

Keeping an Eye on Hidden Markov Models in Gaze Data Classification

Malte Lueken<sup>1</sup>, Simon Kucharsky<sup>1</sup>, & Ingmar Visser<sup>1</sup>

<sup>1</sup> Department of Psychology, University of Amsterdam. Amsterdam, The Netherlands

Author Note

This is an author note.

Correspondence concerning this article should be addressed to Malte Lueken, Postal address. E-mail: malte\_lueken@arcor.de

## Abstract

Eye-tracking allows researchers to infer cognitive processes from eye movements that are classified into distinct events. Parsing the events is typically done by algorithms that transform, filter, classify, and merge raw data samples. Previous algorithms have successfully used hidden Markov models (HMMs) for classification but still inhere weaknesses. Therefore, I developed gazeHMM, an HMM algorithm that has no critical parameters to be set by users, does not require human coded data as input, and classifies fixations, saccades, PSOs, and smooth pursuits. The development was guided by the question of whether HMMs are useful at describing eye movements and whether they improve the event classification. I evaluated gazeHMM in a simulation study and on benchmark data. The simulation study showed that gazeHMM successfully recovered HMM parameters and hidden state sequences. Critical exceptions for good recovery were adding a smooth pursuit like state to gazeHMM and noisy data. For benchmark data, model comparisons with gazeHMM yielded preferred models with more states than expected. I assessed the classification performance of gazeHMM compared to other algorithms by the agreement to human event coding. Here, gazeHMM improved the event classification but did not outperform all other algorithms in most cases. Both the simulation study and benchmark application showed poor classification performance for smooth pursuits. Thus, I advice to classify smooth pursuits only for exploration and recommend gazeHMM with fixations, saccades, and PSOs for application. Future HMM algorithms could use covariates to better capture eye movement processes and explicitly model event durations to improve the classification of smooth pursuits.

*Keywords:* eye movements; eye-tracking; parameter recovery; dependent mixture models

Word count: 12962

## Keeping an Eye on Hidden Markov Models in Gaze Data Classification

### Introduction

### Simulation Study

To assess how well the HMM recovers parameters and state sequences, we conducted a simulation study. The design and analysis of the study were preregistered on the Open Science Framework (<https://doi.org/10.17605/OSF.IO/VDJGP>). This section is majorly copied from the preregistration (with adapted tenses). The study was divided in four parts. Here, we only report the first two parts, the other two parts, which address starting values and missing data, can be found in the supplementary material (URL).

The HMM repeatedly generated data with a set of parameters (true parameter values). The same model was applied to estimate the parameters from the generated data (estimated parameter values). We compared the true with the estimated parameter values to assess whether a parameter was recovered by the model. Additionally, we contrasted the true states of the HMM with the estimated states to judge how accurately the model recovered the states that generated the data.

### Starting Values

The HMM always started with a uniform distribution to estimate the initial state and state transition probabilities. Random starting values for the estimation of shape, scale, and concentration parameters were generated by gamma distributions with a shape parameter of  $\alpha_{start} = 3$  and  $\beta_{start} = \psi_{true}/2$ , with  $\psi_{true}$  being the true value of the parameter to be estimated. This setup ensured that the starting values were positive, their distributions were moderately skewed, and the modes of their distributions equaled the true parameter values. Mean parameters of the von Mises distribution always started at their true values.

## Design

In the first part, we varied the parameters of the HMM. For models with  $k \in \{2, 3, 4\}$  states,  $q \in \{10, 15, 20\}$  parameters were manipulated, respectively. For each parameter, the HMM generated 100 data sets with  $N = 2500$  samples and the parameter varied in a specified interval in equidistant steps. This resulted in  $100 \times (10 + 15 + 20) = 4500$  recoveries. Only one parameter alternated at once, the other parameters were set to their default values. All parameters of the HMM were estimated freely (i.e., there were no fixed parameters in the model). We did not manipulate the initial state probabilities because these are usually irrelevant in the context of eye movement classification. For the transition probabilities, we only simultaneously changed the probabilities for staying in the same state (diagonals of the transition matrix) to reduce the complexity of the simulation. The left over probability mass was split evenly between the probabilities for switching to a different state (per row of the transition matrix). Moreover, we did not modify the mean parameters of the von Mises distributions: As location parameters, they do not alter the shape of the distribution and they are necessary features for the HMM to distinguish between different states.

We defined approximate ranges for each response variable (see supplementary material) and chose true parameter intervals and default values so that they produced samples that roughly corresponded to these ranges. Tables 1 and 2 show the intervals and default values for each parameter in the simulation. Parameters were scaled down by factor 10 (compared to the reported ranges) to improve fitting of the gamma distributions. We set the intervals for shape parameters of the gamma distributions for all events to  $[1, 5]$  to examine how skewness influenced the recovery (shape values above five approach a symmetric distribution). The scale parameters were set so that the respective distribution approximately matched the assumed ranges. Since the concentration parameters of the von Mises distribution are the inverse of standard deviations, they were varied on the inverse scale. An example of simulated data from the HMM with default parameters is visualized

in Figure ??.

In the second part, we manipulated the sample size of the generated data and the amount of noise added to it. The model parameters were set to their default values. For models with  $k \in \{2, 3, 4\}$  states and sample sizes of  $N \in \{500, 2500, 10000\}$ , we generated 100 data sets ( $100 \times 3 \times 3 = 900$  recoveries). These samples sizes roughly corresponded to small, medium, and large eye-tracking data sets for a single participant and trial. To simulate noise, we replaced velocity and acceleration values  $y$  with draws from a gamma distribution with  $\alpha_{noise} = 3$  and  $\beta_{noise} = (y/2)\tau_{noise}$  with  $\tau_{noise} \in [1, 5]$  varying between data sets. This procedure ensured that velocity and acceleration values remained positive and were taken from moderately skewed distributions with modes equal to the original values. To angle, we added white noise from a von Mises distribution with  $\mu_{noise} = 0$  and  $\kappa_{noise} \in 1/[0.1, 10]$  varying between data sets.  $\tau_{noise}$  and  $\kappa_{noise}$  were increased simultaneously in equidistant steps in their intervals.

## Data Analysis

For each parameter, we calculated the root median square proportion deviation (RMdSPD; analogous to root median square percentage errors, see Hyndman & Koehler, 2006) between the true and estimated parameter values:

$$\text{RMdSPD} = \sqrt{\text{Med} \left( \left( \frac{\psi_{true} - \psi_{est}}{\psi_{true}} \right)^2 \right)}.$$

Even though it was not explicitly mentioned in the preregistration, this measure is only appropriate when  $\psi_{true} \neq 0$ . This was not the case for some mean parameters of the von Mises distributions. In those cases, we used  $\psi_{true} = 2\pi$  instead. We treated  $\text{RMdSPD} < 0.1$  as good,  $0.1 \leq \text{RMdSPD} < 0.5$  as moderate, and  $\text{RMdSPD} \geq 0.5$  as bad recovery of a parameter. By taking the median, we reduced the influence of potential outliers in the estimation and using proportions enabled us to compare RMdSPD values across parameters and data sets.

Table 1

*Intervals and Default Parameter Values for the Transition Model in the Simulation*

	Initial state prob.	Trans. prob. same state	Trans. prob. other state
Interval	-	[.01,.99]	$1-a(i=j)/(k-1)$
Default	1/k	0.9	$0.1/(k-1)$

*Note.* The transition probability for staying in the same state is denoted by  $a_{i=j}$  and the probability for switching to a different state by  $a_{i \neq j}$ . The number of states in the model is denoted by  $k$ .

Additionally, we applied a bivariate linear regression with the estimated parameter values as the dependent and the true parameter values as the independent variable to each parameter that has been varied on an interval in part one. Regression slopes closer to one indicated that the model better captured parameter change. Regression intercepts different from zero reflected a bias in parameter estimation.

To assess state recovery, we computed Cohen’s kappa (for all events taken together, not for each event separately) as a measure of agreement between true and estimated states for each generated data set. Cohen’s kappa estimates the agreement between two classifiers accounting for the agreement due to chance. Higher kappa values were interpreted as better model accuracy. We adopted the ranges proposed by Landis and Koch (1977) to interpret kappa values. Models that could not be fitted were excluded from the recovery.

## Results

**Parameter Variation.** In the first part of the simulation, I examined how varying the parameters in the HMM affects the deviation of estimated parameters and accuracy of estimated state sequences. Figure 1 displays the RMdSPD between true and estimated parameters depending on which parameter has been manipulated in the HMM. The

Table 2

*Intervals and Default Parameter Values for the Response Model in the Simulation*

	Velocity		Acceleration		Rel. angle	
	Shape	Scale	Shape	Scale	Mean	Concentration
State 1						
Interval	[1,5]	[0.1,0.6]	[1,5]	[0.05,0.25]	-	-
Default	3	0.35	3	0.25	-	-
State 2						
Interval	[1,5]	[5,15]	[1,5]	[1,5]	-	$1/[0.1,10]$
Default	3	10	3	3	0	1
State 3						
Interval	[1,5]	[0.5,1.5]	[1,5]	[1,5]	-	$1/[0.1,10]$
Default	3	1	3	3	pi	1
State 4						
Interval	[1,5]	[0.5,1.5]	[1,5]	[0.05,0.25]	-	$1/[0.1,10]$
Default	3	1	3	0.15	0	1

*Note.* Shape parameters are denoted by  $\alpha$ , scale parameters by  $\beta$ , mean parameters by  $\mu$ , and concentration parameters by  $\kappa$ . The default values for the uniform distribution in state one were  $\min = 0$  and  $\max = 2\pi$ .

RMdSPD was below 0.1 for all estimated and manipulated parameters, indicating good recovery.<sup>1</sup> The regressions between manipulated true and estimated parameters are shown in Figures 2 and 3. With one outlier at parameter  $\alpha_{acc;1}$ , the estimated parameters matched the true parameters very closely. Thus, parameter change was captured well and the estimation almost unbiased. Considering accuracy, Figure 4 displays Cohen’s kappa between true and estimated hidden state sequences. With two exceptions, kappa values were almost one, suggesting nearly perfect agreement.

For the HMM with three states, the RMdSPD is shown in Figure 5. When response parameters (other than  $a_{i=j}$ ) were manipulated, the RMdSPDs for  $a_{12}$  and  $a_{31}$  were consistently between 0.1 and 0.5. Varying  $\kappa$  in states two and three led to RMdSPDs between 0.1 and 0.5 in the respective states, which we interpreted as moderate recovery. Otherwise, RMdSPDs were consistently lower than 0.1, indicating good recovery. Inspecting the regressions between manipulated true and estimated parameters (see Figures 6 and 7) revealed strong and unbiased linear relationships (intercepts close to zero and slopes close to one). In contrast to the two-state HMM, larger deviations and more outliers were observed. Cohen’s kappa values are presented in Figure 8. For most estimated models, the kappa values between true and estimated state sequences were above 0.95, meaning almost perfect agreement. However, for some models, we observed kappas clustered around zero or -0.33, which suggests that state labels were switched.

The RMdSPDs for the four-state HMM is shown in Figure 9. For estimated transition probabilities and  $\alpha_{vel}$  and  $\beta_{vel}$  parameters in states one and four, RMdSPDs were between 0.1 and 0.5, suggesting moderate recovery. Estimated kappa parameters in state four were also often moderately recovered when parameters in states two, three, and four were varied. Otherwise, RMdSPDs were below 0.1, indicating good recovery. Looking

---

<sup>1</sup> Note that the initial state probability  $\rho_i$  has RMdSPD = 1. Since the HMM only simulated one state sequence, this parameter is always either zero or one (leading to RMdSPD = 1). Therefore, I decided not to include it in the analysis.



