1,100

the cluster-level statistic is computed. This step addresses the issue of multiple comparisons by constructing a distribution of cluster-level statistics under the null hypothesis, providing a baseline against which observed cluster statistics can be compared. By doing so, the method controls the family-wise error rate and ensures that significant findings are not simply due to chance.

3. Significance Testing: The cluster-level statistics from the observed (real) comparison is compared to the null distribution we created above Clusters with statistics falling in the highest or lowest 2.5% of the null distribution are considered significant (e.g., p < 0.05).

To perform CPA, we will load in the `permutes` (Voeten, 2023), `permuco` (Frossard & Renaud, 2021), `foreach` (& Weston, 2022), and `Parallel` (Corporation & Weston, 2022) packages in R. Loading these packages allow us to use the `cluster.glmer()`function to run a cluster permutation (10,000 rimes) across multiple system cores to speed up the process. We run a CPA on the `gaze_sub_id` object where each row in `Looks` denotes whether the AOI was fixated, with values of zero (not fixated) or one (fixated).

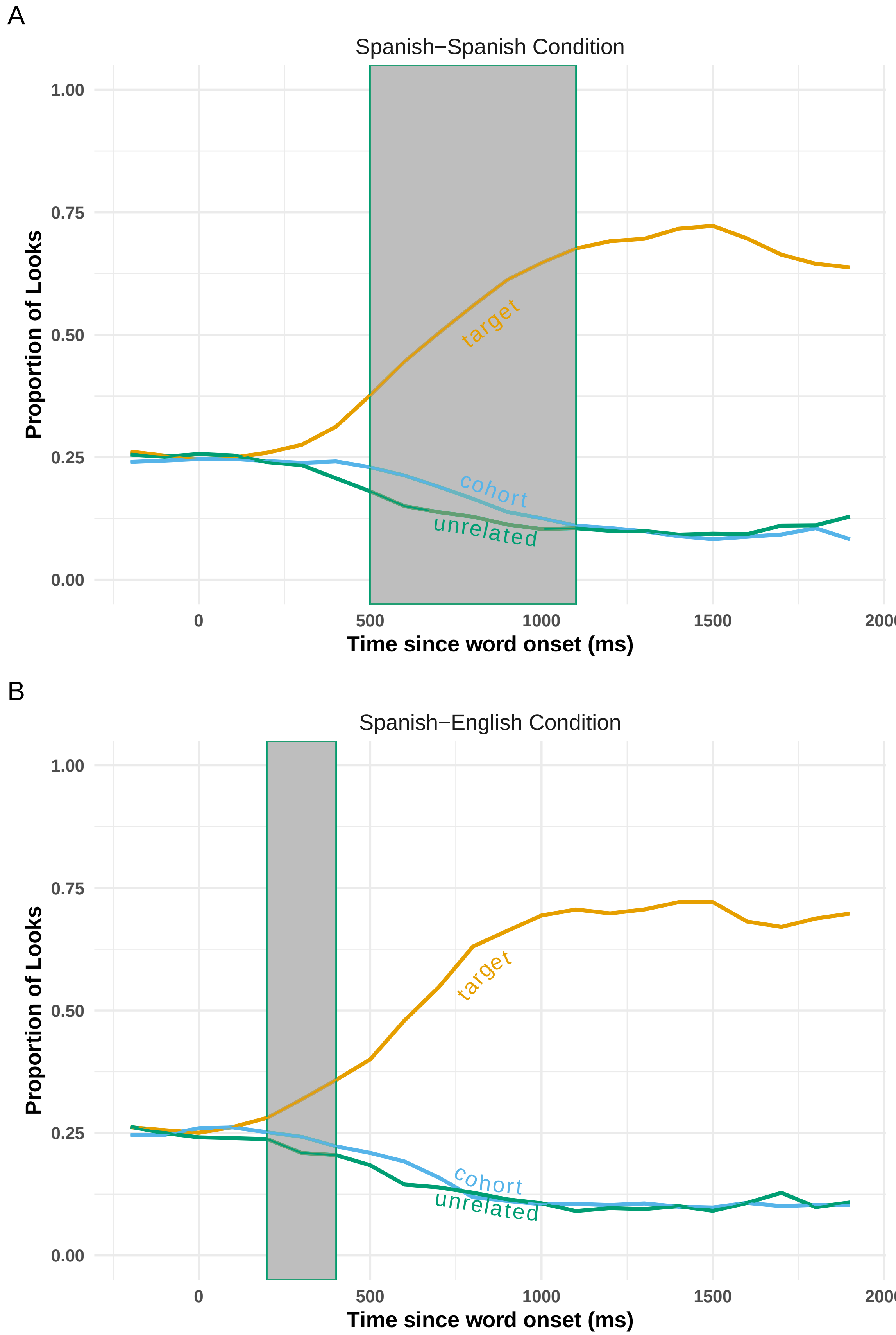
Below you find sample code to perform multilevel CPA in R (please see the Github repository for elaborated code needed to perform CPA.

```
library(permutes) # cpa
library(permuco) # cpa

total_perms <- 1000

cpa.lme <-  permutes::clusterperm.glmer(Looks~ condition1_code + (1|subject) +
↪  (1|trial), data=gaze_sub_L2_cp1, series.var=~time_bin, nperm = total_perms)
```

In the analysis for the Spanish-Spanish condition, one significant cluster was observed between 500 and 1,100 ms, as indicated in the summary statistics from Table 7. The positive `SumStatistic` value associated with this cluster suggests that competition was greater during this time window. This result implies that cohorts in the Spanish-Spanish condition exhibited stronger effects or competition compared to unrelated items. In Figure 10 significant clusters are highlighted for both the Spanish-Spanish and Spanish-English conditions. Both conditions show one significant cluster. Overall, the analysis suggests that both the Spanish-Spanish and Spanish-English conditions demonstrate significant competitor effects.

