# E4-analysis-replication

Rachel Ryskin

7/7/2021

```r
#Define a function for determining if `x,y` value falls in box.
in.box <- function(x, y, left, right, top, bottom, padding){
  is.in.the.box <- x >= left - padding & x <= right + padding & y >= top - padding & y <= bottom + padd
  return(is.in.the.box)
}
```

```r
data.files <- list.files('data/run-2', full.names = TRUE)
data.tables <- lapply(data.files, function(file){
  data.table <- fromJSON(file)
  return(data.table)
})
all.data <- bind_rows(data.tables)
```

```r
task.data <- all.data %>%
  dplyr::filter(compatibility != 'NA', compatibility != 'filler') %>%
  dplyr::select(subject, trial_index, rt, images, webgazer_data, mouse_events, compatibility, audio, ta

# Add a column that uniquely identifies the combination of images and audio shown on the screen
trialID.data <- task.data %>%
  group_by(audio, images) %>%
  slice(1) %>%
  select(audio, images) %>%
  ungroup() %>%
  mutate(trialID = 1:n())

task.data <- task.data %>%
  left_join(trialID.data)
```

```
## Joining with `by = join_by(images, audio)`
```

```r
eyetracking.data <- task.data %>%
  tidyr::unpack(webgazer_targets) %>%
  tidyr::unpack(c(`#jspsych-free-sort-draggable-0`, `#jspsych-free-sort-draggable-1`, `#jspsych-free-so
  unnest(webgazer_data)

mousetracking.data <- task.data %>%
  unnest(mouse_events)
```

```r
#First, create a data set with each trial, adding column for which object the mouse moves over first.
# WTF IS HAPPENING with TRIAL ID??
first.move <- mousetracking.data %>%
  select(subject, trialID, object, type, compatibility, target_instrument, target_animal, images) %>%
  group_by(subject, trialID) %>%
  filter(type=="enter") %>%
  slice(1) %>%
  rowwise() %>%
  mutate(which_loc = which(images == object)-1) %>%
  ungroup() %>%
  mutate(instrument_loc = str_sub(target_instrument, start=-1, end=-1)) %>%
  mutate(animal_loc = str_sub(target_animal, start=-1, end=-1)) %>%
  mutate(first.move.type = case_when(
    which_loc == instrument_loc ~ 'instrument',
    which_loc == animal_loc ~ 'animal',
    TRUE ~ 'other')) %>%
  mutate(is.mouse.instrument = if_else(first.move.type == 'instrument', 1, 0)) %>%
  mutate(compatibility = factor(compatibility))


# Summarize the data by subject, calculating proportion of trials the the first move was to the animal,
first.move.subject.summary <- first.move %>%
  group_by(subject, compatibility) %>%
  summarize(prop.animal = mean(first.move.type == 'animal'),
            prop.instrument = mean(first.move.type == 'instrument'),
            prop.other = mean(first.move.type == 'other')) %>%
  pivot_longer(c('prop.animal', 'prop.instrument', 'prop.other'), names_to="target_type", values_to="pr
```

```
## 'summarise()' has grouped output by 'subject'. You can override using the
## '.groups' argument.
```

```r
# Summarize the condition-level data for a barplot.
first.move.summary <- first.move.subject.summary %>%
  group_by(compatibility, target_type) %>%
  summarize(M=mean(proportion), SE=sd(proportion)/sqrt(n())) %>%
  mutate(bias = factor(compatibility, levels = c("instrument", "equibiased", "modifier") ))
```

```
## 'summarise()' has grouped output by 'compatibility'. You can override using the
## '.groups' argument.
```

```r
first.move.web.summary <- first.move.summary
```

