

¹ Language Without Borders: A Step-by-Step Guide to Analyzing
² Webcam Eye-Tracking Data for L2 Research

³ Jason Geller¹, Yanina Prystauka², Sarah E. Colby³, and Julia R. Drouin⁴

⁴ ¹Department of Psychology and Neuroscience, Boston College

⁵ ²Department of Linguistic, Literary and Aesthetic Studies, University of Bergen

⁶ ³Department of Linguistics, University of Ottawa

⁷ ⁴Division of Speech and Hearing Sciences, University of North Carolina at Chapel Hill

⁸ 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

¹ 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—² sometimes necessitating anesthesia—or the attachment of suction cups to the sclera of the eyes (Płużyczka,³ 2018). These methods were not only cumbersome for researchers, but also uncomfortable and invasive for⁴

5 participants. Over time, such approaches have been replaced by non-invasive, lightweight, and user-friendly
6 systems. Today, modern eye-tracking technology is widely accessible in laboratories worldwide, enabling
7 researchers to tackle critical questions about cognitive processes. This evolution has had a profound impact
8 on fields such as psycholinguistics and bilingualism, opening up new possibilities for understanding how
9 language is processed in real time (Godfroid et al., 2024).

10 **In the last decade, there has been a gradual shift towards conducting more behavioral exper-**
11 **iments online (Anderson et al., 2019; Rodd, 2024). This “onlineification” of behavioral research has**
12 **driven the development of remote eye-tracking methods that do not rely on traditional laboratory set-**
13 **tings. Allowing participants to use their own equipment from anywhere in the world opens the door**
14 **to recruiting more diverse and historically underrepresented populations [@gosling2010]. Behavioral**
15 **research has long struggled with a lack of diverse and representative samples, relying heavily on par-**
16 **ticipants who are predominantly Western, Educated, Industrialized, Rich, and Democratic (WEIRD)**
17 **(Henrich et al., 2010). Additionally, we propose adding able-bodied to this acronym (WEIRD-A) (Pe-**
18 **terson, 2021), to highlight the exclusion of individuals with disabilities who may face barriers to ac-**
19 **cessing research facilities. In language research, this issue is especially pronounced, as studies often**
20 **focus on “modal” listeners and speakers—typically young, monolingual, and neurotypical (Blasi et al.,**
21 **2022; Bylund et al., 2024; McMurray et al., 2010).**

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

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

34 This paper is divided into three parts. First, we introduce automated webcam-based eye-
35 tracking. Second, we review the viability of conducting VWP studies using online eye-tracking meth-
36 ods. Third, we present a detailed tutorial for analyzing webcam-based VWP data with the `webgazeR`
37 package, using our replication experiment to highlight the steps needed for preprocessing.

38 **Webcam eye-tracking with WebGazer.js**

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

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

50 A popular tool for enabling webcam-based eye-tracking is WebGazer.js (Papoutsaki et al.,
51 2016)¹, an open-source, freely available, and actively maintained JavaScript library. WebGazer.js has al-
52 ready been integrated into several popular experimental platforms, including Gorilla, jsPsych, PsychoPy,
53 Labvanced, and PCIbex (Anwyl-Irvine et al., 2020; Kaduk et al., 2024; Leeuw, 2015; Peirce et al., 2019;
54 Zehr & Schwarz, 2018). Because WebGazer.js runs locally on the participant's machine, it does not store we-
55 bcam video recordings, helping alleviate ethical and privacy concerns associated with online eye-tracking.**

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

62 **Eye-tracking in the lab vs. online**

63 Several studies in psychology and psycholinguistics have evaluated the viability of WebGazer.js
64 for online research. Generally, lab-based effects can be successfully replicated in online environments
65 using WebGazer.js (Bogdan et al., 2024; Bramlett & Wiener, 2024, 2025; Özsoy et al., 2023; Prystauka
66 et al., 2024; Slim et al., 2024; Slim & Hartsuiker, 2023; Van der Cruyssen et al., 2024; Vos et al., 2022).

¹It 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.

67 However, a critical finding across online replication studies is that effect sizes are often smaller and
68 more variable than those observed in laboratory settings (Bogdan et al., 2024; Slim et al., 2024; Slim
69 & Hartsuiker, 2023; Van der Cruyssen et al., 2024).

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

77 There are also notable issues with the spatial and temporal accuracy of webcam-based eye-
78 tracking using WebGazer.js. Spatial precision is often lower, with average errors frequently exceeding
79 1° of visual angle (Papoutsaki et al., 2016). Temporal delays are also substantially larger, ranging from
80 200 ms to over 1000 ms (Semmelmann & Weigelt, 2018; Slim et al., 2024; Slim & Hartsuiker, 2023).
81 Additionally, recent work by Bogdan et al. (2024) has documented a systematic bias in gaze estimates
82 favoring centrally located stimuli.

83 Bringing the visual world paradigm (VWP) online

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

87 In the field of language research, few methods have had as enduring an impact as the VWP.
88 Over the past 25 years, the VWP has enabled researchers to address a broad range of topics, including
89 sentence processing (Altmann & Kamide, 1999; Huettig et al., 2011; Kamide et al., 2003), spoken word
90 recognition (Allopenna et al., 1998; Dahan et al., 2001; Huettig & McQueen, 2007; McMurray et al.,
91 2002), bilingual language processing (Hopp, 2013; Ito et al., 2018; Rossi et al., 2019), the effects of
92 brain damage on language (Mirman & Graziano, 2012; Yee et al., 2008), and the impact of hearing
93 loss on lexical access [McMurray et al., 2017].

94 What makes the widespread use of the VWP even more remarkable is the simplicity of the
95 task. In a typical VWP experiment, participants view a display of several objects represented by a
96 picture. As they listen to a spoken word or phrase, their eye movements are recorded in real time.
97 Please note that different versions of this task exist, and implementations may vary slightly across
98 studies, depending on specific research goals and design choices. Despite this, a robust finding in VWP
99 research is that listeners reliably direct their gaze to the picture representing the spoken word, often
100 before the word has been fully articulated, revealing anticipatory or predictive processing.

101 While eye movements are often time-locked to linguistic input, the relationship between
102 eye movements and lexical processing is not one-to-one. Lexical activation interacts with non-

103 lexical factors such as selective attention, visual salience, task demands, working memory, and prior
104 expectations—all of which can shape where and when participants look (Bramlett & Wiener, 2025;
105 Eberhard et al., 1995; Huettig et al., 2011; Kamide et al., 2003). Nonetheless, the VWP remains a
106 powerful and flexible tool for studying online language processing, offering fine-grained insights into
107 how linguistic and cognitive processes unfold moment by moment.

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

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

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

127 Importantly, temporal issues are not observed in every case. Work has begun to address many
128 of these challenges by leveraging updated versions of WebGazer.js and adopting different experimen-
129 tal platforms. For instance, Vos et al. (2022) reported a substantial reduction in temporal delays—
130 approximately 50 ms—when using a newer version of WebGazer.js embedded within the jsPsych
131 framework (Leeuw, 2015). Similarly, studies by Prystauka et al. (2024) and Bramlett and Wiener
132 (2024), which utilized the Gorilla Experiment Builder in combination with the improved WebGazer
133 algorithm, found timing and competition effects closely aligned with those observed in traditional lab-
134 based VWP studies.

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

141 **Bilingual competition: A visual world webcam eye-tracking replication**

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

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

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

- 161 1. **Spanish-Spanish (within) condition:** A Spanish competitor was presented alongside the target
162 word. For example, if the target word spoken was *cielo* (sky), the Spanish competitor was *ciencia*
163 (science).
- 164 2. **Spanish-English (cross-linguistic) condition:** An English competitor was presented for the Spanish
165 target word. For example, if the target word spoken was *botas* (boots), the English competitor
166 was *border*.

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

172 There are two key differences between our dataset and the original study by Sarrett et al. (2022)
173 worth noting. First, Sarrett et al. (2022) focused on adult unbalanced L2 Spanish speakers and posed
174 more fine-grained questions about the time course of competition and resolution and its relationship
175 with L2 language acquisition. Second, unlike Sarrett et al. (2022), who measured Spanish proficiency

176 objectively using LexTALE-esp (Izura et al., 2014)) and ran this study using participants from a Span-
177 ish college course, we relied on participant filtering on Prolific (www.prolific.co) to recruit L2 Spanish
178 speakers.

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

183 *Tutorial Overview*

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

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

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

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

209 To promote transparency and reproducibility, all analyses were conducted in R (R Core Team,
210 2024) using Quarto (Allaire et al., 2024), an open-source publishing system that enables dynamic and
211 reproducible documents. Figures, tables, and text are generated programmatically and embedded di-

212 **rectly in the manuscript, ensuring seamless integration of results. To further enhance computational**
213 **reproducibility, we employed the `rix` package (Rodrigues & Baumann, 2025), which leverages the Nix**
214 **ecosystem (Dolstra & contributors, 2023). This approach captures not only the R package versions**
215 **but also system dependencies at runtime. Researchers can reproduce the exact computational envi-**
216 **ronment by installing the Nix package manager and using the provided `default.nix` file. Detailed**
217 **setup instructions are included in the README file of the accompanying GitHub repository. A video**
218 **tutorial is also provided.**