Figure 10

Average looks in the cross-linguistic VWP task over time for the Spanish-Spanish condition (a) and the Spanish-English condition (b). The shaded rectangles indicate when cohort looks were greater than chance based on the CPA.



Effect size. It is important to address the issue of effect sizes in the context of CPA. Calculating effect sizes for CPA is not straightforward, as the technique is designed to evaluate temporal clusters rather than individual time points. (Slim et al., 2024; but also see Meyer et al., 2021) outline three possible approaches for estimating effect sizes in CPA: (1) computing the effect size within a predefined time window (often the same window used for identifying clusters), (2) calculating an average effect size across the entire cluster, and (3) reporting the maximum effect observed within the cluster. Each method has its trade-offs in terms of interpretability and comparability across studies, and the choice should be guided by theoretical considerations and the research question at hand.

Discussion

Webcam eye-tracking is a relatively nascent technology, and as such, there is limited guidance available for researchers. To ameliorate this, we created a tutorial to assist new users of visual world webcam eye-tracking, using some of the best practices available (e.g., Bramlett & Wiener, 2024). To further facilitate this process, we created the `webgazeR` package, which contains several helper functions designed to streamline data preprocessing, analysis, and visualization.

In this tutorial, we covered the basic steps of running a visual world webcam-based eye-tracking experiment. We highlighted these steps by using data from a cross-linguistic VWP looking at competitive processes in L2 speakers of Spanish. Specifically, we attempted to replicate the experiment by Sarrett et al. (2022) where they observed within- and between L2/L1 competition using carefully crafted materials.

Replication of Sarrett et al. (2022)

While the main purpose of this tutorial was to highlight the steps needed to analyze webcam eye-tracking data, replicating Sarrett et al. (2022) allowed us to not only assess whether within and between L2/L1 competition can be found in a spoken word recognition VWP experiment online, but also provide insight in how to run VWP studies online and the issues associated with it.

Our conceptual replication yielded highly encouraging results, revealing robust competition effects both within-language (Spanish-Spanish) and across-language (Spanish-English) conditions—closely mirroring those reported by Sarrett et al. (2022). However, several key analytic, methodological, and sample differences between our study and theirs warrant discussion.

A major analytic difference lies in how the time course of competition was examined. While Sarrett et al. (2022) employed a non-linear curve-fitting approach (see McMurray et al., 2010), we used cluster-based permutation analysis (CPA). This methodological distinction limits direct comparisons regarding the temporal dynamics of competition. Nonetheless, the overall time course patterns align surprisingly well: our CPA identified a significant cluster starting at 500 ms, while Sarrett et al. (2022) observed effects beginning around 400 ms—suggesting a modest delay of approximately 100 ms in our online data. This delay is still markedly smaller than in previous webcam-based studies (e.g., Semmelmann & Weigelt, 2018; Slim et al., 2024), reflecting progress in online eye-tracking. That said, it's important to note that CPA is not ideally suited for making precise temporal inferences about onset or offset of effects (Fields & Kuperberg, 2019; Ito & Knoeferle, 2023).

Design differences between the studies also play a critical role. In Sarrett et al. (2022), participants previewed the images in each quadrant for 1000 ms, followed by the appearance of a central red dot they clicked to trigger audio playback. After selecting the target, a 250 ms inter-trial interval (ITI) preceded the next trial.

In contrast, our sequence began with a 500 ms fixation cross (serving as the ITI), followed by a longer 1500 ms preview. The images then disappeared, and participants clicked a centrally placed start button to initiate audio playback, at which point the images reappeared. Upon target selection, the next trial began immediately. We also imposed a 5-second timeout for non-responses. Additionally, our study included 250 trials—fewer than the 450 in the original study[2]—but still more than most webcam-based research. Despite the reduced trial count, we observed parallel competition effects in both language conditions, underscoring the robustness of the findings.

Several motivations guided these design adaptations. Online testing introduces greater variability in participants' setups (e.g., device type, connection quality), so we opted for a longer preview period to enhance the likelihood of observing competition effects. Prior work suggests this can boost competition signals in the VWP (Apfelbaum et al., 2021). The start-button mechanism ensured trials began from a centralized gaze position, helping minimize quadrant-based bias. Finally, the timeout feature helped mitigate issues of inattention common in unsupervised online environments.

Participant recruitment also differed. Sarrett et al. (2022) recruited students from a Spanish language course and assessed proficiency using the LexTALE-Spanish test (Izura et al., 2014). Our participants were recruited through Prolific with more limited screening, allowing us only to filter by native language and reported experience with another language. This constraint likely contributed to differences in language profiles between samples. Whereas Sarrett et al. (2022) included L2 learners with verified proficiency, our sample encompassed a broader and more variable group of L2 speakers, with limited verification of language skills (see Table 1 for details). This broader variability may help explain the absence of a sustained cohort competition effect in our study.