As shown in Figure @ref(fig:E4-mouse-moves-fig-web-and-orig), the qualitative results match those of the original. The quantitative patterns of clicks were similar to those observed in the original dataset, though for Instrument-biased verbs, clicks were closer to evenly split between the animal and the instrument relative to the in-lab study where they were very clearly biased toward the instrument. A mixed-effects logistic regression model was used to predict whether the first movement was on the target instrument with the verb bias condition as an orthogonally contrast-coded (instrument vs. equi & modifier: inst = -2/3, equi = 1/3, mod = 1/3; equi vs. modifier: inst = 0, equi = -1/2, mod = 1/2 ) fixed effect.

```r
first.move.orig = read_tsv("original_study_data/Experiment1_clickData.txt")
```

```
## Rows: 1296 Columns: 7
## -- Column specification ----------------------------------------------------
## Delimiter: "\t"
## chr (1): condition
## dbl (6): subject, randomlyGeneratedNumber, order, trialID, clickedTargetAnim...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
eyetracking.window1.summary.by.trial.orig = read_tsv("original_study_data/Experiment1_eye-tracking_verb
  mutate(subject = as.character(subj),
         prop.fixations.animal = TA/(TA+TI+CA+CI),
         #sum_dur = TA+TI+CA+CI,
         condition = factor(cond, levels = c(0, 2, 1, 9), labels = c("instrument", "equibiased", "modifi
         time.window = "post-verb-onset-pre-animal-onset",
         study = "original") %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal, study)
```

```
## Rows: 1944 Columns: 8
## -- Column specification ----------------------------------------------------
## Delimiter: "\t"
## dbl (8): TA, TI, CA, CI, subj, trialID, order, cond
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
eyetracking.window2.summary.by.trial.orig = read_tsv("original_study_data/Experiment1_eye-tracking_anima
  mutate(subject = as.character(subj),
         prop.fixations.animal = TA/(TA+TI+CA+CI),
         condition = factor(cond, levels = c(0, 2, 1, 9), labels = c("instrument", "equibiased", "modifi
         time.window = "post-animal-onset-pre-instrument-onset",
         study = "original") %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal, study)
```

```
## Rows: 1944 Columns: 8
## -- Column specification ----------------------------------------------------
## Delimiter: "\t"
## dbl (8): TA, TI, CA, CI, subj, trialID, order, cond
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
eyetracking.window3.summary.by.trial.orig = read_tsv("original_study_data/Experiment1_eye-tracking_instr
  mutate(subject = as.character(subj),
         prop.fixations.animal = TA/(TA+TI+CA+CI),
         condition = factor(cond, levels = c(0, 2, 1, 9), labels = c("instrument", "equibiased", "modifi
         time.window = "post-instrument-onset",
         study = "original") %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal, study)
```

```
## Rows: 1944 Columns: 8
## -- Column specification ----------------------------------------------------
```

```
## Delimiter: "\t"
## dbl (8): TA, TI, CA, CI, subj, trialID, order, cond
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

# Summarize the data by subject, calculating proportion of trials the the first move was to the animal,
first.move.orig.subject.summary <- first.move.orig %>%
  group_by(subject, condition) %>%
  summarize(prop.animal = mean(clickedTargetAnimalFirst),
            prop.instrument = mean(clickedTargetInstrumentFirst),
            prop.other = mean(clickedTargetAnimalFirst == 0 & clickedTargetInstrumentFirst == 0 )) %>%
  pivot_longer(c('prop.animal', 'prop.instrument', 'prop.other'), names_to="target_type", values_to="pro

## 'summarise()' has grouped output by 'subject'. You can override using the
## '.groups' argument.

# Summarize the condition-level data for a barplot.
first.move.orig.summary <- first.move.orig.subject.summary %>%
  group_by(condition, target_type) %>%
  summarize(M=mean(proportion), SE=sd(proportion)/sqrt(n())) %>%
  mutate(bias = factor(condition, levels = c("Inst", "Equi", "mod"), labels = c("instrument", "equibias
         study = "original")

## 'summarise()' has grouped output by 'condition'. You can override using the
## '.groups' argument.

ggplot(bind_rows(
  first.move.orig.summary %>% mutate(study_type = "Original"),
  first.move.web.summary %>% mutate(study_type = "Web")),
  aes(x=bias, fill=target_type, y=M, ymin=M-SE, ymax=M+SE)) +
  geom_col(position=position_dodge(width=0.9), color = "black") +
  geom_errorbar(position=position_dodge(width=0.9), width=0.2) +
  scale_fill_brewer(palette = "Set1", name = "Location", labels = c("Animal", "Instrument", "Other")) +
  coord_cartesian(ylim = c(0,1)) +
  theme_classic() +
  labs(y = "Proportion of first mouse movements", x = "Bias") +
  facet_wrap(~study_type, ncol = 1) + theme(  # This creates vertical panels
    panel.grid.minor = element_blank(),
    axis.title = element_text(size=14),      # Axis titles
    axis.text = element_text(size=14),       # Axis labels
    legend.title = element_text(size=14),    # Legend title
    legend.text = element_text(size=14)      # Legend text
  )

write_csv(first.move, "output/E4_mouse_data.csv")

contrasts(first.move$compatibility) <- cbind(c(1/3, -2/3, 1/3), c(-1/2, 0, 1/2))

E4_mouse_moves_model <- glmer(is.mouse.instrument ~ compatibility + (1 + compatibility | subject) + (1
                              glmerControl(optimizer = "bobyqa"))
```
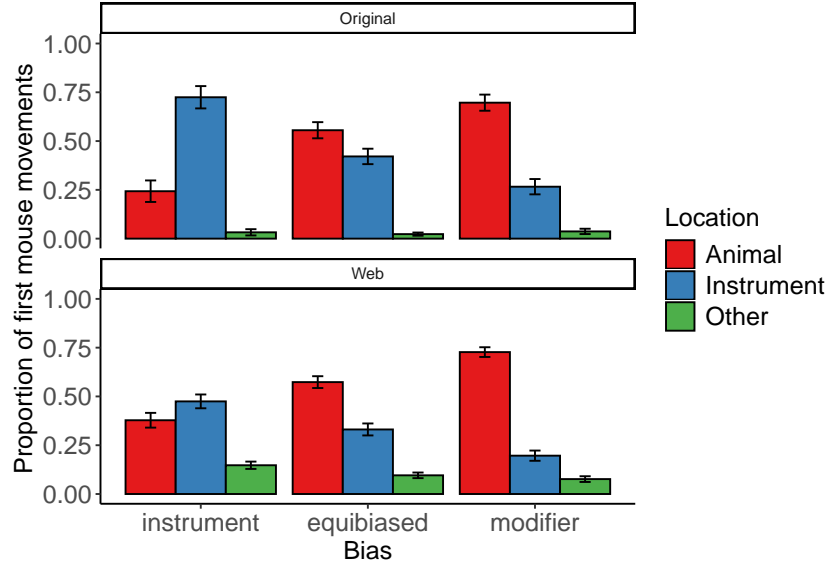
4