219 **Method**

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

222 **Participants**

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

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

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

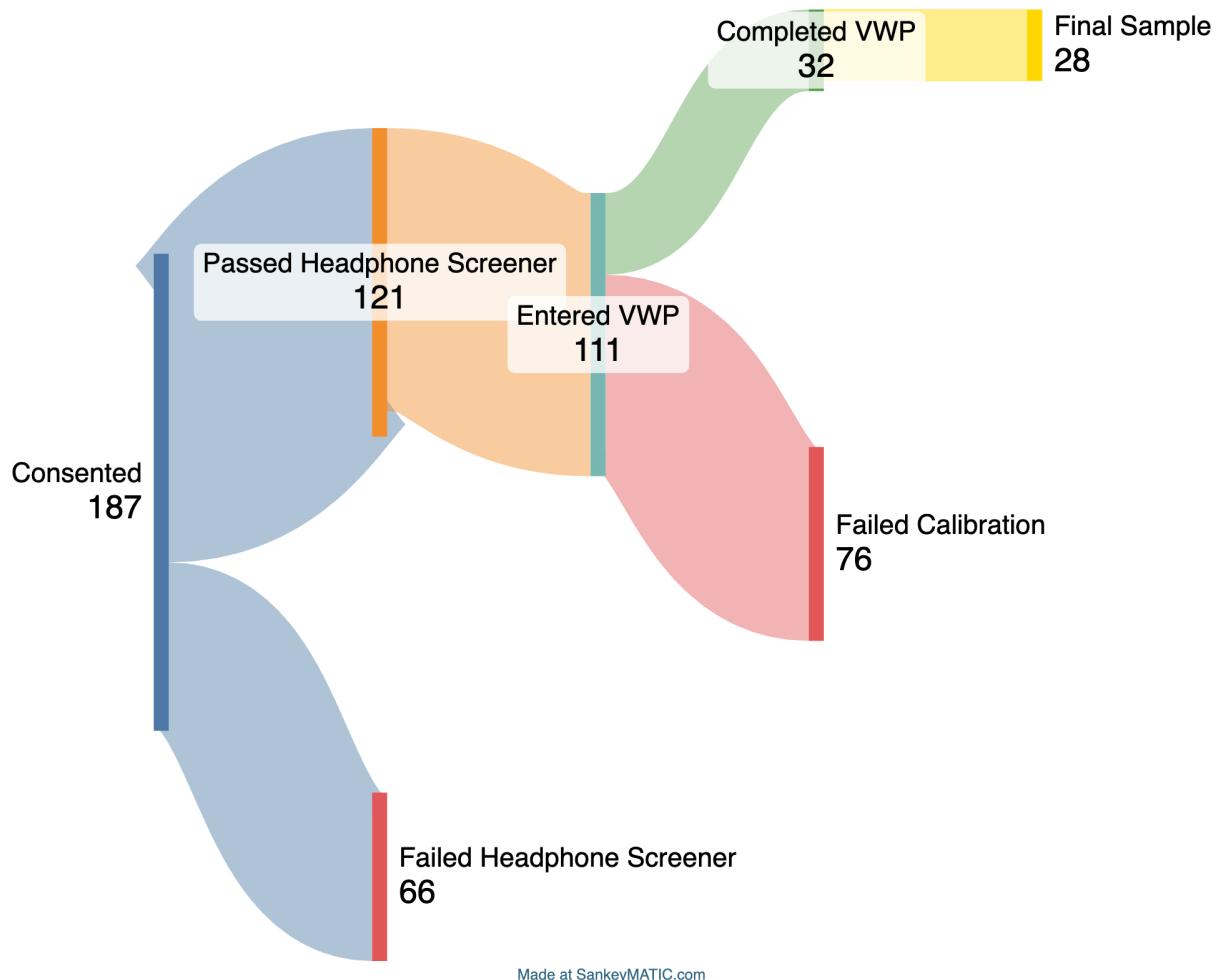
²⁴⁸ information for the full sample that made it to the final survey is provided in Table 1.

```
#| 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."



²⁴⁹ **Materials**

²⁵⁰ **VWP..**

Table 1*Participant demographic variables*

Characteristic	N = 91¹
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)

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

255 The item sets consisted of two types of phonologically-related word pairs: one pair of Spanish-
256 Spanish words and another of Spanish-English words. The Spanish-Spanish pairs were unrelated to the
257 Spanish-English pairs. All the word pairs were carefully controlled on a number of dimensions (see Sarrett
258 et al., 2022). There were three experimental conditions: (1) the Spanish-Spanish (within) condition, where
259 one of the Spanish words was the target and the other was the competitor; (2) the Spanish-English (cross-
260 linguistic) condition, where a Spanish word was the target and its English phonological cohort served as the
261 competitor; and (3) the No Competitor condition, where the Spanish word did not overlap with any other
262 word in the set. The Spanish-Spanish condition had twice as many trials as the other conditions due to the
263 interchangeable nature of the target and competitor words in that pair.

264 **Each item within a set appeared four times as the target word, resulting in a total of 240 trials**
265 (**15 sets × 4 items per set × 4 repetitions**). **Each set included one Spanish–Spanish cohort pair and one**
266 **Spanish–English cohort pair. In the Spanish–Spanish condition, both words in the pair served as mu-**
267 **tual competitors—for example, *cielo* activated *ciencia*, and vice versa. This bidirectional relationship**
268 **yielded 120 trials for the Spanish–Spanish condition.**

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

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

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

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

289 ceiling when wearing headphones but poor when listening in the free field (due to phase cancellation).

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

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

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

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

309 ***Procedure***

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

315 **Only personal computers were permitted for participation. Upon entering the study from Pro-**
316 **lific, participants were presented with a consent form. Once consent was given, participants completed**
317 **a headphone screening test. They had three attempts to pass this test. If unsuccessful by the third at-**
318 **tempt, participants were directed to an early exit screen, followed by the questionnaire. They had three**
319 **attempts to pass this test. If unsuccessful by the third attempt, participants were directed to an early**
320 **exit screen, followed by the questionnaire.**

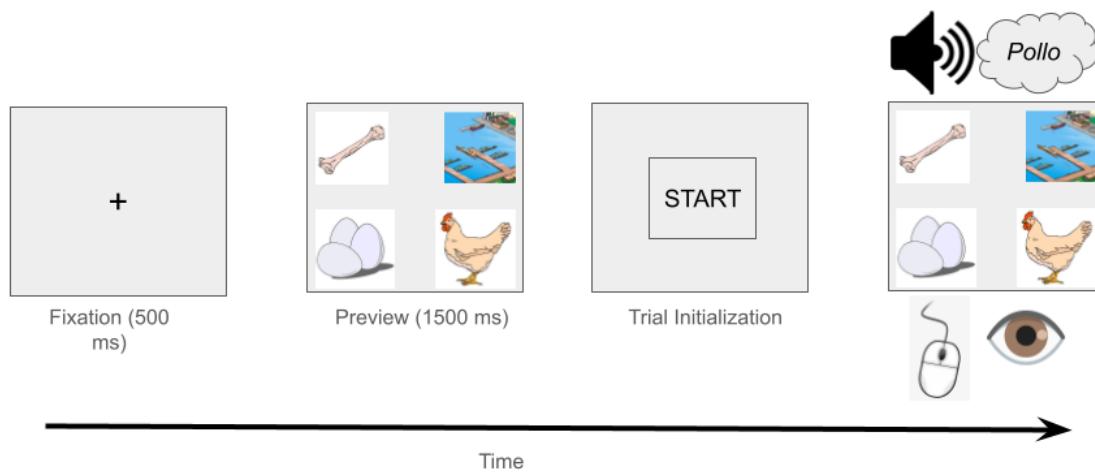
321 **If the headphone screener was passed, participants were next introduced to the VWP task.** This
322 began with instructional videos providing specific guidance on the ideal experiment setup for eye-tracking
323 and calibration procedures. You can view the videos here: <https://osf.io/mgkd2/>. Participants were then
324 required to enter full-screen mode before calibration. A 9-point calibration procedure was used. Calibration

325 occurred every 60 trials for a total of 3 calibrations. Participants had three attempts to successfully complete
 326 each calibration phase. If calibration was unsuccessful, participants were directed to an early exit screen,
 327 followed by the questionnaire.

328 In the main VWP task, each trial began with a 500 ms fixation cross at the center of the screen. This
 329 was followed by a preview screen displaying four images, each positioned in a corner of the screen. After
 330 1500 ms, a start button appeared in the center. Participants clicked the button to confirm they were focused
 331 on the center before the audio played. Once clicked, the audio was played, and the images remained visible.
 332 Participants were instructed to click the image that best matched the spoken target word, while their eye
 333 movements were recorded. Eye movements were only recorded on that screen. Figure 2 displays the VWP
 334 trial sequence.

Figure 2

VWP trial schematic



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

338 Preprocessing data

339 After the data is collected you can begin preprocessing your data. Below we highlight the steps
 340 needed to preprocess your webcam eye-tracking data and get it ready for analysis. For some of this prepro-
 341 cessing we will use the newly created `webgazeR` package (**v. 0.7.1**).

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

- 343 1. Reading in data
- 344 2. Data exclusion
- 345 3. Combining trial- and eye-level data
- 346 4. Assigning areas of interest (AOIs)
- 347 5. Time binning
- 348 1. Downsampling
- 349 2. Upsampling (optional)
- 350 6. Aggregating (optional)
- 351 7. Visualization (optional)

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

354 ***Load packages***

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

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

```
library(remotes) # install github repo

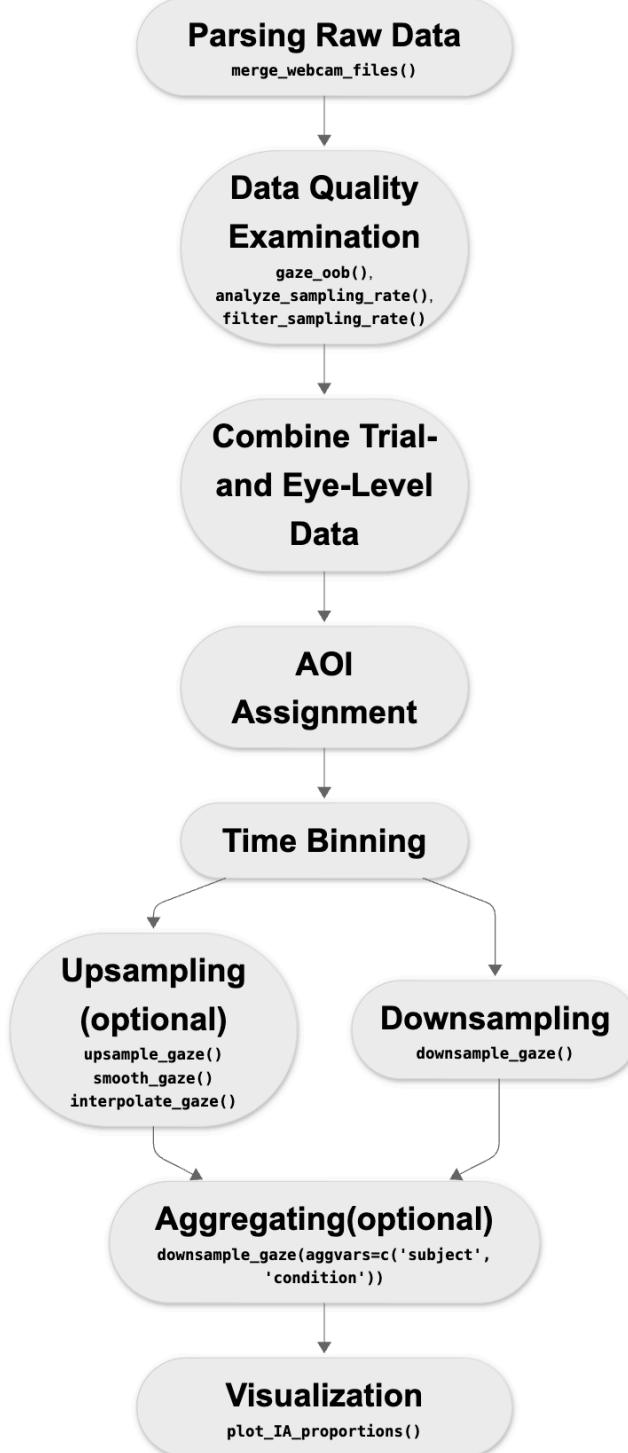
remotes::install_github("jgeller112/webgazeR")
```

361 Once this is installed, `webgazeR` can be loaded along with additional useful packages. The following
362 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
```

Figure 3

Preprocessing steps for webcam eye-tracking data using webgazeR functions



```

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

```

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

365 ***Reading in data***

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

372 The .csv files contain meta-data for each trial, such as what picture were presented on each
 373 trial, which object was the target, reaction times, audio presentation times, what object was clicked on, etc.
 374 To load our data files into our R environment, we use the `here` (Müller, 2020) package to set a relative
 375 rather than an absolute path to our files. We read in the data files from the repository for both versions of
 376 the task and merge the files together. `L2_data` merges both `data_exp_196386-v5_task-scf6.csv` and
 377 `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

```

378 **Eye-tracking data.** Gorilla currently saves each participant's eye-tracking data on a per-trial ba-
 379 sis. The raw subfolder in the project repository contains the eye-tracking files by participant for each trial

380 individually (~/data/L2/raw). Contained in those files, we have information pertaining to each trial such as
 381 participant id, time since trial started, x and y coordinates of looks, convergence (the model’s confidence
 382 in finding a face (and accurately predicting eye movements), face confidence (represents the support vector
 383 machine (SVM) classifier score for the face model fit), and information pertaining to the the AOI screen
 384 coordinates (standardized and user-specific). The `vwp_files_L2` object below contains a list of all the files
 385 contained in the folder. Because `vwp_files_L2` contains trial data as well as calibration data, we remove
 386 the calibration trials and save the non-calibration to to `vwp_paths_filtered_L2`.