In sum, while there are notable differences in methods and samples, the convergence of competition effects across both studies—within and across languages—supports the robustness of these phenomena across diverse research contexts. Still, we view these results as a promising step rather than definitive evidence. A more systematic investigation is needed to fully establish the generalizability of these effects.

[2] The curve-fitting approach used by Sarrett et al. (2022) may have required a larger number of trials to obtain reliable fits. Their study included over 400 trials, while our design was more constrained.

Table 8

Eye-tracking questionnaire items

Question
1.Do you have a history of vision problems (e.g., corrected vision, eye disease, or drooping eyelids)?
2.Are you on any medications currently that can impair your judgement?
If yes, please list below:
4.Does your room currently have natural light?
5.Are you using the built in camera?
If no, what brand of camera are you using?
6.Please estimate how far you think you were sitting from the camera during the experiment (an arm's length from your monitor is about 20 inches (51 cm).
7.Approximately how many times did you look at your phone during the experiment?
8.Approximately how many times did you get up during the experiment?
9.Was the environment you took the experiment in distraction free?
10. When you had to calibrate, were the instructions clear?
11. What additional information would you add to help make things easier to understand?
12. Are you wearing a mask?

Limitations

Recruitment of L2 Speakers

In this study, we used the Prolific platform to recruit L2 Spanish speakers. We specified criteria requiring participants to be native English speakers who were also proficient in Spanish, reside in the United States, and be between the ages of 18 and 36. These criteria yielded a potential recruitment pool of approximately 1,000 participants. While this number is larger than what is typically available for in-lab studies, it is still relatively limited given the overall size of the platform. Notably, English native speakers who are L2 learners of Spanish in the U.S. are not usually considered a particularly niche population, which highlights the extent of the recruitment difficulty. Participant pools are likely to be even more limited when targeting speakers of less commonly studied languages or with specific language backgrounds (e.g. heritage speakers). Moreover, Prolific currently supports only an English user interface, which makes it harder to recruit non-English speakers (Niedermann et al., 2024; Patterson & Nicklin, 2023). For second language research in particular, researchers should be aware of these and other constraints (such as the limited filtering options to control for proficiency) and consider incorporating language background questionnaires and/or proficiency tasks directly into the study design. Ultimately, 181 participants signed up for the study, and recruitment proved to be more challenging than expected. Researchers considering similar studies should be aware of these limitations when targeting niche populations, even on large online platforms. Despite these challenges, the final sample was sufficient for our planned analyses and opened up the possibility to target populations

Table 9

Responses to eye-tracking questions for participants who successfully calibrated (good) vs. participants who had trouble calibrating (bad)

Question	Response	Good	Bad
1.Do you have a history of vision problems (e.g., corrected vision, eye disease, or drooping eyelids)?	No	65.71	64.29
1.Do you have a history of vision problems (e.g., corrected vision, eye disease, or drooping eyelids)?	Yes	34.29	35.71
2.Are you on any medications currently that can impair your judgement?	No	100.00	98.21
2.Are you on any medications currently that can impair your judgement?	Yes	0.00	1.79
4.Does your room currently have natural light?	No	40.00	26.79
4.Does your room currently have natural light?	Yes	60.00	73.21
5.Are you using the built in camera?	No	14.29	8.93
5.Are you using the built in camera?	Yes	85.71	91.07
9.Was the environment you took the experiment in distraction free?	No	11.43	3.57
9.Was the environment you took the experiment in distraction free?	Yes	88.57	96.43

you would be unable to capture otherwise.

Generalizability to other platforms

We demonstrated how to analyze webcam eye-tracking data collected via the Gorilla platform using WebGazer.js. Although we did not validate this pipeline on other platforms that support WebGazer.js—such as PCIbex (Zehr & Schwarz, 2018), jsPsych (Leeuw, 2015), or PsychoPy (Peirce et al., 2019)—we believe the pipeline is generalizable to these and to platforms that use other gaze estimation logarithms, such as Labvanced (Kaduk et al., 2024). To support broader compatibility, the functions in the webgazeR package are designed to work with a variety of file types—including .csv, .tsv, and .xlsx – and work with any dataset that includes five essential columns: subject, trial, x, y, and time. We also provide a helper function, `make_webgazer()`, to assist in renaming columns so your dataset can be adapted to the expected format.

We encourage researchers to test this pipeline in their own studies and report any issues or suggestions on our GitHub repository. We are committed to improving `webgazeR` and welcome feedback that will make the package more flexible, user-friendly, and adaptable to a wider range of experimental platforms.

Power

While we successfully demonstrated competition effects similar to Sarrett's study, we did not conduct an a priori power analysis nor was it our intention. With webcam eye-tracking, it has been recommended running twice the number of participants from the original sample, or powering the study to detect an effect size half as large as the original (Slim & Hartsuiker, 2023; Van der Cruyssen et al., 2024). We did attempt to increase our sample size 2x, but were unable to recruit enough participants through Prolific. However, our sample size is similar to the lab based study. Regardless, researchers should be aware of this and plan accordingly.