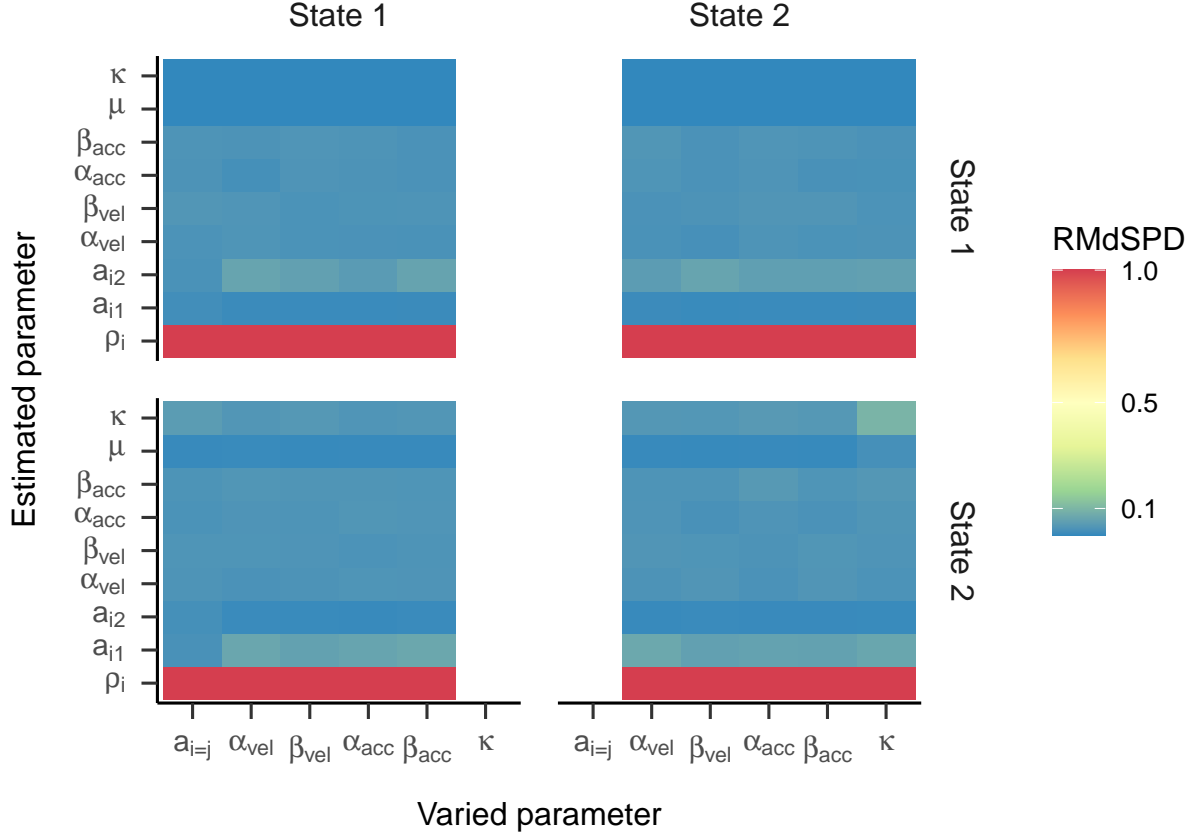


Figure 1. RMdSPD between true and estimated parameters of the two-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong.

at the regressions between true and estimated parameters, Figures 10 and 11 illustrate strong and unbiased relationships. However, there were larger deviations and more outliers than in the previous models, especially for states one and four. Cohen’s kappa ranged majorly between 0.6 and 0.9, meaning moderate to almost perfect agreement between true and estimated state sequences (see Figure 12). Here, some kappa values clustered around 0.25 and zero, which, again, can be interpreted as the result of label switching.

Overall, simulations in part one demonstrated that the HMM recovered parameters

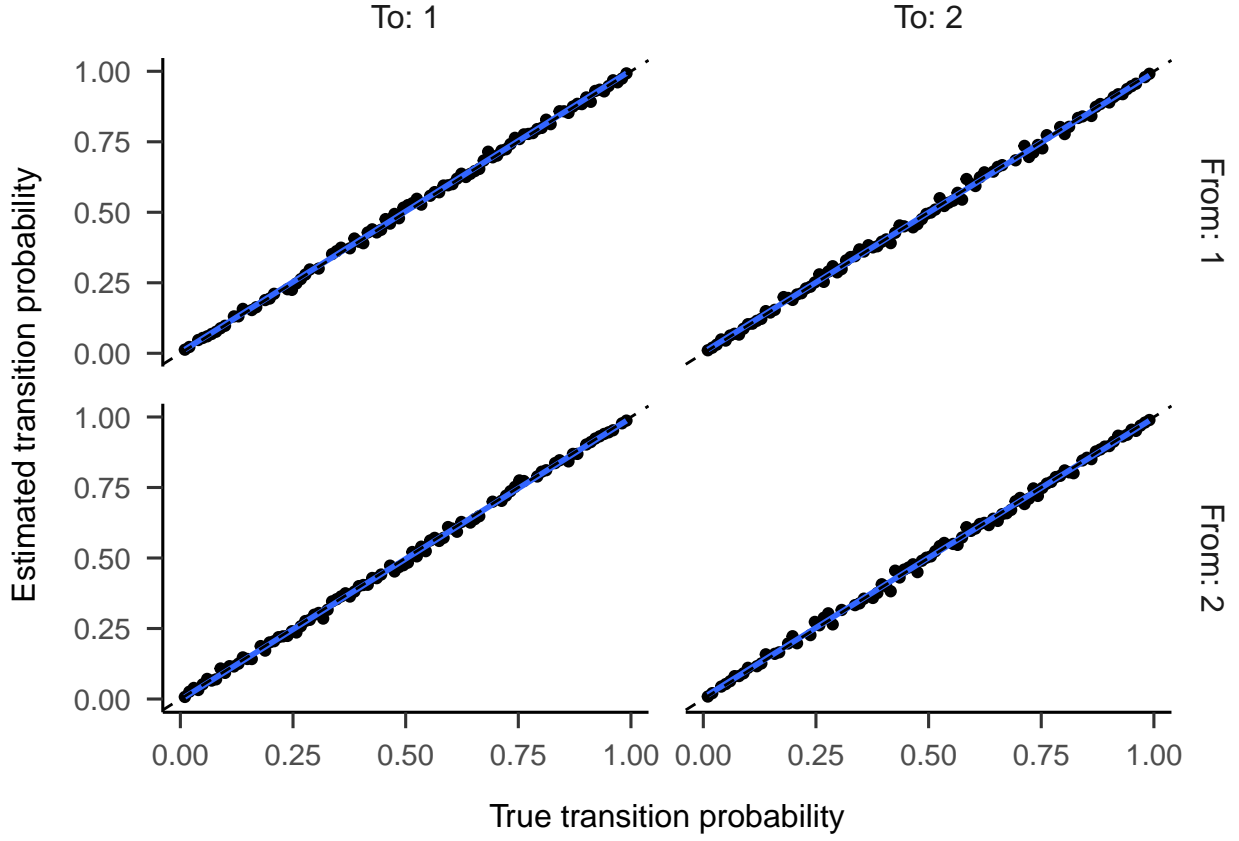


Figure 2. Regression lines between true and estimated transition probabilities for the two-state HMM in part one. Top facet labels show to and right facet labels show from which state the HMM is moving. Dashed lines refer to perfect recovery.

very well when they were manipulated. Deviations from true parameters were mostly small. In the four-state model, estimated transition probabilities for state one and four deviated moderately. Moreover, the HMM estimated state sequences very accurately with decreasing accuracy for the four-state model.

**Sample Size and Noise Variation.** In the second part, we varied the sample size of the HMM and added noise to the generated data. For the two-state HMM, the RMdSPDs were above 0.5 for  $\beta_{vel}$  and  $\beta_{acc}$  in both states (see Figure 13), suggesting bad recovery. The other estimated parameters showed RMdSPDs close to or below 0.1, which means they were recovered well. Increasing the sample size seemed to improve RMdSPDs

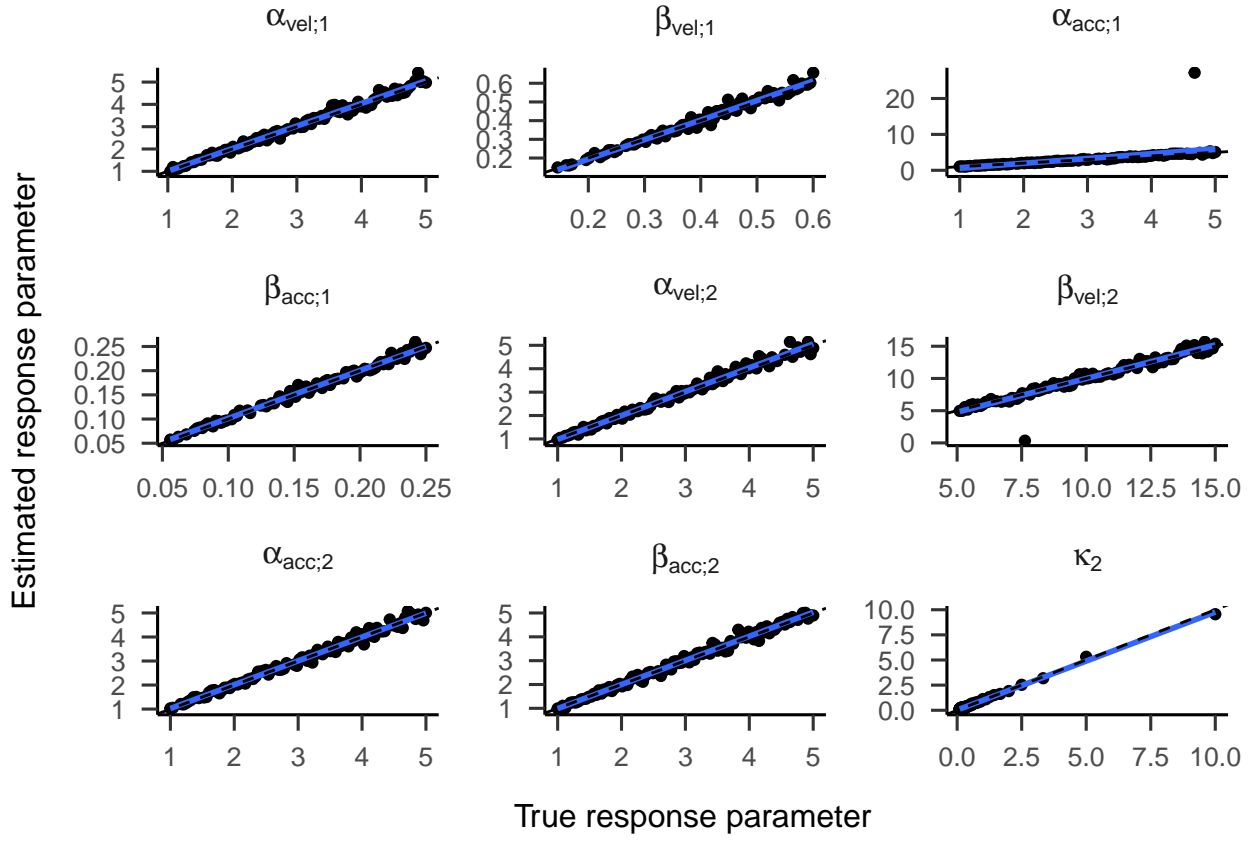


Figure 3. Regression lines between true and estimated response parameters of the two-state HMM in part one. Top facet labels indicate response parameters. Dashed lines refer to perfect recovery.

for most parameters slightly. For  $\beta_{vel}$  and  $\beta_{acc}$  in both states, models with 2500 samples had the lowest RMdSPDs. Accuracy measured by Cohen's kappa was almost perfect with kappa values very close to one (see Figure 14, left plot).