```
# 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))
```

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

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

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

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

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

```
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")
```

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

415 **Successful calibration is critical for reliable eye-tracking measurements, as poor calibration**
 416 **directly compromises the spatial accuracy of gaze data (Blascheck et al., 2017). Requiring a sufficient**
 417 **number of completed trials is crucial for ensuring adequate statistical power and stable individual-level**
 418 **parameter estimates, particularly in tasks with high trial-to-trial variability (Brysbaert & Stevens,**
 419 **2018). We choose 100 trials as this meant participants passed at least two calibration attempts dur-**
 420 **ing the study. Behavioral accuracy (>= 80%) was used as an additional screening measure because**
 421 **low task performance may indicate a lack of attention, misunderstanding of the task, or random re-**
 422 **sponding, all of which could undermine both the behavioral and eye-movement data quality (Bianco**
 423 **et al., 2021). Filtering based on sampling rate ensures that datasets with too few gaze samples (due to**
 424 **technical or environmental issues) are removed, as low sampling rates significantly degrade temporal**
 425 **precision and bias gaze metrics (semmlmann2018?). Finally, we excluded participants with excessive**
 426 **off-screen data (>30%) because this indicates poor gaze tracking, likely caused by head movement,**
 427 **poor lighting, or loss of face detection. At this time, there is no set guide on what constitutes accept-**
 428 **able data loss for webcam-based studies. We felt 30% was a reasonable cut-off. At the trial-level, we**
 429 **also removed incorrect trials and trials where sampling rate was < 5 Hz.**

430 **What we will do first is create a cleaned up version of our behavioral, trial-level data L2_data**
 431 **by creating an object named eye_behav_L2 that selects useful columns from that file and renames**
 432 **stimuli to make them more intuitive. Because most of this will be user-specific, no function is called**
 433 **here. Below we describe the preprocessing done on the behavioral data file. The below code pro-**
 434 **cesses and transforms the L2_data dataset into a cleaned and structured format for further analy-**
 435 **sis. First, the code renames several columns for easier access using janitor::clean_names() (Firke,**
 436 **2023) function. We then select only the columns we need and filter the dataset to include only rows**
 437 **where screen_name is “VWP” and zone_type is called “response_button_image”, representing the**
 438 **picture selected for that trial. Afterward, the function renames additional columns (tlpic to TL, trpic**
 439 **to TR, etc.). We also renamed participant_private_id to subject, spreadsheet_row to trial, and**
 440 **reaction_time to RT. This makes our columns consistent with the edat_L2 above for merging later**

441 **on. Lastly, reaction time (RT) is converted to a numeric format for further numerical analysis.**

442 It is important to note here that what the behavioral spreadsheet denotes as trial is not in fact the trial
443 number used in the eye-tracking files. Thus it is imperative you use `spreadsheet_row` as trial number to
444 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))
```

445 **Audio onset.** Because we are playing audio on each trial and running this experiment from the
 446 browser, audio onset is never going to be consistent across participants. In Gorilla there is an option to
 447 collect advanced audio features (you must make sure you select this when designing the study) such as when
 448 the audio play was requested, played, and ended. We will want to incorporate this timing information into
 449 our analysis pipeline. Gorilla records the onset of the audio which varies by participant. We are extracting
 450 that in the `audio_rt_L2` object by filtering `zone_type` to `content_web_audio` and a response equal to
 451 “AUDIO PLAY EVENT FIRED”. This will tell us when the audio was triggered in the experiment. We are
 452 creating a column called (`RT_audio`) which we will use later on to correct for audio delays. Please note
 453 that on some trials the audio may not play. This is a function of the browser a participant is using and the
 454 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,
 455 make sure you try and request this information, or at the very least acknowledge audio delay.
 456

```
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))
```

457 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"))
```

458 **Trial removal.** As stated above, participants who did not successfully calibrate 3 times or less were
 459 rejected from the experiment. Deciding to remove trials is ultimately up to the researcher. In our case, we
 460 removed participants with less than 100 trials. Let’s take a look at how many participants meet this criterion

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

⁴⁶¹ by probing the trial_data_rt_L2 object. In Table 2 we can see several participants failed some of the cal-
⁴⁶²ibration attempts and do not have an adequate number of trials. Again we make no strong recommendations
⁴⁶³here. 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)))
```

⁴⁶⁴ Let's remove them participants with less than 100 trials from the analysis using the below code.

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

465 **Low accuracy.** In our experiment, we want to make sure accuracy is high (> 80%). Again, we want
 466 participants that are fully attentive in the experiment. In the below code, we keep participants with accuracy
 467 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
```

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

471 **Sampling rate.** While most commercial eye-trackers sample at a constant rate, data captured by
 472 webcams are widely inconsistent. Below is some code to calculate the sampling rate of each participant.
 473 Ideally, you should not have a sampling rate less than 5 Hz. It has been recommended you drop those
 474 values (Bramlett & Wiener, 2024) The below function analyze_sample_rate() calculates the sampling
 475 rate for each subject and each trial in our eye-tracking dataset (edat_L2). The analyze_sample_rate()
 476 function provides overall statistics, including the option to report mean or median (Bramlett & Wiener, 2024)
 477 sampling rate and standard deviation of sampling rates in your experiment. **Sampling rate calculations**
 478 **followed standard procedures (e.g., Bramlett & Wiener, 2024; Prystauka et al., 2024).** The function
 479 also generates a histogram of sampling rates by-subject. Looking at Figure 4, the sampling rate ranges from
 480 5 to 35 Hz with a median sampling rate of 21.56. This corresponds to previous webcam eye-tracking work
 481 (e.g., Bramlett & Wiener, 2024; Prystauka et al., 2024)

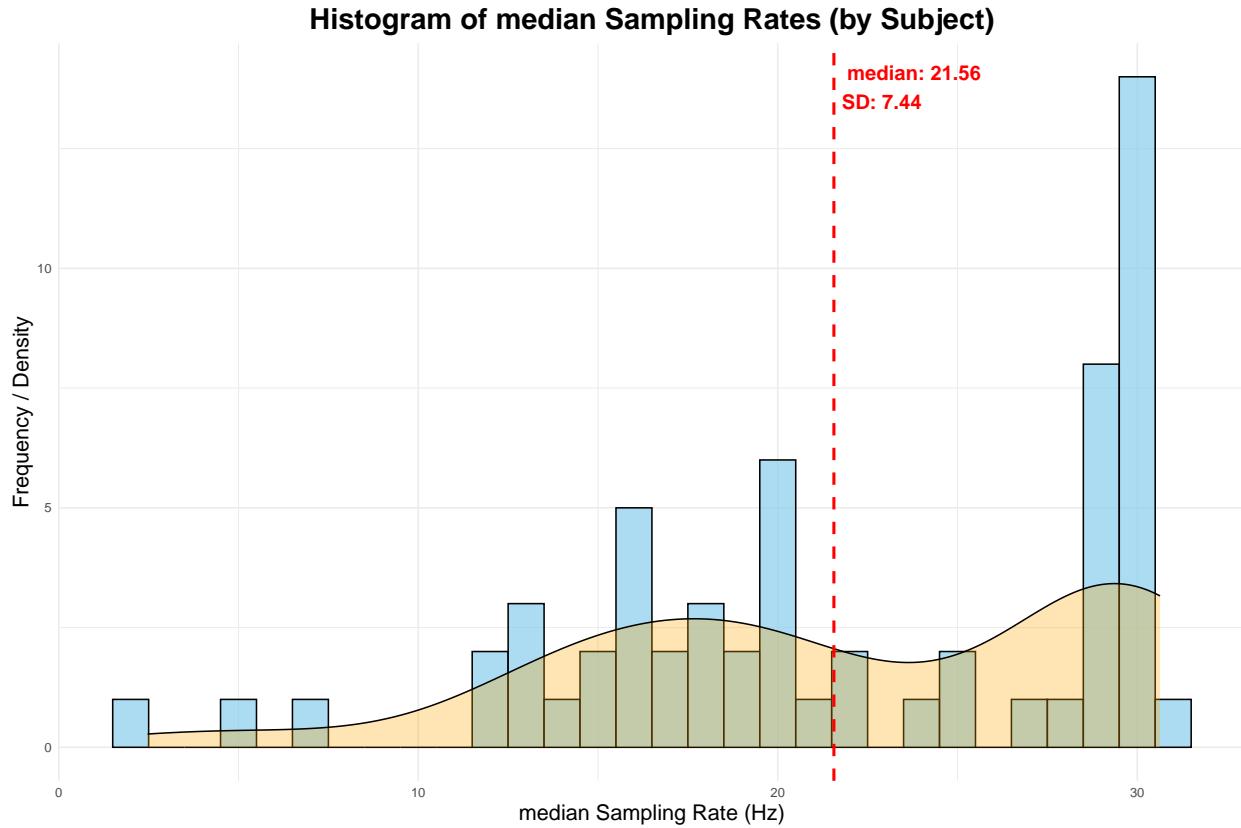
```
samp_rate_L2 <- analyze_sampling_rate(edat_L2, summary_stat="median")
```

482 Overall median Sampling Rate (Hz): 21.56

483 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")
```

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

503 The function returns a summary table showing the total number and percentage of gaze points that
 504 fall outside the bounds, broken down by axis (X, Y), as well as the combined total (see Table 3). It also returns
 505 three additional tibbles: (1) missingness by-subject, (2) missingness by-trial, and (3) a cleaned dataset with
 506 all the data merged, and the problematic rows removed if specified. These outputs can be referenced in a final
 507 report or manuscript. **As shown in Figure 5, no fixation points fall outside the standardized coordinate**
 508 **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)
```

509 We can use the `data_clean` tibble returned by the `gaze_oob()` function to filter out trials and sub-
 510 jects with more than 30% missing data. The value of 30% is just a suggestion and should not be used as a
 511 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)
```

