Kan\_Analysis

Marton Kovacs

19 October, 2020

# Load packages

library(tidyverse)  
library(osfr)  
library(readxl)  
library(readbulk)  
library(ez)  
library(psych)

# Load custom functions

source("R/utils.R")  
source("R/arcsine\_transformation.R")

# Exp1

## Outcome neutral test

### Test sentence reading time

#### Import data

outcome\_neutral\_sentence\_data <- read\_tsv("Data/Processed/Exp1/Kan\_Processed\_Exp1\_OutcomeNeutral\_Sentence\_data.tsv")

## Parsed with column specification:  
## cols(  
## sentence\_region\_no = col\_double(),  
## participant\_id = col\_double(),  
## Congruent = col\_double(),  
## Incongruent = col\_double(),  
## sentence\_region\_type = col\_character()  
## )

#### Nest the data

outcome\_neutral\_sentence\_data <-  
 outcome\_neutral\_sentence\_data %>%   
 nest(data = c(participant\_id, Congruent, Incongruent))

#### Performing the t-tests

We calculate the t test for each sentence region separately.

outcome\_neutral\_sentence\_data <-  
 outcome\_neutral\_sentence\_data %>%   
 mutate(t\_value = map\_dbl(data,  
 ~ t.test(.$Incongruent, .$Congruent, paired = TRUE, alternative = "greater") %>%   
 broom::tidy() %>%   
 pull(statistic)),  
 df = map\_dbl(data,  
 ~ t.test(.$Incongruent, .$Congruent, paired = TRUE, alternative = "greater") %>%   
 broom::tidy() %>%   
 pull(parameter)),  
 p\_value = map\_dbl(data,  
 ~ t.test(.$Incongruent, .$Congruent, paired = TRUE, alternative = "greater") %>%   
 broom::tidy() %>%   
 pull(p.value)),  
 raw\_effect = map\_dbl(data,  
 ~ t.test(.$Incongruent, .$Congruent, paired = TRUE, alternative = "greater") %>%   
 broom::tidy() %>%   
 pull(estimate)))

#### Calculating the SE

outcome\_neutral\_sentence\_data <-  
 outcome\_neutral\_sentence\_data %>%   
 mutate(se = map2\_dbl(raw\_effect, t\_value,  
 ~ abs(.x / .y)))

#### Calculating the Bayes factor

We are calculating the Bf for all the regions, but our main interest is the the temporarily ambiguous and disambiguating region.

Checking the number of observations for each sentence region.

map(outcome\_neutral\_sentence\_data$data, nrow)

## [[1]]  
## [1] 132  
##   
## [[2]]  
## [1] 132  
##   
## [[3]]  
## [1] 132  
##   
## [[4]]  
## [1] 132  
##   
## [[5]]  
## [1] 132  
##   
## [[6]]  
## [1] 132  
##   
## [[7]]  
## [1] 132

Creating a plots to visualize the difference.

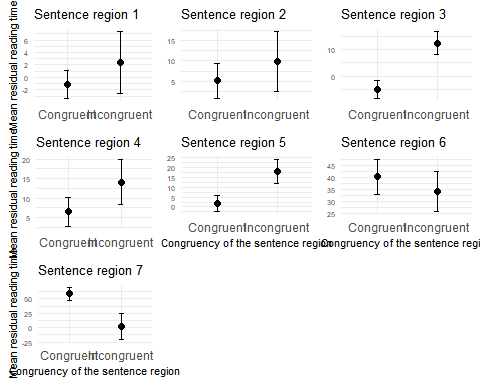
We are creating a plot function for visualization.

sentence\_region\_plot <- function(df, name) {  
 df %>%  
 select(Congruent, Incongruent) %>%   
 gather(key = "congruency", value = "resid\_mean\_reading\_time") %>%   
 group\_by(congruency) %>%   
 summarise(n = n(),  
 mean = mean(resid\_mean\_reading\_time),  
 sd = sd(resid\_mean\_reading\_time, na.rm = T),  
 se = sd / sqrt(n)) %>%   
 ggplot() +  
 aes(x = congruency,  
 y = mean) +  
 geom\_point(size = 2) +  
 geom\_errorbar(aes(ymin = mean - (se \* 1.96),  
 ymax = mean + (se \* 1.96)),  
 width = .1) +  
 labs(title = paste("Sentence region", name),  
 x = "Congruency of the sentence region",  
 y = "Mean residual reading time") +  
 theme\_minimal() +  
 theme(  
 title = element\_text(size = 8),  
 # axis.title = element\_text(size = 5),  
 axis.text.y = element\_text(size = 5)  
 # strip.text.x = element\_text(size = 5)  
 )  
}  
  
outcome\_neutral\_sentence\_data <-   
 outcome\_neutral\_sentence\_data %>%   
 mutate(sentence\_plot = map2(data, as.character(sentence\_region\_no),   
 ~ sentence\_region\_plot(.x, .y)))

## `summarise()` ungrouping output (override with `.groups` argument)  
## `summarise()` ungrouping output (override with `.groups` argument)  
## `summarise()` ungrouping output (override with `.groups` argument)  
## `summarise()` ungrouping output (override with `.groups` argument)  
## `summarise()` ungrouping output (override with `.groups` argument)  
## `summarise()` ungrouping output (override with `.groups` argument)  
## `summarise()` ungrouping output (override with `.groups` argument)

Create one plot to show all of the sentence plots together.

plot\_1 <- outcome\_neutral\_sentence\_data$sentence\_plot[[1]] +  
 theme(axis.title.x = element\_blank())  
plot\_2 <- outcome\_neutral\_sentence\_data$sentence\_plot[[2]] +  
 theme(axis.title.x = element\_blank(),  
 axis.title.y = element\_blank())  
plot\_3 <- outcome\_neutral\_sentence\_data$sentence\_plot[[3]] +  
 theme(axis.title.x = element\_blank(),  
 axis.title.y = element\_blank())  
plot\_4 <- outcome\_neutral\_sentence\_data$sentence\_plot[[4]] +  
 theme(axis.title.x = element\_blank())  
plot\_5 <- outcome\_neutral\_sentence\_data$sentence\_plot[[5]] +  
 theme(axis.title.y = element\_blank())  
plot\_6 <- outcome\_neutral\_sentence\_data$sentence\_plot[[6]] +  
 theme(axis.title.y = element\_blank())  
plot\_7 <- outcome\_neutral\_sentence\_data$sentence\_plot[[7]]  
  
reading\_time\_plot <-  
 cowplot::plot\_grid(plot\_1, plot\_2, plot\_3,  
 plot\_4, plot\_5, plot\_6,  
 plot\_7)  
  
ggsave("Kan\_Exp1\_ReadingTime\_Plot.png", device = "png", plot = reading\_time\_plot, dpi = 300, width = 208, height = 118, units = "mm")  
  
reading\_time\_plot



Assigning SD theory for the Bayes factor analysis.

outcome\_neutral\_sentence\_data <-   
 outcome\_neutral\_sentence\_data %>%   
 mutate(sd\_theory = case\_when(sentence\_region\_no == 3 ~ 40L,  
 sentence\_region\_no == 4 ~ 18L,  
 TRUE ~ 18L))

Running the Bf analysis for all sentence regions.

outcome\_neutral\_sentence\_data <-   
 outcome\_neutral\_sentence\_data %>%   
 mutate(bf = pmap\_dbl(list(se, raw\_effect, df, sd\_theory),  
 ~ Bf(sd = ..1,  
 obtained = ..2,  
 dfdata = ..3,  
 meanoftheory = 0,  
 sdtheory = ..4,  
 dftheory = 10^10,  
 tail = 1)))

Create table with results.

outcome\_neutral\_sentence\_table <-  
 outcome\_neutral\_sentence\_data %>%   
 mutate(raw\_effect = round(raw\_effect, 2),  
 bf = round(bf, 2),  
 t\_value = round(t\_value, 2),  
 p\_value = round(p\_value, 2),  
 se = round(se, 2)) %>%   
 rename(`Sentence region id` = sentence\_region\_no,  
 `t value` = t\_value,  
 DF = df,  
 `p value` = p\_value,  
 `Raw effect` = raw\_effect,  
 SE = se,  
 B = bf) %>%   
 select(-sentence\_region\_type,  
 -data,  
 -sentence\_plot,  
 -sd\_theory)  
  
papaja::apa\_table(  
 outcome\_neutral\_sentence\_table,  
 caption = "Results of the Comparison of the Mean Residual Reading Times in Each Sentence Region",  
 escape = TRUE  
)

(#tab:unnamed-chunk-12)

*Results of the Comparison of the Mean Residual Reading Times in Each Sentence Region*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sentence region id | t value | DF | p value | Raw effect | SE | B |
| 1.00 | 1.25 | 131.00 | 0.11 | 3.55 | 2.83 | 0.60 |
| 2.00 | 1.25 | 131.00 | 0.11 | 4.72 | 3.78 | 0.77 |
| 3.00 | 6.40 | 131.00 | 0.00 | 17.26 | 2.70 | 7,870,174.15 |
| 4.00 | 2.05 | 131.00 | 0.02 | 7.56 | 3.69 | 2.88 |
| 5.00 | 3.98 | 131.00 | 0.00 | 16.32 | 4.10 | 557.44 |
| 6.00 | -1.45 | 131.00 | 0.92 | -6.24 | 4.32 | 0.10 |
| 7.00 | -4.57 | 131.00 | 1.00 | -56.84 | 12.44 | 0.13 |

Calculating the robustness region for temporarily ambiguous region.

temporarily\_data <-   
 outcome\_neutral\_sentence\_data %>%   
 filter(sentence\_region\_type == "temporarly ambiguous")  
  
sd <-   
 temporarily\_data %>%   
 pull(se)  
  
obtained <-   
 temporarily\_data %>%   
 pull(raw\_effect)  
  
df <-   
 temporarily\_data %>%   
 pull(df)  
  
# lower  
Bf(sd = sd,  
 obtained = obtained,  
 dfdata = df,  
 meanoftheory = 0,  
 sdtheory = 0.6,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.056385

# upper  
Bf(sd = sd,  
 obtained = obtained,  
 dfdata = df,  
 meanoftheory = 0,  
 sdtheory = 6125,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.019518

Calculating the robustness region for disambuguating region.

disambiguating\_data <-   
 outcome\_neutral\_sentence\_data %>%   
 filter(sentence\_region\_type == "disambiguating ambiguous")  
  
sd <-   
 disambiguating\_data %>%   
 pull(se)  
  
obtained <-   
 disambiguating\_data %>%   
 pull(raw\_effect)  
  
df <-   
 disambiguating\_data %>%   
 pull(df)  
  
# lower  
Bf(sd = sd,  
 obtained = obtained,  
 dfdata = df,  
 meanoftheory = 0,  
 sdtheory = 17.2,  
 dftheory = 10^10,  
 tail = 1)

## [1] 2.987363

# upper  
Bf(sd = sd,  
 obtained = obtained,  
 dfdata = df,  
 meanoftheory = 0,  
 sdtheory = 173,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3352645