Regarding three states, the RMdSPDs for the  $\beta_{vel}$  and  $\beta_{acc}$  were above 0.5 in all three states (see Figure 15), indicating bad recovery. Again, the other estimated parameters were below or close to 0.1, only  $a_{12}$  and  $a_{31}$  with 500 samples were closer to 0.5. For most parameters across all three states, models with higher sample sizes had lower RMdSPDs. The state recovery of the estimated models was almost perfect with most kappa values above 0.95 (see Figure 14, middle plot). Several outliers clustered around kappas of zero

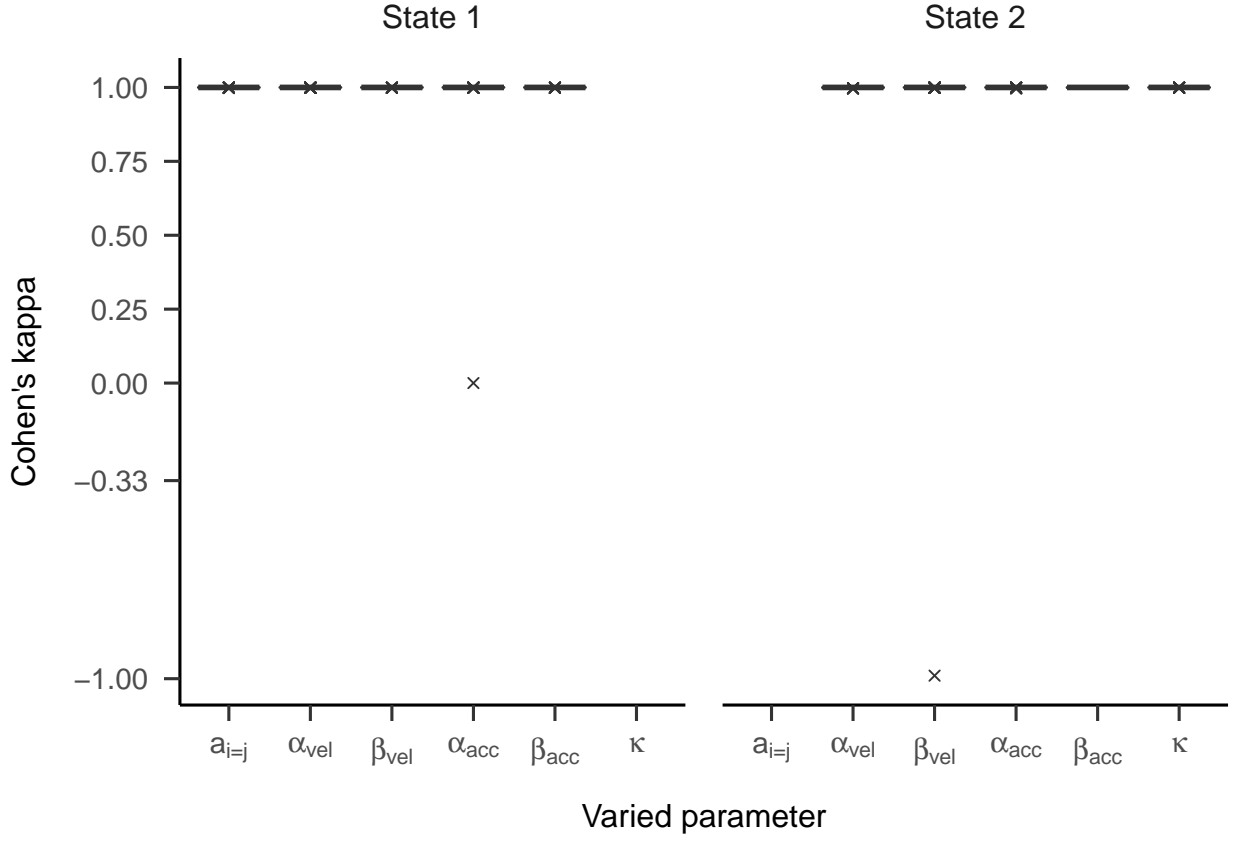


Figure 4. Boxplots displaying Cohen's kappa depending on which parameter of the two-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians. Crosses represent outliers (distance to first/third quartile higher than 1.5 times the inter-quartile range [IQR]).

and -0.33, signaling label switching.

RMdSPDs regarding the four-state HMM are displayed in Figure 16. For states one and four, values for most parameters (including all transition probabilities) were above 0.5, suggesting bad recovery. Similarly,  $\beta_{vel}$  and  $\beta_{acc}$  in states two and three showed bad recovery. For states two and three, higher samples sizes showed slightly lower RMdSPDs. As in the previous part, most Cohen's kappa values ranged between 0.6 and 0.9, meaning substantial to almost perfect agreement between true and estimated states (Figure 14, right plot). Multiple kappa values clustered around 0.25 or zero, which can be explained by

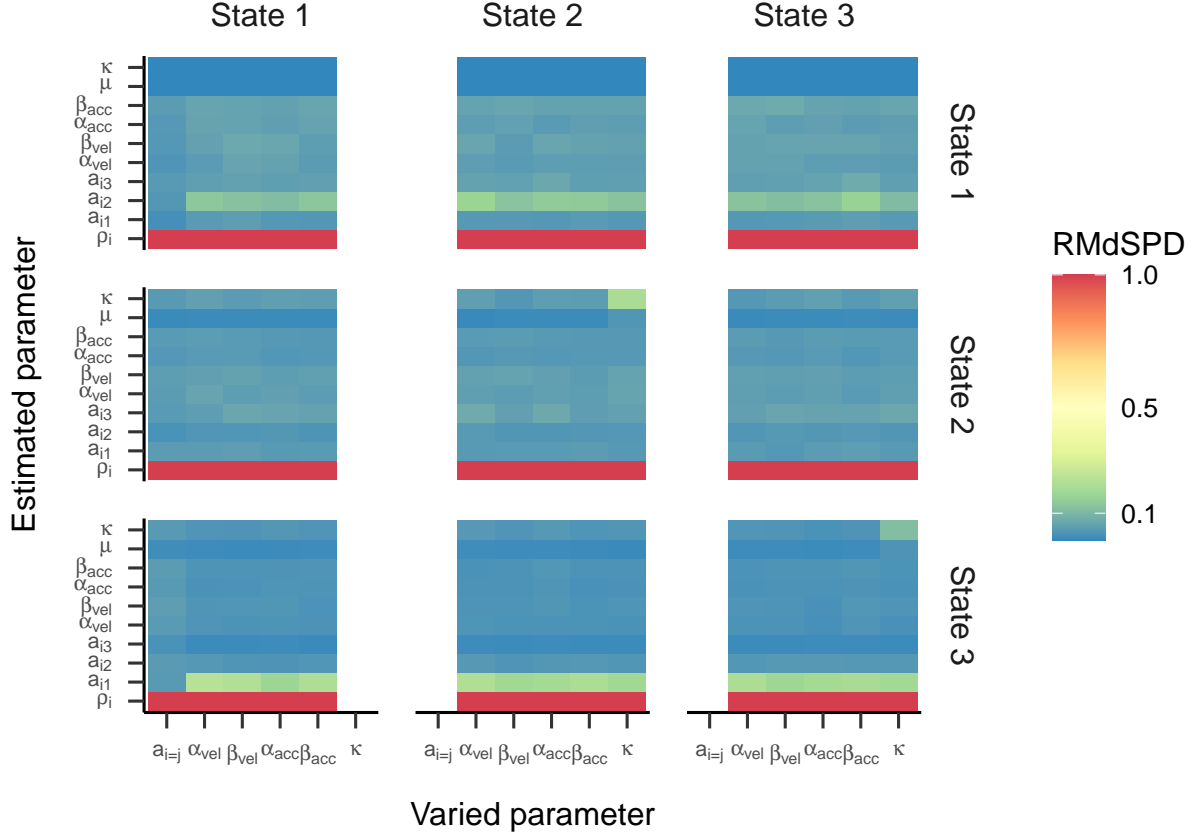


Figure 5. RMdSPD between true and estimated parameters of the three-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong.

label switching.

In general, the HMM recovered parameters well despite noise being added to the data. However, in the four-state model, the parameter recovery for states one and four substantially decreased. In the three- and four-state models, scale parameters of gamma distributions were badly recovered. Increasing the sample size in the HMM slightly improved the recovery of most parameters. The state recovery of the model was slightly lowered when more states were included, but it was neither affected by the noise variability

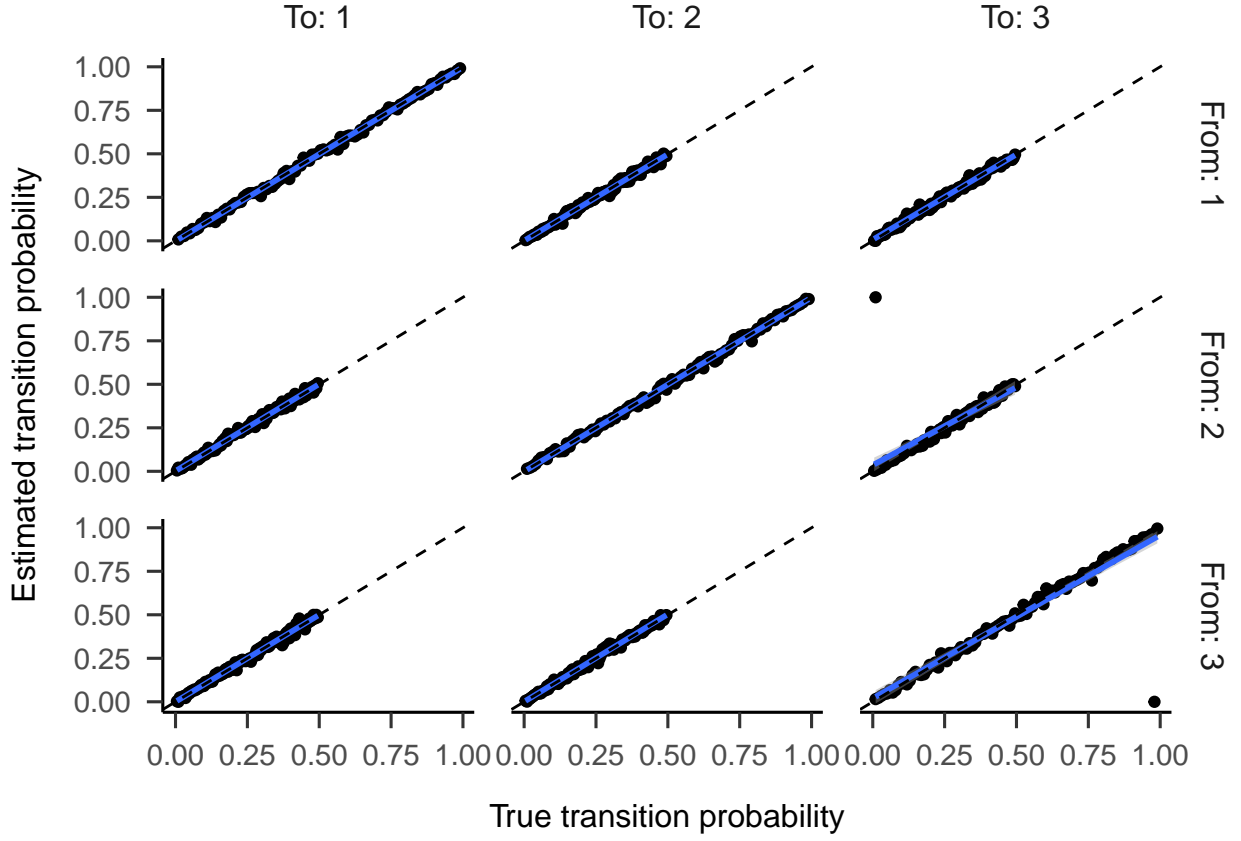


Figure 6. Regression lines between true and estimated transition probabilities for the three-state HMM in part one. Top facet labels show to and right facet labels show from which state the HMM is moving. Dashed lines refer to perfect recovery.