Figure 5

Looks to each quadrant of the screen

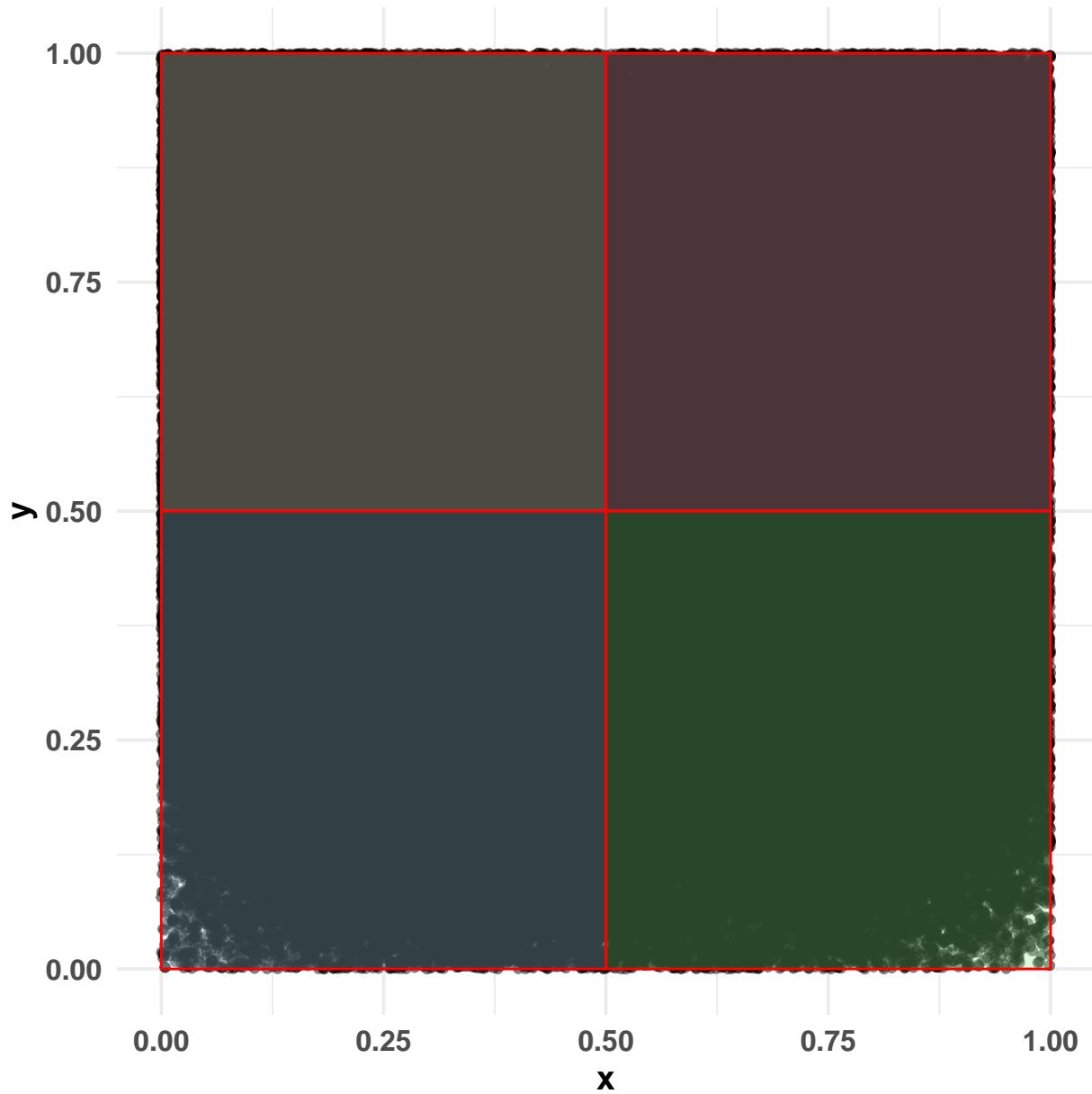


Table 3

Out of bounds gaze statistics by-participant

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

512 *Eye-tracking data*

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

521 Accordingly, we filtered the `edat_L2` dataset to include only rows where `convergence < 0.5` and
 522 `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
```

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

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

Table 4

Quadrant 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

527 Areas of Interest

528 Zone coordinates

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

538 **Quadrant approach.** One way is to make the AOIs as big as possible, dividing the screen into four
 539 quadrants. This approach has been used in several studies [e.g., (Bramlett & Wiener, 2024; Prystauka et al.,
 540 2024). Table 4 lists coordinates for the quadrant approach and Figure 6 shows how each quadrant looks in
 541 standardized space.

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

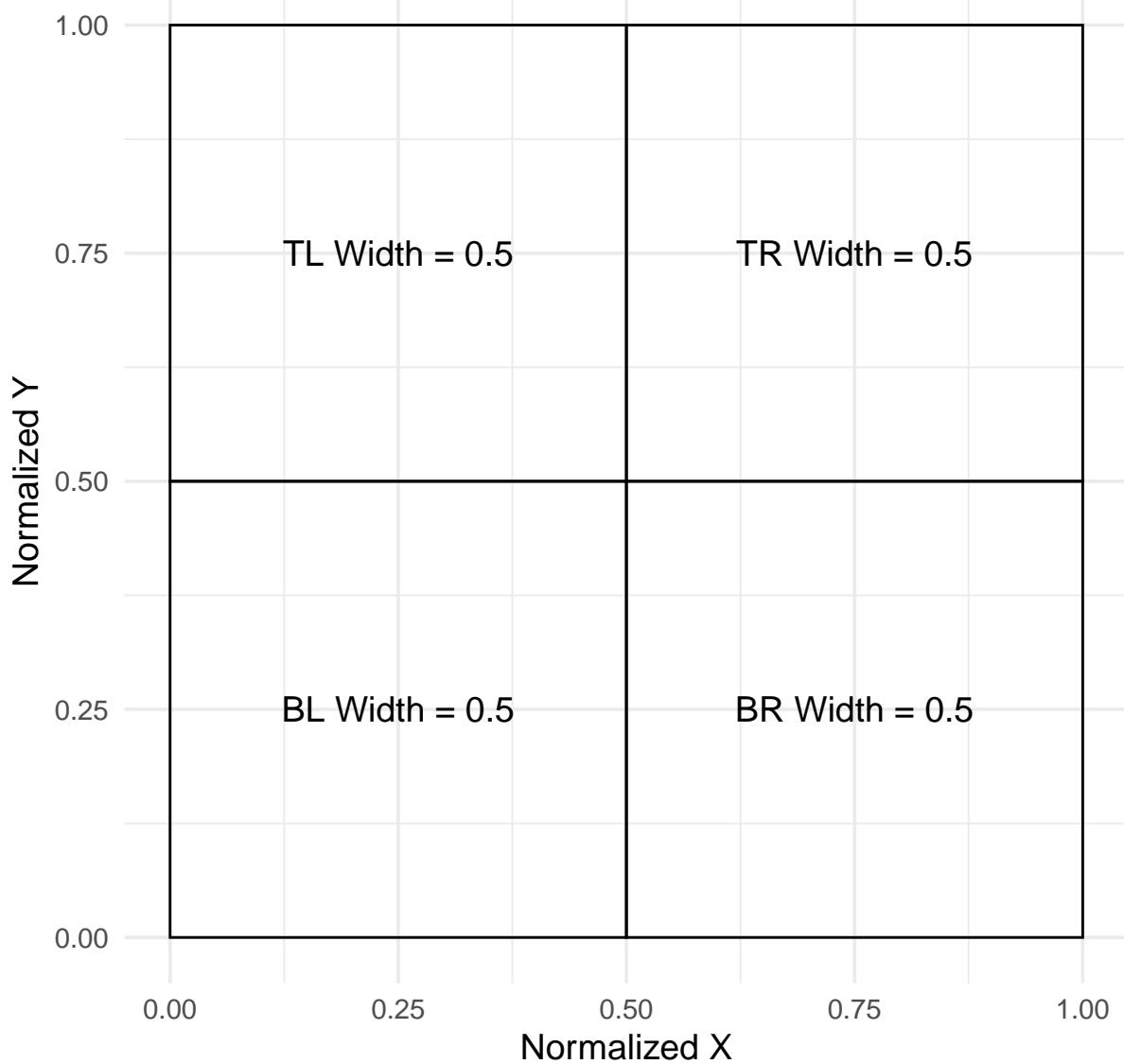
545 For each trial, the images are dynamically placed at different screen locations, and the code maps
 546 each image to its corresponding condition based on these locations.

```
# Assuming your data is in a data frame called dat_L2
dat_L2 <- dat_L2 %>%
  mutate(
    Target = case_when(
      typetl == "target" ~ TL,
```

Figure 6

AOI coordinates in standardized space using the quadrant approach

Quadrants with Width Annotations



```

    typepr == "target" ~ TR,
    typebl == "target" ~ BL,
    typebr == "target" ~ BR,
    TRUE ~ NA_character_ # Default to NA if no match
),
Unrelated = case_when(
    typetl == "unrelated1" ~ TL,
    typepr == "unrelated1" ~ TR,
    typebl == "unrelated1" ~ BL,
    typebr == "unrelated1" ~ BR,
    TRUE ~ NA_character_
),
Unrelated2 = case_when(
    typetl == "unrelated2" ~ TL,
    typepr == "unrelated2" ~ TR,
    typebl == "unrelated2" ~ BL,
    typebr == "unrelated2" ~ BR,
    TRUE ~ NA_character_
),
Cohort = case_when(
    typetl == "cohort" ~ TL,
    typepr == "cohort" ~ TR,
    typebl == "cohort" ~ BL,
    typebr == "cohort" ~ BR,
    TRUE ~ NA_character_
)
)
)

```

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

```

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

555 Once we do this we can use the `assign_aoi()` function to loop through our object called
 556 `dat_colnames_L2` and assign locations (i.e., TR, TL, BL, BR) to where participants looked at on the screen.
 557 This requires the x and y coordinates and the location of our aois `aoi_loc`. Here we are using the quadrant
 558 approach. This function will label non-looks and off screen coordinates with NA. To make it easier to read
 559 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"
  ))
```

560 In `AOI_L2` we label looks to Targets, Unrelated, and Cohort items with 1 (looked) and 0 (no look)
 561 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)
  )

```

562 The locations of looks need to be pivoted into long format—that is, converted from separate columns
 563 into a single column. This transformation makes the data easier to visualize and analyze. We use the
 564 `pivot_longer()` function from the `tidyverse` to combine the columns (Target, Unrelated, Unrelated2,
 565 and Cohort) into a single column called `condition1`. Additionally, we create another column called `Looks`,
 566 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 = "condition",
    values_to = "Looks"
  )

```

567 We further clean up the object by first cleaning up the condition codes. They have a numeral ap-
 568 pended to them and that should be removed. We then adjust the timing in the `gaze_sub_L2_comp` object by
 569 aligning time to the actual audio onset. To achieve this, we subtract `RT_audio` from time for each trial. In
 570 addition, we subtract 300 ms from this to account for the 100 ms of silence at the beginning of each audio
 571 clip and 200 ms to account for the oculomotor delay when planning an eye movement (Viviani, 1990). Ad-
 572 ditionally, we set our interest period between 0 ms (audio onset) and 2000 ms. This was chosen based on the
 573 time course figures in Sarrett et al. (2022) . It is important that you choose your interest area carefully and
 574 preferably you preregister it. The interest period you choose can bias your findings (Peelle & Van Engen,
 575 2021). We also filter out gaze coordinates that fall outside the standardized window, ensuring only valid data
 576 points are retained. The resulting object `gaze_sub_long_L2` provides the corrected time column spanning
 577 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)

```

578 **Samples to bins**

579 ***Downsampling***

580 Downsampling into larger time bins is a common practice in gaze data analysis, as it helps create
 581 a more manageable dataset and reduces noise. When using research grade eye-trackers, downsampling is
 582 an optional step in the preprocessing pipeline. However, with consumer-based webcam eye-tracking it is
 583 recommended you downsample your data so participants have consistent bin sizes (e.g., (Slim et al., 2024;
 584 Slim & Hartsuiker, 2023)). In webgazeR we included the `downsample_gaze()` function to assist with this
 585 process. We apply this function to the `gaze_sub_L2_long` object, and set the `bin.length` argument to 100,
 586 which groups the data into 100-millisecond intervals. This adjustment means that each bin now represents a
 587 100 ms passage of time. We specify `time` as the variable to base these bins on, allowing us to focus on broader
 588 patterns over time rather than individual millisecond fluctuations. There is no agreed upon downsampling
 589 value, but with webcam data larger bins are preferred (see Slim & Hartsuiker, 2023).

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

```

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

```

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

```

# Average Fix for unrelated and unrelated2, then combine with the rest
gaze_sub_L2_avg <- gaze_sub_L2 %>%

```

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

```
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")))
```

596 The above will not include the subject variable. If you want to keep participant-level data we need
 597 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"))
```

598 **Upsampling**

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

604 (e.g., EEG).

605 Our webgazeR package provides several functions to assist with this process. The
 606 upsample_gaze() function allows users to upsample their gaze data to a higher sampling rate (e.g.,
 607 250 Hz or even 1000 Hz). After upsampling, users can apply the smooth_gaze() function to reduce
 608 noise (webgazeR uses a n-point moving average) followed by the interpolate_gaze() function to fill
 609 in missing values using linear interpolation. Below we show you how to use the function, but do not
 610 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" )
```

611 Aggregation

612 Aggregation is an optional step. If you do not plan to analyze proportion data, and instead what time
 613 binned data with binary outcomes preserved please set the aggvars argument to “none.” This will return a
 614 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")
```

615 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))
```