## Crucial tests

### Reaction time analysis

#### Import data

main\_analysis\_rt\_data <- vroom::vroom("Data/Processed/Exp1/Kan\_Processed\_Exp1\_Main\_Rt\_data.tsv")

## Rows: 4,738  
## Columns: 16  
## Delimiter: "\t"  
## chr [ 4]: lab, stim\_type, stimulus, stim\_color  
## dbl [12]: participant\_id, trial\_id, response\_time, is\_previous\_congruent, is\_congruent, is...  
##   
## Use `spec()` to retrieve the guessed column specification  
## Pass a specification to the `col\_types` argument to quiet this message

#### Exploratory data analysis

The number of response per participant.

main\_analysis\_rt\_data %>%   
 count(participant\_id)

## # A tibble: 132 x 2  
## participant\_id n  
## <dbl> <int>  
## 1 109 32  
## 2 1010 34  
## 3 1012 42  
## 4 1013 40  
## 5 1014 30  
## 6 1015 30  
## 7 1016 36  
## 8 1017 37  
## 9 1018 14  
## 10 1021 32  
## # ... with 122 more rows

The number of participants.

main\_analysis\_rt\_data %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 132

#### Figures

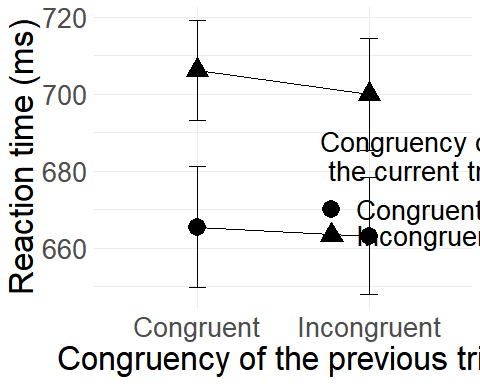
The congruency sequence effect for the reaction time responses.

kan\_exp1\_rt\_cse\_plot <-  
 main\_analysis\_rt\_data %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent")) %>%  
 group\_by(participant\_id, is\_previous\_congruent, is\_congruent) %>%   
 summarise(participant\_mean\_rt = mean(response\_time, na.rm = T)) %>%  
 group\_by(is\_previous\_congruent, is\_congruent) %>%   
 summarise(N = n(),  
 mean\_rt = mean(participant\_mean\_rt, na.rm = T),  
 sd\_rt = sd(participant\_mean\_rt, na.rm = T),  
 se\_rt = sd\_rt / sqrt(N)) %>%   
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_rt,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 6) +  
 geom\_errorbar(aes(ymin = mean\_rt - (se\_rt \* 1.96),  
 ymax = mean\_rt + (se\_rt \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Reaction time (ms)") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 25),  
 axis.text = element\_text(size = 20),  
 legend.text = element\_text(size = 20),  
 legend.title = element\_text(size = 20),  
 legend.position = c(.85, .4))

## `summarise()` regrouping output by 'participant\_id', 'is\_previous\_congruent' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent' (override with `.groups` argument)

kan\_exp1\_rt\_cse\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp1\_Rt\_Cse\_Plot.png", width = 14.4, height = 8, plot = kan\_exp1\_rt\_cse\_plot)

#### Data preprocessing

We are calculating the mean reaction time per participant per condition (ci, ic, cc, ii).

main\_analysis\_rt\_aggregate <-  
 main\_analysis\_rt\_data %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent) %>%   
 summarise(rt\_conditional\_mean = mean(response\_time, na.rm = T)) %>%  
 ungroup() %>%   
 mutate(is\_congruent = as.factor(is\_congruent),  
 is\_previous\_congruent = as.factor(is\_previous\_congruent),  
 participant\_id = as.factor(participant\_id))

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

The number of conditions per participant. Because we excluded the incorrect trials, it can happen that a participant does no have reaction time responses from all the 4 conditions.

main\_analysis\_rt\_aggregate %>%   
 count(participant\_id) %>%   
 arrange(n)

## # A tibble: 132 x 2  
## participant\_id n  
## <fct> <int>  
## 1 109 4  
## 2 1010 4  
## 3 1012 4  
## 4 1013 4  
## 5 1014 4  
## 6 1015 4  
## 7 1016 4  
## 8 1017 4  
## 9 1018 4  
## 10 1021 4  
## # ... with 122 more rows

Save the [participant\_id] of those participants who do not have data from all the four conditions.

missing\_condition <-  
 main\_analysis\_rt\_aggregate %>%   
 count(participant\_id) %>%   
 filter(n != 4) %>%  
 select(participant\_id)

The number of participants who has missing conditions.

nrow(missing\_condition)

## [1] 0

Calculating the raw congruency, previous congruency and interaction effect for each participant.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = rt\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((ci + ii) / 2) - ((cc + ic) / 2),  
 previous\_congruency\_effect = ((ci + cc) / 2) - ((ii + ic) / 2),  
 interaction\_effect = (ci - cc) - (ii - ic))

Running the repeated measure ANOVA.

anova\_rt <-  
 aov(rt\_conditional\_mean ~ is\_congruent \* is\_previous\_congruent + Error(participant\_id / (is\_congruent \* is\_previous\_congruent)), data = main\_analysis\_rt\_aggregate)

Saving the F value and Df for each effect of interest.

anova\_rt\_f\_value <-  
 anova\_rt %>%   
 broom::tidy() %>%   
 select(term, statistic) %>%   
 transmute(term = case\_when(term == "is\_congruent" ~ "main\_effect\_congruent",  
 term == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 term == "is\_congruent:is\_previous\_congruent" ~ "interaction\_effect",  
 term == "Residuals" ~ term),  
 f\_value = statistic) %>%   
 filter(!is.na(f\_value))

Show the results of the ANOVA.

anova\_rt %>%   
 broom::tidy() %>%   
 rename(f\_value = statistic,  
 p\_value = p.value) %>%   
 mutate(f\_value = round(f\_value, 2),  
 p\_value = round(p\_value, 2)) %>%   
 filter(!is.na(f\_value))

## # A tibble: 3 x 7  
## stratum term df sumsq meansq f\_value p\_value  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 participant\_id:is\_congr~ is\_congruent 1 1.98e5 1.98e5 87.0 0   
## 2 participant\_id:is\_previ~ is\_previous\_cong~ 1 2.42e3 2.42e3 1.97 0.16  
## 3 participant\_id:is\_congr~ is\_congruent:is\_~ 1 4.98e2 4.98e2 0.35 0.56

Calculating the raw effects summarized for every participant.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T),  
 n\_participant = n()) %>%  
 gather(key = "term", value = "raw\_effect", -n\_participant)

Joining the calculated F values and Dfs with the raw effects.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%  
 inner\_join(., anova\_rt\_f\_value, by = "term")

Calculating the SE from the raw effect and the F value.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

#### Running the analysis

Saving the values needed for the analysis.

# Obtained sd  
interaction\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# Obtained effect  
interaction\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# Df  
interaction\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(n\_participant)

Running the Bf analysis with the model described in the paper.

Bf(sd = interaction\_se,  
 obtained = interaction\_effect,  
 dfdata = interaction\_df - 1,  
 meanoftheory = 0,  
 sdtheory = 30,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3655853

Robustness region.

# Lower  
## The lower boundary is 0  
  
# Upper  
Bf(sd = interaction\_se,  
 obtained = interaction\_effect,  
 dfdata = interaction\_df - 1,  
 meanoftheory = 0,  
 sdtheory = 33,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3346875

### Outcome neutral test

#### Test Stroop effect

This is an outcome neutral test, therefore, the results of this test do not effect our main conclusions.

Saving the values needed for the analysis.

# Obtained sd  
main\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(se)  
  
# Obtained effect  
main\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(raw\_effect)  
  
# Df  
main\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(n\_participant)

Running the Bf analysis with the model described in the paper.

Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df - 1,  
 meanoftheory = 0,  
 sdtheory = 28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 4.613872e+13

Robustness region.

# Lower  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df - 1,  
 meanoftheory = 0,  
 sdtheory = 0.8,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.098987

# Upper  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df - 1,  
 meanoftheory = 0,  
 sdtheory = 14510,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.015894

### Supporting test of interest

#### Test previous congruency effect

This is a supporting test of interest. We use the same prior as for the main Stroop effect test.

Saving the values needed for the analysis.

# Obtained sd  
main\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(se)  
  
# Obtained effect  
main\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(raw\_effect)  
  
# Df  
main\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(n\_participant)

Running the Bf analysis with the model described in the paper.

Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df - 1,  
 meanoftheory = 0,  
 sdtheory = 28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.5260381

Robustness region.

# Lower  
## 0 as the Bf is inconclusive  
  
# Upper  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df - 1,  
 meanoftheory = 0,  
 sdtheory = 44,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.340425

### Accuracy analysis

#### Import data

main\_analysis\_acc\_data <-  
 read\_tsv("Data/Processed/Exp1/Kan\_Processed\_Exp1\_Main\_Acc\_data.tsv")

## Parsed with column specification:  
## cols(  
## participant\_id = col\_double(),  
## lab = col\_character(),  
## trial\_id = col\_double(),  
## stim\_type = col\_character(),  
## stimulus = col\_character(),  
## stim\_color = col\_character(),  
## response\_time = col\_double(),  
## is\_previous\_congruent = col\_double(),  
## is\_congruent = col\_double(),  
## is\_previous\_correct = col\_double(),  
## is\_correct = col\_double()  
## )

#### Exploratory data analysis

The number of response per participant.

main\_analysis\_acc\_data %>%   
 group\_by(participant\_id) %>%   
 count()

## # A tibble: 132 x 2  
## # Groups: participant\_id [132]  
## participant\_id n  
## <dbl> <int>  
## 1 109 42  
## 2 1010 42  
## 3 1012 42  
## 4 1013 42  
## 5 1014 42  
## 6 1015 42  
## 7 1016 42  
## 8 1017 42  
## 9 1018 42  
## 10 1021 42  
## # ... with 122 more rows

The number of participants.

main\_analysis\_acc\_data %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 132

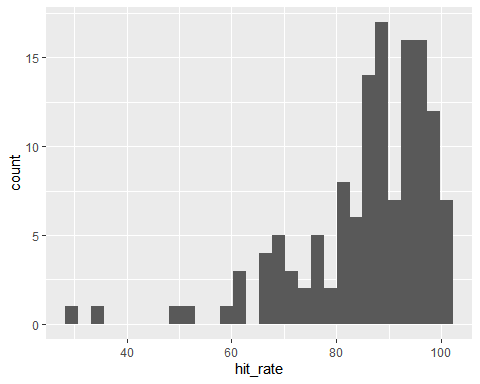
#### The distribution of hit rates

Calculating the hit rate for each participant.

main\_analysis\_acc\_data %>%   
 group\_by(participant\_id) %>%   
 summarise(acc = mean(is\_correct),  
 hit\_rate = acc \* 100) %>%   
 arrange(acc) %>%   
 ggplot() +  
 aes(x = hit\_rate) %>%   
 geom\_histogram()