$\tau_{noise}$  nor the sample size.

### Validation Study

To validate gazeHMM, we applied the algorithm on two benchmark data sets. As starting values, we used  $\rho = 1/k$  for the initial state model as well as  $a_{i=j} = 0.9$  and  $a_{i \neq j} = 0.1/k$  for the transition model. The values for the response model are displayed in Table 3. In contrast to the simulation study, generating random starting values often led to bad model fits and label switching between states. To improve the fitting of the gamma

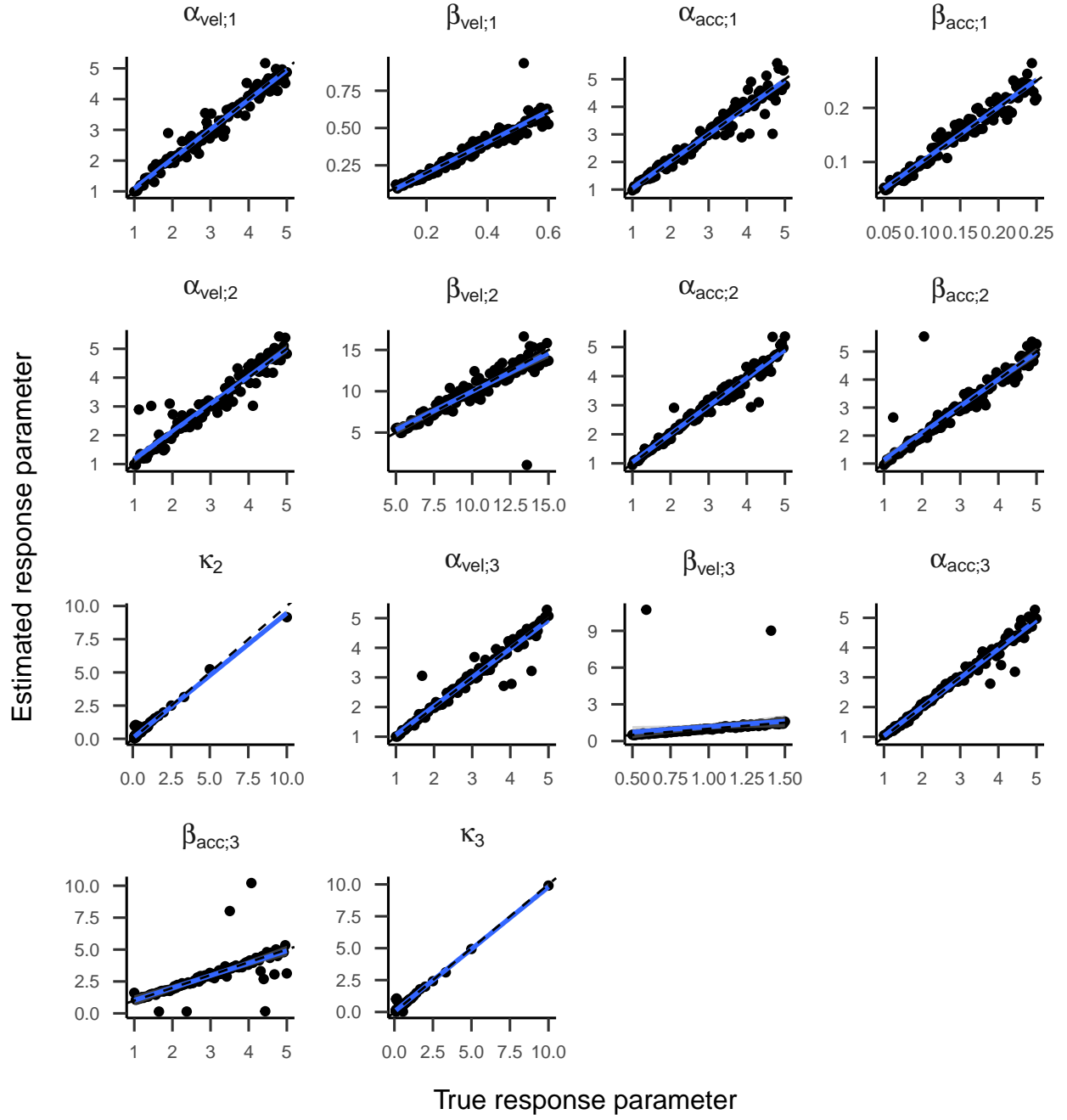


Figure 7. Regression lines between true and estimated response parameters of the three-state HMM in part one. Top facet labels indicate response parameters. Dashed lines refer to perfect recovery.

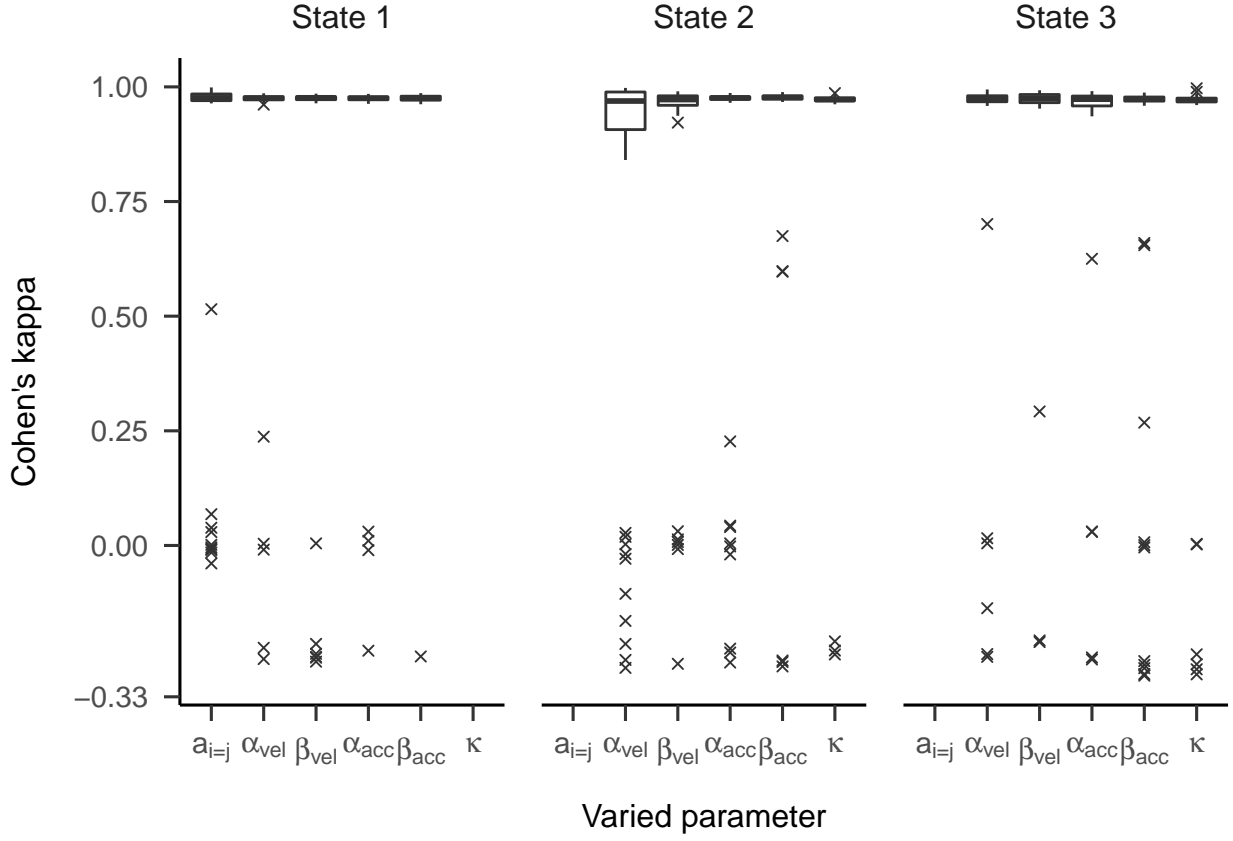


Figure 8. Boxplots displaying Cohen’s kappa depending on which parameter of the three-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

distributions, velocity and acceleration signals were scaled down by factor  $100^2$  (so were the starting values for their gamma distributions).

<sup>2</sup> Scaling down by factor 100 differs from the simulation study (scaling down by 10). The algorithm allows the user to manually specify this factor and in this case, factor 100 led to better model fits than factor 10. Thus, I chose it exploratorily.



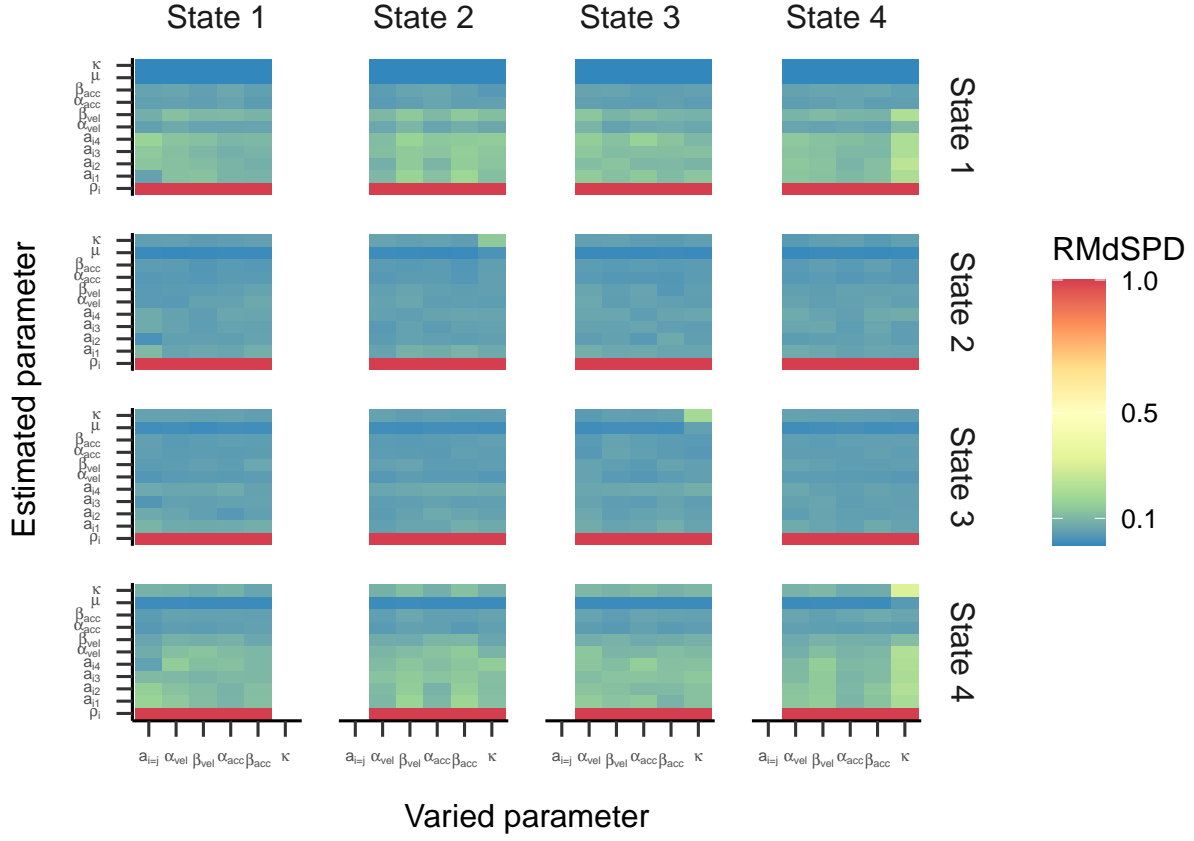


Figure 9. RMdSPD between true and estimated parameters of the four-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong.