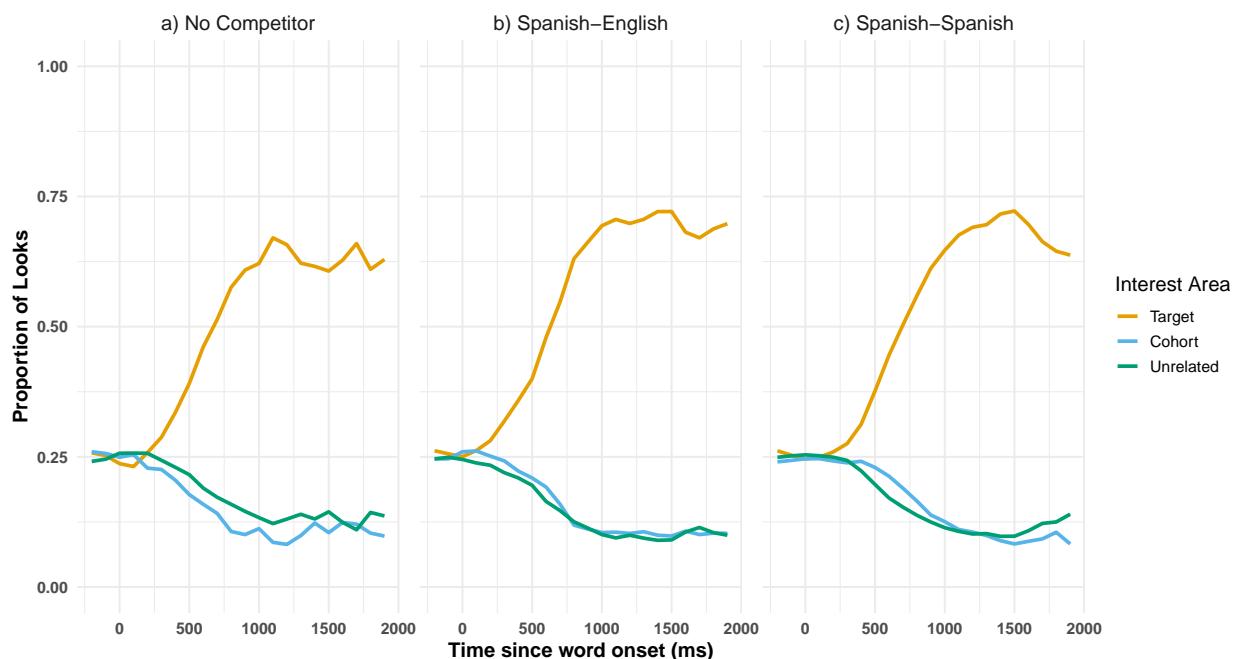
616 Visualizing time course data

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

623 Below, we have plotted the time-course data for each condition in Figure 7. By default, the graphs
 624 utilize a color-blind-friendly palette from the `ggokabeito` package (Barrett, 2021). However, you can set
 625 the argument `use_color = FALSE` to generate a non-colored version of the figure, where different line types
 626 and shapes differentiate conditions. Additionally, since these are `ggplot` objects, you can further customize
 627 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



628 Gorilla provided coordinates

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

Table 6

Gorilla provided standarized 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

633 We will use the function `extract_aois()` to get the standarized coordinates for each quadrant on
 634 screen. You can use the `zone_names` argument to get the zones you want to use. In our example, we want the
 635 TL, BR, BL TR coordinates. We input the object from above `vwp_paths_filtered_L2` that contains all our
 636 eye-tracking files and extract the coordinates we want. These are labeled in Table 6. In Figure 8 we can see
 637 that the AOIs are a bit smaller than then when using the quadrant approach. We can take these coordinates
 638 and use them in our analysis. Looking at Figure 9, we see the data is a bit nosier than the quadrant approach,
 639 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"))
```

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

640 Modeling data

641 After you have preprocessed the data, the next step is analysis. When analyzing VWP data there are
 642 many analytic approaches to choose from (e.g., growth curve analysis (GCA), cluster permutation analysis
 643 (CPA), generalized additive mixed models (GAMMS), logistic multilevel models, divergent point analysis
 644 (DPA), etc.), and a lot has already been written describing these methods and a lot of great tutorials exist
 645 showing how to apply these methods to visual world fixation data from the lab (Coretta & Casillas, 2024;
 646 see Ito & Knoeferle, 2023; Stone et al., 2021) and online (Bramlett & Wiener, 2024).

647 This paper's goal, however, is to not evaluate different analytic approaches and tell readers what they
 648 should use. All methods have their strengths and weaknesses (see Ito & Knoeferle, 2023). Nevertheless,
 649 statistical modeling should be guided by the questions researchers have and thus serious thought needs to be

Figure 8

Gorilla provided standardized coordinates for the four quadrants on the screen

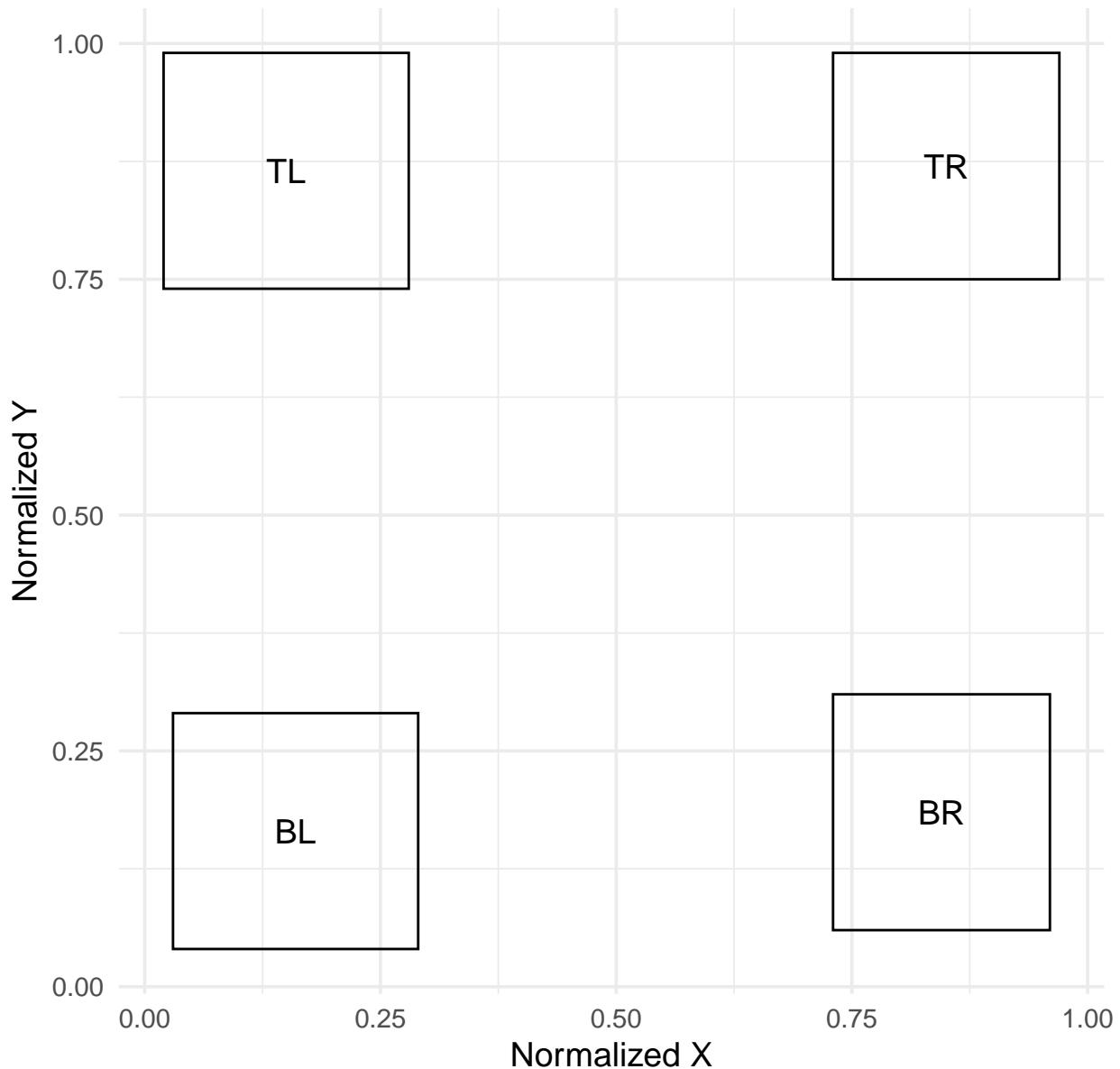
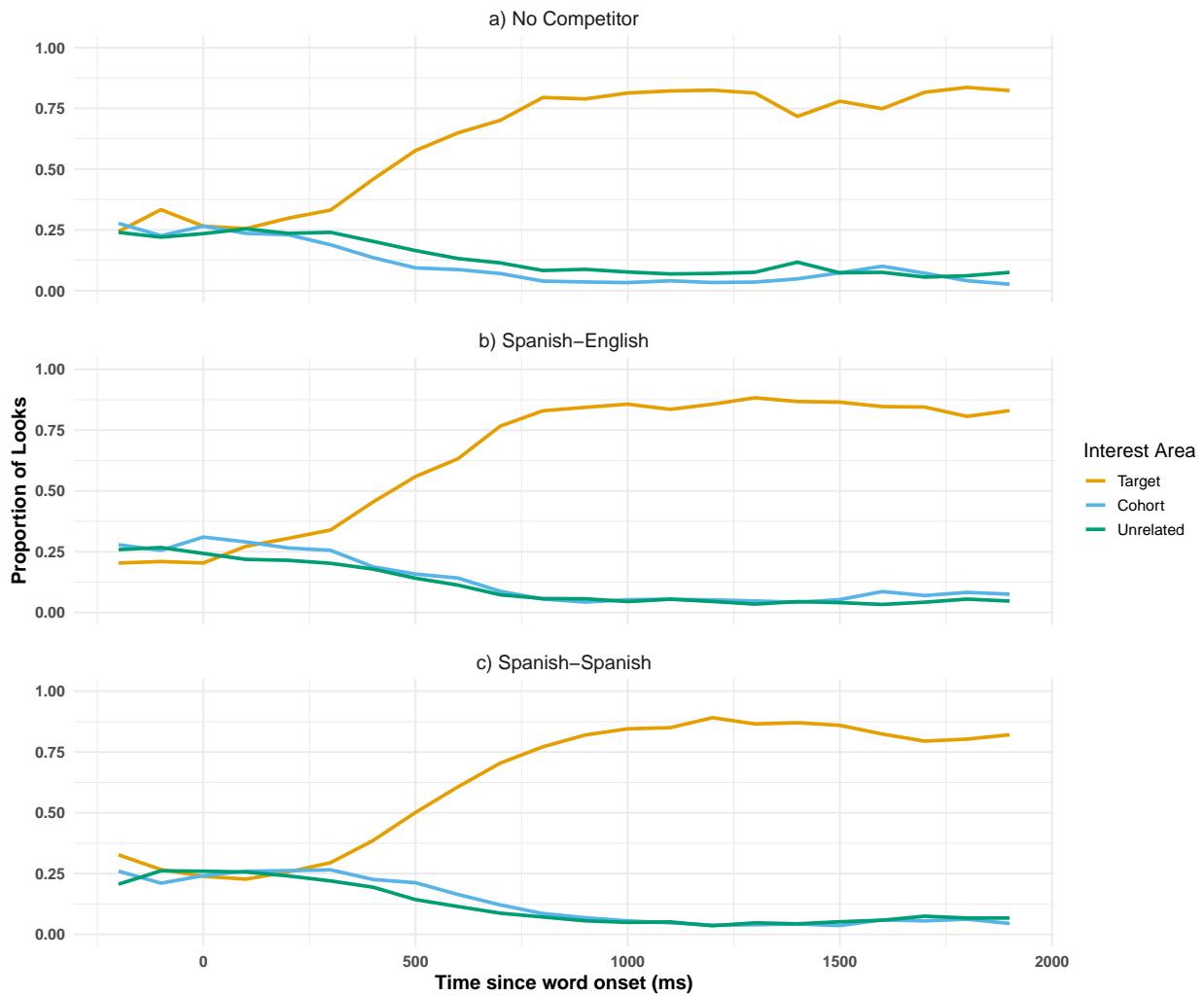


Figure 9

Comparison of competition effects with Gorilla standardized coordinates



650 given to the proper analysis. In the VWP, there are two general questions one might be interested in: (1) Are
 651 there any overall difference in fixations between conditions and (2) Are there any time course differences in
 652 fixations between conditions (and/or groups).

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

659 **CPA**

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

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

668 The clustering procedure involves three main steps:

- 669 1. Cluster Formation: With our data, a multilevel logistic model is conducted for every data point (con-
670 dition by time). Please note that any statistical test can be run here. Adjacent data points that surpass
671 the mass univariate significance threshold (e.g., $p < .05$) are combined into clusters. The cluster-
672 level statistic, typically the sum of the t-values (or F-values) within the cluster, is computed labeled
673 as SumStatitic is output below). By clustering adjacent significant data points, this step accounts for
674 autocorrelation by considering temporal dependencies rather than treating each data point as indepen-
675 dent.
- 676 2. Null Distribution Creation: Next, the same analysis is run as in step 1. However, the analysis is based
677 on randomly permuting or shuffling the conditions within subjects. This principle of exchangeability is
678 important here, as it suggests that the condition labels can be exchanged without altering the underlying
679 data structure. This randomization is repeated n times (e.g., 1000 shuffles), and for each permutation,
680 the cluster-level statistic is computed. This step addresses the issue of multiple comparisons by con-
681 structing a distribution of cluster-level statistics under the null hypothesis, providing a baseline against
682 which observed cluster statistics can be compared. By doing so, the method controls the family-wise
683 error rate and ensures that significant findings are not simply due to chance.
- 684 3. Significance Testing: The cluster-level statistics from the observed (real) comparison is compared to
685 the null distribution we created above. Clusters with statistics falling in the highest or lowest 2.5% of
686 the null distribution are considered significant (e.g., $p < 0.05$).

687 To perform CPA, we will load in the `permutes` (Voeten, 2023), `permuco` (Frossard & Renaud,
688 2021), `foreach` (& Weston, 2022), and `Parallel` (Corporation & Weston, 2022) packages in R. Loading
689 these packages allow us to use the `cluster.glmer()` function to run a cluster permutation (10,000 rimes)
690 across multiple system cores to speed up the process. We run a CPA on the `gaze_sub_id` object where each
691 row in `Looks` denotes whether the AOI was fixated, with values of zero (not fixated) or one (fixated).

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

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

```
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.

