**Introduction**

Humans have the capacity to learn from changes in their environment and adjust their knowledge accordingly (Nassar et al., 2019).

**Methods & Procedure**

*Procedure*

6 healthy adults, including three males and three females (*m*age = 25.17, *SD* = 2,03 years), from an EEG course at the University of Hamburg volunteered to participate in this study and were compensated with course credits. The participants could choose their preferred appointment to come to the lab via an online registration form. After arrival, the participants’ heads were measured to find the suitable the EEG cap. Then the cap was populated on a model’s head followed by applying the populated cap on the participant’s head. At last, they were presented by a learning task. The whole procedure of applying the EEG cap and performing the task was done in a 180 minute time window for each participant.

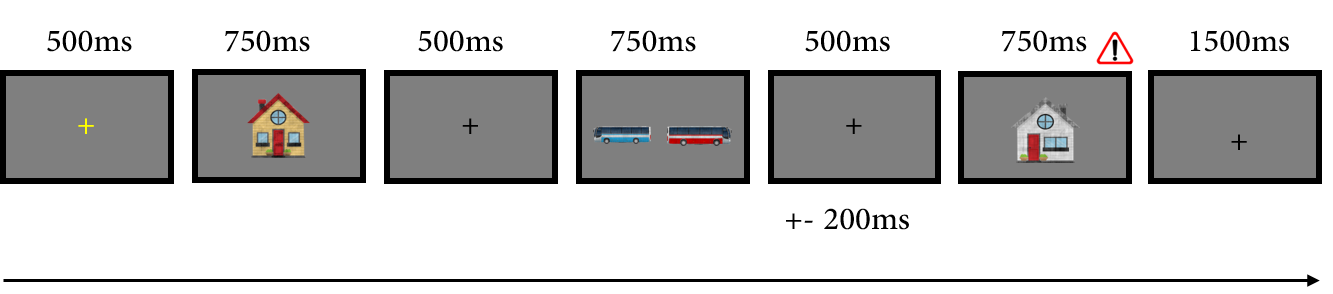
*Learning Task*

The participants were presented by a learning task -the bus task-.The task began with a yellow fixation cross in the middle of the monitor for 500ms. Then house was appeared to indicate the choice of interest for 750ms followed by a fixation cross for 500ms. After that, the busses were appeared for 750ms that the participants had to choose by pressing the corresponding buttons. Then a fixation cross was appeared after 300ms or 700ms of the bus choice for 500ms and the house associated with the chosen bus was presented as a feedback, accompanied by a warning sign solely in practice trials. At last, a 1500ms long fixation cross was presented to end the trial. If a participant pressed a button before the onset of the buses, a prompt popped up with the text “too fast!”. In the opposite condition, if the participant did not choose any of the busses a prompt appeared with the text “too slow!”.

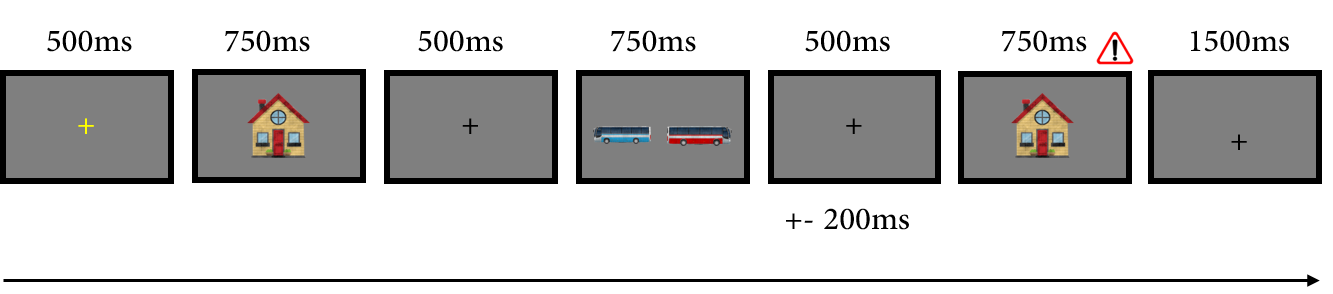
Figure 1.