## Data Sets

We chose two data sets for validation: One was published in a study by Andersson, Larsson, Holmqvist, Stridh, and Nyström (2017) and has been widely used for validation purposes (e.g., Pekkanen & Lappi, 2017). It contains eye-tracking data from three conditions: A static condition, where subjects had to look freely at images, and two dynamic conditions, where they had to follow a constantly moving dot or objects in a video. The data was sampled with 500 Hz and two human coders (MN and RA) labeled

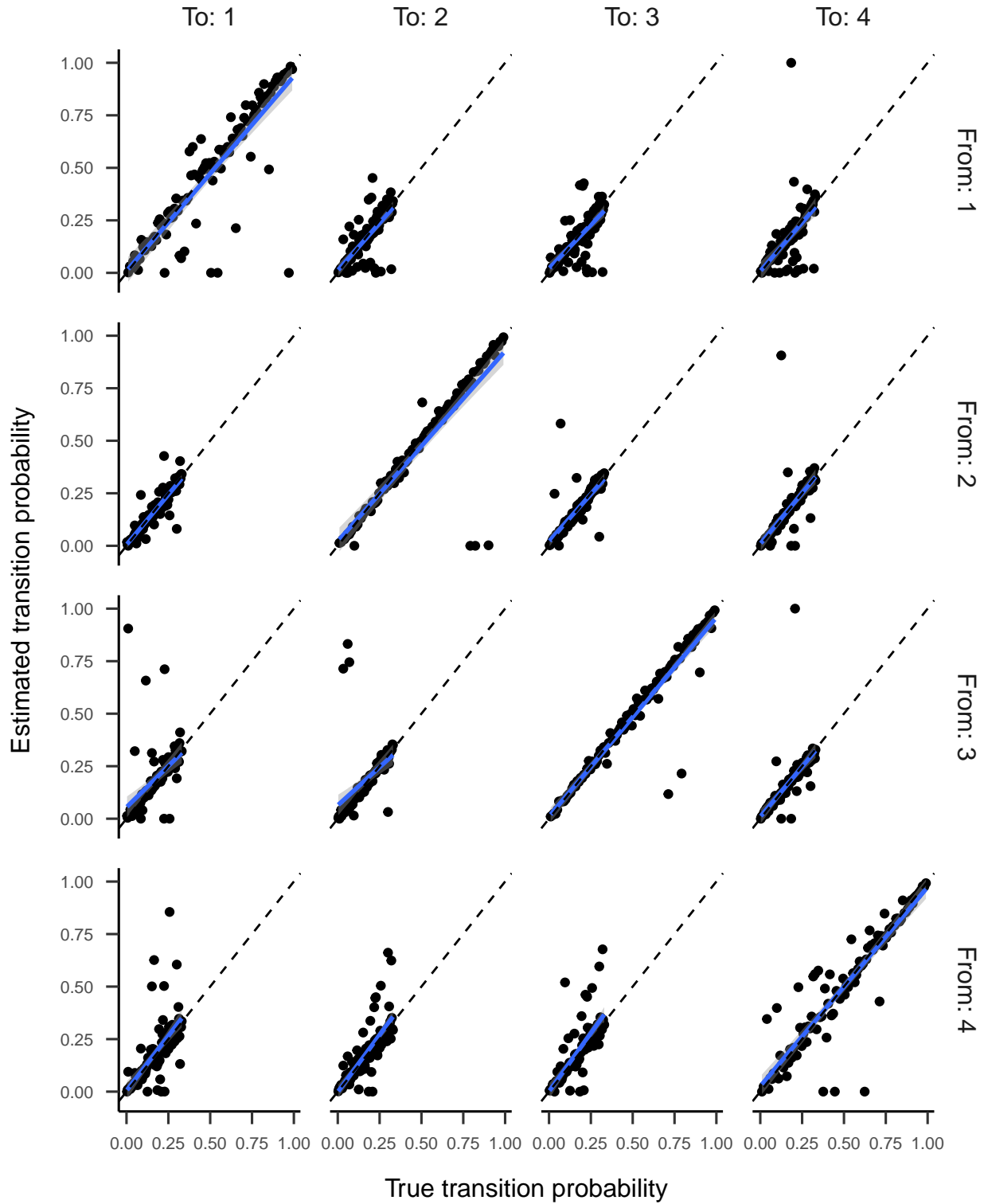


Figure 10. Regression lines between true and estimated transition probabilities for the four-state HMM in part one. Top facet labels show to and right facet labels show from which state the HMM is moving. Dashed lines refer to perfect recovery.

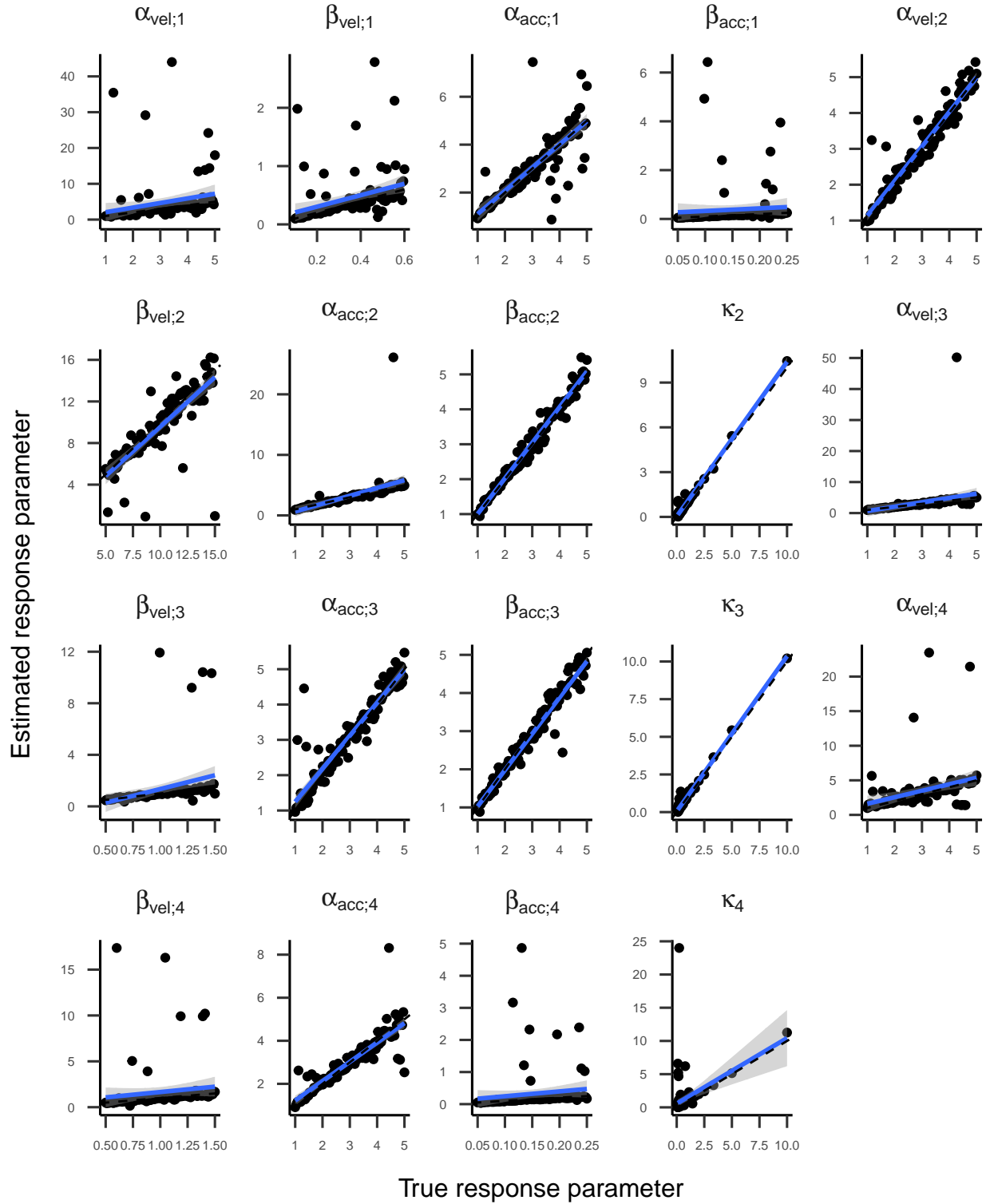


Figure 11. Regression lines between true and estimated response parameters of the four-state HMM in part one. Top facet labels indicate response parameters. Dashed lines refer to perfect recovery.

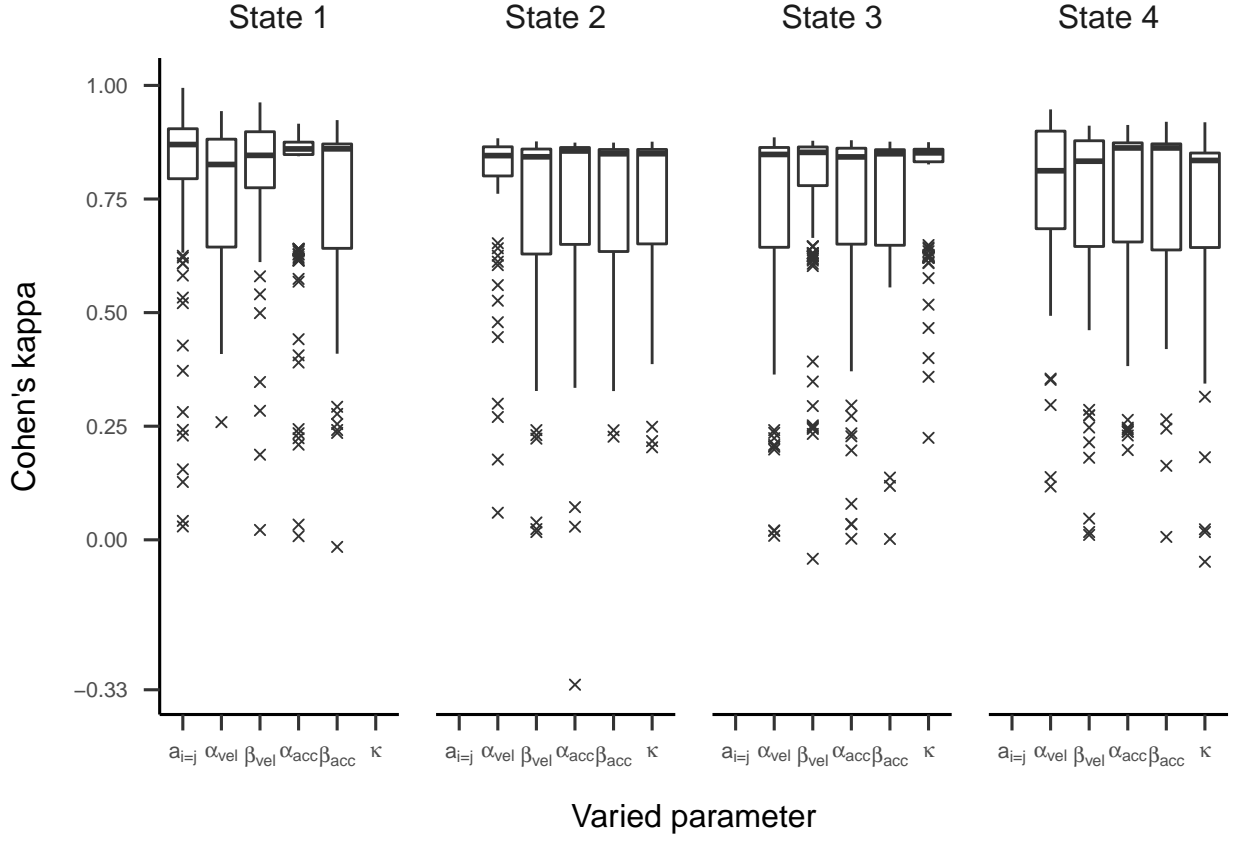


Figure 12. Boxplots displaying Cohen’s kappa depending on which parameter of the four-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

them as belonging to six different eye movement events: Fixation, saccade, PSO, smooth pursuit, blink, or other. Andersson et al. (2017) used the data to compare 10 different classification algorithms. We used their results to compare these 10 algorithms and the two human coders with gazeHMM.