We strongly urge researchers to perform power analyses and justify their sample sizes (Lakens, 2022). While tools like G*Power (Faul et al., 2007) are available for this purpose, we recommend power simulations using Monte Carlo or resampling methods on pilot or sample data (see Prystauka et al., 2024; Slim & Hartsuiker, 2023). Several excellent R packages, such as `mixedpower` (Kumle et al., 2021) and `SIMR` (Green & MacLeod, 2016) make such simulations straightforward and accessible.

Recommendations and ways forward

While our findings support the promise of webcam eye-tracking for language research, several challenges remain that researchers should consider. One of the most significant issues is data loss due to poor calibration. In our study, we excluded approximately 75% of participants due to calibration failure. These attrition rates are in line with some previous reports (e.g., Slim & Hartsuiker, 2023), though others have found substantially lower rates (Bramlett & Wiener, 2025; Prystauka et al., 2024). With this valuation, it is important to understand the factors that lead to better quality data.

To address this, we included a post-task questionnaire assessing participants' setups and their experiences with the experiment. These questions, included in Table 8, provide insights that informed the following recommendations, which we also base on our experimental design and personal experience.

In our experimental design, participants were branched based on whether they successfully completed the experiment or failed calibration at any point. Table 9 highlights the comparisons between good and poor calibrators. For the sake of brevity, we will discuss some recommendations based on questionnaire responses and personal experience that will hopefully improve research using webcam eye-tracking.

Prioritize external webcams

Our data suggest that participants using external webcams were significantly more likely to complete the calibration successfully than those using built-in laptop cameras. External webcams typically offer higher resolution and frame rates—both critical for accurate gaze estimation (Slim & Hartsuiker, 2023) Researchers should, whenever possible, encourage participants to use external webcams and may consider administering a brief pre-experiment questionnaire to screen for webcam type and exclude low-quality setups.

Optimize environmental conditions

Poor calibration was often reported in environments with natural light. Ambient lighting introduces variability that can degrade tracking performance. We recommend that researchers instruct participants to complete studies in rooms with consistent artificial lighting and minimal glare or shadows.

In addition to lighting, head movement and distance from the screen are critical for achieving reliable eye-tracking. Excessive movement or leaning in and out of the camera's view can disrupt the face mesh tracking used by WebGazer.js. Participants should be advised to remain still and maintain a consistent, moderate distance from the screen—approximately 50–70 cm, depending on their camera setup. We asked individuals to provide an approximate distance from their screens, (arms length) but it is not clear how accurate this is. Providing clear guidance (e.g., via an instructional video) may help mitigate these issues and improve overall tracking fidelity).

A different platform, Labvanced (Kaduk et al., 2024), for example, offers additional eye-tracking functionality including a virtual chinrest to ensure head movement is restricted to an acceptable range and warns users if they deviate from this range. Together this might make for a better eye-tracking experience with less data thrown out. This should be investigated further.

Conduct a priori power analysis

To ensure adequate statistical power, researchers should conduct a priori power analyses either via GUI like GPower or perform Monte Carlo simulations/resampling on pilot data. This step is particularly important for online studies, where sample variability can be higher than in controlled lab environments. To this point, you will have to over-enroll your study due to the high attrition rate to reach your target goal, so please plan accordingly.

Collect detailed post-experiment feedback

Gathering detailed feedback about participants' setups—such as webcam type, browser, lighting conditions, and perceived ease of use—can provide valuable information about what contributes to successful calibration. These insights can inform more effective participant instructions and refined inclusion criteria for future studies.

By implementing these strategies, researchers can improve the quality and consistency of data collected through webcam-based eye-tracking. These recommendations aim to maximize the utility and reproducibility of remote eye-tracking research, particularly in language processing contexts.

Conclusions

This work highlights the steps required to process webcam eye-tracking data, demonstrating the potential of webcam-based eye-tracking for robust psycholinguistic experimentation. By providing a standardized pipeline for processing eye-tracking data, we aim to give researchers a clear and practical path for collecting and analyzing visual world webcam eye-tracking data. An interactive demo of the preprocessing pipeline—using data from a monolingual VWP—is available at the webgazeR website (https://jgeller112.github.io/webgazeR/vignettes/webgazeR_vignette.html), where users can explore the code and workflow firsthand.

Moreover, our findings demonstrate the feasibility of conducting high-quality online experiments, paving the way for future research to address more nuanced questions about L2 processing and language comprehension more broadly. Additionally, further refinement of webcam eye-tracking methodologies could enhance data precision and extend their applicability to more complex experimental designs. This is an exciting time for eye-tracking research, with its boundaries continuously expanding. We eagerly anticipate the advancements and possibilities that the future of webcam eye-tracking will bring.

References

Allaire, J. J., Teague, C., Scheidegger, C., Xie, Y., Dervieux, C., & Woodhull, G. (2024). *Quarto* (Version 1.6) [Computer software]. https://doi.org/10.5281/zenodo.5960048

Allopenna, P. D., Magnuson, J. S., & Tanenhaus, M. K. (1998). *Tracking the time c ourse of spoken word recognition using eye movements: Evidence for c ontinuous mapping models* (pp. 419–439).