## `summarise()` ungrouping output (override with `.groups` argument)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



#### Figures

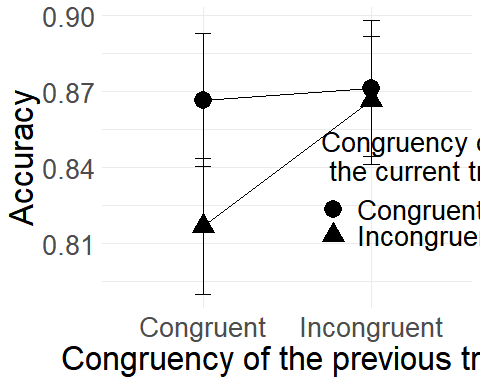
The congruency sequence effect for the accuracy responses.

kan\_exp1\_acc\_cse\_plot <-  
 main\_analysis\_acc\_data %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent")) %>%  
 group\_by(participant\_id, is\_previous\_congruent, is\_congruent) %>%   
 summarise(participant\_mean\_acc = mean(is\_correct, na.rm = T)) %>%   
 group\_by(is\_previous\_congruent, is\_congruent) %>%   
 summarise(N = n(),  
 mean\_acc = mean(participant\_mean\_acc, na.rm = T),  
 sd\_acc = sd(participant\_mean\_acc, na.rm = T),  
 se\_acc = sd\_acc / sqrt(N)) %>%  
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_acc,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 6) +  
 geom\_errorbar(aes(ymin = mean\_acc - (se\_acc \* 1.96),  
 ymax = mean\_acc + (se\_acc \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Accuracy") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 25),  
 axis.text = element\_text(size = 20),  
 legend.text = element\_text(size = 20),  
 legend.title = element\_text(size = 20),  
 legend.position = c(.85, .4))

## `summarise()` regrouping output by 'participant\_id', 'is\_previous\_congruent' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent' (override with `.groups` argument)

kan\_exp1\_acc\_cse\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp1\_Acc\_Cse\_Plot.png", width = 14.4, height = 8, plot = kan\_exp1\_acc\_cse\_plot)

#### Data preprocessing

We are calculating the accuracy for each condition (cc, ci, ic, ii) for each participant.

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc\_data %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent) %>%   
 summarise(acc\_conditional\_mean = mean(is\_correct, na.rm = T)) %>%   
 ungroup()

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

We convert the predictor variables for the ANOVA to factors.

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc\_aggregate %>%   
 mutate(is\_congruent = as\_factor(is\_congruent),  
 is\_previous\_congruent = as\_factor(is\_previous\_congruent),  
 participant\_id = as\_factor(participant\_id))

We are running the ANOVA described in the paper.

anova\_acc <-  
 ezANOVA(data = main\_analysis\_acc\_aggregate, dv = acc\_conditional\_mean, wid = participant\_id, within = .(is\_congruent, is\_previous\_congruent), type = 3)

We are converting the output of the ANOVA to tibble format.

anova\_acc <-  
 as\_tibble(anova\_acc$ANOVA)

Show the results of the ANOVA.

anova\_acc

## # A tibble: 3 x 7  
## Effect DFn DFd F p `p<.05` ges  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 is\_congruent 1 131 8.52 0.00413 \* 0.00788  
## 2 is\_previous\_congruent 1 131 10.6 0.00142 \* 0.00772  
## 3 is\_congruent:is\_previous\_congruent 1 131 6.50 0.0119 \* 0.00536

We are extracting the F value and the df for each term in the ANOVA.

anova\_acc\_f\_value <-  
 anova\_acc %>%   
 filter(Effect %in% c("is\_congruent",  
 "is\_previous\_congruent",  
 "is\_congruent:is\_previous\_congruent")) %>%   
 transmute(term = case\_when(Effect == "is\_congruent" ~ "main\_effect\_congruent",  
 Effect == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 Effect == "is\_congruent:is\_previous\_congruent" ~ "interaction\_effect"),  
 f\_value = F,  
 df = DFd)

We calculate the raw effects for each term for each participant.

main\_analysis\_acc\_participant\_level\_raw\_effect <-  
 main\_analysis\_acc\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = acc\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((cc + ic) / 2) - ((ci + ii) / 2),  
 previous\_congruency\_effect = ((ci + cc) / 2) - ((ii + ic) / 2),  
 interaction\_effect = (cc - ci) - (ic - ii))

We are aggregating the raw effects of the participants.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_participant\_level\_raw\_effect %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T)) %>%   
 gather(key = "term", value = "raw\_effect")

We join the F value and Dfs to the raw effects for each term.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%  
 inner\_join(., anova\_acc\_f\_value, by = "term")

From the raw effect and the F value we calculate the SE.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

#### Running the analysis

Saving the values needed for the analysis.

# Obtained sd  
interaction\_se <-   
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# Obtained effect  
interaction\_effect <-   
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# Df  
interaction\_df <-   
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(df)

Running the Bf analysis with the model described in the paper.

Bf(sd = interaction\_se,  
 obtained = interaction\_effect,  
 dfdata = interaction\_df,  
 meanoftheory = 0,  
 sdtheory = .03,  
 dftheory = 10^10,  
 tail = 1)

## [1] 10.59967

Robustness region.

# Lower  
Bf(sd = interaction\_se,  
 obtained = interaction\_effect,  
 dfdata = interaction\_df,  
 meanoftheory = 0,  
 sdtheory = .0095,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.111778

# Upper  
Bf(sd = interaction\_se,  
 obtained = interaction\_effect,  
 dfdata = interaction\_df,  
 meanoftheory = 0,  
 sdtheory = .28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.034073

#### Arcsine transformed accuracy

We run the calculations with arcsine transformed proportions. We will include these findings in the supplementary materials.

The calculation is the same as before expect that we use arcsine transformation on the conditional means. We are going to use a formula describe in the original paper to calculate the arcsine transformed values, but we will also use a more common way to calculate the transformed values. Copying the whole accuracy analysis part would take up a lot of space in the code and would be error prone therefore, we created simple functions to calculate the Bf with the arcsine transformed data.

The method the original paper used to transform the raw proportions.

arcsine\_original\_method <- quo(asin(( 2 \* acc\_conditional\_mean) - 1))

A more common way to calculate.

arcsine\_common\_method <- quo(asin(sqrt(acc\_conditional\_mean)))

Calculating the Bf with the two methods.

arcsine\_two\_way(  
 df = main\_analysis\_acc\_data,  
 expression = arcsine\_original\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

## [1] 3.06691

arcsine\_two\_way(  
 df = main\_analysis\_acc\_data,  
 expression = arcsine\_common\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

## [1] 8.006314

**Note: The accuracy main effect of the current trial was not significant in the original study so we do not attempt to replicate it.**

### Supporting tests of interest

As the presence of the interaction effect in the accuracy analysis was supported by the Bayes factor we run further tests to investigate the CSE.

We run further comparisons to test whether congruency sequence effect modulated the performance on congruent, on incongruent trials or on both. To this aim, we test whether the type of the preceding trial modulates the accuracy rates of congruent and incongruent trials. To model the H1s, we utilize the parameters of the test of the interaction.

Comparing cC and iC trials.

cong\_prev <-   
 main\_analysis\_acc\_participant\_level\_raw\_effect %>%   
 t.test(.$cc, .$ic, paired = TRUE, alternative = "greater", data = .) %>%   
 broom::tidy()  
  
cong\_prev$estimate <- unname(cong\_prev$estimate)  
cong\_prev$statistic <- unname(cong\_prev$statistic)  
cong\_prev$parameter <- unname(cong\_prev$parameter)

Running the Bf analysis with the model described in the paper.

Bf(sd = abs(cong\_prev$estimate / cong\_prev$statistic),  
 obtained = cong\_prev$estimate,  
 dfdata = cong\_prev$parameter,  
 meanoftheory = 0,  
 sdtheory = .03,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3004279

Robustness region.

# Lower  
Bf(sd = abs(cong\_prev$estimate / cong\_prev$statistic),  
 obtained = cong\_prev$estimate,  
 dfdata = cong\_prev$parameter,  
 meanoftheory = 0,  
 sdtheory = .027,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3265478

# Upper  
## Inf

Comparing iI and Ci trials.

incong\_prev <-   
 main\_analysis\_acc\_participant\_level\_raw\_effect %>%   
 t.test(.$ii, .$ci, paired = TRUE, alternative = "greater", data = .) %>%   
 broom::tidy()  
  
incong\_prev$estimate <- unname(incong\_prev$estimate)  
incong\_prev$statistic <- unname(incong\_prev$statistic)  
incong\_prev$parameter <- unname(incong\_prev$parameter)

Running the Bf analysis with the model described in the paper.

Bf(sd = abs(incong\_prev$estimate / incong\_prev$statistic),  
 obtained = incong\_prev$estimate,  
 dfdata = incong\_prev$parameter,  
 meanoftheory = 0,  
 sdtheory = .03,  
 dftheory = 10^10,  
 tail = 1)

## [1] 960.6346

Robustness region.

# Lower  
Bf(sd = abs(incong\_prev$estimate / incong\_prev$statistic),  
 obtained = incong\_prev$estimate,  
 dfdata = incong\_prev$parameter,  
 meanoftheory = 0,  
 sdtheory = .0036,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.08249

# Upper  
Bf(sd = abs(incong\_prev$estimate / incong\_prev$statistic),  
 obtained = incong\_prev$estimate,  
 dfdata = incong\_prev$parameter,  
 meanoftheory = 0,  
 sdtheory = 14.3,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.070943

# Exp 2

## Reaction time analysis

### Import data

main\_analysis\_rt\_data <-  
 read\_tsv("Data/Processed/Exp2/Kan\_Processed\_Exp2\_Main\_Rt\_data.tsv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## participant\_id = col\_character(),  
## lab = col\_character(),  
## procedure = col\_character(),  
## conflict\_condition = col\_character(),  
## stim\_type = col\_character(),  
## stimulus = col\_character(),  
## stim\_color = col\_character(),  
## reversal = col\_logical(),  
## reversal\_group = col\_character()  
## )

## See spec(...) for full column specifications.

### Exploratory data analysis

The number of response per participant.

main\_analysis\_rt\_data %>%   
 count(participant\_id)

## # A tibble: 163 x 2  
## participant\_id n  
## <chr> <int>  
## 1 0001 33  
## 2 0002 41  
## 3 0003 33  
## 4 0004 24  
## 5 0005 23  
## 6 0006 36  
## 7 0007 45  
## 8 0008 33  
## 9 0009 39  
## 10 0010 56  
## # ... with 153 more rows

The number of participants.

main\_analysis\_rt\_data %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 163

Comparing the descriptive statistics of the low and high reversal groups. These values are calculated from the mean number of reversals for each participant. The calculation based on trial level reversal times can be found in the **Kan\_Raw\_Processed** file.

main\_analysis\_rt\_data %>%   
 group\_by(reversal\_group) %>%   
 summarise(n = n(),  
 all\_mean\_reversal = mean(mean\_reversal),  
 all\_sd\_reversal = sd(mean\_reversal, na.rm = TRUE))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 4  
## reversal\_group n all\_mean\_reversal all\_sd\_reversal  
## <chr> <int> <dbl> <dbl>  
## 1 high 3082 18.8 16.3   
## 2 low 3365 3.00 2.29