The second data set was published in Ehinger2019 and has to our knowledge not yet been used for validation. Here, we only use tasks four and five out of 10 tasks, because these are qualitatively different to the first data set. In task four, subjects were instructed

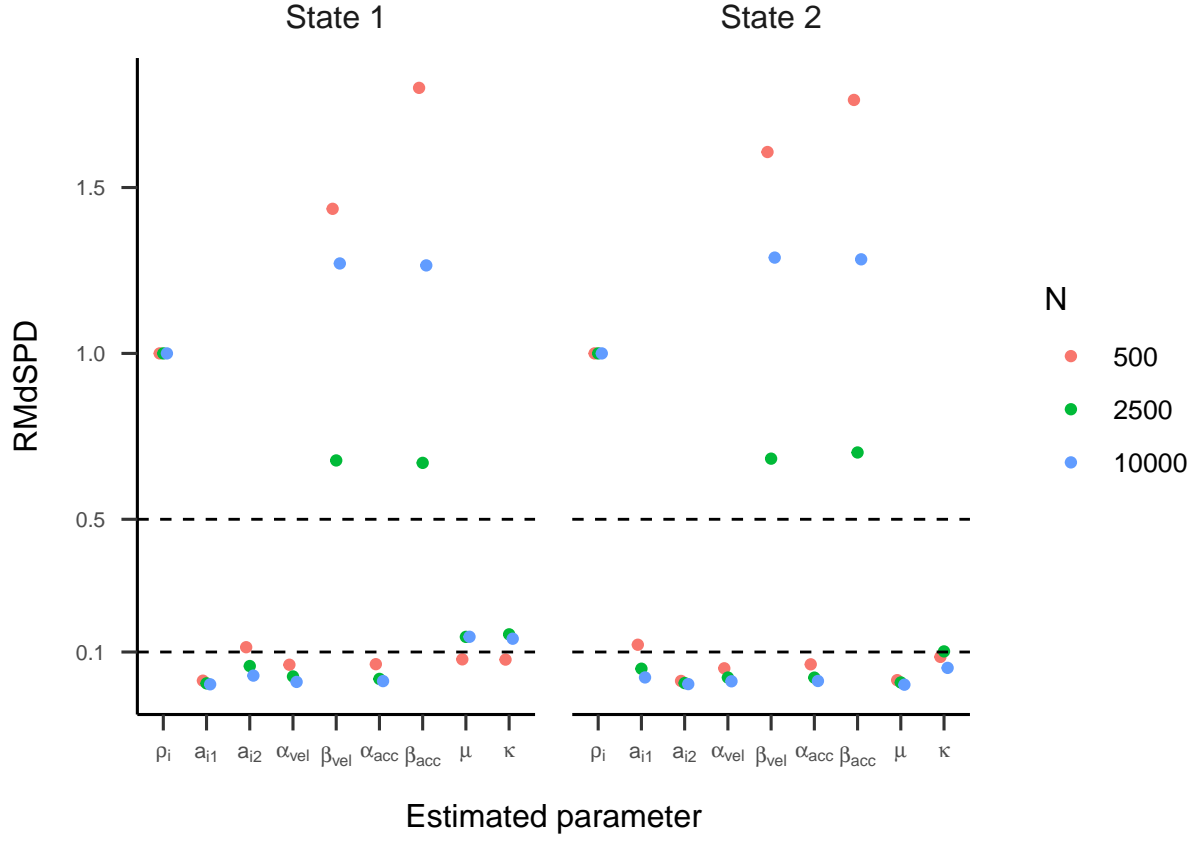


Figure 13. RMdSPD between true and estimated parameters of the two-state HMM in part two of the simulation. Colours indicate different sizes of generated data. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

to fixate a central target for 20 s. Task 5 was set up similarly, but subjects had to blink when they heard one out of seven beeps (with a beep duration of 100 ms and 1.5 s intervals in between). Eye movements were recorded with 250-500 Hz. For simplicity, we only used data obtained by the EyeLink eye-tracker.

**Model Comparison.** To examine whether gazeHMM accurately describes eye movements, I applied gazeHMM with one, two, three, four, and five states on two benchmark data sets and compared the Schwarz weights of the HMMs. The starting values for the fifth state were chosen to represent distributions for microsaccade samples (velocity

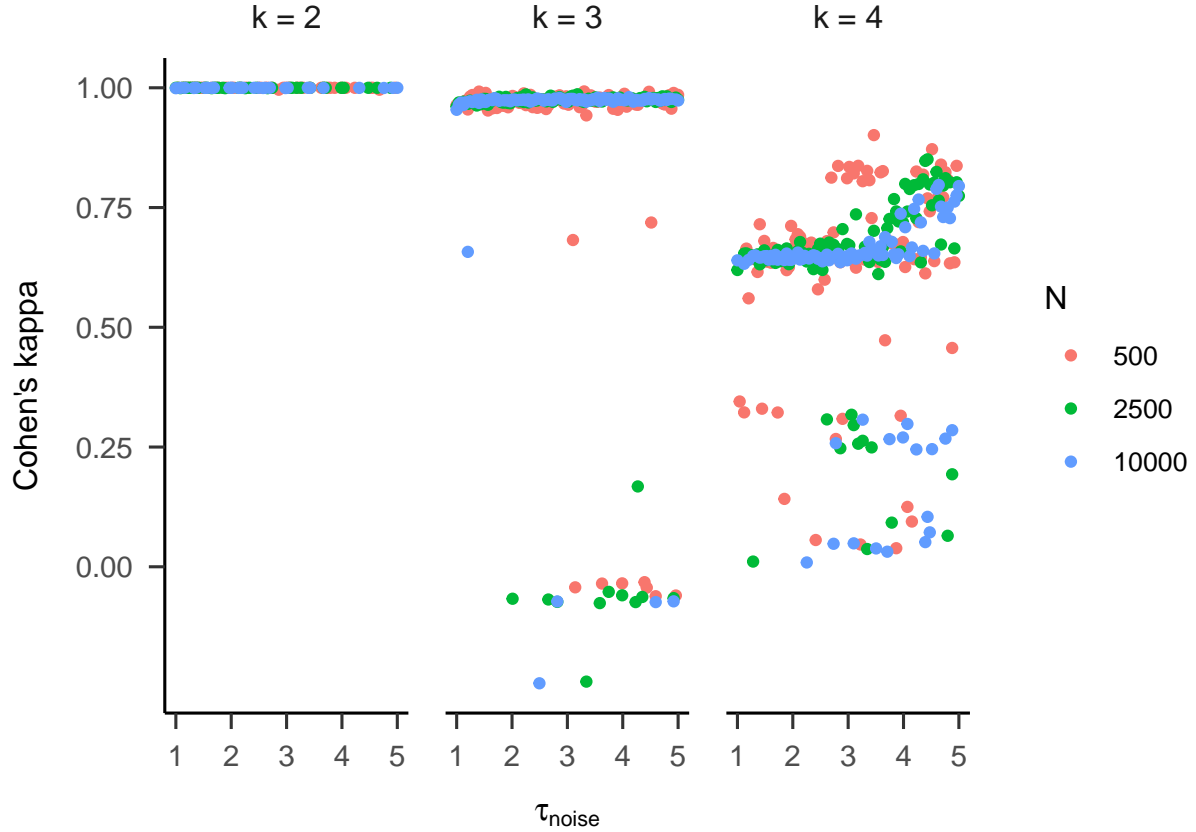


Figure 14. Cohen's kappa depending on the variation of noise added to the data generated by the HMM. Colours indicate different sizes of generated data. Top facet labels indicate the number of states in the HMM.

and sample-to-sample angle as for smooth pursuits, acceleration as for saccades and PSOs). Schwarz weights are the result of transforming a set of BIC values (Wagenmakers & Farrell, 2004). They can be interpreted as the probability for a model to have generated the data it was fitted to. For the Andersson et al. (2017) data set, I expected the highest weights for the three-state models in the image condition. For the moving dots and video conditions, I predicted the highest weights for the four-state model. Figure 17 shows the Schwarz weights for different subjects and models in the image condition. For all subjects, the five-state model displayed the highest weight, suggesting that it most likely generated the data. In the moving dots condition, the five-state model had the highest weights for

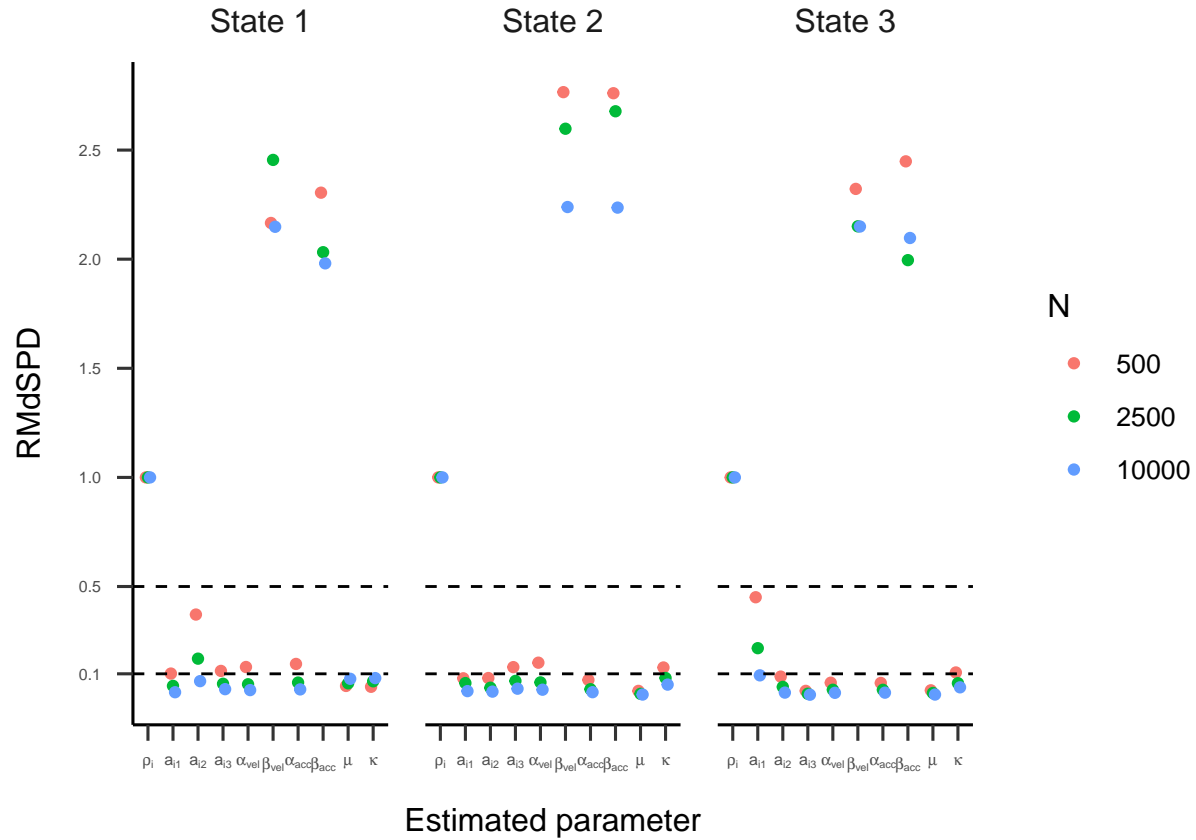


Figure 15. RMdSPD between true and estimated parameters of the three-state HMM in part two of the simulation. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

most subjects, but the one-, three, and four-state models exhibited the highest weights for two subjects each (see Figure 18). The video condition had the same pattern as the image condition, as the five-state model consistently had the highest weights (see Figure 19). In all three conditions, response parameters of the fifth state were mostly similar to those of the smooth pursuit state.