Altmann, G. T. M., & Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, *73*(3), 247–264. https://doi.org/10.1016/S0010-0277(99)00059-1

Anderson, C. A., Allen, J. J., Plante, C., Quigley-McBride, A., Lovett, A., & Rokkum, J. N. (2019). The MTurkification of Social and Personality Psychology. *Personality & Social Psychology Bulletin*, *45*(6), 842–850. https://doi.org/10.1177/0146167218798821

Anwyl-Irvine, A. L., Massonnié, J., Flitton, A., Kirkham, N., & Evershed, J. K. (2020). Gorilla in our midst: An online behavioral experiment builder. *Behavior Research Methods*, *52*(1), 388–407. https://doi.org/10.3758/s13428-019-01237-x

Apfelbaum, K. S., Klein-Packard, J., & McMurray, B. (2021). The pictures who shall not be named: Empirical support for benefits of preview in the visual world paradigm. *Journal of Memory and Language*, *121*, 104279. https://doi.org/10.1016/j.jml.2021.104279

Barrett, M. (2021). *Ggokabeito: 'Okabe-ito' scales for 'ggplot2' and 'ggraph'*. https://CRAN.R-project.org/package=ggokabeito

Bianco, R., Mills, G., Kerangal, M. de, Rosen, S., & Chait, M. (2021). Reward enhances online participants' engagement with a demanding auditory task. *Trends in Hearing*, *25*, 23312165211025941. https://doi.org/10.1177/23312165211025941

Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., & Ertl, T. (2017). Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum*, *36*(8), 260–284. https://doi.org/10.1111/cgf.13079

Blasi, D. E., Henrich, J., Adamou, E., Kemmerer, D., & Majid, A. (2022). Over-reliance on english hinders cognitive science. *Trends in Cognitive Sciences*, *26*(12), 1153–1170. https://doi.org/10.1016/j.tics.2022.09.015

Bogdan, P. C., Dolcos, S., Buetti, S., Lleras, A., & Dolcos, F. (2024). Investigating the suitability of online eye tracking for psychological research: Evidence from comparisons with in-person data using emotion–attention interaction tasks. *Behavior Research Methods*, *56*(3), 2213–2226. https://doi.org/10.3758/s13428-023-02143-z

Boxtel, W. S. van, Linge, M., Manning, R., Haven, L. N., & Lee, J. (2024). Online eye tracking for aphasia: A feasibility study comparing web and lab tracking and implications for clinical use. *Brain and Behavior*, *14*(11), e70112. https://doi.org/10.1002/brb3.70112

Bramlett, A. A., & Wiener, S. (2024). The art of wrangling. *Linguistic Approaches to Bilingualism*. https://doi.org/https://doi.org/10.1075/lab.23071.bra

Bramlett, A. A., & Wiener, S. (2025). Individual differences modulate prediction of Italian words based on lexical stress: a close replication and LASSO extension of Sulpizio and McQueen (2012). *Journal of Cultural Cognitive Science*, *9*(1), 55–81. https://doi.org/10.1007/s41809-024-00162-6

Brysbaert, M., & Stevens, M. (2018). Power analysis and effect size in mixed effects models: A tutorial. *Journal of Cognition*, *1*(1). https://doi.org/10.5334/joc.10

Bylund, E., Khafif, Z., & Berghoff, R. (2024). Linguistic and geographic diversity in research on second language acquisition and multilingualism: An analysis of selected journals. *Applied Linguistics*, *45*(2), 308–329. https://doi.org/10.1093/applin/amad022

Carter, B. T., & Luke, S. G. (2020). Best practices in eye tracking research. *International Journal of Psychophysiology*, *155*, 49–62. https://doi.org/10.1016/j.ijpsycho.2020.05.010

Chen-Sankey, J., Elhabashy, M., Gratale, S., Geller, J., Mercincavage, M., Strasser, A. A., Delnevo, C. D., Jeong, M., & Wackowski, O. A. (2023). Examining Visual Attention to Tobacco Marketing Materials Among Young Adult Smokers: Protocol for a Remote Webcam-Based Eye-Tracking Experiment. *JMIR Research Protocols*, *12*, e43512. https://doi.org/10.2196/43512

Colby, S. E., & McMurray, B. (2023). Efficiency of spoken word recognition slows across the adult lifespan. *Cognition*, *240*, 105588. https://doi.org/10.1016/j.cognition.2023.105588

Cooper, R. M. (1974). The control of eye fixation by the meaning of spoken language: A new methodology for the real-time investigation of speech perception, memory, and language processing. *Cognitive Psychology*, *6*(1), 84–107. https://doi.org/10.1016/0010-0285(74)90005-X

Coretta, S., & Casillas, J. V. (2024). A tutorial on generalised additive mixed effects models for bilingualism research. *Linguistic Approaches to Bilingualism*. https://doi.org/10.1075/lab.23076.cor

Corporation, M., & Weston, S. (2022). *doParallel: Foreach parallel adaptor for the 'parallel' package*. https://CRAN.R-project.org/package=doParallel

Csárdi, G., Hester, J., Wickham, H., Chang, W., Morgan, M., & Tenenbaum, D. (2024). *Remotes: R package installation from remote repositories, including 'GitHub'*. https://CRAN.R-project.org/package=remotes

Dahan, D., Magnuson, J. S., & Tanenhaus, M. K. (2001). Time course of frequency effects in spoken-word recognition: Evidence from eye movements. *Cognitive Psychology*, *42*(4), 317–367. https://doi.org/10.1006/cogp.2001.0750

Degen, J., Kursat, L., & Leigh, D. D. (2021). Seeing is believing: Testing an explicit linking assumption for visual world eye-tracking in psycholinguistics. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *43*.