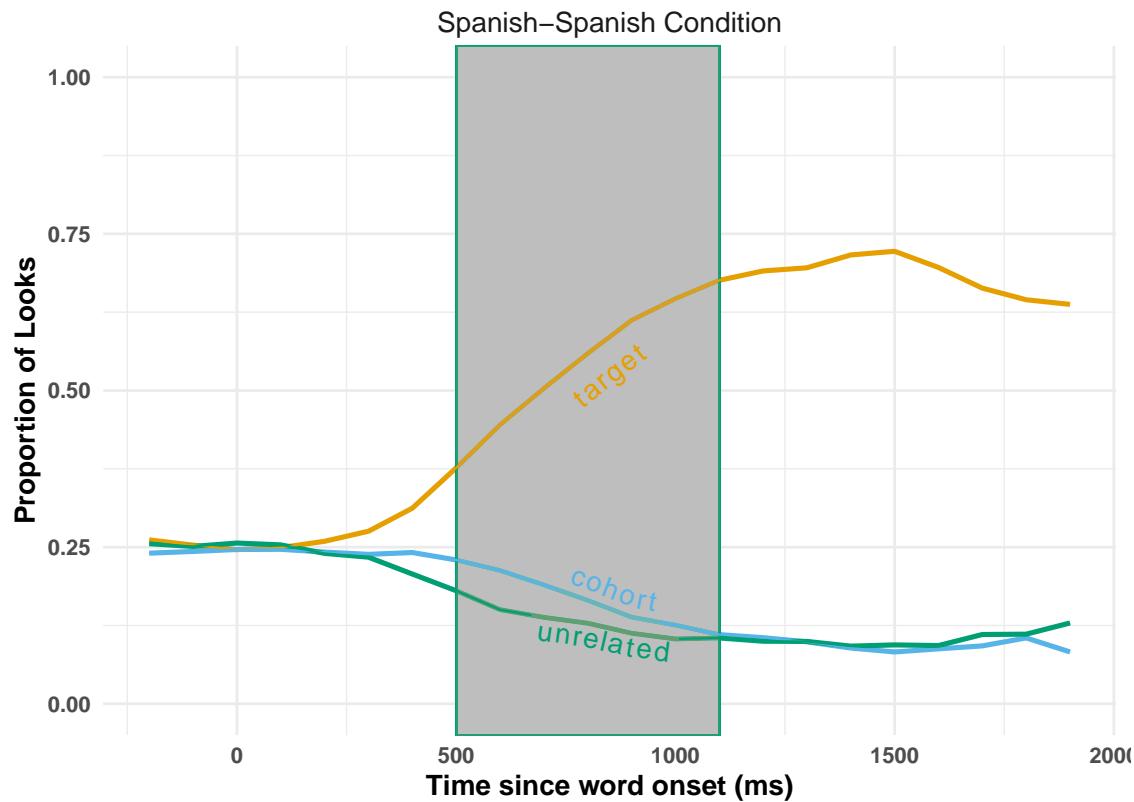
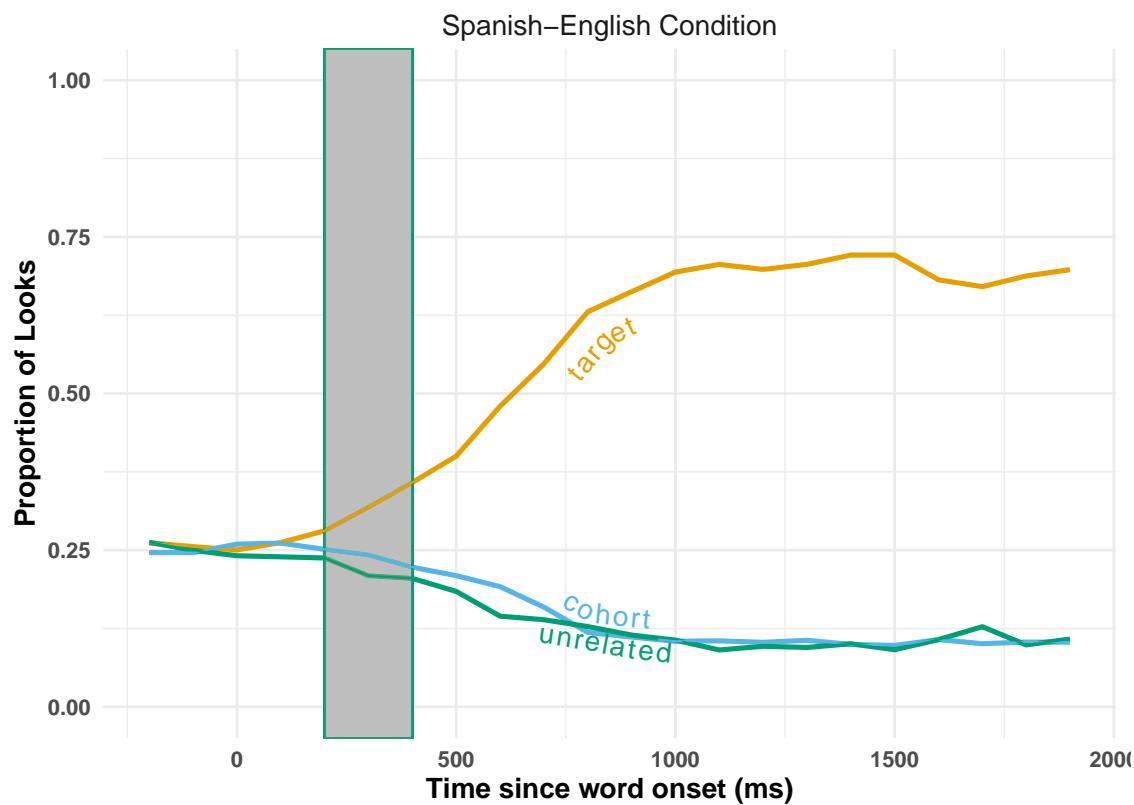
Effect size. It is important to address the issue of effect sizes in the context of CPA. Calculating effect sizes for CPA is not straightforward, as the technique is designed to evaluate temporal clusters rather than individual time points. (Slim et al., 2024; but also see Meyer et al., 2021) outline 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.

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.

A**B**

715 In this tutorial, we covered the basic steps of running a visual world webcam-based eye-tracking
716 experiment. We highlighted these steps by using data from a cross-linguistic VWP looking at competitive
717 processes in L2 speakers of Spanish. Specifically, we attempted to replicate the experiment by Sarrett et al.
718 (2022) where they observed within- and between L2/L1 competition using carefully crafted materials.

719 **Replication of Sarrett et al. (2022)**

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

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

728 A major analytic difference lies in how the time course of competition was examined. While Sarrett
729 et al. (2022) employed a non-linear curve-fitting approach (see McMurray et al., 2010), we used cluster-
730 based permutation analysis (CPA). This methodological distinction limits direct comparisons regarding the
731 temporal dynamics of competition. Nonetheless, the overall time course patterns align surprisingly well: our
732 CPA identified a significant cluster starting at 500 ms, while Sarrett et al. (2022) observed effects beginning
733 around 400 ms—suggesting a modest delay of approximately 100 ms in our online data. This delay is still
734 markedly smaller than in previous webcam-based studies (e.g., Semmelmann & Weigelt, 2018; Slim et al.,
735 2024), reflecting progress in online eye-tracking. That said, it's important to note that CPA is not ideally
736 suited for making precise temporal inferences about onset or offset of effects (Fields & Kuperberg, 2019; Ito
737 & Knoeferle, 2023).

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

742 In contrast, our sequence began with a 500 ms fixation cross (serving as the ITI), followed by a longer
743 1500 ms preview. The images then disappeared, and participants clicked a centrally placed start button to
744 initiate audio playback, at which point the images reappeared. Upon target selection, the next trial began
745 immediately. We also imposed a 5-second timeout for non-responses. Additionally, our study included 250
746 trials—fewer than the 450 in the original study²—but still more than most webcam-based research. Despite
747 the reduced trial count, we observed parallel competition effects in both language conditions, underscoring
748 the robustness of the findings.

²The 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.

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.

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 8

Eye-tracking questionnaire items

Question
1. Do you have a history of vision problems (e.g., corrected vision, eye strain, etc.)?
2. Are you on any medications currently that can impact your eyesight?
If yes, please list below:
4. Does your room currently have natural light?
5. Are you using the built in camera or an external 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 (in feet)?
7. Approximately how many times did you look at your phone or computer screen?
8. Approximately how many times did you get up during the experiment?
9. Was the environment you took the experiment in quiet or noisy?
10. When you had to calibrate, were the instructions clear?
11. What additional information would you add to help make the experiment better?
12. Are you wearing a mask?

⁷⁸⁶ **you would be unable to capture otherwise.**

⁷⁸⁷ ***Generalizability to other platforms***

⁷⁸⁸ We demonstrated how to analyze webcam eye-tracking data collected via the Gorilla platform using
⁷⁸⁹ WebGazer.js. Although we did not validate this pipeline on other platforms that support WebGazer.js—
⁷⁹⁰ such as PCIbex (Zehr & Schwarz, 2018), jsPsych (Leeuw, 2015), or PsychoPy (Peirce et al., 2019)—we
⁷⁹¹ believe the pipeline is generalizable to these and to platforms that use other gaze estimation logarithms,
⁷⁹² such as Labvanced [@kaduk2024]. To support broader compatibility, the functions in the webgazeR
⁷⁹³ package are designed to work with a variety of file types—including .csv, .tsv, and .xlsx – and work
⁷⁹⁴ with any dataset that includes five essential columns: subject, trial, x, y, and time. We also provide a
⁷⁹⁵ helper function, `make_webgazer()`, to assist in renaming columns so your dataset can be adapted to
⁷⁹⁶ the expected format.

⁷⁹⁷ 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.

Table 9

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

Question
1. Do you have a history of vision problems (e.g., corrected vision, eye disease, or drooping eyelids)?
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?
2. Are you on any medications currently that can impair your judgement?
4. Does your room currently have natural light?
4. Does your room currently have natural light?
5. Are you using the built in camera?
5. Are you using the built in camera?
9. Was the environment you took the experiment in distraction free?
9. Was the environment you took the experiment in distraction free?

801 **Power**

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

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

814 ****Recommendations and ways forward****

815 While our findings support the promise of webcam eye-tracking for language research, several chal-
 816 lenges remain that researchers should consider. One of the most significant issues is data loss due to poor
 817 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.

828 *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.

834 *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 [@kaduk2024], 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.

