Cross-cultural relationships between music, emotion, and visual imagery: A comparative study of Iran, Canada, and Japan [Stage 1 Registered Report]

Code and analyses: Simulation-based power analysis

Shafagh Hadavi[1] Junji Kuroda[2] Taiki Shimozono[2] Juan David Leongómez[3]
Patrick E. Savage[1]

05 July, 2023

[1] Graduate School of Media and Policy / Faculty of Environment and Information Studies, Keio University, Fujisawa, Japan.
[2] YAMAHA Corporation, Hamamatsu, Japan.
[3] Human Behaviour and Evolution Lab, Faculty of Psychology, Universidad El Bosque, Bogota, Colombia.

Correspondence to: SH: shafagh@keio.jp. PES: psavage@sfc.keio.ac.jp.

Descripción

This document contains all code, and step by step explanations for all analyses, figures and tables (including supplementary figures and tables) for:

hadavi, S., Kuroda, J., Shimozono, T., Leongómez, J. D., & Savage, P. E. (in prep). *Cross-cultural relationships between music, emotion, and visual imagery: A comparative study of Iran, Canada, and Japan* [Stage 1 Registered Report]. https://doi.org/10.31234/osf.io/26yg5

Data available from the Open Science Framework (OSF): https://doi.org/10.XXXXX/OSF.IO/XXXXX. This document and its underlying code were created in `R Markdown` by Juan David Leongómez using LaTeX.

Contents

1 Preliminaries

1.1 Load packages

This file was created using `knitr` (Xie, 2014), mostly using `tidyverse` (Wickham et al., 2019) syntax. As such, data wrangling was mainly done using packages such as `dplyr` (Wickham et al., 2022), and most figures were created using `ggplot2` (Wickham, 2016).

Cumulative Link Mixed Models (CLMM) were fitted using `ordinal` (Christensen, 2022), and contrasts and interactions were explored using `emmeans` (Lenth, 2022).

All packages used in this file can be directly installed from the Comprehensive R Archive Network (CRAN). For a complete list of packages used to create this file, and their versions, see section 5, at the end of the document.

```
library(tidyverse)
library(ggpubr)
library(emmeans)
library(ordinal)
```

1.2 Custom functions

To simulate ordinal data with a specific distribution, we used the function `clmm_generate_data`, created by Borders et al. (2022) as part of their tutorial (https://osf.io/e6usd/). To run this function, a number of other custom functions need to be defined and implemented as well. The original code can be found in the `clmm-power-library.R` file (https://osf.io/tjpkf). Here, it is run from source, but in a version oif the file changing the `mc.cores` parameter to 1, to avoid errors.

```
source("clmm-power-library.R", local = knitr::knit_global())
```

2 Data

2.1 Pilot data

To simulate data, first we looked at the distribution of the pilot data.

```
# Load data
sh1 <- read.csv("../sh1.csv") |>
  mutate(Tempo = str_replace(Tempo, "low", "Low")) |>
  mutate(Tempo = str_replace(Tempo, "high", "High")) |>
  mutate_if(is.character,as.factor) |>
  mutate(Tempo = fct_relevel(Tempo, c("Low", "High")))

# Assign unique id to pairs
uniqueval <- unique(sh1[, c("Participant", "Music.country", "Solo.group")])
sh1$groupid <- 0
for (i in 1:dim(uniqueval)[1]) {
  idx <- sh1$Participant == uniqueval$Participant[i] &
    sh1$Music.country == uniqueval$Music.country[i] &
    sh1$Solo.group == uniqueval$Solo.group[i]
  sh1$groupid[idx] <- i
}
```

```r
# Select only relevant variables
data <- sh1  |>
  select(1:9,20)

# Plot distribution of results for density ratings
p1a <- ggplot(data, aes(x = density.tempo, y = after_stat(density),
                       fill = Tempo, color = Tempo)) +
  scale_fill_hue(direction = -1) + scale_colour_hue(direction = -1) +
  geom_histogram(alpha = 0.3, position = "identity", binwidth = 1) +
  labs(y= "Probability", x = "Rating", title = "Density") +
  geom_text(aes(label = format(after_stat(density), digits = 1), y= after_stat(density)),
            stat= "bin", binwidth = 1,
            vjust = -0.2,
            show.legend = FALSE) +
  facet_wrap(~Solo.group)

# Plot distribution of results for arousal ratings
p1b <- ggplot(data, aes(x = arousal.tempo, y = after_stat(density),
                       fill = Tempo, color = Tempo)) +
  scale_fill_hue(direction = -1) + scale_colour_hue(direction = -1) +
  geom_histogram(alpha = 0.3, position = "identity", binwidth = 1) +
  labs(y= NULL, x = "Rating", title = "Arousal") +
  geom_text(aes(label = format(after_stat(density), digits = 1), y= after_stat(density)),
            stat= "bin", binwidth = 1,
            vjust = -0.2,
            show.legend = FALSE) +
  facet_wrap(~Solo.group)

# Arrange plots
p1 <- ggarrange(p1a, p1b,
                common.legend = TRUE,
                legend = "bottom",
                labels = "AUTO")
p1
```