Figure 1: Proportion of first mouse movements by location and verb bias in the original dataset (Ryskin et al., 2017) and the current data collected online.

```
E4_mouse_moves_model_tab = broom.mixed::tidy(E4_mouse_moves_model)
E4_mouse_moves_model_c1 = E4_mouse_moves_model_tab %>% filter(term == "compatibility1")
E4_mouse_moves_model_c2 = E4_mouse_moves_model_tab %>% filter(term == "compatibility2")
```

Participants and items were entered as varying intercepts with by-participant varying slopes for verb bias condition.[1] Participants were more likely to first move their mouse over target instruments in the instrument-biased condition relative to the equi-biased and modifier-biased condition ($b$ = -1.5, $SE$ = 0.25, $p < 0.01$). Further, participants were more likely to first move their mouse over target instruments in the equi-biased condition relative to the modifier-biased condition ($b$ = -1.1, $SE$ = 0.29, $p < 0.01$)

Gaze fixations were time-locked to the auditory stimulus on a trial by trial basis and categorized as being directed towards one of the four items in the display if the x, y coordinates fell within a rectangle containing the image. In order to assess how verb bias impacted sentence disambiguation as the sentence unfolded, the proportion of fixations was computed in three time windows: the verb-to-animal window (from verb onset + 200 ms to animal onset + 200 ms), the animal-to-instrument window (from animal onset + 200 ms to instrument onset + 200 ms), and the post-instrument window (from instrument onset + 200 ms to instrument onset + 1500ms + 200 ms). Results were qualitatively similar to those in the original are shown in Figure @ref(fig:E4-gaze-timecourse-fig), though proportions of fixations to the target animal were much lower in the web version of the study. This may reflect the fact that participants in the web study are less attentive and/or the quality of the webgazer eye-tracking system is lower, relative to the Eyelink 1000 which was used for the original study. Eyegaze results are shown in more detail in Figure @ref(fig:E4-gaze-timecourse-fig).

```
#First figure out which object they are looking at.
#Calculate gaze in ROIs.
eyetracking.data.with.roi <- eyetracking.data %>%
  #filter(!subject %in% bad.eyetracking.data.subjects) %>%
  mutate(in.roi.0 = in.box(x,y,`#jspsych-free-sort-draggable-0.left`, `#jspsych-free-sort-draggable-0.ri
  mutate(in.roi.1 = in.box(x,y,`#jspsych-free-sort-draggable-1.left`, `#jspsych-free-sort-draggable-1.ri
  mutate(in.roi.2 = in.box(x,y,`#jspsych-free-sort-draggable-2.left`, `#jspsych-free-sort-draggable-2.ri
```