### Figures

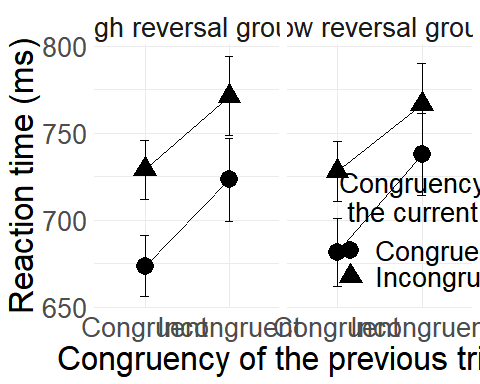
The congruency sequence effect for the reaction time responses with reversal groups.

kan\_exp2\_rt\_cse\_rev\_plot <-  
 main\_analysis\_rt\_data %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent"),  
 reversal\_group = case\_when(reversal\_group == "low" ~ "Low reversal group",  
 reversal\_group == "high" ~ "High reversal group")) %>%  
 group\_by(participant\_id, reversal\_group, is\_previous\_congruent, is\_congruent) %>%   
 summarise(par\_mean\_rt = mean(response\_time, na.rm = T)) %>%   
 group\_by(is\_previous\_congruent, is\_congruent, reversal\_group) %>%   
 summarise(N = n(),  
 mean\_rt = mean(par\_mean\_rt, na.rm = T),  
 sd\_rt = sd(par\_mean\_rt, na.rm = T),  
 se\_rt = sd\_rt / sqrt(N)) %>%  
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_rt,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 6) +  
 geom\_errorbar(aes(ymin = mean\_rt - (se\_rt \* 1.96),  
 ymax = mean\_rt + (se\_rt \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Reaction time (ms)") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 facet\_wrap( ~ reversal\_group) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 25),  
 axis.text = element\_text(size = 20),  
 legend.text = element\_text(size = 20),  
 legend.title = element\_text(size = 20),  
 strip.text.x = element\_text(size = 20),  
 legend.position = c(.9, .3))

## `summarise()` regrouping output by 'participant\_id', 'reversal\_group', 'is\_previous\_congruent' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent', 'is\_congruent' (override with `.groups` argument)

kan\_exp2\_rt\_cse\_rev\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp2\_Rt\_Cse\_Rev\_Plot.png", width = 14.4, height = 8, plot = kan\_exp2\_rt\_cse\_rev\_plot)

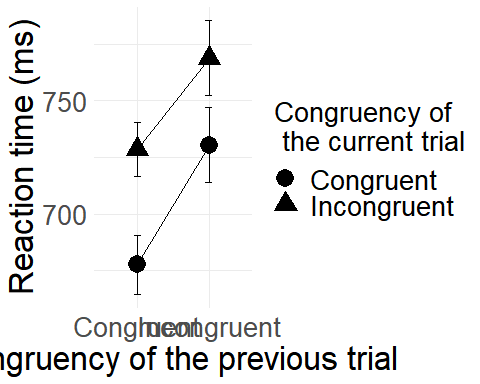
The congruency sequence effect for the reaction time responses without reversal groups.

kan\_exp2\_rt\_cse\_plot <-  
 main\_analysis\_rt\_data %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent")) %>%  
 group\_by(participant\_id, is\_previous\_congruent, is\_congruent) %>%   
 summarise(par\_mean\_rt = mean(response\_time, na.rm = T)) %>%   
 group\_by(is\_previous\_congruent, is\_congruent) %>%   
 summarise(N = n(),  
 mean\_rt = mean(par\_mean\_rt, na.rm = T),  
 sd\_rt = sd(par\_mean\_rt, na.rm = T),  
 se\_rt = sd\_rt / sqrt(N)) %>%  
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_rt,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 6) +  
 geom\_errorbar(aes(ymin = mean\_rt - (se\_rt \* 1.96),  
 ymax = mean\_rt + (se\_rt \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Reaction time (ms)") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 25),  
 axis.text = element\_text(size = 20),  
 legend.text = element\_text(size = 20),  
 legend.title = element\_text(size = 20),  
 strip.text.x = element\_text(size = 20))

## `summarise()` regrouping output by 'participant\_id', 'is\_previous\_congruent' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent' (override with `.groups` argument)

kan\_exp2\_rt\_cse\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp2\_Rt\_Cse\_Plot.png", width = 14.4, height = 8, plot = kan\_exp2\_rt\_cse\_plot)

### Crucial tests

### Two way interaction

#### Data preprocessing

We are calculating the mean reaction time per participant per condition (ci, ic, cc, ii).

main\_analysis\_rt\_aggregate <-  
 main\_analysis\_rt\_data %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent) %>%   
 summarise(rt\_conditional\_mean = mean(response\_time, na.rm = T)) %>%  
 ungroup()

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

Before the ANOVA we transform the predictor variables to factors.

main\_analysis\_rt\_aggregate <-  
 main\_analysis\_rt\_aggregate %>%  
 mutate(is\_congruent = as.factor(is\_congruent),  
 is\_previous\_congruent = as.factor(is\_previous\_congruent),  
 participant\_id = as.factor(participant\_id))

The number of participants for the rt analysis.

main\_analysis\_rt\_aggregate %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 163

We are running the ANOVA.

anova\_rt <-  
 ezANOVA(data = main\_analysis\_rt\_aggregate, dv = rt\_conditional\_mean, wid = participant\_id, within = .(is\_congruent, is\_previous\_congruent), type = 3)

Transforming the results of the ANOVA to a datatable.

anova\_rt <-  
 as\_tibble(anova\_rt$ANOVA)

Show the results of the ANOVA.

anova\_rt

## # A tibble: 3 x 7  
## Effect DFn DFd F p `p<.05` ges  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 is\_congruent 1 162 123. 1.28e-21 "\*" 0.0518   
## 2 is\_previous\_congruent 1 162 65.8 1.15e-13 "\*" 0.0563   
## 3 is\_congruent:is\_previous\_congruent 1 162 2.95 8.77e- 2 "" 0.00109

Extracting the F value and df from the ANOVA.

anova\_rt\_f\_value <-  
 anova\_rt %>%   
 filter(Effect %in% c("is\_congruent",  
 "is\_previous\_congruent",  
 "is\_congruent:is\_previous\_congruent")) %>%   
 transmute(term = case\_when(Effect == "is\_congruent" ~ "main\_effect\_congruent",  
 Effect == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 Effect == "is\_congruent:is\_previous\_congruent" ~ "interaction\_effect"),  
 f\_value = F,  
 df = DFd)

Calculating the raw effects per participant.

main\_analysis\_rt\_participant\_level\_raw\_effect <-  
 main\_analysis\_rt\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = rt\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((ci + ii) / 2) - ((cc + ic) / 2),  
 previous\_congruency\_effect = ((ii + ic) / 2) - ((ci + cc) / 2),  
 interaction\_effect = (ci - cc) - (ii - ic),  
 prev\_cong = ci - cc,  
 prev\_incon = ii - ic,  
 current\_incongruent = ci - ii)

We are calculating the mean raw effect sizes.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_participant\_level\_raw\_effect %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T),  
 prev\_cong\_mean = mean(prev\_cong, na.rm = T),  
 prev\_incon\_mean = mean(prev\_incon, na.rm = T)) %>%   
 gather(key = "term", value = "raw\_effect")

We are joining the F values and dfs with the raw effect sizes.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%  
 inner\_join(., anova\_rt\_f\_value, by = "term")

We are calculating the SE from the raw effect size and F value.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

Assigning variables to calculate Bayes factor.

# sd  
rt\_sd <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# obtained  
rt\_obtained <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# df  
rt\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(df)

Running the Bf analysis.

Bf(sd = rt\_sd,  
 obtained = rt\_obtained,  
 dfdata = rt\_df,  
 meanoftheory = 0,  
 sdtheory = 30,  
 dftheory = 10^10,  
 tail = 1)

## [1] 1.811392

Robustness region.

# Lower  
## 0 because it is inconclusive  
  
# Upper  
Bf(sd = rt\_sd,  
 obtained = rt\_obtained,  
 dfdata = rt\_df,  
 meanoftheory = 0,  
 sdtheory = 180,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3399338

### Outcome neutral test

#### Test Stroop effect

This is an outcome neutral test, therefore, the results of this test do not effect our main conclusions.

Saving the values needed for the analysis.

# Obtained sd  
main\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(se)  
  
# Obtained effect  
main\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(raw\_effect)  
  
# Df  
main\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(df)

Running the Bf analysis with the model described in the paper.

Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 8.080785e+18

Robustness region.

# Lower  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 0.69,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.110053

# Upper  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 16991,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.085931

### Supporting test of interest

#### Test previous congruency effect

This is a supporting test of interest. We use the same prior as for the main Stroop effect test.

Saving the values needed for the analysis.

# Obtained sd  
main\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(se)  
  
# Obtained effect  
main\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(raw\_effect)  
  
# Df  
main\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(df)

Running the Bf analysis with the model described in the paper.

Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 125944052981

Robustness region.

# Lower  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 1.08,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.092565

# Upper  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 17280,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.010143

### Three way interaction

#### Data preprocessing

We are calculating the mean reaction time per participant per condition (ci, ic, cc, ii).

main\_analysis\_rt\_aggregate <-  
 main\_analysis\_rt\_data %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent, reversal\_group) %>%   
 summarise(rt\_conditional\_mean = mean(response\_time, na.rm = T)) %>%  
 ungroup()

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent', 'is\_previous\_congruent' (override with `.groups` argument)

Before the ANOVA we transform the predictor variables to factors.

main\_analysis\_rt\_aggregate <-  
 main\_analysis\_rt\_aggregate %>%  
 mutate(is\_congruent = as.factor(is\_congruent),  
 is\_previous\_congruent = as.factor(is\_previous\_congruent),  
 participant\_id = as.factor(participant\_id),  
 reversal\_group = as.factor(reversal\_group))

The number of participants per reversal group.

main\_analysis\_rt\_aggregate %>%   
 group\_by(reversal\_group) %>%   
 count()

## # A tibble: 2 x 2  
## # Groups: reversal\_group [2]  
## reversal\_group n  
## <fct> <int>  
## 1 high 324  
## 2 low 328

The number of participants for the rt analysis.

main\_analysis\_rt\_aggregate %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 163

We are running the ANOVA.

anova\_rt <-  
 ezANOVA(data = main\_analysis\_rt\_aggregate, dv = rt\_conditional\_mean, wid = participant\_id, within = .(is\_congruent, is\_previous\_congruent), between = reversal\_group, type = 3)

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

Transforming the results of the ANOVA to a datatable.

anova\_rt <-  
 as\_tibble(anova\_rt$ANOVA)

Show the results of the ANOVA.