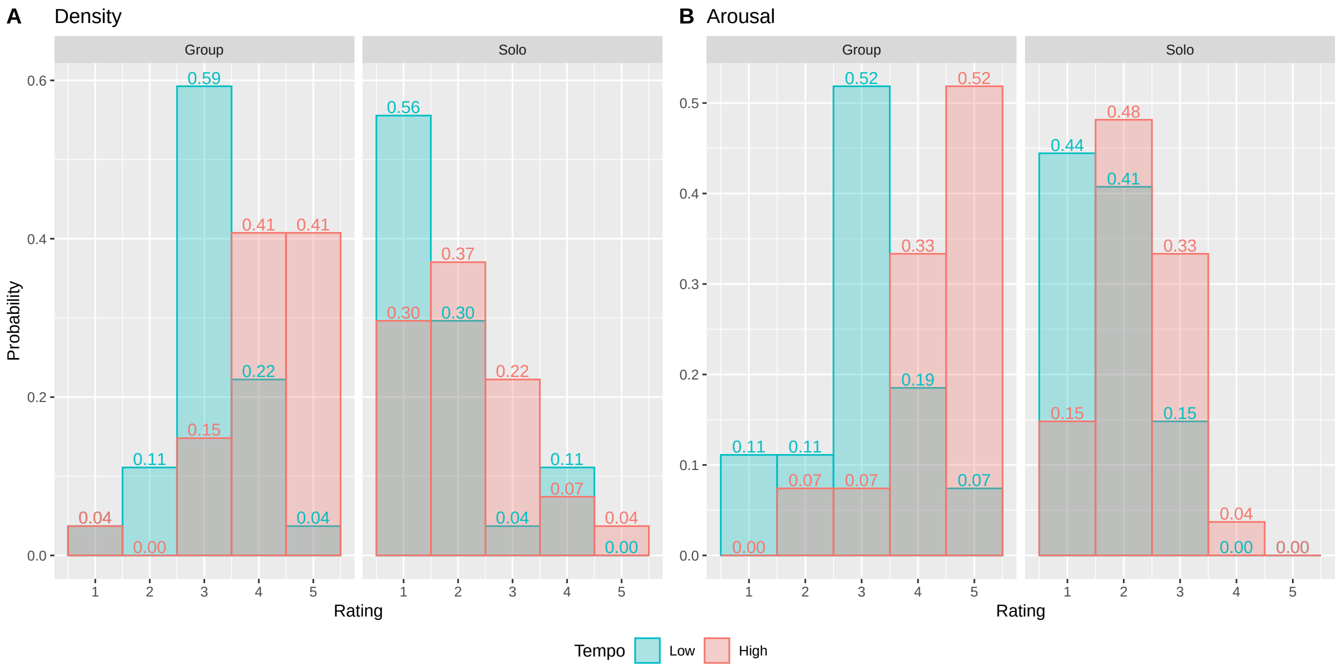

Figure S1. Distribution of results from pilot data for ratings of both Density (**A**) and Arousal (**B**), by Instrumentation (Group, Solo), and Tempo (High, Low).

2.2 Data simulation

While basing a power analysis on pilot data is problematic and can bias the sample size estimation (see e.g., Albers & Lakens, 2018), we used this distribution as a starting point.

The `clmm_generate_data` function (Borders et al., 2022) allows to simulate data with random and fixed effects with an ordinal outcome. In this case, we simulated data for Tempo, the within-subject variable for which there was an *a-priori* prediction, replicating the distribution of the first level (Low) using a specified probability of each rating score (using the argument `control_distribution`). The second level of that within-subject variable (Tempo = High) is generated based on a hypothesized effect (using the argument `effect`). Other important factors such as participant country and music country were randomly assigned, as there are no *a-priori* predictions regarding these factors.

2.2.1 Replication of the pilot data distribution

To obtain distributions similar to the probabilities observed in the pilot data, we first simulated individual data for Density and Arousal ratings, separated by Instrumentation type (Group, Solo). In all cases, we simulated data from 10000 participants, and changed the value assigned to the `effect` argument until the distribution of the second level of the within-subject variable (Tempo = High) resembled the pilot data.

```r
# Density ratings for group music
dat.den.gr.PILOT <- clmm_generate_data(n_participants = 10000,
                                       n_trials = 3,
                                       control_distribution = c(.04, .11, .59, .22, .04),
                                       effect = 3.2,
                                       participant_variation = 1,
                                       within_subject = TRUE,
                                       control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Density.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Group", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))

# Density ratings for solo music
dat.den.so.PILOT <- clmm_generate_data(n_participants = 10000,
                                n_trials = 3,
                                control_distribution = c(.55, .30, .04, .10, .01),
                                effect = 1.2,
                                participant_variation = 1,
                                within_subject = TRUE,
                                control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Density.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Solo", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))

# Arousal ratings for group music
dat.aro.gr.PILOT <- clmm_generate_data(n_participants = 10000,
                                n_trials = 3,
                                control_distribution = c(.11, .11, .52, .19, .07),
                                effect = 3.1,
                                participant_variation = 1,
                                within_subject = TRUE,
                                control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Arousal.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Group", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))

# Arousal ratings for solo music
dat.aro.so.PILOT <- clmm_generate_data(n_participants = 10000,
                                n_trials = 3,
                                control_distribution = c(.43, .41, .14, .01, .01),
                                effect = 1.6,
                                participant_variation = 1,
                                within_subject = TRUE,
                                control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Arousal.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Solo", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))
```

We then merged these four data frames to obtain a simulated data frame.

```
dat.sim.PILOT <- left_join(bind_rows(dat.den.gr.PILOT, dat.den.so.PILOT),
                           bind_rows(dat.aro.gr.PILOT, dat.aro.so.PILOT),
                           relationship = "many-to-many") |>
  mutate(Tempo = fct_relevel(Tempo, c("Low", "High")))
```

The distribution of this replication of the pilot data distribution is represented in S2

```
# Plot distribution of results for density ratings
p2a <- ggplot(dat.sim.PILOT, aes(x = Density.tempo, y = after_stat(density),
                                 fill = Tempo, color = Tempo)) +
  scale_fill_hue(direction = -1) + scale_colour_hue(direction = -1) +
  geom_histogram(alpha = 0.3, position = "identity", binwidth = 1) +
  labs(y= "Probability", x = "Rating") +
  geom_text(aes(label = format(after_stat(density), digits = 1), y= after_stat(density)),
            stat= "bin", binwidth = 1,
            vjust = -0.2,
            show.legend = FALSE) +
  facet_wrap(~Solo.group)

# Plot distribution of results for arousal ratings
p2b <- ggplot(dat.sim.PILOT, aes(x = Arousal.tempo, y = after_stat(density),
                                 fill = Tempo, color = Tempo)) +
  scale_fill_hue(direction = -1) + scale_colour_hue(direction = -1) +
  geom_histogram(alpha = 0.3, position = "identity", binwidth = 1) +
  labs(y= NULL, x = "Rating") +
  geom_text(aes(label = format(after_stat(density), digits = 1), y= after_stat(density)),
            stat= "bin", binwidth = 1,
            vjust = -0.2,
            show.legend = FALSE) +
  facet_wrap(~Solo.group)

# Arrange plots
p2 <- ggarrange(p2a +
                  labs(subtitle = "Density"),
                p2b  +
                  labs(subtitle = "Arousal"),
                common.legend = TRUE,
                legend = "bottom",
                labels = "AUTO")
p2
```