[1]lme4 syntax: `glmer(is.mouse.over.instrument ~ verb_bias + (1 + verb_bias | participant) + (1 | item), family="binomial", data=d)`

```r
  mutate(in.roi.3 = in.box(x,y,`#jspsych-free-sort-draggable-3.left`, `#jspsych-free-sort-draggable-3.ri
  mutate(in.roi.instrument = case_when(
    target_instrument == '#jspsych-free-sort-draggable-0' | target_instrument == '#jspsych-freesort-dra
    target_instrument == '#jspsych-free-sort-draggable-1' | target_instrument == '#jspsych-freesort-dra
    target_instrument == '#jspsych-free-sort-draggable-2' | target_instrument == '#jspsych-freesort-dra
    target_instrument == '#jspsych-free-sort-draggable-3' | target_instrument == '#jspsych-freesort-dra
  )) %>%
  mutate(in.roi.animal = case_when(
    target_animal == '#jspsych-free-sort-draggable-0' | target_animal == '#jspsych-freesort-draggable-0
    target_animal == '#jspsych-free-sort-draggable-1' | target_animal == '#jspsych-freesort-draggable-1
    target_animal == '#jspsych-free-sort-draggable-2' | target_animal == '#jspsych-freesort-draggable-2
    target_animal == '#jspsych-free-sort-draggable-3' | target_animal == '#jspsych-freesort-draggable-3
  ))


# load audio data
audio.info <- read_csv('info/audio_timing.csv')


#Calculate average animal onset
animal.onset <- audio.info %>% pull(onset_noun) %>% mean()
instrument.onset <- audio.info %>% pull(onset_instrument) %>% mean()


# Merge in audio timing information
eyetracking.data.with.roi <- eyetracking.data.with.roi %>%
  mutate(sound = str_split(audio, pattern="/", simplify = T)[,4])

eyetracking.data.with.roi <- eyetracking.data.with.roi %>%
  left_join(audio.info, by="sound")

# Add time window information
eyetracking.data.with.time.windows <- eyetracking.data.with.roi %>%
  mutate(time.window = case_when(
    t < onset_verb + 200 ~ "pre-verb-onset",
    t <= onset_noun + 200 ~ "post-verb-onset-pre-animal-onset",
    t <= onset_instrument + 200 ~ "post-animal-onset-pre-instrument-onset",
    t <= onset_instrument + 1500 + 200 ~ "post-instrument-onset",
    TRUE ~ "end"
  ),
  time.from.verb = t - onset_verb)


#Add time window
eyetracking.data.with.time.windows <- eyetracking.data.with.time.windows %>%
  mutate(t.window = floor(time.from.verb/50)*50)


#Summarize data for plotting
eyetracking.figure.2.data <- eyetracking.data.with.time.windows %>%
  filter(between(t.window , -200, 4000)) %>%
  group_by(subject, compatibility, t.window) %>%
  summarize(p.animal = mean(in.roi.animal), p.instrument = mean(in.roi.instrument)) %>%
  pivot_longer(c('p.animal', 'p.instrument'), names_to="object_type", values_to="prop_fixations") %>%
  #mutate(prop_fixations = if_else(is.na(prop_fixations), 0, prop_fixations)) %>%
  group_by(compatibility, t.window, object_type) %>%
  summarize(M=mean(prop_fixations), SE=sd(prop_fixations)/sqrt(n()))
```

```
## 'summarise()' has grouped output by 'subject', 'compatibility'. You can
## override using the '.groups' argument.
## 'summarise()' has grouped output by 'compatibility', 't.window'. You can
## override using the '.groups' argument.
```

```r
first.move = read_csv("output/E4_mouse_data.csv")
```

```
## Rows: 3078 Columns: 13
## -- Column specification ---------------------------------------------------
## Delimiter: ","
## chr (7): subject, object, type, compatibility, target_instrument, target_ani...
## dbl (5): trialID, which_loc, instrument_loc, animal_loc, is.mouse.instrument
## lgl (1): images
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```r
eyetracking.window.summary.by.trial = read_csv( "output/E4_eye-tracking_data.csv") %>%
  rename("condition" = compatibility) %>%
  select(subject, trialID, condition, time.window, prop.fixations.animal) %>%
  mutate(study = "web")
```

```
