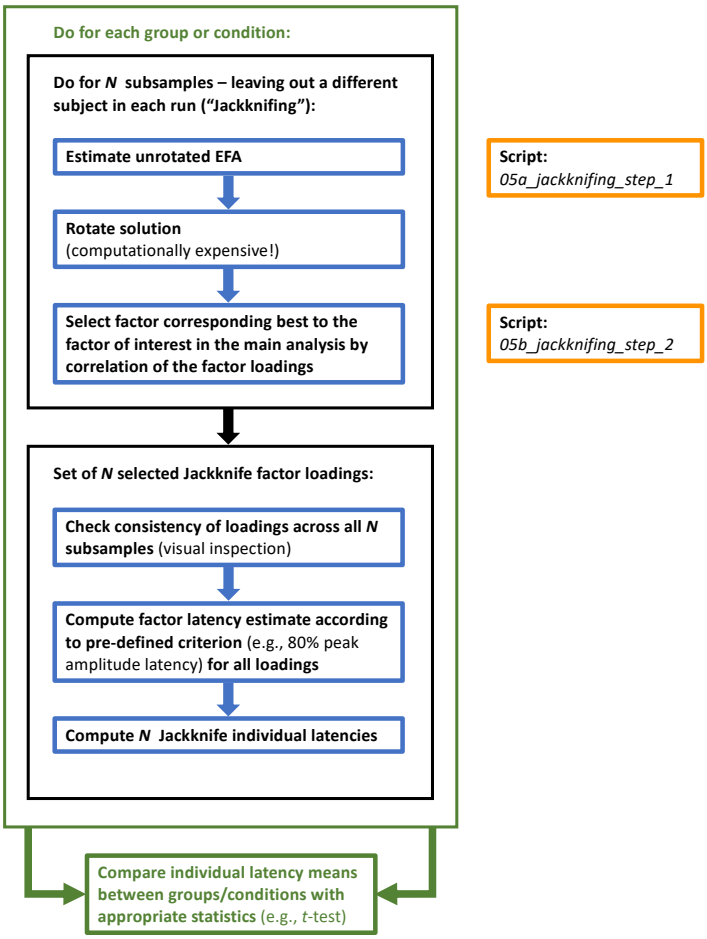
Analysis of Latency Effects

Apart from the analysis of amplitude differences between conditions, we were also interested in latency differences between the adult and the child group. As outlined before, latency differences can only be investigated for separate PCAs because, in a combined PCA, the time courses (i.e., factor loadings) are per definition equal due to the inherent assumption of strict measurement invariance[1]. We adopted a Jackknife approach to test for latency effects that has been applied in ERP analysis quite successfully (Kiesel et al., 2008; J. Miller et al., 1998; Ulrich & Miller, 2001). Instead of applying the leave-one-out Jackknifing resampling approach to the observed ERP, one could simply apply it to the factor-wise reconstruction of the ERP. However, remember that the loadings represent the time courses of the factors' activity. Therefore, all information about the latency of a factor is contained in the loadings – and the reconstructed ERP would just be a rescaled representation of the same information. Thus, we did not reconstruct the ERPs in an intermediate step but applied the Jackknife procedure directly to the estimated unstandardized loadings from the (rotated) PCAs. This procedure is implemented in the scripts *05a_jackknifing_step_1.R* and *05b_jackknifing_step_2.R* (see Figure 11 for an overview).

[1] That is, when latency differences between conditions are of interest, separate PCAs would need to be conducted per condition as well. In that case, it is important to note that the resampling scheme for dependent samples would need to be *blocked* to account for dependencies between repeated measurements. That is, the resampling should still be conducted in a participant-wise manner – *not* separately for both conditions (e.g., Lahiri, 2002).

Figure 1

Overview of all processing steps and accompanying scripts for the Jackknifing approach



In general, Jackknifing is a resampling procedure which can be used to estimate the uncertainty of estimated parameters and to obtain confidence intervals in a non-parametric way (i.e., without specific knowledge regarding the sampling distribution). At its heart lies a resampling step in which subsamples are repeatedly drawn from the original sample similarly to bootstrap procedures (Efron & Tibshirani, 1993; Quenouille, 1949; Tukey, 1958). Put simply, the estimated parameter is then computed in all subsamples and the variation of the estimates across the subsamples can be used to obtain an estimate of the standard errors (or

confidence intervals) for the respective parameter. In a classic „*N* - 1"-Jackknife, the subsamples are generated by removing one of the observations from the sample. That is, for an original sample of size *N*, there are *N* subsamples (of size *N* – 1) in which one specific observation is removed from the dataset. Jackknifing can be conceived as a computationally more efficient approximation of bootstrapping which would involve drawing many more subsamples but with replacement (Efron, 1979; Efron & Hastie, 2016; Rodgers, 1999). Because any resampling-based procedure involves the re-computation of the PCA including the rotation, bootstrapping would be too computationally expansive with many more resamples to be drawn. If the computational resources were available, however, bootstrapping would be a valid alternative.