In contrast to my hypothesis, applying different HMMs to the Andersson et al. (2017) data suggests that the five-state model has most likely generated the data. Except for the moving dots condition, there was little variation in the Schwarz weights, indicating large differences in the model likelihoods. For the Ehinger, Groß, Ibs, and Peter (2019)

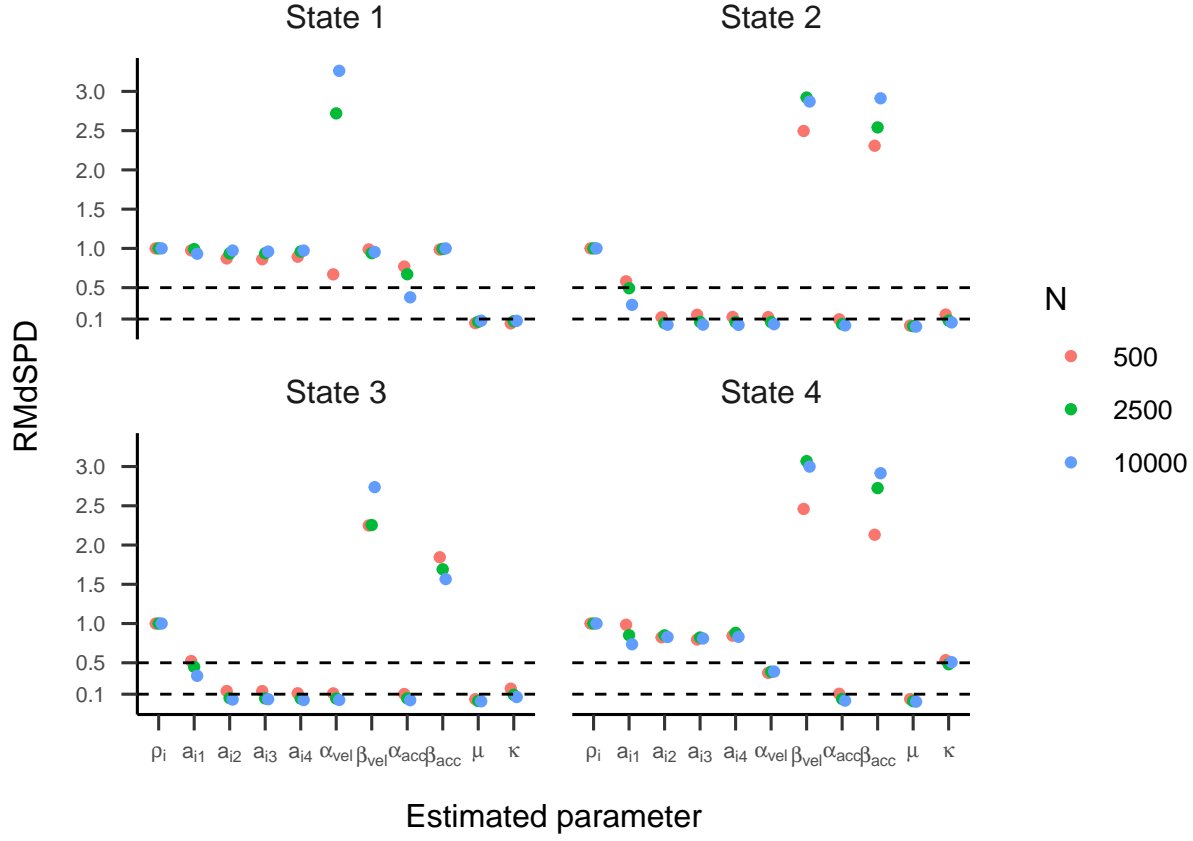


Figure 16. RMdSPD between true and estimated parameters of the three-state HMM in part two of the simulation. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

data, I predicted a one-state model to be preferred. Here, the model comparison for the yielded similar results (see Appendix A), contradicting my hypothesis.

A recent model recovery study showed that the BIC tended to prefer overly complex HMMs when they were misspecified (e.g., the conditional independence assumption was violated; Pohle, Langrock, Beest, & Schmidt, 2017). Instead, the integrated completed likelihood (ICL) criterion (Biernacki, Celeux, & Govaert, 2000) performed better in choosing the correct data generating model. Therefore, I computed the weighted ICL criterion for the HMMs fitted to the Andersson et al. (2017) data set as an exploratory analysis. Using the ICL as the model selection criterion yielded very similar results to the



Table 3

*Starting Values for the Response Model for Fitting gazeHMM to Benchmark Data*

	Velocity		Acceleration		Rel. angle	
	Shape	Scale	Shape	Scale	Mean	Concentration
Fixation	10	10	10	10	-	-
Saccade	50	50	50	50	0	10
PSO	50	50	50	50	3.14	10
Pursuit	20	20	20	20	0	10
Event 5	20	20	50	50	0	10

*Note.* Starting values for velocity and acceleration signals are shown before scaling down by factor 100. Values for event 5 were chosen so that they approximately match plausible distributions for microsaccades.

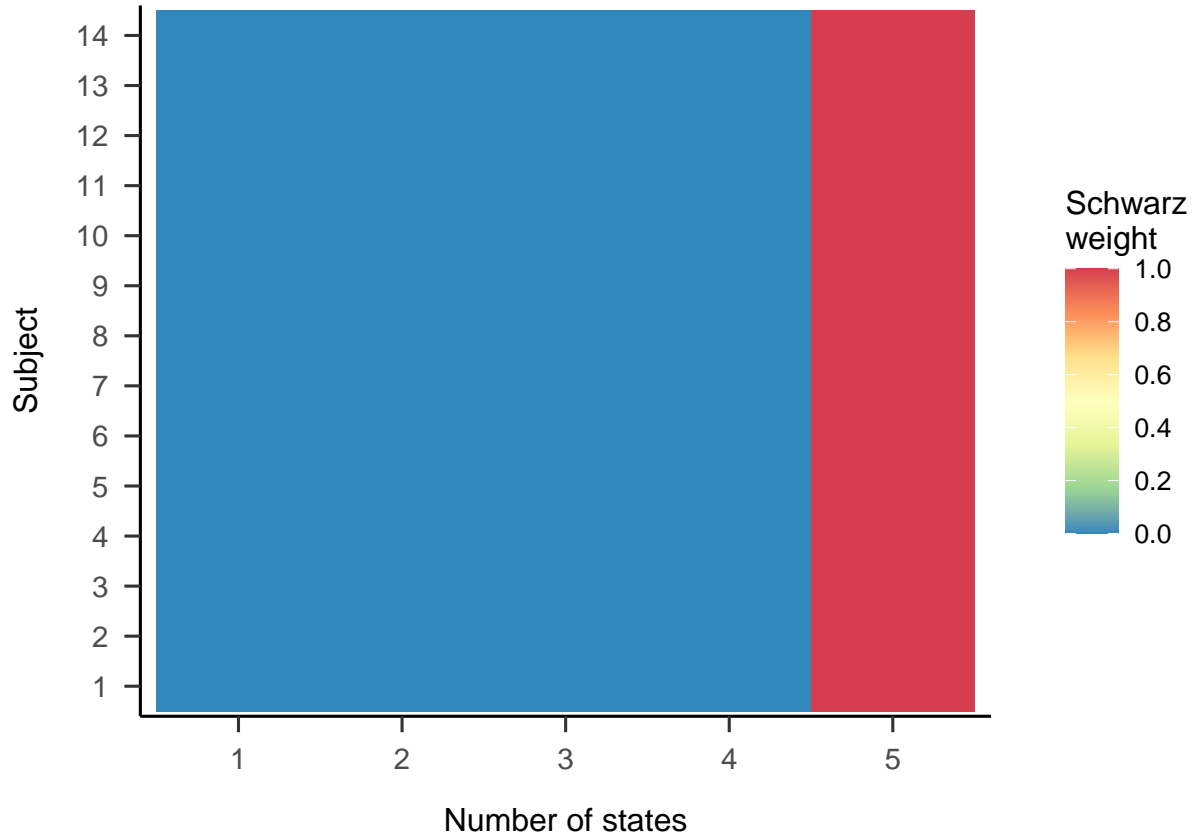
BIC (see Appendix A). The preference for the five-state model was even more consistent across conditions and subjects.

### **Comparison to Other Algorithms.**

#### ***Event Durations.***

The hypothesis that gazeHMM improves the classification performance of eye movement events was investigated by comparing it against other algorithms. Therefore, I applied gazeHMM with four events to the Andersson et al. (2017) data set. I calculated the RMSD<sup>3</sup> of event durations as described in the original article by Andersson et al. for all algorithms included in the study plus gazeHMM: First, the mean and standard deviation of

<sup>3</sup> In my proposal, I declared to compute the RMSE instead of RMSD. However, since I conducted an in-sample comparison and it was used by Andersson et al. (2017), the term *deviation* is more appropriate

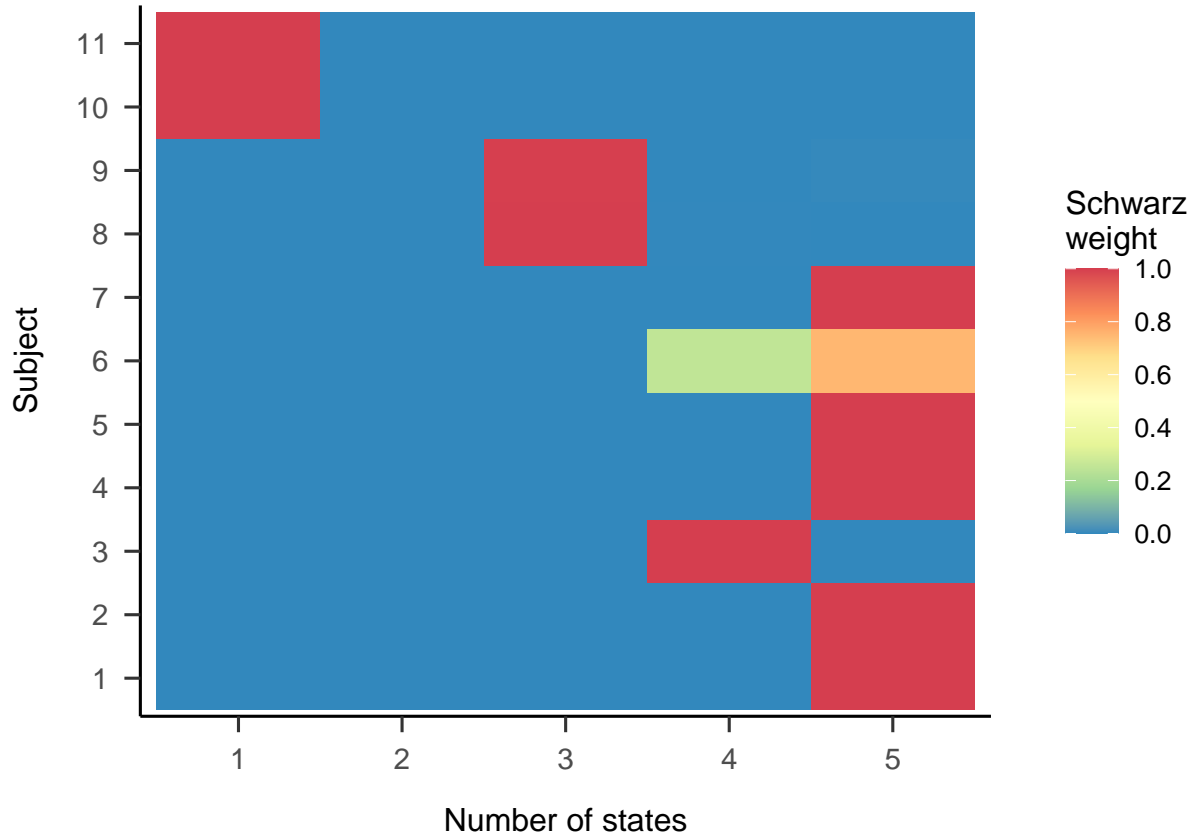


*Figure 17.* Schwarz weights displayed for each subject and HMMs with different numbers of states. Models were applied to the image condition of the Andersson et al. (2017) data set. Higher weights indicate better model fit.

event durations as well as the number of events were normalized to  $[0,1]$  for each coder and algorithm. Second, the sum of the root square deviation between each algorithm and the average of the two human coders is computed. This metric can only be relatively interpreted (Andersson et al., 2017).