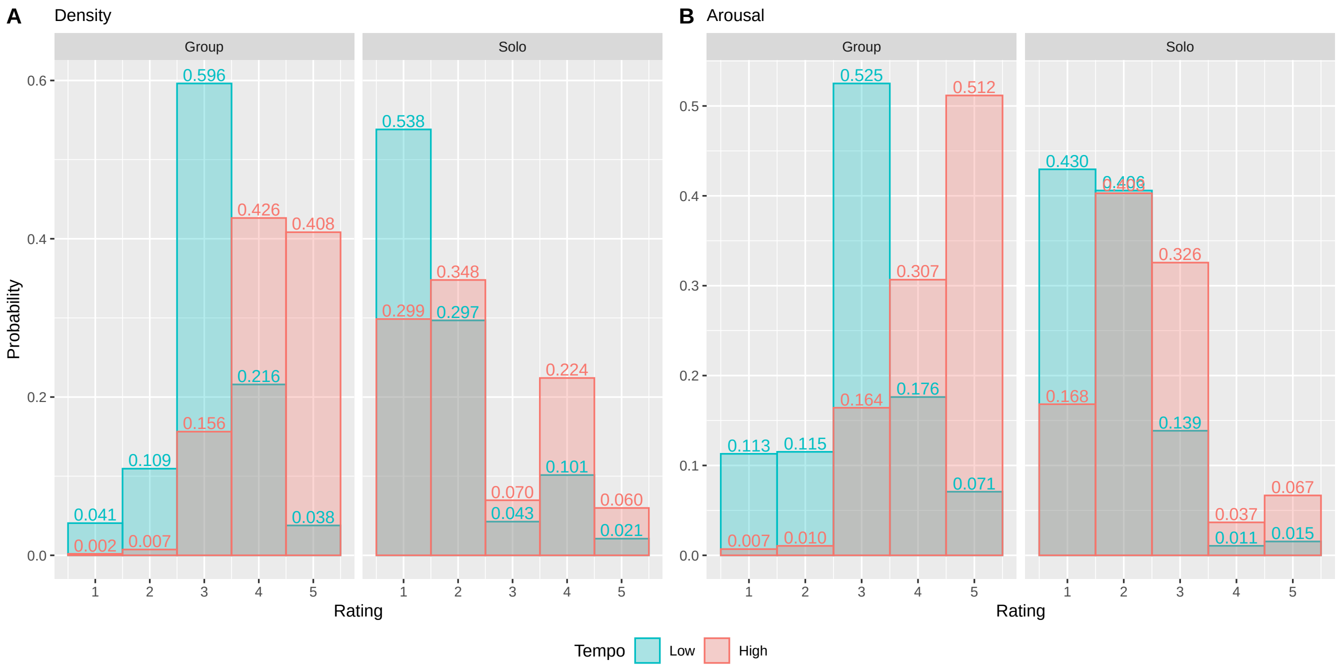

Figure S2. Distribution of results from a simulated replication of the pilot data distribution for ratings of both Density (**A**) and Arousal (**B**), by Instrumentation (Group, Solo), and Tempo (High, Low).

2.2.2 Simulation of data with a more conservative effect

While the data simulated in the previous section resembled the distribution of the pilot data, using these data would be problematic for at least two reasons:

First, the effects of Tempo were extremely large (3.2, 1.2, 3.1, 1.6), particularly for ratings of Group instrumentation (3.2, and 3.1). With such differences, a very high statistical power of about $1 - \beta = 0.95$ can be achieved with 2 or 3 participants. And second, as mentioned before, estimating effects from pilot data is problematic, as it biases the results (Albers & Lakens, 2018).

For this reason, we decided to maintain the distributions of probabilities for the first Tempo level (Low), as it is a goiod starting point to estimate the distribution of ratings that participants would assign, but simulated the second level (Tempo = High) with a much more conservative effect of 1 in all cases.

```
# Density ratings for group music
dat.den.gr <- clmm_generate_data(n_participants = 10000,
                                 n_trials = 3,
                                 control_distribution = c(.04, .11, .59, .22, .04),
                                 effect = 1,
                                 participant_variation = 1,
                                 within_subject = TRUE,
                                 control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Density.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Group", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))
```

```
# Density ratings for solo music
dat.den.so <- clmm_generate_data(n_participants = 10000,
                                 n_trials = 3,
                                 control_distribution = c(.55, .30, .04, .10, .01),
                                 effect = 1,
                                 participant_variation = 1,
                                 within_subject = TRUE,
                                 control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Density.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Solo", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))

# Arousal ratings for group music
dat.aro.gr <- clmm_generate_data(n_participants = 10000,
                                 n_trials = 3,
                                 control_distribution = c(.11, .11, .52, .19, .07),
                                 effect = 1,
                                 participant_variation = 1,
                                 within_subject = TRUE,
                                 control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Arousal.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Group", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))

# Arousal ratings for solo music
dat.aro.so <- clmm_generate_data(n_participants = 10000,
                                 n_trials = 3,
                                 control_distribution = c(.43, .41, .14, .01, .01),
                                 effect = 1,
                                 participant_variation = 1,
                                 within_subject = TRUE,
                                 control_weight = .5) |>
  mutate(group = ifelse(group == 0, "Low", "High")) |>
  dplyr::rename(Arousal.tempo = pas) |>
  dplyr::rename(Tempo = group) |>
  dplyr::rename(Participant = id) |>
  mutate(Participant = paste0("p", Participant)) |>
  select(1:3) |>
  mutate(pair = rep(1:3, times = 20000)) |>
  mutate(Participant.country = rep(rep(c("Iran", "Canada", "Japan"), each = 10000), 2)) |>
  mutate(Music.country = rep(c("Iran", "Canada", "Japan"), times = 20000)) |>
  mutate(Solo.group = rep("Solo", TIMES = 60000)) |>
  select(c(1,6,5,2,7,3))
```

Again, we then merged these four data frames to obtain a final simulated data frame.

```
dat.sim <- left_join(bind_rows(dat.den.gr, dat.den.so),
                     bind_rows(dat.aro.gr, dat.aro.so),
                     relationship = "many-to-many") |>
  mutate(Tempo = fct_relevel(Tempo, c("Low", "High")))

# Assign unique id to pairs
uniqueval <- unique(dat.sim[, c("Participant", "Music.country", "Solo.group")])
dat.sim$groupid <- 0
for (i in 1:dim(uniqueval)[1]) {
  idx <- dat.sim$Participant == uniqueval$Participant[i] &
    dat.sim$Music.country == uniqueval$Music.country[i] &
    dat.sim$Solo.group == uniqueval$Solo.group[i]
  dat.sim$groupid[idx] <- i
}
```