Here, we are interested in obtaining Jackknife standard errors of latency estimates, in order to conduct a test for group differences in the latency. The estimated latencies for the factors depend on the loading estimates from temporal PCA. That is, we need to estimate the PCA model for each of the *N* subsamples – including the factor rotation with multiple random starts. Importantly, *N* is the number of participants (not electrodes and conditions). Data from both conditions and all electrode sites are essentially repeated measures and it is important to resample according to the sampling scheme that created the original sample – that is, in a participant-wise manner – to account for dependencies between repeated measures. Then, we can use the loading estimates from the subsamples to test for latency differences using the procedure described for peak-based analyses (Kiesel et al., 2008). Because this step is computationally very expensive and takes several hours to complete, we separated the estimation of the PCA model in the subsamples from their application for the test. The script *05a_jackknifing_step_1.R* is solely concerned with generating the subsamples and obtaining the rotated PCA model – essentially the same way described above for the original sample. To account for the whole uncertainty underlying the PCA model estimation, one would also need to repeat the step of determining the number of factors in each subsample. This type of

modelling uncertainty is usually not accounted for by standard errors in PCA (Zhang, 2014). However, changing the number of factors – possibly underextracting in some subsamples – carries the risk of changing the subsample solutions in unforeseeable ways and would require manual inspection and verification of the solution in each subsample (e.g., by also investigating the topographies). Therefore, we decided to keep the number of factors to be extracted constant across all subsamples.

```
# Load participant average data
load("erpdata.Rdata")

# Group labels to iterate through
groups <- c("ad", "ch")
# How many factors should be extracted per group?
# This should match the number of factors from the PCA over all participants.
nFac <- c(23,21)

# The following code essentially replicates the estimation of the PCA model
# in step 2 for every Jackknife subsample.
# We only increased the number of maximum iterations for the rotation as well
# as the number of random starts to prevent suboptimal results due to
# local optima or non-convergence in single samples.
[...]
```

After obtaining the rotated PCA model for all subsamples, we proceed in the script *05b_jackknifing_step_2.R.* In short, first, we go through all subsamples and extract the rotated loadings. Second, as described for the matching of factors between groups, the factors are matched by time course similarity (i.e., correlation of the loadings with the loadings of the original sample PCA). The solutions from the subsamples should be considerably more comparable than solutions from different groups – therefore, we only check for temporal similarity. Then, we compute a latency estimate by determining at which sampling point the loadings exceed 80% of the peak loading (Kiesel et al., 2008) and convert this from sampling points to the unit seconds. Finally, we compute a so-called *pseudo-value*[2] for the individual

[2] Please note that there are two equivalent ways to think about the Jackknife: First, as in bootstrapping, the statistic of interest can be computed for all the subsamples but than an adjustment is needed to account for the reduced variability (because Jackknife resamples are much more similar than bootstrapped resamples) – this is demonstrated by Kiesel et al. (2008). Second, one can compute pseudo-values which inherently make this adjustment as is proposed by Smulders (2010). It is important to note that both approaches yield equivalent

latency of the participant excluded in the specific subsample (Smulders, 2010) which can be used for statistical inference in the final step. In addition, we save a diagnostic plot which enables us to check the consistency of the loading estimates across subsamples.

```
# The following code loads all jackknifeFits and extracts individual
# latency estimates for the requested components
# In addition, a diagnostic plot of the subsample loadings is provided
# so that problems with factor alignment or stability of the factor solution
can be detected
indLatEst <- lapply(jackknifeFits, FUN = function(iFit){
...
  # apply convenience function to extract loadings and peak latencies
  out <- evalIndFits(rotFit, rotFitAll, iFacIdx, crit)
  # convert indices to seconds
  lat <- (out["latIdx",] - 1) / srate + xmin
  # estimate individual latencies; Smulders, 2010, eq. (1)
  indLat <- N * mean(lat) - (N - 1) * lat
...
})
```