anova\_rt

## # A tibble: 7 x 7  
## Effect DFn DFd F p `p<.05` ges  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 reversal\_group 1 161 9.98e-2 7.53e- 1 "" 4.47e-4  
## 2 is\_congruent 1 161 1.25e+2 8.01e-22 "\*" 5.20e-2  
## 3 is\_previous\_congruent 1 161 6.54e+1 1.38e-13 "\*" 5.64e-2  
## 4 reversal\_group:is\_congruent 1 161 3.16e+0 7.72e- 2 "" 1.39e-3  
## 5 reversal\_group:is\_previous\_cong~ 1 161 1.46e-2 9.04e- 1 "" 1.34e-5  
## 6 is\_congruent:is\_previous\_congru~ 1 161 2.93e+0 8.90e- 2 "" 1.09e-3  
## 7 reversal\_group:is\_congruent:is\_~ 1 161 5.51e-1 4.59e- 1 "" 2.04e-4

Extracting the F value and df from the ANOVA.

anova\_rt\_f\_value <-  
 anova\_rt %>%   
 filter(Effect %in% c("is\_congruent",  
 "is\_previous\_congruent",  
 "reversal\_group:is\_congruent:is\_previous\_congruent")) %>%   
 transmute(term = case\_when(Effect == "is\_congruent" ~ "main\_effect\_congruent",  
 Effect == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 Effect == "reversal\_group:is\_congruent:is\_previous\_congruent" ~ "interaction\_effect"),  
 f\_value = F,  
 df = DFd)

Calculating the raw effects per participant.

main\_analysis\_rt\_participant\_level\_raw\_effect <-  
 main\_analysis\_rt\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = rt\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((ci + ii) / 2) - ((cc + ic) / 2),  
 previous\_congruency\_effect = ((ii + ic) / 2) - ((ci + cc) / 2),  
 interaction\_effect = (ci - cc) - (ii - ic),  
 prev\_cong = ci - cc,  
 prev\_incon = ii - ic,  
 current\_incongruent = ci - ii)

We are calculating the mean raw effect sizes.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_participant\_level\_raw\_effect %>%   
 group\_by(reversal\_group) %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T),  
 prev\_cong\_mean = mean(prev\_cong, na.rm = T),  
 prev\_incon\_mean = mean(prev\_incon, na.rm = T)) %>%   
 gather(key = "term", value = "raw\_effect", -reversal\_group) %>%  
 spread(key = "reversal\_group", value = "raw\_effect")

## `summarise()` ungrouping output (override with `.groups` argument)

We are extracting the low raw effect sizes from the high raw effect sizes according to the hypothesis.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%   
 ungroup() %>%   
 mutate(raw\_effect = high - low) %>%   
 select(-high,  
 -low)

We are joining the F values and dfs with the raw effect sizes.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%  
 inner\_join(., anova\_rt\_f\_value, by = "term")

We are calculating the se from the raw effect size and f value.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

Assigning variables to calculate bayes factor.

# sd  
rt\_sd <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# obtained  
rt\_obtained <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# df  
rt\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(df)

Running the Bf analysis.

Bf(sd = rt\_sd,  
 obtained = rt\_obtained,  
 dfdata = rt\_df,  
 meanoftheory = 0,  
 sdtheory = 30,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.2798037

Robustness region.

# Lower  
Bf(sd = rt\_sd,  
 obtained = rt\_obtained,  
 dfdata = rt\_df,  
 meanoftheory = 0,  
 sdtheory = 24.3,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3322246

# Upper  
## Inf

### Correlation

We are calculating the correlation between the mean number of reversals per participant and the size of the conflict adaptation operationalized by the current incongruent difference.

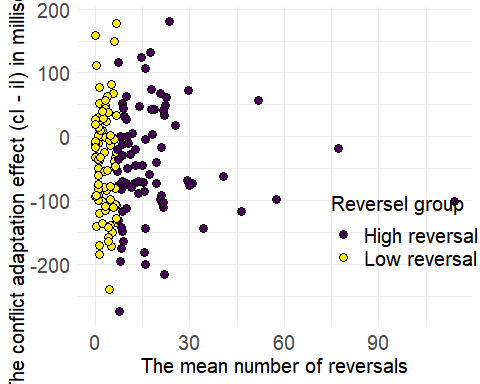
First we create a dataset for the comparison.

main\_analysis\_rt\_cor\_data <-  
 main\_analysis\_rt\_data %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent, reversal\_group, mean\_reversal) %>%   
 summarise(rt\_conditional\_mean = mean(response\_time, na.rm = T)) %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = rt\_conditional\_mean) %>%   
 mutate(current\_incongruent = ci - ii) %>%   
 select(participant\_id,  
 current\_incongruent,  
 mean\_reversal,  
 reversal\_group)

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent', 'is\_previous\_congruent', 'reversal\_group' (override with `.groups` argument)

We are visualizing the comparison on a figure.

kan\_exp2\_correlation\_rt\_plot <-  
 main\_analysis\_rt\_cor\_data %>%   
 mutate(reversal\_group = case\_when(reversal\_group == "low" ~ "Low reversal group",  
 reversal\_group == "high" ~ "High reversal group")) %>%   
 ggplot() +  
 aes(x = mean\_reversal,  
 y = current\_incongruent,  
 fill = reversal\_group) +  
 geom\_jitter(colour = "black", pch = 21, size = 3) +  
 guides(shape = guide\_legend(title = "Reversal group")) +  
 labs(x = "The mean number of reversals",  
 y = "The conflict adaptation effect (cI - iI) in milliseconds",  
 fill = "Reversel group") +  
 viridis::scale\_fill\_viridis(discrete = TRUE) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 15),  
 axis.text = element\_text(size = 15),  
 legend.text = element\_text(size = 15),  
 legend.title = element\_text(size = 15),  
 legend.position = c(.9, .3))  
  
kan\_exp2\_correlation\_rt\_plot



Save the plot.

ggsave("Figures/Kan\_Exp2\_Correlation\_Rt\_Plot.png", width = 14.4, height = 8, plot = kan\_exp2\_correlation\_rt\_plot)

Running the correlation.

cor.test(main\_analysis\_rt\_cor\_data$current\_incongruent,  
 main\_analysis\_rt\_cor\_data$mean\_reversal,  
 method = "pearson")

##   
## Pearson's product-moment correlation  
##   
## data: main\_analysis\_rt\_cor\_data$current\_incongruent and main\_analysis\_rt\_cor\_data$mean\_reversal  
## t = -0.25675, df = 161, p-value = 0.7977  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.1734118 0.1339064  
## sample estimates:  
## cor   
## -0.02023055

## Accuracy analysis

### Importing data

main\_analysis\_acc <- read\_tsv("Data/Processed/Exp2/Kan\_Processed\_Exp2\_Main\_Acc\_data.tsv")

## Parsed with column specification:  
## cols(  
## participant\_id = col\_character(),  
## lab = col\_character(),  
## loop\_num = col\_double(),  
## trial\_id = col\_double(),  
## procedure = col\_character(),  
## conflict\_condition = col\_character(),  
## stim\_type = col\_character(),  
## stimulus = col\_character(),  
## stim\_color = col\_character(),  
## is\_correct = col\_double(),  
## response\_time = col\_double(),  
## is\_previous\_congruent = col\_double(),  
## is\_congruent = col\_double(),  
## is\_previous\_correct = col\_double(),  
## mean\_reversal = col\_double(),  
## reversal\_group = col\_character()  
## )

### Exploratory data analysis

The number of trials per participant.

main\_analysis\_acc %>%   
 group\_by(participant\_id) %>%   
 count()

## # A tibble: 163 x 2  
## # Groups: participant\_id [163]  
## participant\_id n  
## <chr> <int>  
## 1 0001 42  
## 2 0002 42  
## 3 0003 46  
## 4 0004 49  
## 5 0005 44  
## 6 0006 44  
## 7 0007 46  
## 8 0008 42  
## 9 0009 46  
## 10 0010 74  
## # ... with 153 more rows

The number of participants.

main\_analysis\_acc %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 163

The number and frequency of correct trials. There are different number of trials per participant because each participant was assigned to a different set of trails (out of 6) randomly. These sets contained different number of test trials that satisfy the criteria to be the part of the analysis.

main\_analysis\_acc %>%   
 group\_by(participant\_id, is\_correct) %>%   
 count() %>%   
 group\_by(participant\_id) %>%   
 mutate(sum\_n = sum(n),  
 freq = n / sum\_n \* 100) %>%  
 filter(is\_correct == 1L)

## # A tibble: 163 x 5  
## # Groups: participant\_id [163]  
## participant\_id is\_correct n sum\_n freq  
## <chr> <dbl> <int> <int> <dbl>  
## 1 0001 1 33 42 78.6  
## 2 0002 1 41 42 97.6  
## 3 0003 1 33 46 71.7  
## 4 0004 1 24 49 49.0  
## 5 0005 1 23 44 52.3  
## 6 0006 1 36 44 81.8  
## 7 0007 1 45 46 97.8  
## 8 0008 1 33 42 78.6  
## 9 0009 1 39 46 84.8  
## 10 0010 1 56 74 75.7  
## # ... with 153 more rows

### Figures

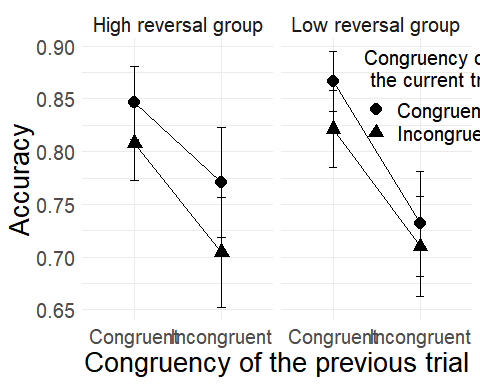
Figure with reversal groups.

kan\_exp2\_acc\_cse\_rev\_plot <-  
 main\_analysis\_acc %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent"),  
 reversal\_group = case\_when(reversal\_group == "low" ~ "Low reversal group",  
 reversal\_group == "high" ~ "High reversal group")) %>%  
 group\_by(is\_previous\_congruent, is\_congruent, reversal\_group, participant\_id) %>%   
 summarise(par\_mean\_acc = mean(is\_correct, na.rm = T)) %>%   
 group\_by(is\_previous\_congruent, is\_congruent, reversal\_group) %>%   
 summarise(N = n(),  
 mean\_acc = mean(par\_mean\_acc, na.rm = T),  
 sd\_acc = sd(par\_mean\_acc, na.rm = T),  
 se\_acc = sd\_acc / sqrt(N)) %>%  
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_acc,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 4) +  
 geom\_errorbar(aes(ymin = mean\_acc - (se\_acc \* 1.96),  
 ymax = mean\_acc + (se\_acc \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Accuracy") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 facet\_wrap( ~ reversal\_group) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 20),  
 axis.text = element\_text(size = 15),  
 legend.text = element\_text(size = 15),  
 legend.title = element\_text(size = 15),  
 strip.text.x = element\_text(size = 15),  
 legend.position = c(.9, .8))