The distribution of these simulated data is represented in S3

```
# Plot distribution of results for density ratings
p3a <- ggplot(dat.sim, aes(x = Density.tempo, y = after_stat(density),
                           fill = Tempo, color = Tempo)) +
  scale_fill_hue(direction = -1) + scale_colour_hue(direction = -1) +
  geom_histogram(alpha = 0.3, position = "identity", binwidth = 1) +
  labs(y= "Probability", x = "Rating") +
  geom_text(aes(label = format(after_stat(density), digits = 1), y= after_stat(density)),
            stat= "bin", binwidth = 1,
            vjust = -0.2,
            show.legend = FALSE) +
  facet_wrap(~Solo.group)

# Plot distribution of results for arousal ratings
p3b <- ggplot(dat.sim, aes(x = Arousal.tempo, y = after_stat(density),
                           fill = Tempo, color = Tempo)) +
  scale_fill_hue(direction = -1) + scale_colour_hue(direction = -1) +
  geom_histogram(alpha = 0.3, position = "identity", binwidth = 1) +
  labs(y= NULL, x = "Rating") +
  geom_text(aes(label = format(after_stat(density), digits = 1), y= after_stat(density)),
            stat= "bin", binwidth = 1,
            vjust = -0.2,
            show.legend = FALSE) +
  facet_wrap(~Solo.group)

# Arrange plots
p3 <- ggarrange(p3a +
                  labs(subtitle = "Density"),
                p3b  +
                  labs(subtitle = "Arousal"),
                common.legend = TRUE,
                legend = "bottom",
                labels = "AUTO")
p3
```

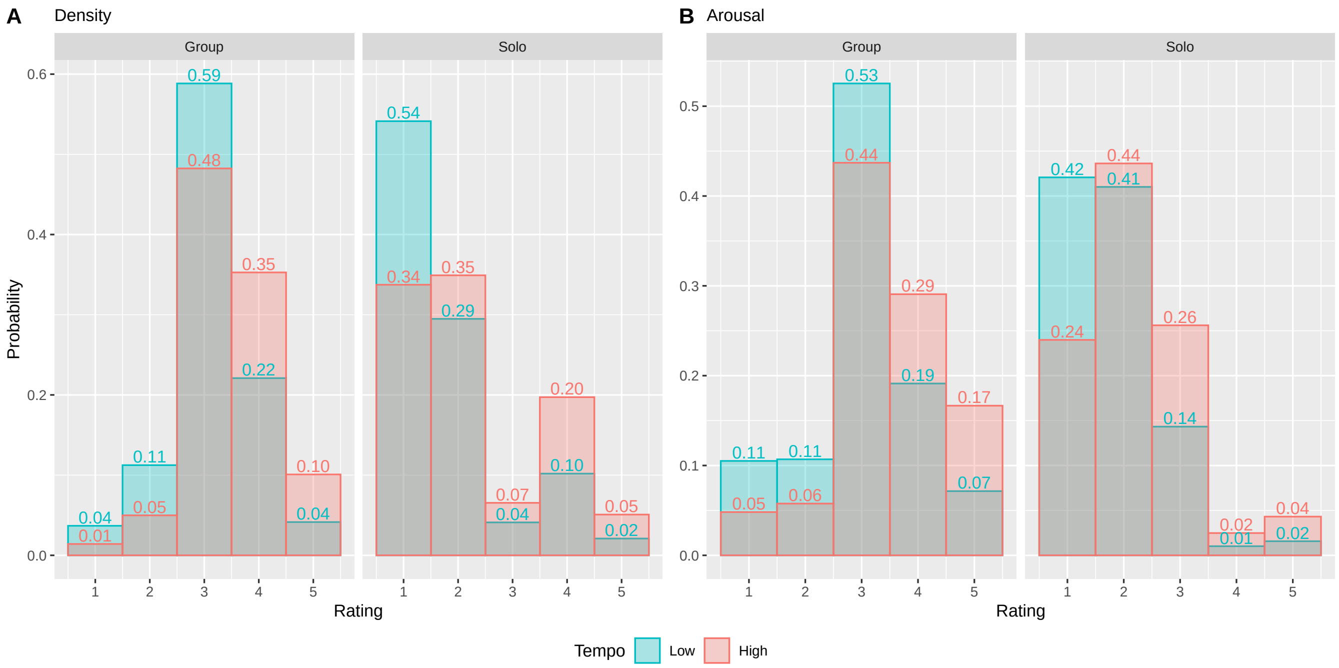

Figure S3. Distribution of results from a conservative simulated distribution for ratings of both Density (**A**) and Arousal (**B**), by Instrumentation (Group, Solo), and Tempo (High, Low).

2.3 Comparisons of the distribuition of results between pilot and simulated data

```
p4 <- ggarrange(ggarrange(p1a + labs(title = "Pilot data"),
                          p1b + labs(title = " "),
                          legend = "none"),
                ggarrange(p2a + labs(title = "Simulation (pilot data replication)"),
                          p2b + labs(title = " "),
                          legend = "none"),
                ggarrange(p3a + labs(title = "Final simulation (conservative effect)"),
                          p3b + labs(title = " "),
                          common.legend = TRUE, legend = "bottom"),
                common.legend = TRUE,
                labels = "AUTO",
                nrow = 3,
                heights = c(1,1,1.1))
p4
```