Dolstra, E., & contributors, T. N. (2023). *Nix* (Version 2.15.3) [Computer software]. https://nixos.org/

Eberhard, K. M., Spivey-Knowlton, M. J., Sedivy, J. C., & Tanenhaus, M. K. (1995). Eye movements as a window into real-time spoken language comprehension in natural contexts. *Journal of Psycholinguistic Research*, *24*(6), 409–436. https://doi.org/10.1007/BF02143160

Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, *39*(2), 175–191. https://doi.org/10.3758/BF03193146

Fields, E. C., & Kuperberg, G. R. (2019). Having your cake and eating it too: Flexibility and power with mass univariate statistics for ERP data. *Psychophysiology*. https://doi.org/10.1111/psyp.13468

Firke, S. (2023). *Janitor: Simple tools for examining and cleaning dirty data*. https://CRAN.R-project.org/package=janitor

Frossard, J., & Renaud, O. (2021). *Permutation tests for regression, {ANOVA}, and comparison of signals: The {permuco} package*. *99*. https://doi.org/10.18637/jss.v099.i15

Geller, J., & Prystauka, Y. (2024). *webgazeR: Tools for processing webcam eye tracking data*. https://github.com/jgeller112/webgazeR

Godfroid, A., Finch, B., & Koh, J. (2024). Reporting Eye-Tracking Research in Second Language Acquisition and Bilingualism: A Synthesis and Field-Specific Guidelines. *Language Learning*, *n/a*(n/a). https://doi.org/10.1111/lang.12664

Gosling, S. D., Sandy, C. J., John, O. P., & Potter, J. (2010). Wired but not WEIRD: The promise of the Internet in reaching more diverse samples. *Behavioral and Brain Sciences*, *33*(2-3), 94–95. https://doi.org/10.1017/S0140525X10000300

Green, P., & MacLeod, C. J. (2016). SIMR: an R package for power analysis of generalized linear mixed models by simulation. *Methods in Ecology and Evolution*, *7*(4), 493–498. https://doi.org/10.1111/2041-210X.12504

Henrich, J., Heine, S. J., & Norenzayan, A. (2010). Most people are not WEIRD. *Nature*, *466*(7302), 29.

https://doi.org/10.1038/466029a

Hooge, I. T. C., Hessels, R. S., Niehorster, D. C., Andersson, R., Skrok, M. K., Konklewski, R., Stremplewski, P., Nowakowski, M., Tamborski, S., Szkulmowska, A., Szkulmowski, M., & Nyström, M. (2024). Eye tracker calibration: How well can humans refixate a target? *Behavior Research Methods*, *57*(1), 23. https://doi.org/10.3758/s13428-024-02564-4

Hopp, H. (2013). Grammatical gender in adult L2 acquisition: Relations between lexical and syntactic variability. *Second Language Research*, *29*(1), 33–56. https://doi.org/10.1177/0267658312461803

Huang, Y., & Snedeker, J. (2020). Evidence from the visual world paradigm raises questions about unaccusativity and growth curve analyses. *Cognition*, *200*, 104251. https://doi.org/10.1016/j.cognition.2020.104251

Huettig, F., & McQueen, J. M. (2007). The tug of war between phonological, semantic and shape information in language-mediated visual search. *Journal of Memory and Language*, *57*(4), 460–482. https://doi.org/10.1016/j.jml.2007.02.001

Huettig, F., Rommers, J., & Meyer, A. S. (2011). Using the visual world paradigm to study language processing: a review and critical evaluation. *Acta Psychologica*, *137*(2), 151–171. https://doi.org/10.1016/j.actpsy.2010.11.003

Ito, A., & Knoeferle, P. (2023). Analysing data from the psycholinguistic visual-world paradigm: Comparison of different analysis methods. *Behavior Research Methods*, *55*(7), 3461–3493. https://doi.org/10.3758/s13428-022-01969-3

Ito, A., Pickering, M. J., & Corley, M. (2018). Investigating the time-course of phonological prediction in native and non-native speakers of english: A visual world eye-tracking study. *Journal of Memory and Language*, *98*, 1–11. https://doi.org/10.1016/j.jml.2017.09.002

Izura, C., Cuetos, F., & Brysbaert, M. (2014). Lextale-Esp: a test to rapidly and efficiently assess the Spanish vocabulary size. *PSICOLOGICA*, *35*(1), 49–66. http://hdl.handle.net/1854/LU-5774107

Ju, M., & Luce, P. A. (2004). Falling on sensitive ears: Constraints on bilingual lexical activation. *Psychological Science*, *15*(5), 314–318. https://doi.org/10.1111/j.0956-7976.2004.00675.x

Kaduk, T., Goeke, C., Finger, H., & König, P. (2024). Webcam eye tracking close to laboratory standards: Comparing a new webcam-based system and the EyeLink 1000. *Behavior Research Methods*, *56*(5), 5002–5022. https://doi.org/10.3758/s13428-023-02237-8