## `summarise()` regrouping output by 'is\_previous\_congruent', 'is\_congruent', 'reversal\_group' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent', 'is\_congruent' (override with `.groups` argument)

kan\_exp2\_acc\_cse\_rev\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp2\_Acc\_Cse\_Rev\_Plot.png", width = 14.4, height = 8, plot = kan\_exp2\_acc\_cse\_rev\_plot)

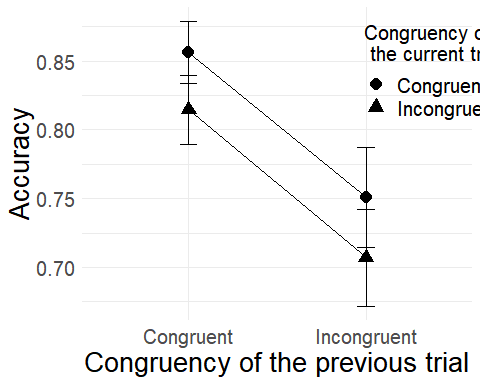
Figure without reversal groups.

kan\_exp2\_acc\_cse\_plot <-  
 main\_analysis\_acc %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent")) %>%  
 group\_by(participant\_id, is\_previous\_congruent, is\_congruent) %>%   
 summarise(par\_mean\_acc = mean(is\_correct, na.rm = T)) %>%   
 group\_by(is\_previous\_congruent, is\_congruent) %>%   
 summarise(N = n(),  
 mean\_acc = mean(par\_mean\_acc, na.rm = T),  
 sd\_acc = sd(par\_mean\_acc, na.rm = T),  
 se\_acc = sd\_acc / sqrt(N)) %>%  
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_acc,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 4) +  
 geom\_errorbar(aes(ymin = mean\_acc - (se\_acc \* 1.96),  
 ymax = mean\_acc + (se\_acc \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Accuracy") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 20),  
 axis.text = element\_text(size = 15),  
 legend.text = element\_text(size = 15),  
 legend.title = element\_text(size = 15),  
 strip.text.x = element\_text(size = 15),  
 legend.position = c(.9, .8))

## `summarise()` regrouping output by 'participant\_id', 'is\_previous\_congruent' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent' (override with `.groups` argument)

kan\_exp2\_acc\_cse\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp2\_Acc\_Cse\_Plot.png", width = 14.4, height = 8, plot = kan\_exp2\_acc\_cse\_plot)

### Crucial tests

### Two way interaction

#### Data preprocessing

we are calculating the conditional mean of the correct trials per participant and per condition (CC, II, CI, IC).

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent) %>%   
 summarise(acc\_conditional\_mean = mean(is\_correct, na.rm = T)) %>%   
 ungroup()

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

Transfer independent variables used in the ANOVA to factors.

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc\_aggregate %>%   
 mutate(is\_congruent = as\_factor(is\_congruent),  
 is\_previous\_congruent = as\_factor(is\_previous\_congruent),  
 participant\_id = as\_factor(participant\_id))

Running the mixed ANOVA.

anova\_acc <-  
 ezANOVA(data = main\_analysis\_acc\_aggregate, dv = acc\_conditional\_mean, wid = participant\_id, within = .(is\_congruent, is\_previous\_congruent), type = 3)

Transforming the output of the ANOVA to a tibble.

anova\_acc <-  
 as\_tibble(anova\_acc$ANOVA)

Show the results of the ANOVA.

anova\_acc

## # A tibble: 3 x 7  
## Effect DFn DFd F p `p<.05` ges  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 is\_congruent 1 162 20.5 1.16e- 5 "\*" 1.16e-2  
## 2 is\_previous\_congruent 1 162 72.1 1.23e-14 "\*" 6.76e-2  
## 3 is\_congruent:is\_previous\_congr~ 1 162 0.0102 9.20e- 1 "" 5.92e-6

Extract the F value and df2 from the ANOVA.

anova\_acc\_f\_value <-  
 anova\_acc %>%   
 filter(Effect %in% c("is\_congruent",  
 "is\_previous\_congruent",  
 "is\_congruent:is\_previous\_congruent")) %>%   
 transmute(term = case\_when(Effect == "is\_congruent" ~ "main\_effect\_congruent",  
 Effect == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 Effect == "is\_congruent:is\_previous\_congruent" ~ "interaction\_effect"),  
 f\_value = F,  
 df = DFd)

We calculate the raw effects per participant.

main\_analysis\_acc\_participant\_level\_raw\_effect <-  
 main\_analysis\_acc\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = acc\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((cc + ic) / 2) - ((ci + ii) / 2),  
 previous\_congruency\_effect = ((ci + cc) / 2) - ((ii + ic) / 2),  
 interaction\_effect = (cc - ci) - (ic - ii),  
 current\_incongruent = ii - ci)

We summarize the participant level data for further analysis.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_participant\_level\_raw\_effect %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T)) %>%   
 gather(key = "term", value = "raw\_effect")

We join the calculated raw effects with the F value and the df from the ANOVA.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%  
 inner\_join(., anova\_acc\_f\_value, by = "term")

We valvulate the SE corresponding to the raw effects.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

Saving values needed for the Bayes factor analysis.

# SE  
acc\_sd <-  
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# Raw effect  
acc\_obtained <-  
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# Degrees of freedom  
acc\_df <-  
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(df)

Calculating the Bayes factor.

Bf(sd = acc\_sd,  
 obtained = acc\_obtained,  
 dfdata = acc\_df,  
 meanoftheory = 0,  
 sdtheory = .03,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.5031245

Robustness region.

# Lower  
## 0 because it is inconclusive  
  
# Upper  
Bf(sd = acc\_sd,  
 obtained = acc\_obtained,  
 dfdata = acc\_df,  
 meanoftheory = 0,  
 sdtheory = .049,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.336111

#### Arcsine transformed accuracy

Calculating the Bf with the two methods.

arcsine\_two\_way(  
 df = main\_analysis\_acc,  
 expression = arcsine\_original\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

## [1] 0.8493062

arcsine\_two\_way(  
 df = main\_analysis\_acc,  
 expression = arcsine\_common\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

## [1] 0.6495672

### Three way interaction

#### Data preprocessing

We are calculating the conditional mean of the correct trials per participant and per condition (CC, II, CI, IC).

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc %>%   
 group\_by(participant\_id, reversal\_group, is\_congruent, is\_previous\_congruent) %>%   
 summarise(acc\_conditional\_mean = mean(is\_correct, na.rm = T)) %>%   
 ungroup()

## `summarise()` regrouping output by 'participant\_id', 'reversal\_group', 'is\_congruent' (override with `.groups` argument)

The number of participants in each reversal group.

main\_analysis\_acc\_aggregate %>%   
 distinct(participant\_id, .keep\_all = T) %>%   
 group\_by(reversal\_group) %>%   
 count()

## # A tibble: 2 x 2  
## # Groups: reversal\_group [2]  
## reversal\_group n  
## <chr> <int>  
## 1 high 81  
## 2 low 82

Transfer independent variables used in the ANOVA to factors.

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc\_aggregate %>%   
 mutate(is\_congruent = as\_factor(is\_congruent),  
 is\_previous\_congruent = as\_factor(is\_previous\_congruent),  
 participant\_id = as\_factor(participant\_id),  
 reversal\_group = as\_factor(reversal\_group))

Running the mixed ANOVA.

anova\_acc <-  
 ezANOVA(data = main\_analysis\_acc\_aggregate, dv = acc\_conditional\_mean, wid = participant\_id, within = .(is\_congruent, is\_previous\_congruent), between = reversal\_group, type = 3)

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

Transforming the output of the ANOVA to a tibble.

anova\_acc <-  
 as\_tibble(anova\_acc$ANOVA)

Show the results of the ANOVA.

anova\_acc

## # A tibble: 7 x 7  
## Effect DFn DFd F p `p<.05` ges  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 reversal\_group 1 161 4.42e-5 9.95e- 1 "" 1.79e-7  
## 2 is\_congruent 1 161 2.05e+1 1.13e- 5 "\*" 1.17e-2  
## 3 is\_previous\_congruent 1 161 7.23e+1 1.18e-14 "\*" 6.76e-2  
## 4 reversal\_group:is\_congruent 1 161 1.02e+0 3.13e- 1 "" 5.88e-4  
## 5 reversal\_group:is\_previous\_co~ 1 161 1.80e+0 1.82e- 1 "" 1.80e-3  
## 6 is\_congruent:is\_previous\_cong~ 1 161 1.20e-2 9.13e- 1 "" 6.96e-6  
## 7 reversal\_group:is\_congruent:i~ 1 161 1.82e+0 1.79e- 1 "" 1.06e-3

Extract the F value and df2 from the ANOVA.

anova\_acc\_f\_value <-  
 anova\_acc %>%   
 filter(Effect %in% c("is\_congruent",  
 "is\_previous\_congruent",  
 "reversal\_group:is\_congruent:is\_previous\_congruent")) %>%   
 transmute(term = case\_when(Effect == "is\_congruent" ~ "main\_effect\_congruent",  
 Effect == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 Effect == "reversal\_group:is\_congruent:is\_previous\_congruent" ~ "interaction\_effect"),  
 f\_value = F,  
 df = DFd)

We calculate the raw effects per participant.

main\_analysis\_acc\_participant\_level\_raw\_effect <-  
 main\_analysis\_acc\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = acc\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((cc + ic) / 2) - ((ci + ii) / 2),  
 previous\_congruency\_effect = ((ci + cc) / 2) - ((ii + ic) / 2),  
 interaction\_effect = (cc - ci) - (ic - ii),  
 current\_incongruent = ii - ci)

We summarise the participant level data for further analysis.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_participant\_level\_raw\_effect %>%   
 group\_by(reversal\_group) %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T)) %>%   
 gather(key = "term", value = "raw\_effect", -reversal\_group) %>%  
 spread(key = "reversal\_group", value = "raw\_effect")

## `summarise()` ungrouping output (override with `.groups` argument)

We calculate the differences of the effects between the low and high reversal group. We extract the high reversal group raw effect from the low reversal group effect because according to the theory the high group experienced greater conflict, therefore the CSE should be stronger there.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%   
 ungroup() %>%   
 mutate(raw\_effect = high - low) %>%   
 select(-high,  
 -low)

We join the calculated raw effects with the F value and the df from the ANOVA.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%  
 inner\_join(., anova\_acc\_f\_value, by = "term")

We valvulate the SE corresponding to the raw effects.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

Saving values needed for the Bayes factor analysis.

# SE  
acc\_sd <-  
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# Raw effect  
acc\_obtained <-  
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# Degrees of freedom  
acc\_df <-  
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(df)

Calculating the Bayes factor.

Bf(sd = acc\_sd,  
 obtained = acc\_obtained,  
 dfdata = acc\_df,  
 meanoftheory = 0,  
 sdtheory = .03,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.4532576

Robustness region.

# Lower  
## 0 because inconclusive  
  
# Upper  
Bf(sd = acc\_sd,  
 obtained = acc\_obtained,  
 dfdata = acc\_df,  
 meanoftheory = 0,  
 sdtheory = .045,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3341171

#### Arcsine transformed accuracy

We run the calculations with arcsine transformed proportions. We will include these findings in the supplementary materials.