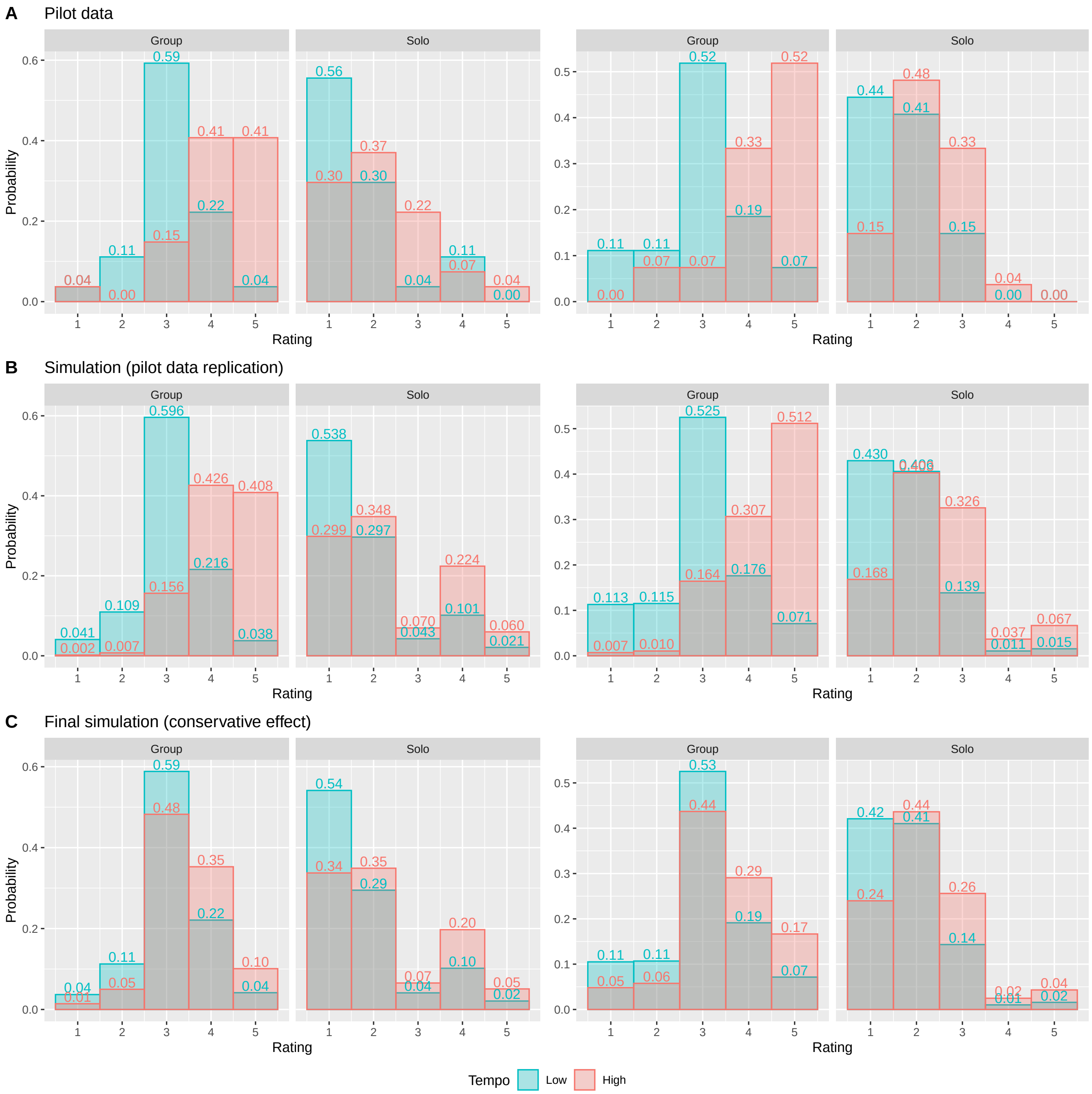

Figure S4. Distribution of results of ratings of both Density and Arousal (**B**), by Instrumentation (Group, Solo), and Tempo (High, Low). **A** pilot data; **B** simulation replicating pilot data distribution; **C** simulation with a more conservative effect.

3 Power analysis

Using the final, simulated population ($N = 10000$ participants, with 6 paired observations per participant), we conducted a simulation-based power analysis. To do this, we first defined the desired number of simulations, the sample size (per country) of each simulation, and the α value (significance level) for all statistical tests:

```
# Number of simulations to run
num_sims.clmm = 100

# Sample size for each simulation (number of participants per country)
```

```
sample_size.clmm = 25

# Significance level
alpha.clmm = 0.05
```

Then, models were fitted from the defined 100 random samples extracted from the simulated population.

From each sample, two Cumulative Link Mixed Models (CLMM) were fitted: one for Density ratings, and one for arousal ratings. All models were fitted with the following call: `clmm(DV ~ Tempo * Solo.group * Participant.country * Music.country + (1 + Tempo | Participant)`.

This is, all models had the same structure, which included the main effects and all possible interactions between Tempo (Low, High), Instrumentation (Group, Solo), Participant country (Iran, Canada, Japan), and Music country (Iran, Canada, Japan) as fixed effects, as well as random intercepts and random slopes between Tempo conditions for each participant.

For each random sample, we adjusted the number of participants per country (in this case, $n = 25$) to achieve the target statistical power of at least $1 - \beta = 0.95$ for both the Density and Arousal models.

3.1 Select random samples

```
# Get participant IDs
part <- dat.sim  |>
  group_by(Participant.country) |>
  distinct(Participant) |>
  ungroup()

# Select samples of random participants
samples.clmm <- map_dfr(seq_len(num_sims.clmm), ~part |>
                        sample_n(sample_size.clmm) |>
                        mutate(sample.clmm = as.factor(.x)))

# Final data base of data for each participant selected for each sample
samples.clmm.long <- left_join(samples.clmm, dat.sim, by = c("Participant",
                                                             "Participant.country"),
                                relationship = "many-to-many") |>
  mutate(Density.tempo = as.factor(Density.tempo)) |>
  mutate(Arousal.tempo = as.factor(Arousal.tempo)) |>
  mutate(Tempo = as.factor(Tempo))
```

3.2 Density power analysis

```
# Create empty data frame
clmm.comps.den <- data.frame(Sample = 1:num_sims.clmm) #create empty data frame

# Loop to fit a model for each random sample and extract the p-value of the Tempo contrasts
for (i in 1:num_sims.clmm){
  samp <- samples.clmm.long |>
    filter(sample.clmm == i)
  tryCatch(mod <- clmm(Density.tempo ~ Tempo * Solo.group * Participant.country * Music.country +
                      (1 + Tempo | Participant), data = samp),
           error = function(e) {
             message('Error')
             print(e)
           })
  tryCatch(comp <- emmeans(mod, pairwise~Tempo),
           error = function(e) {
            message('Error')
            print(e)
          })
  clmm.comps.den$p[i] <- round(data.frame(comp$contrast)$p.value, 3)
  clmm.comps.den$`Statistical signicance`[i] <- ifelse(clmm.comps.den$p[i] <= alpha.clmm,
                                                       "Significant", "Non-significant")
}
```

```
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
```

```
# Calculate power (proportion of simulations in which the Tempo contrast was significant)
power.clmm.den <- sum(clmm.comps.den$`Statistical signicance` == "Significant")/num_sims.clmm
```