**Depiction of Trials**

**a**



**b**



*Note*. The whole trial lasted from 3,550ms to 3,950ms. In figure (a) an expectancy violation is depicted, which indicates a surprising condition. The participant chose bus associated with the yellow house, however, the white house appeared on the screen as a feedback. In figure (b) no expectancy violation is illustrated, because the house associated with the chosen bus is appeared.

The task comprised two conditions namely oddball and reversal learning, which both had two associations. In the reversal learning condition 80 percent of the trials are common trials, which follow the instructed ans usual task structure. The remaining 20 percent of the trials are those, in which an association is switched to another called rare trials. In the oddball condition, the same logic applies in terms of the number of associations. However, in this condition 80 percent of the trials belong to one association forming the oddball common trials and the remainder shape the oddball rare trials, in which the task switches from on association to another similar to the rare trials in thee reversal learning condition. To compare the associations in each condition, one block is required in reversal learning condition, as opposed to two blocks of different associations in the oddball condition.

Each participant completed 7 blocks of practice trials and 9 experimental blocks, with 60 trials in the reversal learning blocks and 30 trials in the oddball blocks.

*EEG Data Preprocessing*

The preprocessing steps were mainly inspired by the instructions of Luck (2014) in MATLAB (Mathworks, Natick MA) through EEGLAB package (Delorme & Makeig, 2004). The EEG data were recorded using 64 electrodes in a 10-20 system with a BrainVision recorder, while the ground electrode was located at Fpz and the reference electord at FCz. Data were rereferenced to the average of left and right mastoids. The data was resampled to 500 Hz. During the initial visual inspection no bad channels were identified in any participant, therefore no channels were interpolated. The raw data was filtered with a 0.5 Hz to 30 Hz band pass and segmented into 1500 ms epochs, starting at 500 ms before the stimilui onset until 1000 ms after it in each trial. . In average 10 epochs per participant -less than 3 percent of the epochs- were removed in the secondary visual inspection.

The data were baseline corrected at 500 ms before the stimuli until its onset. To remove the artifacts an independent component analysis (ICA) was utilized in two steps. ICA was first ran without removing any components followed by manually rejecting the artifacts. Artifacts such as eye blinks, heart beat, head movements, and impedances were removed in this stage, however, no additional components were excluded due to the potential risk of mistaking genuine neural activity for artifacts.

*Behavioral Data Preprocessing*

All of the behavioral data preprocessing was done in RStudio using MuMIn (Bartoń, 2010), broom (Robinson et al., 2014), tidyr (Wickham et al., 2014), ggplot2 (Wickham, 2016), lme4 (Bates et al., 2015), lmerTest (Kuznetsova et al., 2017), broom.mixed (Bolker & Robinson, 2018), and dplyr (Wickham et al., 2023).

References

Wickham H, François R, Henry L, Müller K, Vaughan D (2023). dplyr: A Grammar of Data Manipulation. R package version 1.1.4, https://dplyr.tidyverse.org.

Robinson D, Hayes A, Couch S, Ogle D (2014). broom: Convert Statistical Analysis Objects into Tidy Data Frames. R package version 0.7.0, https://CRAN.R-project.org/package=broom.

Bolker B, Robinson D (2018). broom.mixed: Tidying Methods for Mixed Models. R package version 0.2.4, https://CRAN.R-project.org/package=broom.mixed.

Wickham H (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, https://ggplot2.tidyverse.org.

Wickham H, Henry L, Girlich M (2014). tidyr: Tidy Messy Data. R package version 1.2.0, https://CRAN.R-project.org/package=tidyr.

Bates D, Mächler M, Bolker B, Walker S (2015). Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48, https://doi.org/10.18637/jss.v067.i01.

Kuznetsova A, Brockhoff PB, Christensen RHB (2017). lmerTest Package: Tests in Linear Mixed Effects Models. Journal of Statistical Software, 82(13), 1-26, https://doi.org/10.18637/jss.v082.i13.

Bartoń K (2010). MuMIn: Multi-Model Inference. R package version 1.43.17, https://CRAN.R-project.org/package=MuMIn.