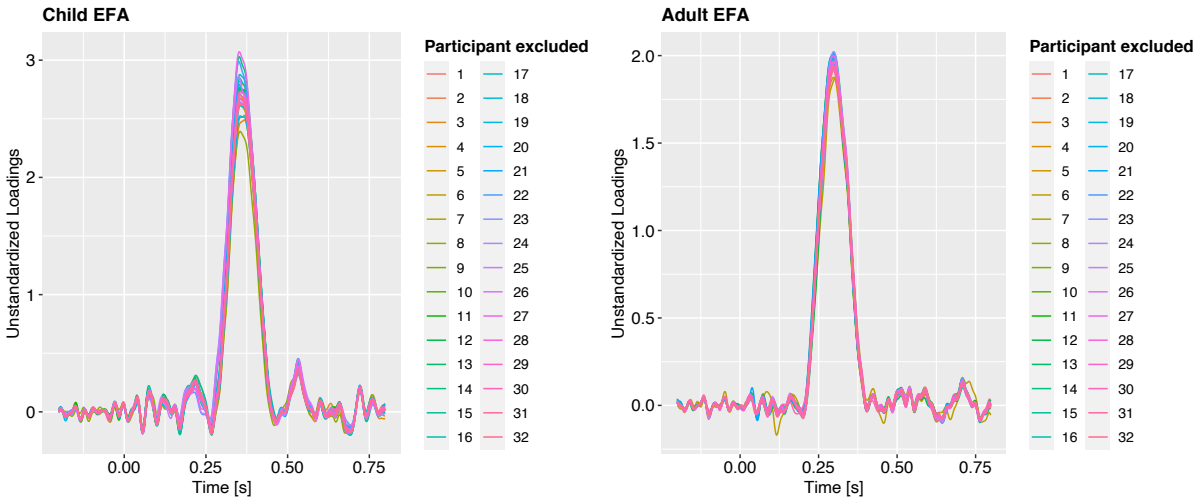
We illustrate the analysis of latency differences for the late P3a. Figure A1 depicts the estimated loadings from the subsamples and reveals remarkable consistency across all subsamples. In case this step reveals large variability of the solutions across samples, one should first increase the number of random starts for the rotation and re-run the Jackknife resampling. If the problem persists, one must inspect the rotated solutions of the subsamples more closely. For instance, if there are several factors with strong temporal overlap, the automatic matching of the factors could fail and mistake one factor for the other in some subsamples (e.g., the early P3a might be mismatched with the late P3a). When this occurs, the matching would need to be rectified manually (e.g., by matching the factors manually). Although we did not encounter a specific example for this problem so far, we hope that this diagnostic plot helps the reader to detect such problems. For the late P3a in our dataset, the plot revealed very consistent loading estimates across subsamples. Therefore, we proceeded to

conclusions. Further details are beyond the scope of this paper but are also documented in the more general statistical literature (e.g., R. G. Miller, 1974; Zhou et al., 2011; Appendix B).

submit the pseudo-values from both groups to a *t*-test for independent samples and we found a significantly prolonged latency of the late P3a in children (0.334 s) compared to adults (0.270 s), $t(58.08) = -4.83$, $p < .001$.

Figure A1

Loading estimates across all Jackknife subsamples for child PCA (left panel) and adult PCA (right panel)



Note. These plots contain the factor loadings of the late P3a factor for all subsamples. The line colors indicate which participant was left out in the specific subsample. One can clearly see that matching by temporal similarity worked extremely well and that the solutions are very consistent across subsamples. If certain subsamples yield a noticeably different loading pattern, they should be inspected to see if the factor was simply mismatched based on temporal similarity alone (which can be corrected manually) or if the different loadings reflect genuine differences in the factor solutions.

```
############### Show diagnostic plots ###############
# Ideally, the individual loadings should be highly similar.
# Any indication that the factor structure varies drastically between
# subsamples should be inspected closely!
indLatEst[[1]]$diagPlot # adult PCA
indLatEst[[2]]$diagPlot # child PCA
###############

############### Subject latencies to a test ###############
# frequentist and Bayesian t tests
t.test(indLatEst[[1]]$indLat, indLatEst[[2]]$indLat)
ttestBF(indLatEst[[1]]$indLat, indLatEst[[2]]$indLat)
###############
```