3.3 Arousal power analysis

```
# Create empty data frame
clmm.comps.aro <- data.frame(Sample = 1:num_sims.clmm) #create empty dataframe

# Loop to fit a model for each random sample and extract the p-value of the Tempo contrasts
for (i in 1:num_sims.clmm){
  samp <- samples.clmm.long |>
    filter(sample.clmm == i)
  tryCatch(mod <- clmm(Arousal.tempo ~ Tempo * Solo.group * Participant.country * Music.country +
                         (1 + Tempo | Participant), data = samp),
           error = function(e) {
             message('Error')
             print(e)
           })
  tryCatch(comp <- emmeans(mod, pairwise~Tempo),
           error = function(e) {
             message('Error')
             print(e)
           })
  clmm.comps.aro$p[i] <- round(data.frame(comp$contrast)$p.value, 3)
  clmm.comps.aro$`Statistical signicance`[i] <- ifelse(clmm.comps.aro$p[i] <= alpha.clmm,
                                                       "Significant", "Non-significant")
}
```

```
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
## <simpleError in vcov.clm(object, method = "Cholesky"): Cannot compute vcov:
## Hessian is not positive definite>
```

```
# Calculate power (proportion of simulations in which the Tempo contrast was significant)
power.clmm.aro <- sum(clmm.comps.aro$`Statistical signicance` == "Significant")/num_sims.clmm
```

3.4 Achieved power

```
p5a <- ggplot(clmm.comps.den, aes(x = p, fill = `Statistical signicance`)) +
  scale_fill_hue(direction = -1) +
  geom_histogram(bins = 1/alpha.clmm, breaks = seq(0, 1, alpha.clmm)) +
  labs(y = "Count", x = "p-value", title = "Density") +
  annotate("text", x = 0.5, y = sum(clmm.comps.den$`Statistical signicance` == "Significant") * 0.9,
           label = paste0("Power = ", round(power.clmm.den, 3),
                          "\nSample size = ", sample_size.clmm,
                          " participants per country\n(6 paired responses per participant)"))

p5b <- ggplot(clmm.comps.aro, aes(x = p, fill = `Statistical signicance`)) +
  scale_fill_hue(direction = -1) +
  geom_histogram(bins = 1/alpha.clmm, breaks = seq(0, 1, alpha.clmm)) +
  labs(y = "Count", x = "p-value", title = "Arousal") +
  annotate("text", x = 0.5, y = sum(clmm.comps.aro$`Statistical signicance` == "Significant") * 0.9,
           label = paste0("Power = ", round(power.clmm.aro, 3),
                          "\nSample size = ", sample_size.clmm,
                          " participants per country\n(6 paired responses per participant)"))

p5 <- ggarrange(p5a, p5b,
                common.legend = TRUE,
                legend = "bottom",
                labels = "AUTO")
p5
```

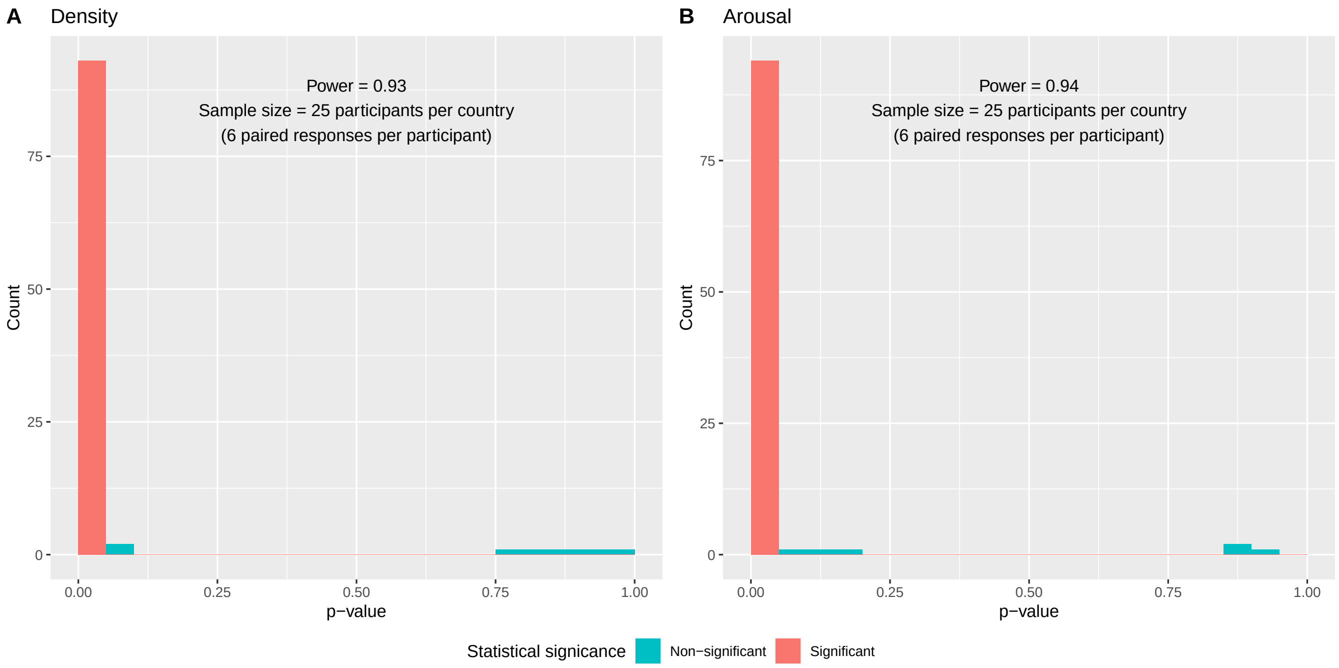

Figure S5. Histogram of *p*-values for the contrasts between Tempo conditions (Low, High) for all simulations. The estatistical power is the proportion of significant results. Significant results are in red.