849 *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*

852 important for online studies, where sample variability can be higher than in controlled lab environments. To
853 this point, you will have to over-enroll your study due to the high attrition rate to reach your target goal, so
854 please plan accordingly.**

855 ****Collect detailed post-experiment feedback****

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

860 By implementing these strategies, researchers can improve the quality and consistency of data col-
861 lected through webcam-based eye-tracking. These recommendations aim to maximize the utility and repro-
862 ducibility of remote eye-tracking research, particularly in language processing contexts.**

863 **Conclusions**

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

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

877

References

- 878 Allaire, J. J., Teague, C., Scheidegger, C., Xie, Y., Dervieux, C., & Woodhull, G. (2024). *Quarto* (Version
879 1.6) [Computer software]. <https://doi.org/10.5281/zenodo.5960048>
- 880 Allopenna, P. D., Magnuson, J. S., & Tanenhaus, M. K. (1998). *Tracking the time course of spoken word
881 recognition using eye movements: Evidence for continuous mapping models* (pp. 419–439).
- 882 Altmann, G. T. M., & Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of
883 subsequent reference. *Cognition*, 73(3), 247–264. [https://doi.org/10.1016/S0010-0277\(99\)00059-1](https://doi.org/10.1016/S0010-0277(99)00059-1)
- 884 Anderson, C. A., Allen, J. J., Plante, C., Quigley-McBride, A., Lovett, A., & Rokkum, J. N. (2019). The
885 MTurkification of Social and Personality Psychology. *Personality & Social Psychology Bulletin*, 45(6),

- 886 842–850. <https://doi.org/10.1177/0146167218798821>
- 887 Anwyl-Irvine, A. L., Massonnié, J., Flitton, A., Kirkham, N., & Evershed, J. K. (2020). Gorilla in our
888 midst: An online behavioral experiment builder. *Behavior Research Methods*, 52(1), 388–407. <https://doi.org/10.3758/s13428-019-01237-x>
- 890 Apfelbaum, K. S., Klein-Packard, J., & McMurray, B. (2021). The pictures who shall not be named: Em-
891 pirical support for benefits of preview in the visual world paradigm. *Journal of Memory and Language*,
892 121, 104279. <https://doi.org/10.1016/j.jml.2021.104279>
- 893 Barrett, M. (2021). *Ggokabeito: 'Okabe-ito' scales for 'ggplot2' and 'ggraph'*. <https://CRAN.R-project.org/package=ggokabeito>
- 894 Bianco, R., Mills, G., Kerangal, M. de, Rosen, S., & Chait, M. (2021). Reward enhances online participants'
895 engagement with a demanding auditory task. *Trends in Hearing*, 25, 23312165211025941. <https://doi.org/10.1177/23312165211025941>
- 896 Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., & Ertl, T. (2017). Visualization of Eye
897 Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum*, 36(8), 260–284. <https://doi.org/10.1111/cgf.13079>
- 898 Blasi, D. E., Henrich, J., Adamou, E., Kemmerer, D., & Majid, A. (2022). Over-reliance on english hinders
899 cognitive science. *Trends in Cognitive Sciences*, 26(12), 1153–1170. <https://doi.org/10.1016/j.tics.2022.09.015>
- 900 Bogdan, P. C., Dolcos, S., Buetti, S., Lleras, A., & Dolcos, F. (2024). Investigating the suitability of
901 online eye tracking for psychological research: Evidence from comparisons with in-person data using
902 emotion–attention interaction tasks. *Behavior Research Methods*, 56(3), 2213–2226. <https://doi.org/10.3758/s13428-023-02143-z>
- 903 Boxtel, W. S. van, Linge, M., Manning, R., Haven, L. N., & Lee, J. (2024). Online eye tracking for aphasia:
904 A feasibility study comparing web and lab tracking and implications for clinical use. *Brain and Behavior*,
905 14(11), e70112. <https://doi.org/10.1002/brb3.70112>
- 906 Bramlett, A. A., & Wiener, S. (2024). The art of wrangling. *Linguistic Approaches to Bilingualism*.
907 <https://doi.org/https://doi.org/10.1075/lab.23071.bra>
- 908 Bramlett, A. A., & Wiener, S. (2025). Individual differences modulate prediction of Italian words based on
909 lexical stress: a close replication and LASSO extension of Sulpizio and McQueen (2012). *Journal of
910 Cultural Cognitive Science*, 9(1), 55–81. <https://doi.org/10.1007/s41809-024-00162-6>
- 911 Brysbaert, M., & Stevens, M. (2018). Power analysis and effect size in mixed effects models: A tutorial.
912 *Journal of Cognition*, 1(1). <https://doi.org/10.5334/joc.10>
- 913 Bylund, E., Khafif, Z., & Berghoff, R. (2024). Linguistic and geographic diversity in research on second
914 language acquisition and multilingualism: An analysis of selected journals. *Applied Linguistics*, 45(2),
915 308–329. <https://doi.org/10.1093/applin/amad022>
- 916 Carter, B. T., & Luke, S. G. (2020). Best practices in eye tracking research. *International Journal of Psy-
917 chophysiology*, 155, 49–62. <https://doi.org/10.1016/j.ijpsycho.2020.05.010>
- 918 Chen-Sankey, J., Elhabashy, M., Gratale, S., Geller, J., Mercincavage, M., Strasser, A. A., Delnevo, C. D.,
919 Jeong, M., & Wackowski, O. A. (2023). Examining Visual Attention to Tobacco Marketing Materials
920 Among Young Adult Smokers: Protocol for a Remote Webcam-Based Eye-Tracking Experiment. *JMIR*
921

- 926 *Research Protocols*, 12, e43512. <https://doi.org/10.2196/43512>
- 927 Colby, S. E., & McMurray, B. (2023). Efficiency of spoken word recognition slows across the adult lifespan.
928 *Cognition*, 240, 105588. <https://doi.org/10.1016/j.cognition.2023.105588>
- 929 Cooper, R. M. (1974). The control of eye fixation by the meaning of spoken language: A new methodol-
930 ogy for the real-time investigation of speech perception, memory, and language processing. *Cognitive*
931 *Psychology*, 6(1), 84–107. [https://doi.org/10.1016/0010-0285\(74\)90005-X](https://doi.org/10.1016/0010-0285(74)90005-X)
- 932 Coretta, S., & Casillas, J. V. (2024). A tutorial on generalised additive mixed effects models for bilingualism
933 research. *Linguistic Approaches to Bilingualism*. <https://doi.org/10.1075/lab.23076.cor>
- 934 Corporation, M., & Weston, S. (2022). *doParallel: Foreach parallel adaptor for the 'parallel' package*.
935 <https://CRAN.R-project.org/package=doParallel>
- 936 Csárdi, G., Hester, J., Wickham, H., Chang, W., Morgan, M., & Tenenbaum, D. (2024). *Remotes: R pack-
937 age installation from remote repositories, including 'GitHub'*. <https://CRAN.R-project.org/package=remotes>
- 938 Dahan, D., Magnuson, J. S., & Tanenhaus, M. K. (2001). Time course of frequency effects in spoken-word
939 recognition: Evidence from eye movements. *Cognitive Psychology*, 42(4), 317–367. <https://doi.org/10.1006/cogp.2001.0750>
- 940 Degen, J., Kursat, L., & Leigh, D. D. (2021). Seeing is believing: Testing an explicit linking assumption
941 for visual world eye-tracking in psycholinguistics. *Proceedings of the Annual Meeting of the Cognitive
942 Science Society*, 43.
- 943 Dolstra, E., & contributors, T. N. (2023). *Nix* (Version 2.15.3) [Computer software]. <https://nixos.org/>
- 944 Eberhard, K. M., Spivey-Knowlton, M. J., Sedivy, J. C., & Tanenhaus, M. K. (1995). Eye movements as a
945 window into real-time spoken language comprehension in natural contexts. *Journal of Psycholinguistic
946 Research*, 24(6), 409–436. <https://doi.org/10.1007/BF02143160>
- 947 Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G*Power 3: A flexible statistical power analysis
948 program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, 39(2), 175–
949 191. <https://doi.org/10.3758/BF03193146>
- 950 Fields, E. C., & Kuperberg, G. R. (2019). Having your cake and eating it too: Flexibility and power with
951 mass univariate statistics for ERP data. *Psychophysiology*. <https://doi.org/10.1111/psyp.13468>
- 952 Firke, S. (2023). *Janitor: Simple tools for examining and cleaning dirty data*. <https://CRAN.R-project.org/package=janitor>
- 953 Frossard, J., & Renaud, O. (2021). *Permutation tests for regression, {ANOVA}, and comparison of signals:
954 The {permuco} package*. 99. <https://doi.org/10.18637/jss.v099.i15>
- 955 Geller, J., & Prystauka, Y. (2024). *webgazeR: Tools for processing webcam eye tracking data*. <https://github.com/jgeller112/webgazeR>
- 956 Godfroid, A., Finch, B., & Koh, J. (2024). Reporting Eye-Tracking Research in Second Language Ac-
957 quisition and Bilingualism: A Synthesis and Field-Specific Guidelines. *Language Learning*, n/a(n/a).
958 <https://doi.org/10.1111/lang.12664>
- 959 Green, P., & MacLeod, C. J. (2016). SIMR: an R package for power analysis of generalized linear mixed
960 models by simulation. *Methods in Ecology and Evolution*, 7(4), 493–498. <https://doi.org/10.1111/2041-210X.12504>
- 961
- 962
- 963
- 964
- 965

- 966 Henrich, J., Heine, S. J., & Norenzayan, A. (2010). Most people are not WEIRD. *Nature*, 466(7302), 29.
967 <https://doi.org/10.1038/466029a>
- 968 Hooge, I. T. C., Hessels, R. S., Niehorster, D. C., Andersson, R., Skrok, M. K., Konklewski, R., Stremplewski,
969 P., Nowakowski, M., Tamborski, S., Szkulmowska, A., Szkulmowski, M., & Nyström, M. (2024). Eye
970 tracker calibration: How well can humans refixate a target? *Behavior Research Methods*, 57(1), 23.
971 <https://doi.org/10.3758/s13428-024-02564-4>
- 972 Hopp, H. (2013). Grammatical gender in adult L2 acquisition: Relations between lexical and syntactic
973 variability. *Second Language Research*, 29(1), 33–56. <https://doi.org/10.1177/0267658312461803>
- 974 Huang, Y., & Snedeker, J. (2020). Evidence from the visual world paradigm raises questions about unac-
975 cusativity and growth curve analyses. *Cognition*, 200, 104251. <https://doi.org/10.1016/j.cognition.2020.104251>
- 976 Huettig, F., & McQueen, J. M. (2007). The tug of war between phonological, semantic and shape information
977 in language-mediated visual search. *Journal of Memory and Language*, 57(4), 460–482. <https://doi.org/10.1016/j.jml.2007.02.001>
- 978 Huettig, F., Rommers, J., & Meyer, A. S. (2011). Using the visual world paradigm to study language pro-
979 cessing: a review and critical evaluation. *Acta Psychologica*, 137(2), 151–171. <https://doi.org/10.1016/j.actpsy.2010.11.003>
- 980 Ito, A., & Knoeferle, P. (2023). Analysing data from the psycholinguistic visual-world paradigm: Compar-
981 ison of different analysis methods. *Behavior Research Methods*, 55(7), 3461–3493. <https://doi.org/10.3758/s13428-022-01969-3>
- 982 Ito, A., Pickering, M. J., & Corley, M. (2018). Investigating the time-course of phonological prediction in
983 native and non-native speakers of english: A visual world eye-tracking study. *Journal of Memory and
984 Language*, 98, 1–11. <https://doi.org/10.1016/j.jml.2017.09.002>
- 985 Izura, C., Cuetos, F., & Brysbaert, M. (2014). Lextale-Esp: a test to rapidly and efficiently assess the Spanish
986 vocabulary size. *PSICOLOGICA*, 35(1), 49–66. <http://hdl.handle.net/1854/LU-5774107>
- 987 Ju, M., & Luce, P. A. (2004). Falling on sensitive ears: Constraints on bilingual lexical activation. *Psycho-
988 logical Science*, 15(5), 314–318. <https://doi.org/10.1111/j.0956-7976.2004.00675.x>
- 989 Kaduk, T., Goeke, C., Finger, H., & König, P. (2024). Webcam eye tracking close to laboratory standards:
990 Comparing a new webcam-based system and the EyeLink 1000. *Behavior Research Methods*, 56(5),
991 5002–5022. <https://doi.org/10.3758/s13428-023-02237-8>
- 992 Kamide, Y., Altmann, G. T. M., & Haywood, S. L. (2003). The time-course of prediction in incremental
993 sentence processing: Evidence from anticipatory eye movements. *Journal of Memory and Language*,
994 49(1), 133–156. [https://doi.org/10.1016/S0749-596X\(03\)00023-8](https://doi.org/10.1016/S0749-596X(03)00023-8)
- 995 Kret, M. E., & Sjak-Shie, E. E. (2018). Preprocessing pupil size data: Guidelines and code. *Behavior
996 Research Methods*, 1–7. <https://doi.org/10.3758/s13428-018-1075-y>
- 997 Kumle, L., Võ, M. L.-H., & Draschkow, D. (2021). Estimating power in (generalized) linear mixed models:
998 An open introduction and tutorial in R. *Behavior Research Methods*, 53(6), 2528–2543. <https://doi.org/10.3758/s13428-021-01546-0>
- 999 Lakens, D. (2022). Sample size justification. *Collabra: Psychology*, 8(1). <https://doi.org/10.1525/collabra.33267>