Kamide, Y., Altmann, G. T. M., & Haywood, S. L. (2003). The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye movements. *Journal of Memory and Language*, *49*(1), 133–156. https://doi.org/10.1016/S0749-596X(03)00023-8

Kret, M. E., & Sjak-Shie, E. E. (2018). Preprocessing pupil size data: Guidelines and code. *Behavior Research Methods*, 1–7. https://doi.org/10.3758/s13428-018-1075-y

Kumle, L., Võ, M. L.-H., & Draschkow, D. (2021). Estimating power in (generalized) linear mixed models: An open introduction and tutorial in R. *Behavior Research Methods*, *53*(6), 2528–2543. https://doi.org/10.3758/s13428-021-01546-0

Lakens, D. (2022). Sample size justification. *Collabra: Psychology*, *8*(1). https://doi.org/10.1525/collabra.33267

Leeuw, J. R. de. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser.

Behavior Research Methods, *47*(1), 1–12. https://doi.org/10.3758/s13428-014-0458-y

Madsen, J., Júlio, S. U., Gucik, P. J., Steinberg, R., & Parra, L. C. (2021). Synchronized eye movements predict test scores in online video education. *Proceedings of the National Academy of Sciences*, *118*(5), e2016980118. https://doi.org/10.1073/pnas.2016980118

Magnuson, J. S., Dixon, J. A., Tanenhaus, M. K., & Aslin, R. N. (2007). The Dynamics of Lexical Competition During Spoken Word Recognition. *Cognitive Science*, *31*(1), 133–156. https://doi.org/10.1080/03640210709336987

Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, *164*(1), 177–190. https://doi.org/10.1016/j.jneumeth.2007.03.024

McMurray, B., Farris-Trimble, A., & Rigler, H. (2017). Waiting for lexical access: Cochlear implants or severely degraded input lead listeners to process speech less incrementally. *Cognition*, *169*, 147–164. https://doi.org/10.1016/j.cognition.2017.08.013

McMurray, B., Samelson, V. M., Lee, S. H., & Tomblin, J. B. (2010). Individual differences in online spoken word recognition: Implications for SLI. *Cognitive Psychology*, *60*(1), 1–39. https://doi.org/10.1016/j.cogpsych.2009.06.003

McMurray, B., Tanenhaus, M. K., & Aslin, R. N. (2002). Gradient effects of within-category phonetic variation on lexical access. *Cognition*, *86*(2), B33–B42. https://doi.org/10.1016/S0010-0277(02)00157-9

Meyer, M., Lamers, D., Kayhan, E., Hunnius, S., & Oostenveld, R. (2021). Enhancing reproducibility in developmental EEG research: BIDS, cluster-based permutation tests, and effect sizes. *Developmental Cognitive Neuroscience*, *52*, 101036. https://doi.org/10.1016/j.dcn.2021.101036

Microsoft, & Weston, S. (2022). *Foreach: Provides foreach looping construct*. https://CRAN.R-project.org/package=foreach

Miller, J. (2023). Outlier exclusion procedures for reaction time analysis: The cures are generally worse than the disease. *Journal of Experimental Psychology: General*, *152*(11), 3189–3217. https://doi.org/10.1037/xge0001450

Milne, A. E., Zhao, S., Tampakaki, C., Bury, G., & Chait, M. (2021). Sustained pupil responses are modulated by predictability of auditory sequences. *The Journal of Neuroscience*, *41*(28), 6116–6127. https://doi.org/10.1523/JNEUROSCI.2879-20.2021

Mirman, D., & CRC Press. (n.d.). *Growth curve analysis and visualization using r*.

Mirman, D., & Graziano, K. M. (2012). Individual differences in the strength of taxonomic versus thematic relations. *Journal of Experimental Psychology: General*, *141*(4), 601–609. https://doi.org/10.1037/a0026451

Müller, K. (2020). *Here: A simpler way to find your files*. https://CRAN.R-project.org/package=here

Niedermann, J. P., Sucholutsky, I., Marjieh, R., Çelen, E., Griffiths, T., Jacoby, N., & Rijn, P. van. (2024). Studying the Effect of Globalization on Color Perception using Multilingual Online Recruitment and Large Language Models. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *46*(0). https://escholarship.org/uc/item/4hs755zz

Özsoy, O., Çiçek, B., Özal, Z., Gagarina, N., & Sekerina, I. A. (2023). Turkish-german heritage speakers' predictive use of case: Webcam-based vs. In-lab eye-tracking. *Frontiers in Psychology*, *14*, 1155585. https://doi.org/10.3389/fpsyg.2023.1155585

Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). *Webgazer: Scalable webcam eye tracking using user interactions*. 38393845.