4 Analysis script

Here, using a randomly selected sample, we fit one model as if it was the data collected from the experiment. This is not only to make sure that the data are sufficient and the structure is adequate to fit the model, but to plan in advance all the code.

```
# Select a random sample
ex.data <- samples.clmm.long %>%
  filter(sample.clmm == sample(1:num_sims.clmm, 1, replace = TRUE))
```

Because only the effect of Tempo was predicted and simulated *a-priori*, all other effects and interactions are completely random.

```
p6a <- ggplot(ex.data, aes(x = Tempo, y = Density.tempo, color = Participant.country, shape = Music.countr
  geom_violin(aes(group = Tempo), draw_quantiles = 0.5) +
  geom_point(position = position_dodge(0.3), size = 3) +
  geom_line(aes(group = factor(groupid)),
            linewidth = 0.5, linetype = "dashed", position = position_dodge(0.3), alpha = 0.5) +
  labs(y= "Density", x = "Tempo", title = "Density") +
  scale_x_discrete(limits = c("Low", "High")) +
  scale_size_discrete(breaks = c("Solo", "Group")) +
  theme(plot.title = element_text(face = "bold", hjust = 0.5)) +
  facet_wrap(~Solo.group)

p6b <- ggplot(ex.data, aes(x = Tempo, y = Arousal.tempo, color = Participant.country, shape = Music.countr
   geom_violin(aes(group = Tempo), draw_quantiles = 0.5) +
  geom_point(position = position_dodge(0.3), size = 3) +
  geom_line(aes(group = factor(groupid)),
            linewidth = 0.5, linetype = "dashed", position = position_dodge(0.3), alpha = 0.5) +
  labs(y= "Arousal", x = "Tempo", title = "Arousal") +
  scale_x_discrete(limits = c("Low", "High")) +
  scale_size_discrete(breaks = c("Solo", "Group")) +
  theme(plot.title = element_text(face = "bold", hjust = 0.5)) +
  facet_wrap(~Solo.group)

p6 <- ggarrange(p6a, p6b,
                common.legend = TRUE,
                legend = "bottom")
p6
```

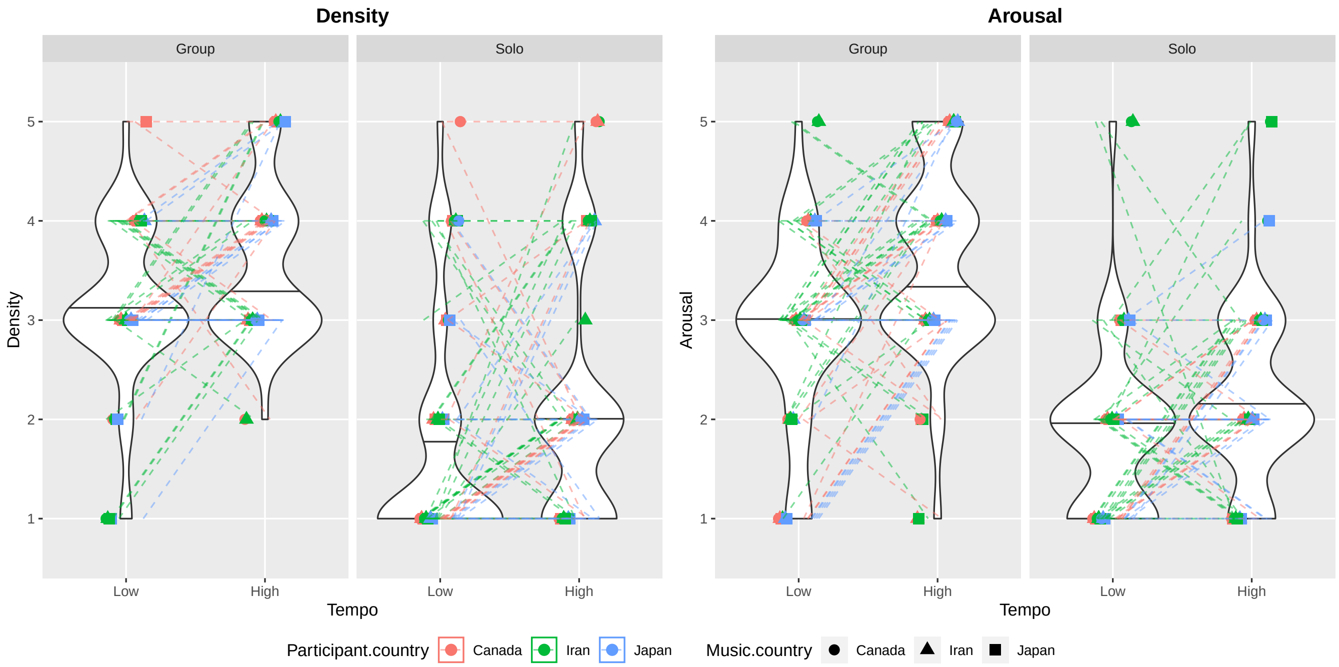

Figure S6. Data for Density and Arousal ratings from a ramdomly selected, simulated sample. Dashed linesrepresent the within-subject effect of the Tempo manipulation.

We fitted the model with the same structure: `clmm(DV ~ Tempo * Solo.group * Participant.country * Music.country + (1 + Tempo | Participant)`. Estimated marginal means and confidence intervals for this simulation are represented in S7.

```
# Density model
model.den <- clmm(Density.tempo ~ Tempo * Solo.group * Participant.country * Music.country + (1 | Particip
              data = ex.data,
              contrasts = TRUE)

# Arousal model
model.aro <- clmm(Arousal.tempo ~ Tempo * Solo.group * Participant.country * Music.country + (1 | Particip
              data = ex.data,
              contrasts = TRUE)

p7a <- emmip(model.den, Solo.group ~ Tempo | Participant.country + Music.country,
      CIs = TRUE) +
  labs(title = "Density")

p7b <- emmip(model.aro, Solo.group ~ Tempo | Participant.country + Music.country,
      CIs = TRUE) +
  labs(title = "Arousal")

p7 <- ggarrange(p7a, p7b,
                common.legend = TRUE,
                legend = "bottom")
```

```
p7
```