## Rows: 14667 Columns: 7
## -- Column specification ---------------------------------------------------
## Delimiter: ","
## chr (4): subject, sound, compatibility, time.window
## dbl (3): trialID, prop.fixations.animal, prop.fixations.instrument
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```r
eyetracking.window.summary.by.trial.both = bind_rows(eyetracking.window1.summary.by.trial.orig, eyetrack
    filter(time.window != "end", time.window != "pre-verb-onset", condition != "filler" )

eyetracking.window.summary.by.subj.both = bind_rows(eyetracking.window1.summary.by.trial.orig, eyetracki
    filter(time.window != "end", time.window != "pre-verb-onset", condition != "filler" ) %>%
  group_by(study, condition, time.window, subject) %>%
  summarize(mean_prop = mean(prop.fixations.animal, na.rm = T)) %>%
  mutate(time.window = factor(time.window,
                       levels = c("post-verb-onset-pre-animal-onset",
                                   "post-animal-onset-pre-instrument-onset",
                                   "post-instrument-onset"),
                       labels = c('Verb to animal \n("Rub the")',
                                   'Animal onset\n("frog with the")',
                                   'Instrument+1500ms\n(feather...")')),
         condition  = factor(condition,
                       levels = c("modifier", "equibiased", "instrument"),
                       labels = c("Modifier", "Equi-biased", "Instrument"))
         ) %>%
  ungroup()
```

```
## 'summarise()' has grouped output by 'study', 'condition', 'time.window'. You
## can override using the '.groups' argument.
```

```
eyetracking.window.summary.both = eyetracking.window.summary.by.subj.both %>%
  group_by(study, condition, time.window) %>%
  summarize(mean_prop = mean(mean_prop, na.rm = T),
            se_prop = sd(mean_prop, na.rm = T)/sqrt(n()))
```

```
## 'summarise()' has grouped output by 'study', 'condition'. You can override
## using the '.groups' argument.
```

```
ggplot(eyetracking.window.summary.by.subj.both)+
  stat_summary(aes(x=time.window, color=condition, y = mean_prop, shape = study),
               geom = 'point', fun = 'mean', size = 3)+
 stat_summary(aes(x=as.numeric(time.window), y = mean_prop, linetype = study, group=paste(condition, st
             geom = 'line', fun = 'mean', alpha = 0.5, color = "grey")+
   stat_summary(aes(x=as.numeric(time.window), y = mean_prop, color = condition, group = paste(condition
               geom = 'errorbar', fun.data = 'mean_cl_boot', width = 0.05)+
  #facet_wrap(~time.window)+
  scale_color_brewer(palette = "Set1")+
  scale_shape_manual(values = c(1,16))+
  scale_linetype_manual(values = c("dashed","solid"))+
  coord_cartesian(ylim = c(0,.8))+
  theme_classic()+
  labs(y = "Proportion of fixations", x = NULL) +      theme(
    panel.grid.minor = element_blank(),
    axis.title = element_text(size=14),        # Axis titles
    axis.text = element_text(size=14),         # Axis labels
    legend.title = element_text(size=14),      # Legend title
    legend.text = element_text(size=14)        # Legend text
  )
```

```
## Warning: Computation failed in 'stat_summary()'.
## Caused by error in 'fun.data()':
## ! The package "Hmisc" is required.
```

```
orig.data.time.window.1 <- eyetracking.window1.summary.by.trial.orig %>%
  filter(condition != "filler") %>%
  mutate(condition = fct_drop(condition))
contrasts(orig.data.time.window.1$condition) <- cbind(c(-2/3, 1/3, 1/3), c(0, -1/2, 1/2))
```

```
model.time.window.1 <- lmer(prop.fixations.animal ~ condition + (1 | subject) + (1 | trialID), data=ori
summary(model.time.window.1)
```

```
fig2<-ggplot(eyetracking.figure.2.data %>%
  mutate(compatibility = factor(compatibility,
                                levels = c("modifier", "equibiased", "instrument" ),
                                labels = c("Modifier", "Equi-biased", "Instrument" ))),