Patterson, A. S., & Nicklin, C. (2023). L2 self-paced reading data collection across three contexts: In-person, online, and crowdsourcing. *Research Methods in Applied Linguistics*, *2*(1), 100045. https://doi.org/10.1016/j.rmal.2023.100045

Peelle, J. E., & Van Engen, K. J. (2021). Time stand still: Effects of temporal window selection on eye tracking analysis. *Collabra: Psychology*, *7*(1), 25961. https://doi.org/10.1525/collabra.25961

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*(1), 195–203. https://doi.org/10.3758/s13428-018-01193-y

Peterson, R. J. (2021). We need to address ableism in science. *Molecular Biology of the Cell*, *32*(7), 507–510. https://doi.org/10.1091/mbc.E20-09-0616

Płużyczka, M. (2018). The First Hundred Years: a History of Eye Tracking as a Research Method. *Applied Linguistics Papers*, *25/4*, 101–116. http://cejsh.icm.edu.pl/cejsh/element/bwmeta1.element.desklight-98576d43-39e3-4981-8c1c-717962cf29da

Prystauka, Y., Altmann, G. T. M., & Rothman, J. (2024). Online eye tracking and real-time sentence processing: On opportunities and efficacy for capturing psycholinguistic effects of different magnitudes and diversity. *Behavior Research Methods*, *56*(4), 3504–3522. https://doi.org/10.3758/s13428-023-02176-4

R Core Team. (2024). *R: A language and environment for statistical computing* (Version 4.4.2). R Foundation for Statistical Computing. https://www.R-project.org/

Rodd, J. M. (2024). Moving experimental psychology online: How to obtain high quality data when we can't see our participants. *Journal of Memory and Language*, *134*, 104472. https://doi.org/10.1016/j.jml.2023.104472

Rodrigues, B., & Baumann, P. (2025). *Rix: Reproducible data science environments with 'nix'*. https://docs.ropensci.org/rix/

Rossi, E., Krass, K., & Kootstra, G. J. (2019). *Psycholinguistic Methods in Multilingual Research* (pp. 75–99). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119387725.ch4

Sarrett, M. E., Shea, C., & McMurray, B. (2022). Within- and between-language competition in adult second language learners: Implications for language proficiency. *Language, Cognition and Neuroscience*, *37*(2), 165–181. https://doi.org/10.1080/23273798.2021.1952283

Seedorff, M., Oleson, J., & McMurray, B. (2018). Detecting when timeseries differ: Using the bootstrapped differences of timeseries (BDOTS) to analyze visual world paradigm data (and more). *Journal of Memory and Language*, *102*, 55–67. https://doi.org/10.1016/J.JML.2018.05.004

Semmelmann, K., & Weigelt, S. (2018). Online webcam-based eye tracking in cognitive science: A first look. *Behavior Research Methods*, *50*(2), 451–465. https://doi.org/10.3758/s13428-017-0913-7

Slim, M. S., & Hartsuiker, R. J. (2023). Moving visual world experiments online? A web-based replication of Dijkgraaf, Hartsuiker, and Duyck (2017) using PCIbex and WebGazer.js. *Behavior Research Methods*, *55*(7), 3786–3804. https://doi.org/10.3758/s13428-022-01989-z

Slim, M. S., Kandel, M., Yacovone, A., & Snedeker, J. (2024). Webcams as windows to the mind? A direct comparison between in-lab and web-based eye-tracking methods. *Open Mind*, *8*, 1369–1424. https:

//doi.org/10.1162/opmi_a_00171

Spivey, M. J., & Marian, V. (1999). Cross talk between native and second languages: Partial activation of an irrelevant lexicon. *Psychological Science*, *10*(3), 281–284. https://doi.org/10.1111/1467-9280.00151

Stone, K., Lago, S., & Schad, D. J. (2021). Divergence point analyses of visual world data: applications to bilingual research. *Bilingualism: Language and Cognition*, *24*(5), 833–841. https://doi.org/10.1017/S1366728920000607

Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K. M., & Sedivy, J. C. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science (New York, N.Y.)*, *268*(5217), 1632–1634. http://www.ncbi.nlm.nih.gov/pubmed/7777863

Trueswell, J. C. (2008). *Using eye movements as a developmental measure within psycholinguistics* (I. A. Sekerina, E. M. Fernández, & H. Clahsen, Eds.; pp. 73–96). John Benjamins Publishing Company. https://doi.org/10.1075/lald.44.05tru

Van der Cruyssen, I., Ben-Shakhar, G., Pertzov, Y., Guy, N., Cabooter, Q., Gunschera, L. J., & Verschuere, B. (2024). The validation of online webcam-based eye-tracking: The replication of the cascade effect, the novelty preference, and the visual world paradigm. *Behavior Research Methods*, *56*(5), 4836–4849. https://doi.org/10.3758/s13428-023-02221-2

Viviani, P. (1990). Eye movements in visual search: cognitive, perceptual and motor control aspects. *Reviews of Oculomotor Research*, *4*, 353–393.

Voeten, C. C. (2023). *Permutes: Permutation tests for time series data*. https://CRAN.R-project.org/package=permutes

Vos, M., Minor, S., & Ramchand, G. C. (2022). Comparing infrared and webcam eye tracking in the Visual World Paradigm. *Glossa Psycholinguistics*, *1*(1). https://doi.org/10.5070/G6011131

Wickham, H. (2017). *Tidyverse: Easily install and load the 'tidyverse'*. https://CRAN.R-project.org/package=tidyverse

Yee, E., Blumstein, S., & Sedivy, J. C. (2008). Lexical-semantic activation in broca's and wernicke's aphasia: Evidence from eye movements. *Journal of Cognitive Neuroscience*, *20*(4), 592–612. https://doi.org/10.1162/jocn.2008.20056

Zehr, J., & Schwarz, F. (2018). *PennController for internet based experiments (IBEX)*. https://doi.org/10.17605/OSF.IO/MD832