The calculation is the same as before expect that we use arcsine transformation on the conditional means.

arcsine\_three\_way(  
 df = main\_analysis\_acc,  
 expression = arcsine\_original\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'reversal\_group', 'is\_congruent' (override with `.groups` argument)

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

## `summarise()` ungrouping output (override with `.groups` argument)

## [1] 0.7785031

arcsine\_three\_way(  
 df = main\_analysis\_acc,  
 expression = arcsine\_common\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'reversal\_group', 'is\_congruent' (override with `.groups` argument)

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

## `summarise()` ungrouping output (override with `.groups` argument)

## [1] 0.608429

### Correlation

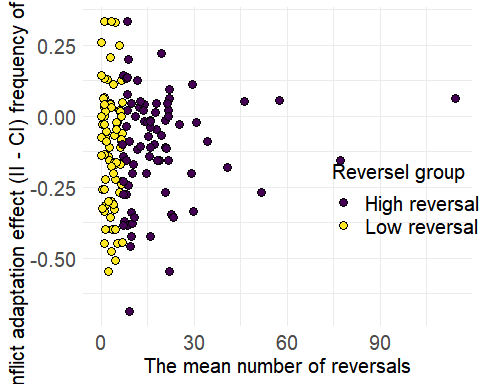
Creating dataset for the comparison between the mean number of experienced reversals and the raw interaction effect per participant.

main\_analysis\_acc\_cor <-  
 main\_analysis\_acc %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent, reversal\_group, mean\_reversal) %>%   
 summarise(acc\_conditional\_mean = mean(is\_correct, na.rm = T)) %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = acc\_conditional\_mean) %>%   
 mutate(current\_incongruent = ii - ci) %>%   
 select(participant\_id,  
 current\_incongruent,  
 mean\_reversal,  
 reversal\_group)

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent', 'is\_previous\_congruent', 'reversal\_group' (override with `.groups` argument)

Visualizing the correlation on a figure.

kan\_exp2\_correlation\_acc\_plot <-  
 main\_analysis\_acc\_cor %>%   
 mutate(reversal\_group = case\_when(reversal\_group == "low" ~ "Low reversal group",  
 reversal\_group == "high" ~ "High reversal group")) %>%   
 ggplot() +  
 aes(x = mean\_reversal,  
 y = current\_incongruent,  
 fill = reversal\_group) +  
 geom\_jitter(colour="black", pch=21, size = 3) +  
 guides(shape = guide\_legend(title = "Reversal group")) +  
 labs(x = "The mean number of reversals",  
 y = "The conflict adaptation effect (II - CI) frequency of correct trials",  
 fill = "Reversel group") +  
 viridis::scale\_fill\_viridis(discrete = TRUE) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 15),  
 axis.text = element\_text(size = 15),  
 legend.text = element\_text(size = 15),  
 legend.title = element\_text(size = 15),  
 legend.position = c(.9, .4))  
  
kan\_exp2\_correlation\_acc\_plot



Save the plot.

ggsave("Figures/Kan\_Exp2\_Correlation\_Acc\_Plot.png", width = 14.4, height = 8, plot = kan\_exp2\_correlation\_acc\_plot)

Running the correlation test.

cor.test(main\_analysis\_acc\_cor$current\_incongruent,  
 main\_analysis\_acc\_cor$mean\_reversal,  
 method = "pearson")

##   
## Pearson's product-moment correlation  
##   
## data: main\_analysis\_acc\_cor$current\_incongruent and main\_analysis\_acc\_cor$mean\_reversal  
## t = 0.71098, df = 161, p-value = 0.4781  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.09862328 0.20787811  
## sample estimates:  
## cor   
## 0.05594536

# Exp 3

## Reaction time analysis

### Import data

main\_analysis\_rt\_data <-  
 read\_tsv("Data/Processed/Exp3/Kan\_Processed\_Exp3\_Main\_Rt\_data.tsv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## participant\_id = col\_character(),  
## lab = col\_character(),  
## procedure = col\_character(),  
## conflict\_condition = col\_character(),  
## stim\_type = col\_character(),  
## stimulus = col\_character(),  
## stim\_color = col\_character(),  
## reversal = col\_logical(),  
## score = col\_logical()  
## )

## See spec(...) for full column specifications.

### Exploratory data analysis

The number of response per participant.

main\_analysis\_rt\_data %>%   
 group\_by(participant\_id) %>%   
 count()

## # A tibble: 80 x 2  
## # Groups: participant\_id [80]  
## participant\_id n  
## <chr> <int>  
## 1 0001 67  
## 2 0002 61  
## 3 0003 39  
## 4 0005 36  
## 5 0006 65  
## 6 0008 28  
## 7 0010 36  
## 8 0011 45  
## 9 0013 35  
## 10 0014 40  
## # ... with 70 more rows

The number of participants.

main\_analysis\_rt\_data %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 80

### Figures

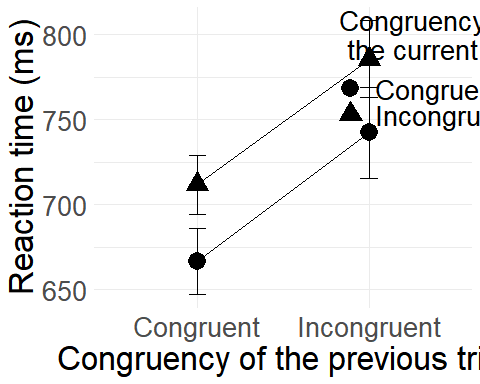
The congruency sequence effect for the reaction time responses.

kan\_exp3\_rt\_cse\_plot <-  
 main\_analysis\_rt\_data %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent")) %>%  
 group\_by(participant\_id, is\_previous\_congruent, is\_congruent) %>%   
 summarise(par\_mean\_rt = mean(response\_time, na.rm = T)) %>%   
 group\_by(is\_previous\_congruent, is\_congruent) %>%   
 summarise(N = n(),  
 mean\_rt = mean(par\_mean\_rt, na.rm = T),  
 sd\_rt = sd(par\_mean\_rt, na.rm = T),  
 se\_rt = sd\_rt / sqrt(N)) %>%  
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_rt,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 6) +  
 geom\_errorbar(aes(ymin = mean\_rt - (se\_rt \* 1.96),  
 ymax = mean\_rt + (se\_rt \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Reaction time (ms)") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 25),  
 axis.text = element\_text(size = 20),  
 legend.text = element\_text(size = 20),  
 legend.title = element\_text(size = 20),  
 strip.text.x = element\_text(size = 20),  
 legend.position = c(.9, .8))

## `summarise()` regrouping output by 'participant\_id', 'is\_previous\_congruent' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent' (override with `.groups` argument)

kan\_exp3\_rt\_cse\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp3\_Rt\_Cse\_Plot.png", width = 14.4, height = 8, plot = kan\_exp3\_rt\_cse\_plot)

### Crucial tests

#### Data preprocessing

We are calculating the mean reaction time per participant per condition (ci, ic, cc, ii).

main\_analysis\_rt\_aggregate <-  
 main\_analysis\_rt\_data %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent) %>%   
 summarise(rt\_conditional\_mean = mean(response\_time, na.rm = T)) %>%  
 ungroup()

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

Before the ANOVA we transform the predictor variables to factors.

main\_analysis\_rt\_aggregate <-  
 main\_analysis\_rt\_aggregate %>%  
 mutate(is\_congruent = as.factor(is\_congruent),  
 is\_previous\_congruent = as.factor(is\_previous\_congruent),  
 participant\_id = as.factor(participant\_id))

We are running the ANOVA.

anova\_rt <-  
 ezANOVA(data = main\_analysis\_rt\_aggregate, dv = rt\_conditional\_mean, wid = participant\_id, within = .(is\_congruent, is\_previous\_congruent), type = 3)

Transforming the results of the ANOVA to a datatable.

anova\_rt <-  
 as\_tibble(anova\_rt$ANOVA)

Show the results of the ANOVA.

anova\_rt

## # A tibble: 3 x 7  
## Effect DFn DFd F p `p<.05` ges  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 is\_congruent 1 79 51.7 3.18e-10 "\*" 4.77e-2  
## 2 is\_previous\_congruent 1 79 98.1 1.68e-15 "\*" 1.25e-1  
## 3 is\_congruent:is\_previous\_congru~ 1 79 0.0285 8.66e- 1 "" 1.77e-5

Extracting F value and df from the ANOVA.

anova\_rt\_f\_value <-  
 anova\_rt %>%   
 filter(Effect %in% c("is\_congruent",  
 "is\_previous\_congruent",  
 "is\_congruent:is\_previous\_congruent")) %>%   
 transmute(term = case\_when(Effect == "is\_congruent" ~ "main\_effect\_congruent",  
 Effect == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 Effect == "is\_congruent:is\_previous\_congruent" ~ "interaction\_effect"),  
 f\_value = F,  
 df = DFd)

Calculating the raw effects per participant.

main\_analysis\_rt\_participant\_level\_raw\_effect <-  
 main\_analysis\_rt\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = rt\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((ci + ii) / 2) - ((cc + ic) / 2),  
 previous\_congruency\_effect = ((ii + ic) / 2) - ((ci + cc) / 2),  
 interaction\_effect = (ci - cc) - (ii - ic),  
 prev\_cong = ci - cc,  
 prev\_incon = ii - ic,  
 current\_incongruent = ci - ii)

Summarizing the mean raw effects.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_participant\_level\_raw\_effect %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T)) %>%   
 gather(key = "term", value = "raw\_effect")

Joining the raw effects with the F value and df.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%  
 inner\_join(., anova\_rt\_f\_value, by = "term")

Calculating SE.

main\_analysis\_rt\_raw\_effect <-  
 main\_analysis\_rt\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

#### Running the analysis

Saving the values needed for the analysis.

# Obtained sd  
rt\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# Obtained effect  
rt\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# Df  
rt\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(df)

Running the Bayes factor analysis.

Bf(sd = rt\_se,  
 obtained = rt\_effect,  
 dfdata = rt\_df,  
 meanoftheory = 0,  
 sdtheory = 30,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.358782

Robustness region.

# Lower  
## 0 because it was inconclusive  
  
# Upper  
Bf(sd = rt\_se,  
 obtained = rt\_effect,  
 dfdata = rt\_df,  
 meanoftheory = 0,  
 sdtheory = 32,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3387923

### Outcome neutral test

#### Test Stroop effect

This is an outcome neutral test, therefore, the results of this test do not effect our main conclusions.

Saving the values needed for the analysis.

# Obtained sd  
main\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(se)  
  
# Obtained effect  
main\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(raw\_effect)  
  
# Df  
main\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_congruent") %>%   
 pull(df)

Running the Bf analysis with the model described in the paper.

Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 73996186

Robustness region.

# Lower  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 1.5,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.014758

# Upper  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 15800,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.01088

### Supporting test of interest

#### Test previous congruency effect

This is a supporting test of interest. We use the same prior as for the main Stroop effect test.