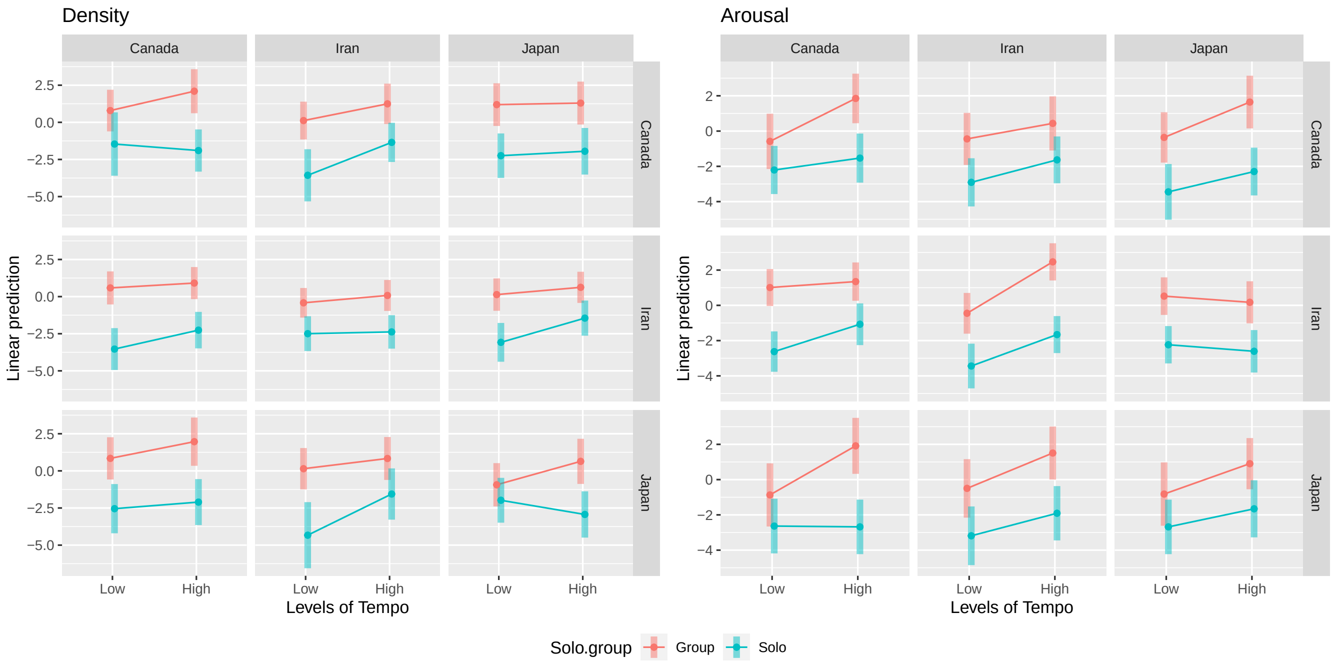

Figure S7. Estimated marginal means and confidence intervals for the within-subject effects of Tempo from a ramdomly selected, simulated sample. Columns represent data from simulated participants from each country, while rows represent the simulated response to music from each country.

5 Session info (for reproducibility)

```
library(pander)
pander(sessionInfo(), locale = FALSE)
```

R version 4.3.1 (2023-06-16 ucrt)

Platform: x86_64-w64-mingw32/x64 (64-bit)

attached base packages: *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

other attached packages: *pander(v.0.6.5)*, *ordinal(v.2022.11-16)*, *emmeans(v.1.8.6)*, *ggpubr(v.0.6.0)*, *lubridate(v.1.9.2)*, *forcats(v.1.0.0)*, *stringr(v.1.5.0)*, *dplyr(v.1.1.2)*, *purrr(v.1.0.1)*, *readr(v.2.1.4)*, *tidyr(v.1.3.0)*, *tibble(v.3.2.1)*, *ggplot2(v.3.4.2)*, *tidyverse(v.2.0.0)* and *knitr(v.1.43)*

loaded via a namespace (and not attached): *gtable(v.0.3.3)*, *xfun(v.0.39)*, *rstatix(v.0.7.2)*, *lattice(v.0.21-8)*, *tzdb(v.0.4.0)*, *numDeriv(v.2016.8-1.1)*, *vctrs(v.0.6.3)*, *tools(v.4.3.1)*, *generics(v.0.1.3)*, *fansi(v.1.0.4)*, *highr(v.0.10)*, *ucminf(v.1.2.0)*, *pkgconfig(v.2.0.3)*, *Matrix(v.1.5-4.1)*, *lifecycle(v.1.0.3)*, *compiler(v.4.3.1)*, *farver(v.2.1.1)*, *tinytex(v.0.45)*, *munsell(v.0.5.0)*, *codetools(v.0.2-19)*, *carData(v.3.0-5)*, *htmltools(v.0.5.5)*, *yaml(v.2.3.7)*, *pillar(v.1.9.0)*, *car(v.3.1-2)*, *MASS(v.7.3-60)*, *abind(v.1.4-5)*, *nlme(v.3.1-162)*, *tidyselect(v.1.2.0)*, *digest(v.0.6.31)*, *mvtnorm(v.1.2-2)*, *stringi(v.1.7.12)*, *bookdown(v.0.34)*, *labeling(v.0.4.2)*, *cowplot(v.1.1.1)*, *fastmap(v.1.1.1)*, *grid(v.4.3.1)*, *colorspace(v.2.1-0)*, *cli(v.3.6.1)*, *beepr(v.1.3)*, *magrittr(v.2.0.3)*, *utf8(v.1.2.3)*, *broom(v.1.0.5)*, *withr(v.2.5.0)*, *scales(v.1.2.1)*, *backports(v.1.4.1)*, *timechange(v.0.2.0)*, *estimability(v.1.4.1)*, *rmarkdown(v.2.22)*, *audio(v.0.1-10)*, *gridExtra(v.2.3)*, *ggsignif(v.0.6.4)*, *hms(v.1.1.3)*, *evaluate(v.0.21)*, *rlang(v.1.1.1)*, *Rcpp(v.1.0.10)*, *xtable(v.1.8-4)*, *glue(v.1.6.2)*, *rstudioapi(v.0.14)* and *R6(v.2.5.1)*

Supplementary references

Albers, C., & Lakens, D. (2018). When power analyses based on pilot data are biased: Inaccurate effect size estimators and follow-up bias. *Journal of Experimental Social Psychology*, *74*, 187–195. https://doi.org/10.101

6/j.jesp.2017.09.004

Borders, J. C., Grande, A. A., & Troche, M. S. (2022). *Power Analysis with Ordinal Outcomes: A Tutorial*. https://osf.io/8sc5e

Christensen, R. H. B. (2022). *Ordinal—regression models for ordinal data*. https://CRAN.R-project.org/package=ordinal

Lenth, R. V. (2022). *Emmeans: Estimated marginal means, aka least-squares means*. https://CRAN.R-project.org/package=emmeans

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L., & Müller, K. (2022). *Dplyr: A grammar of data manipulation*. https://CRAN.R-project.org/package=dplyr

Xie, Y. (2014). Knitr: A comprehensive tool for reproducible research in R. In V. Stodden, F. Leisch, & R. D. Peng (Eds.), *Implementing reproducible computational research*. Chapman and Hall/CRC. https://doi.org/10.1201/9781315373461-1