  aes(x=t.window, y=M, ymin=M-SE, ymax=M+SE, color=compatibility, fill=compatibility, linetype=object_ty
  geom_ribbon(color=NA, alpha=0.3)+
  geom_line(size=1)+
  scale_color_manual(values = c( "#377eb8", "#e41a1c","#4daf4a"))+
  scale_fill_manual(values = c("#377eb8", "#e41a1c", "#4daf4a"))+
  scale_linetype(labels = c("Animal", "Instrument") )+
```
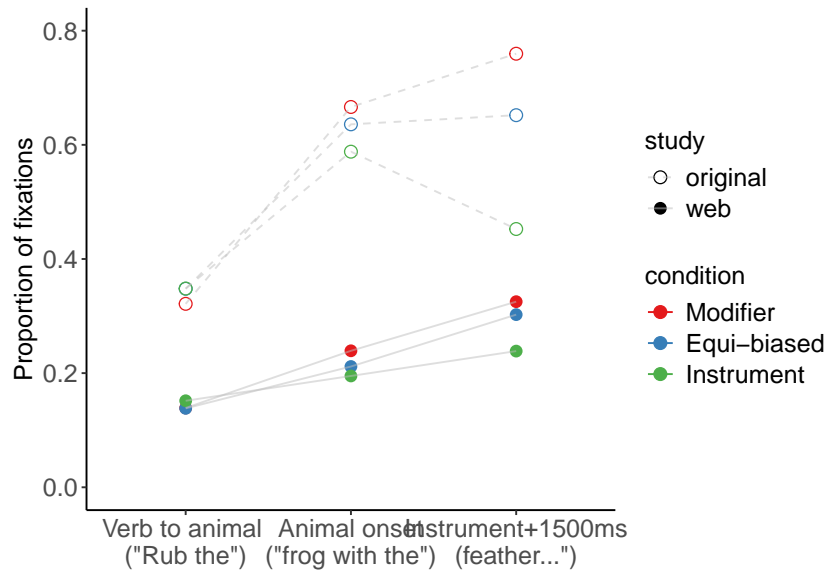
Figure 2: Proportion of target fixations by verb bias in the original dataset (Ryskin et al., 2017) and the current data collected online. Error bars reflect bootstrapped 95% CIs over subject means

```
theme_classic() +
geom_vline(xintercept = animal.onset + 200) +
geom_vline(xintercept = instrument.onset + 200)+
labs(y = "Proportion of looks", x = "Time relative to verb onset (ms)")+
guides(color = guide_legend("Verb bias"),fill = guide_legend("Verb bias"), linetype = guide_legend("Ga
  panel.grid.minor = element_blank(),
  axis.title = element_text(size=14),      # Axis titles
  axis.text = element_text(size=14),       # Axis labels
  legend.title = element_text(size=14),    # Legend title
  legend.text = element_text(size=14)      # Legend text
)
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
fig2
```

```
saveRDS(fig2, "output/ETfig.rds")
ggsave("output/ETfig.png", plot = fig2, height = 4, width = 6)
```

```
#Summarize fixations on target and instrument
eyetracking.window.summary.by.trial <- eyetracking.data.with.time.windows %>%
  group_by(subject, trialID, sound, compatibility, time.window) %>%
  summarize(prop.fixations.animal = sum(in.roi.animal) / n(),
            prop.fixations.instrument = sum(in.roi.instrument) / n()) %>%
  mutate(compatibility = factor(compatibility))
```
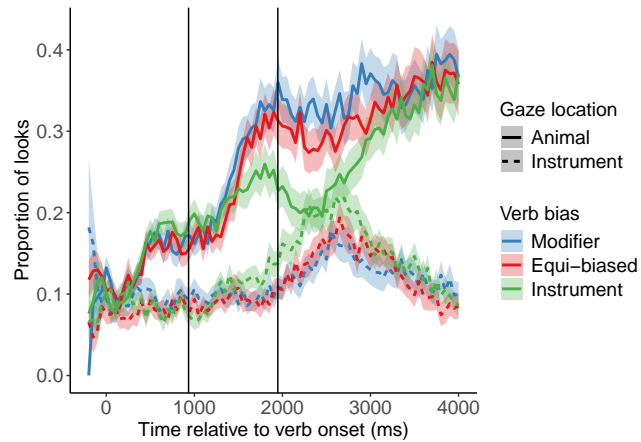
Figure 3: Timecourse of eye-gaze to target animal and target instrument by verb bias condition. Vertical lines indicate average onsets of animal and instrument offset by 200ms.

```
## 'summarise()' has grouped output by 'subject', 'trialID', 'sound',
## 'compatibility'. You can override using the '.groups' argument.
```

```r
write_csv(eyetracking.window.summary.by.trial, "output/E4_eye-tracking_data.csv")
```

```r
# Add orthogonal contrasts to model
contrasts(eyetracking.window.summary.by.trial$compatibility) <- cbind(c(-2/3, 1/3, 1/3), c(0, -1/2, 1/2)
```

```r
data.time.window.1 <- eyetracking.window.summary.by.trial %>% filter(time.window == "post-verb-onset-pr
```

```r
model.time.window.1 <- lmer(prop.fixations.animal ~ compatibility + (1  | subject) + (1 | trialID), data
                         control = lmerControl(optimizer = "bobyqa",
                                               optCtrl = list(maxfun = 2e6)))
```

```r
data.time.window.2 <- eyetracking.window.summary.by.trial %>% filter(time.window == "post-animal-onset-p
```

```r
model.time.window.2 <- lmer(prop.fixations.animal ~ compatibility + (1  | subject) + (1 | trialID), data
```

```r
data.time.window.3 <- eyetracking.window.summary.by.trial %>% filter(time.window == "post-instrument-ons
```

```r
model.time.window.3 <- lmer(prop.fixations.animal ~ compatibility + (1 | subject) + (1 | trialID), data=
```

```r
E4_ET_model1_tab = broom.mixed::tidy(model.time.window.1)
E4_ET_model2_tab = broom.mixed::tidy(model.time.window.2)
E4_ET_model3_tab = broom.mixed::tidy(model.time.window.3)

E4_ET_model1_c1 = E4_ET_model1_tab %>% filter(term == "compatibility1")
E4_ET_model1_c2 = E4_ET_model1_tab %>% filter(term == "compatibility2")

E4_ET_model2_c1 = E4_ET_model2_tab %>% filter(term == "compatibility1")
E4_ET_model2_c2 = E4_ET_model2_tab %>% filter(term == "compatibility2")

E4_ET_model3_c1 = E4_ET_model3_tab %>% filter(term == "compatibility1")
E4_ET_model3_c2 = E4_ET_model3_tab %>% filter(term == "compatibility2")
```