Saving the values needed for the analysis.

# Obtained sd  
main\_se <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(se)  
  
# Obtained effect  
main\_effect <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(raw\_effect)  
  
# Df  
main\_df <-   
 main\_analysis\_rt\_raw\_effect %>%   
 filter(term == "main\_effect\_previous\_congruent") %>%   
 pull(df)

Running the Bf analysis with the model described in the paper.

Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 1.985061e+12

Robustness region.

# Lower  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 1.8,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.023546

# Upper  
Bf(sd = main\_se,  
 obtained = main\_effect,  
 dfdata = main\_df,  
 meanoftheory = 0,  
 sdtheory = 27740,  
 dftheory = 10^10,  
 tail = 1)

## [1] 3.015082

## Accuracy analysis

### Importing data

main\_analysis\_acc <- read\_tsv("Data/Processed/Exp3/Kan\_Processed\_Exp3\_Main\_Acc\_data.tsv")

## Parsed with column specification:  
## cols(  
## participant\_id = col\_character(),  
## lab = col\_character(),  
## loop\_num = col\_double(),  
## trial\_id = col\_double(),  
## procedure = col\_character(),  
## conflict\_condition = col\_character(),  
## stim\_type = col\_character(),  
## stimulus = col\_character(),  
## stim\_color = col\_character(),  
## is\_correct = col\_double(),  
## response\_time = col\_double(),  
## is\_previous\_congruent = col\_double(),  
## is\_congruent = col\_double(),  
## is\_previous\_correct = col\_double(),  
## mean\_acc = col\_double()  
## )

### Exploratory data analysis

The number and frequency of correct trials.

main\_analysis\_acc %>%   
 group\_by(participant\_id, is\_correct) %>%   
 count() %>%   
 group\_by(participant\_id) %>%   
 mutate(sum\_n = sum(n),  
 freq = n / sum\_n \* 100) %>%  
 filter(is\_correct == 1L)

## # A tibble: 80 x 5  
## # Groups: participant\_id [80]  
## participant\_id is\_correct n sum\_n freq  
## <chr> <dbl> <int> <int> <dbl>  
## 1 0001 1 67 74 90.5  
## 2 0002 1 61 74 82.4  
## 3 0003 1 39 46 84.8  
## 4 0005 1 36 42 85.7  
## 5 0006 1 65 74 87.8  
## 6 0008 1 28 42 66.7  
## 7 0010 1 36 44 81.8  
## 8 0011 1 45 49 91.8  
## 9 0013 1 35 49 71.4  
## 10 0014 1 40 42 95.2  
## # ... with 70 more rows

The number of participants.

main\_analysis\_acc %>%   
 distinct(participant\_id) %>%   
 count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 80

### Figures

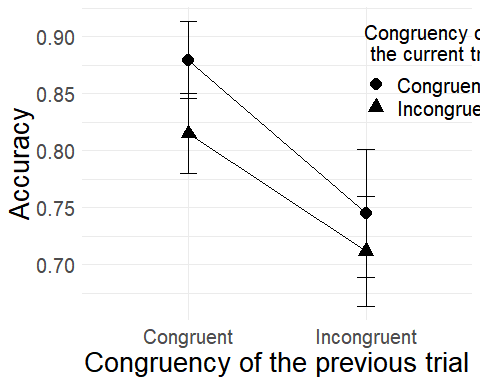
Congruency sequence effect of accuracy.

kan\_exp3\_acc\_congruency\_plot <-  
 main\_analysis\_acc %>%   
 mutate(is\_previous\_congruent = case\_when(is\_previous\_congruent == 0L ~ "Incongruent",  
 is\_previous\_congruent == 1L ~ "Congruent"),  
 is\_congruent = case\_when(is\_congruent == 0L ~ "Incongruent",  
 is\_congruent == 1L ~ "Congruent")) %>%  
 group\_by(participant\_id, is\_previous\_congruent, is\_congruent) %>%   
 summarise(par\_mean\_acc = mean(is\_correct, na.rm = T)) %>%   
 group\_by(is\_previous\_congruent, is\_congruent) %>%   
 summarise(N = n(),  
 mean\_acc = mean(par\_mean\_acc, na.rm = T),  
 sd\_acc = sd(par\_mean\_acc, na.rm = T),  
 se\_acc = sd\_acc / sqrt(N)) %>%  
 ggplot() +  
 aes(x = is\_previous\_congruent,  
 y = mean\_acc,  
 shape = is\_congruent,  
 group = is\_congruent) +  
 geom\_path() +  
 geom\_point(size = 4) +  
 geom\_errorbar(aes(ymin = mean\_acc - (se\_acc \* 1.96),  
 ymax = mean\_acc + (se\_acc \* 1.96)),  
 width = .1) +  
 guides(color = FALSE) +  
 xlab("Congruency of the previous trial")+  
 ylab("Accuracy") +  
 guides(shape = guide\_legend(title = "Congruency of \n the current trial")) +  
 theme\_minimal() +  
 theme(axis.title = element\_text(size = 20),  
 axis.text = element\_text(size = 15),  
 legend.text = element\_text(size = 15),  
 legend.title = element\_text(size = 15),  
 strip.text.x = element\_text(size = 15),  
 legend.position = c(.9, .8))

## `summarise()` regrouping output by 'participant\_id', 'is\_previous\_congruent' (override with `.groups` argument)

## `summarise()` regrouping output by 'is\_previous\_congruent' (override with `.groups` argument)

kan\_exp3\_acc\_congruency\_plot



Saving the figure.

ggsave("Figures/Kan\_Exp3\_Acc\_Cse\_Plot.png", width = 14.4, height = 8, plot = kan\_exp3\_acc\_congruency\_plot)

### Data preprocessing

Calculating the mean accuracy for each participant in each condition.

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc %>%   
 group\_by(participant\_id, is\_congruent, is\_previous\_congruent) %>%   
 summarise(acc\_conditional\_mean = mean(is\_correct, na.rm = T)) %>%   
 ungroup()

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

Transforming grouping variables to factors.

main\_analysis\_acc\_aggregate <-  
 main\_analysis\_acc\_aggregate %>%   
 mutate(is\_congruent = as\_factor(is\_congruent),  
 is\_previous\_congruent = as\_factor(is\_previous\_congruent),  
 participant\_id = as\_factor(participant\_id))

Running the ANOVA.

anova\_acc <-  
 ezANOVA(data = main\_analysis\_acc\_aggregate, dv = acc\_conditional\_mean, wid = participant\_id, within = .(is\_congruent, is\_previous\_congruent), type = 3)

Show the results of the ANOVA.

anova\_acc

## $ANOVA  
## Effect DFn DFd F p p<.05  
## 2 is\_congruent 1 79 22.093069 1.082208e-05 \*  
## 3 is\_previous\_congruent 1 79 46.408294 1.703329e-09 \*  
## 4 is\_congruent:is\_previous\_congruent 1 79 1.469172 2.290891e-01   
## ges  
## 2 0.014434048  
## 3 0.080415117  
## 4 0.001542679

Transforming the results of the ANOVA to a datatable.

anova\_acc <-  
 as\_tibble(anova\_acc$ANOVA)

Extracting F value and df.

anova\_acc\_f\_value <-  
 anova\_acc %>%   
 filter(Effect %in% c("is\_congruent",  
 "is\_previous\_congruent",  
 "is\_congruent:is\_previous\_congruent")) %>%   
 transmute(term = case\_when(Effect == "is\_congruent" ~ "main\_effect\_congruent",  
 Effect == "is\_previous\_congruent" ~ "main\_effect\_previous\_congruent",  
 Effect == "is\_congruent:is\_previous\_congruent" ~ "interaction\_effect"),  
 f\_value = F,  
 df = DFd)

Calculating raw effects for each participant.

main\_analysis\_acc\_participant\_level\_raw\_effect <-  
 main\_analysis\_acc\_aggregate %>%   
 ungroup() %>%   
 mutate(condition = case\_when(is\_previous\_congruent == 0L & is\_congruent == 0L ~ "ii",  
 is\_previous\_congruent == 0L & is\_congruent == 1L ~ "ic",  
 is\_previous\_congruent == 1L & is\_congruent == 0L ~ "ci",  
 is\_previous\_congruent == 1L & is\_congruent == 1L ~ "cc",  
 TRUE ~ NA\_character\_)) %>%   
 select(-is\_previous\_congruent, -is\_congruent) %>%   
 spread(key = condition, value = acc\_conditional\_mean) %>%   
 mutate(congruency\_effect = ((cc + ic) / 2) - ((ci + ii) / 2),  
 previous\_congruency\_effect = ((ci + cc) / 2) - ((ii + ic) / 2),  
 interaction\_effect = (cc - ci) - (ic - ii),  
 current\_incongruent = ii - ci)

Summarizing mean raw effects.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_participant\_level\_raw\_effect %>%   
 summarise(main\_effect\_congruent = mean(congruency\_effect, na.rm = T),  
 main\_effect\_previous\_congruent = mean(previous\_congruency\_effect, na.rm = T),  
 interaction\_effect = mean(interaction\_effect, na.rm = T)) %>%   
 gather(key = "term", value = "raw\_effect")

Joining F values and dfs and raw effects.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%  
 inner\_join(., anova\_acc\_f\_value, by = "term")

Calculating SE.

main\_analysis\_acc\_raw\_effect <-  
 main\_analysis\_acc\_raw\_effect %>%   
 mutate(se = abs(raw\_effect / sqrt(f\_value)))

#### Running the analysis

Saving the values needed for the analysis.

# Obtained sd  
acc\_se <-   
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(se)  
  
# Obtained effect  
acc\_effect <-   
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(raw\_effect)  
  
# Df  
acc\_df <-   
 main\_analysis\_acc\_raw\_effect %>%   
 filter(term == "interaction\_effect") %>%   
 pull(df)

Running the Bayes factor analysis.

Bf(sd = acc\_se,  
 obtained = acc\_effect,  
 dfdata = acc\_df,  
 meanoftheory = 0,  
 sdtheory = .03,  
 dftheory = 10^10,  
 tail = 1)

## [1] 1.63487

Robustness region.

# Lower  
## 0 because it is inconclusive  
  
# Upper  
Bf(sd = acc\_se,  
 obtained = acc\_effect,  
 dfdata = acc\_df,  
 meanoftheory = 0,  
 sdtheory = .28,  
 dftheory = 10^10,  
 tail = 1)

## [1] 0.3399575

#### Arcsine transformed accuracy

We run the calculations with arcsine transformed proportions. We will include these findings in the supplementary materials.

The calculation is the same as before expect that we use arcsine transformation on the conditional means.

arcsine\_two\_way(  
 df = main\_analysis\_acc,  
 expression = arcsine\_original\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

## [1] 1.600753

arcsine\_two\_way(  
 df = main\_analysis\_acc,  
 expression = arcsine\_common\_method  
 )

## `summarise()` regrouping output by 'participant\_id', 'is\_congruent' (override with `.groups` argument)

## [1] 2.179013