- 1006 Leeuw, J. R. de. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser.
1007 *Behavior Research Methods*, 47(1), 1–12. <https://doi.org/10.3758/s13428-014-0458-y>
- 1008 Madsen, J., Júlio, S. U., Gucik, P. J., Steinberg, R., & Parra, L. C. (2021). Synchronized eye movements
1009 predict test scores in online video education. *Proceedings of the National Academy of Sciences*, 118(5),
1010 e2016980118. <https://doi.org/10.1073/pnas.2016980118>
- 1011 Magnuson, J. S., Dixon, J. A., Tanenhaus, M. K., & Aslin, R. N. (2007). The Dynamics of Lexical Com-
1012 petition During Spoken Word Recognition. *Cognitive Science*, 31(1), 133–156. <https://doi.org/10.1080/03640210709336987>
- 1013 Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *Journal of
1014 Neuroscience Methods*, 164(1), 177–190. <https://doi.org/10.1016/j.jneumeth.2007.03.024>
- 1015 McMurray, B., Samelson, V. M., Lee, S. H., & Tomblin, J. B. (2010). Individual differences in online
1016 spoken word recognition: Implications for SLI. *Cognitive Psychology*, 60(1), 1–39. <https://doi.org/10.1016/j.cogpsych.2009.06.003>
- 1017 McMurray, B., Tanenhaus, M. K., & Aslin, R. N. (2002). Gradient effects of within-category phonetic vari-
1018 ation on lexical access. *Cognition*, 86(2), B33–B42. [https://doi.org/10.1016/S0010-0277\(02\)00157-9](https://doi.org/10.1016/S0010-0277(02)00157-9)
- 1019 Meyer, M., Lamers, D., Kayhan, E., Hunnius, S., & Oostenveld, R. (2021). Enhancing reproducibility in
1020 developmental EEG research: BIDS, cluster-based permutation tests, and effect sizes. *Developmental
1021 Cognitive Neuroscience*, 52, 101036. <https://doi.org/10.1016/j.dcn.2021.101036>
- 1022 Microsoft, & Weston, S. (2022). *Foreach*: Provides foreach looping construct. <https://CRAN.R-project.org/package=foreach>
- 1023 Miller, J. (2023). Outlier exclusion procedures for reaction time analysis: The cures are generally worse
1024 than the disease. *Journal of Experimental Psychology: General*, 152(11), 3189–3217. <https://doi.org/10.1037/xge0001450>
- 1025 Milne, A. E., Zhao, S., Tampakaki, C., Bury, G., & Chait, M. (2021). Sustained pupil responses are
1026 modulated by predictability of auditory sequences. *The Journal of Neuroscience*, 41(28), 6116–6127.
1027 <https://doi.org/10.1523/JNEUROSCI.2879-20.2021>
- 1028 Mirman, D., & Graziano, K. M. (2012). Individual differences in the strength of taxonomic versus the-
1029 matic relations. *Journal of Experimental Psychology: General*, 141(4), 601–609. <https://doi.org/10.1037/a0026451>
- 1030 Müller, K. (2020). *Here*: A simpler way to find your files. <https://CRAN.R-project.org/package=here>
- 1031 Niedermann, J. P., Sucholutsky, I., Marjeh, R., Çelen, E., Griffiths, T., Jacoby, N., & Rijn, P. van. (2024).
1032 Studying the Effect of Globalization on Color Perception using Multilingual Online Recruitment and
1033 Large Language Models. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 46(0).
1034 <https://escholarship.org/uc/item/4hs755zz>
- 1035 Özsoy, O., Çiçek, B., Özal, Z., Gagarina, N., & Sekerina, I. A. (2023). Turkish-german heritage speakers'
1036 predictive use of case: Webcam-based vs. In-lab eye-tracking. *Frontiers in Psychology*, 14, 1155585.
1037 <https://doi.org/10.3389/fpsyg.2023.1155585>
- 1038 Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). *Webgazer: Scalable
1039 webcam eye tracking using user interactions*. 38393845.
- 1040 Patterson, A. S., & Nicklin, C. (2023). L2 self-paced reading data collection across three contexts: In-person,
1041

- 1046 online, and crowdsourcing. *Research Methods in Applied Linguistics*, 2(1), 100045. <https://doi.org/10.1016/j.rmal.2023.100045>
- 1047
- 1048 Peele, J. E., & Van Engen, K. J. (2021). Time stand still: Effects of temporal window selection on eye
1049 tracking analysis. *Collabra: Psychology*, 7(1), 25961. <https://doi.org/10.1525/collabra.25961>
- 1050 Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv,
1051 J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, 51(1), 195–
1052 203. <https://doi.org/10.3758/s13428-018-01193-y>
- 1053 Peterson, R. J. (2021). We need to address ableism in science. *Molecular Biology of the Cell*, 32(7), 507–510.
1054 <https://doi.org/10.1091/mbc.E20-09-0616>
- 1055 Płużyczka, M. (2018). The First Hundred Years: a History of Eye Tracking as a Research Method.
1056 *Applied Linguistics Papers*, 25/4, 101–116. <http://cejsh.icm.edu.pl/cejsh/element/bwmeta1.element.desklight-98576d43-39e3-4981-8c1c-717962cf29da>
- 1057
- 1058 Prystauka, Y., Altmann, G. T. M., & Rothman, J. (2024). Online eye tracking and real-time sentence pro-
1059 cessing: On opportunities and efficacy for capturing psycholinguistic effects of different magnitudes and
1060 diversity. *Behavior Research Methods*, 56(4), 3504–3522. <https://doi.org/10.3758/s13428-023-02176-4>
- 1061 R Core Team. (2024). *R: A language and environment for statistical computing* (Version 4.4.2). R Founda-
1062 tion for Statistical Computing. <https://www.R-project.org/>
- 1063 Rodd, J. M. (2024). Moving experimental psychology online: How to obtain high quality data when we
1064 can't see our participants. *Journal of Memory and Language*, 134, 104472. <https://doi.org/10.1016/j.jml.2023.104472>
- 1065
- 1066 Rodrigues, B., & Baumann, P. (2025). *Rix: Reproducible data science environments with 'nix'*. <https://docs.ropensci.org/rix/>
- 1067
- 1068 Rossi, E., Krass, K., & Kootstra, G. J. (2019). *Psycholinguistic Methods in Multilingual Research* (pp. 75–
1069 99). John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781119387725.ch4>
- 1070 Sarrett, M. E., Shea, C., & McMurray, B. (2022). Within- and between-language competition in adult second
1071 language learners: Implications for language proficiency. *Language, Cognition and Neuroscience*, 37(2),
1072 165–181. <https://doi.org/10.1080/23273798.2021.1952283>
- 1073 Semmelmann, K., & Weigelt, S. (2018). Online webcam-based eye tracking in cognitive science: A first
1074 look. *Behavior Research Methods*, 50(2), 451–465. <https://doi.org/10.3758/s13428-017-0913-7>
- 1075 Slim, M. S., & Hartsuiker, R. J. (2023). Moving visual world experiments online? A web-based replication
1076 of Dijkgraaf, Hartsuiker, and Duyck (2017) using PCIbex and WebGazer.js. *Behavior Research Methods*,
1077 55(7), 3786–3804. <https://doi.org/10.3758/s13428-022-01989-z>
- 1078 Slim, M. S., Kandel, M., Yacovone, A., & Snedeker, J. (2024). Webcams as windows to the mind? A direct
1079 comparison between in-lab and web-based eye-tracking methods. *Open Mind*, 8, 1369–1424. https://doi.org/10.1162/opmi_a_00171
- 1080
- 1081 Spivey, M. J., & Marian, V. (1999). Cross talk between native and second languages: Partial activation of
1082 an irrelevant lexicon. *Psychological Science*, 10(3), 281–284. <https://doi.org/10.1111/1467-9280.00151>
- 1083 Stone, K., Lago, S., & Schad, D. J. (2021). Divergence point analyses of visual world data: applications
1084 to bilingual research. *Bilingualism: Language and Cognition*, 24(5), 833–841. <https://doi.org/10.1017/S1366728920000607>
- 1085

- 1086 Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K. M., & Sedivy, J. C. (1995). Integration of visual
1087 and linguistic information in spoken language comprehension. *Science (New York, N.Y.)*, 268(5217),
1088 1632–1634. <http://www.ncbi.nlm.nih.gov/pubmed/7777863>
- 1089 Trueswell, J. C. (2008). *Using eye movements as a developmental measure within psycholinguistics* (I. A.
1090 Sekerina, E. M. Fernández, & H. Clahsen, Eds.; pp. 73–96). John Benjamins Publishing Company.
1091 <https://doi.org/10.1075/lald.44.05tru>
- 1092 Van der Cruyssen, I., Ben-Shakhar, G., Pertzov, Y., Guy, N., Cabooter, Q., Gunschera, L. J., & Verschuere,
1093 B. (2024). The validation of online webcam-based eye-tracking: The replication of the cascade effect,
1094 the novelty preference, and the visual world paradigm. *Behavior Research Methods*, 56(5), 4836–4849.
1095 <https://doi.org/10.3758/s13428-023-02221-2>
- 1096 Viviani, P. (1990). Eye movements in visual search: cognitive, perceptual and motor control aspects. *Reviews
1097 of Oculomotor Research*, 4, 353–393.
- 1098 Voeten, C. C. (2023). *Permutest: Permutation tests for time series data*. [https://CRAN.R-project.org/
package=permutes](https://CRAN.R-project.org/
1099 package=permutes)
- 1100 Vos, M., Minor, S., & Ramchand, G. C. (2022). Comparing infrared and webcam eye tracking in the Visual
1101 World Paradigm. *Glossa Psycholinguistics*, 1(1). <https://doi.org/10.5070/G6011131>
- 1102 Wickham, H. (2017). *Tidyverse: Easily install and load the 'tidyverse'*. [https://CRAN.R-project.org/
package=tidyverse](https://CRAN.R-project.org/
1103 package=tidyverse)
- 1104 Yee, E., Blumstein, S., & Sedivy, J. C. (2008). Lexical-semantic activation in broca's and wernicke's aphasia:
1105 Evidence from eye movements. *Journal of Cognitive Neuroscience*, 20(4), 592–612. [https://doi.org/10.1162/jocn.2008.20056](https://doi.org/10.
1106 1162/jocn.2008.20056)
- 1107 Zehr, J., & Schwarz, F. (2018). *PennController for internet based experiments (IBEX)*. [https://doi.org/10.17605/OSF.IO/MD832](https://doi.org/10.
1108 17605/OSF.IO/MD832)