Mixed-effects linear regression models were used to predict the proportions of fixations to the target animal vs. instrument within each time window with the verb bias condition as an orthogonally contrast-coded (instrument vs. equi & modifier: inst = -2/3, equi = 1/3, mod = 1/3; equi vs. modifier: inst = 0, equi = -1/2, mod = 1/2 ) fixed effect. Participants and items were entered as varying intercepts.[2]

In the *verb-to-noun* window, participants did not look more at the target animal in any of the verb bias conditions (Instrument vs. Equi and Modifier: $b$ = -0.01, $SE$ = 0.02, $p$ = 0.59; Equi vs. Modifier: $b$ = 0, $SE$ = 0.02, $p$ = 1 ).

In the *noun-to-instrument* window, participants looked more at the target animal in the modifier-biased condition and equi-biased conditions relative to the instrument-biased condition ( $b$ = 0.03, $SE$ = 0.01, $p$ < 0.01) and in the modifier biased relative to the equi-biased condition ( $b$ = 0.02, $SE$ = 0.01, $p$ < 0.05).

In the *post-instrument* window, participants looked more at the target animal in the modifier-biased condition and the equi-biased conditions relative to the instrument-biased condition ( $b$ = 0.08, $SE$ = 0.02, $p$ < 0.01) but not significantly so in the modifier biased condition relative to the equi-biased condition ( $b$ = 0.03, $SE$ = 0.02, $p$ = 0.15).

---

[2]`lme4` syntax: `lmer(prop.fix.target.animal ~ verb_bias + (1 + verb_bias | participant) + (1 | item), data=d)`. A model with by-participant varying slopes for verb bias condition was first attempted but did not converge.