Table 4 shows that, in all three conditions, gazeHMM has a higher RMSD than all the the other algorithms for classified fixations. The high RMSD values were mostly due to a large amount of very short fixations.

For saccades, gazeHMM has a relatively low RMSD in the image and video condition  
(i.e., RMSD).



*Figure 18.* Schwarz weights displayed for each subject and HMMs with different numbers of states. Models were applied to the moving dots condition of the Andersson et al. (2017) data set. Higher weights indicate better model fit.

and a relatively high RMSD in the moving dots condition (see Table 5). Here, gazeHMM overestimated the duration of saccades, as indicated by high mean and SD durations.

For PSOs, gazeHMM had the highest RMSD in the image and video condition (see Table 6). In the moving dots condition, its RMSD was lower than for NH (Nyström & Holmqvist, 2010) but higher than for LNS (Larsson, Nystrom, & Stridh, 2013). Here, gazeHMM underestimated the number of PSOs and their duration compared to the human coders.

No other algorithm in the study was designed to parse smooth pursuits, but gazeHMM classified a large number of short smooth pursuits, causing a substantially

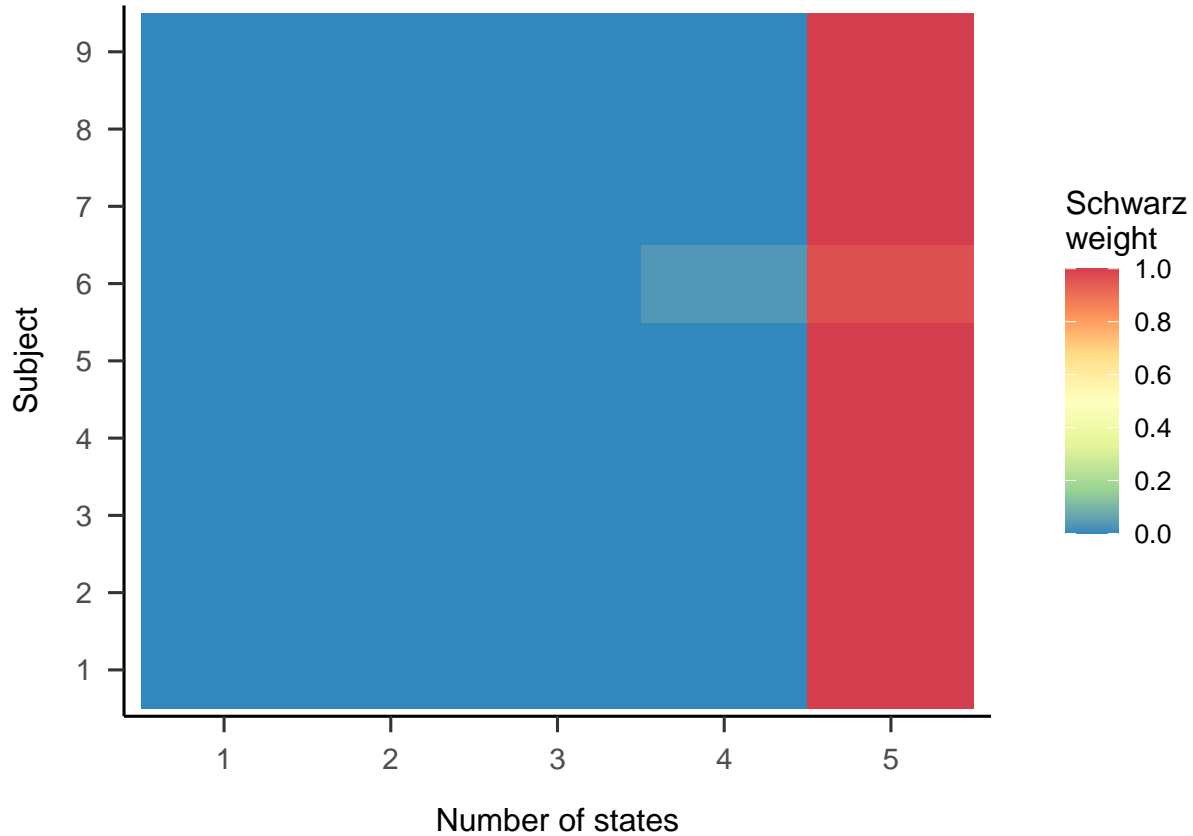


Figure 19. Schwarz weights displayed for each subject and HMMs with different numbers of states. Models were applied to the video condition of the Andersson et al. (2017) data set. Higher weights indicate better model fit.

higher RMSD than the human coders (see Figure 7).

In sum, gazeHMM did not outperform all other algorithms regarding the RMSD of event durations compared to human coders. The duration of fixations and smooth pursuits was largely underestimated and the number of these two events largely overestimated. For saccades and PSOs, gazeHMM estimated relatively accurate durations and event numbers, but did not reach the best algorithms.

### ***Sample-to-sample Agreement.***

Additionally, I compared gazeHMM to the other algorithms by using Cohen's kappa as a measure of sample-to-sample agreement between algorithms and human coders (see

Table 4

*Fixation Duration Descriptives and RMSD Between Algorithms and Human Coders*

Algorithm	Image				Moving dots				Video			
	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	0.275	0.285	403	0.02	0.19	0.09	12	0.089	0.338	0.303	82	0.305
coderRA	0.271	0.287	391	0.02	0.167	0.092	21	0.089	0.255	0.185	81	0.305
gazeHMM	0.024	0.032	2457	2.076	0.017	0.018	455	1.657	0.023	0.026	1124	1.757
CDT	0.397	0.559	251	0.914	0.06	0.127	165	0.773	0.213	0.297	211	0.344
EM	-	-	-	-	-	-	-	-	-	-	-	-
IDT	0.399	0.328	242	0.485	0.323	0.146	8	0.592	0.554	0.454	48	0.779
IKF	0.174	0.239	513	0.406	0.217	0.184	72	0.526	0.258	0.296	169	0.219
IMST	0.304	0.293	333	0.124	0.268	0.14	12	0.412	0.526	0.825	71	1.169
IHMM	0.133	0.216	701	0.644	0.214	0.286	67	0.83	0.234	0.319	194	0.316
IVT	0.114	0.204	827	0.774	0.203	0.282	71	0.796	0.202	0.306	227	0.391
NH	0.258	0.299	292	0.112	0.38	0.333	30	1.355	0.429	0.336	83	0.366
BIT	0.209	0.136	423	0.467	0.189	0.113	67	0.212	0.248	0.215	170	0.211
LNS	-	-	-	-	-	-	-	-	-	-	-	-

*Note.* Durations are displayed in seconds. gazeHMM classified four events. RMSD = root mean square deviation. Table adapted from Andersson et al. (2017).

Table 8). Absolute kappa values were interpreted according to Landis and Koch (1977).

For fixations, absolute kappa values in all three conditions indicated a slight to fair agreement between gazeHMM and human coders. Compared to the other algorithms, the agreement of gazeHMM was the lowest in the image condition but the highest in the moving dots and video condition.

For saccades, gazeHMM showed kappa values that correspond to a moderate to substantial agreement and were relatively high compared to the other algorithms. However, gazeHMM never reached the highest agreement.

For PSOs, absolute kappa values showed slight to fair agreement agreement to human coders. It was relatively low in the moving dots condition and second-rate in the image and video conditions compared to NH and LNS.

The agreement for smooth pursuits was moderate in the moving dots condition, fair

Table 5

*Saccade Duration Descriptives and RMSD Between Algorithms and Human Coders*

Algorithm	Image				Moving dots				Video			
	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	0.032	0.016	377	0.1	0.023	0.01	47	0.054	0.027	0.012	117	0.071
coderRA	0.034	0.014	374	0.1	0.022	0.011	47	0.054	0.027	0.011	127	0.071
gazeHMM	0.04	0.024	388	0.501	0.035	0.026	59	1.151	0.039	0.021	122	0.547
CDT	-	-	-	-	-	-	-	-	-	-	-	-
EM	0.025	0.022	787	1.209	0.017	0.014	93	0.837	0.02	0.016	252	0.906
IDT	0.025	0.015	258	0.408	0.032	0.014	10	0.803	0.024	0.053	41	1.426
IKF	0.062	0.037	353	1.494	0.06	0.026	29	1.736	0.055	0.02	107	1.025
IMST	0.017	0.01	335	0.625	0.013	0.005	18	0.822	0.018	0.01	76	0.496
IHMM	0.048	0.026	368	0.747	0.041	0.017	27	0.931	0.042	0.018	109	0.618
IVT	0.041	0.022	373	0.434	0.036	0.014	28	0.671	0.036	0.016	112	0.395
NH	0.05	0.02	344	0.615	0.043	0.016	42	0.745	0.044	0.018	104	0.696
BIT	-	-	-	-	-	-	-	-	-	-	-	-
LNS	0.029	0.012	390	0.235	0.026	0.011	53	0.16	0.028	0.012	122	0.039

*Note.* Durations are displayed in seconds. gazeHMM classified four events. RMSD = root mean square deviation. Table adapted from Andersson et al. (2017).

Table 6

*PSO Duration Descriptives and RMSD Between Algorithms and Human Coders*

Algorithm	Image				Moving dots				Video			
	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	0.025	0.014	313	0.229	0.015	0.005	33	0.786	0.023	0.013	97	0.744
coderRA	0.025	0.012	310	0.229	0.015	0.008	28	0.786	0.021	0.012	89	0.744
gazeHMM	0.02	0.02	213	2.208	0.004	0.006	23	1.193	0.025	0.019	66	1.944
NH	0.028	0.013	237	1.079	0.024	0.012	17	2.068	0.028	0.013	78	1.333
LNS	0.025	0.009	319	0.457	0.02	0.009	31	0.595	0.024	0.01	87	0.755

*Note.* Durations are displayed in seconds. gazeHMM classified four events. RMSD = root mean square deviation. Table adapted from Andersson et al. (2017).

Table 7

*Smooth Pursuit Duration Descriptives and RMSD Between Algorithms and Human Coders*

Algorithm	Image				Moving dots				Video			
	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
coderMN	0.363	0.187	3	0.235	0.363	0.236	48	0.329	0.559	0.391	51	0.135
coderRA	0.299	0.18	17	0.235	0.367	0.333	45	0.329	0.516	0.376	70	0.135
gazeHMM	0.02	0.022	2648	2.882	0.022	0.022	493	2.835	0.025	0.026	1178	2.933

*Note.* Durations are displayed in seconds. gazeHMM classified four events. RMSD = root mean square deviation. Table adapted from Andersson et al. (2017).

in the video condition, and slight in the image condition. No other algorithm in the study was designed to detect smooth pursuits.

In sum, gazeHMM revealed a relatively high sample-to-sample agreement to human coders for saccades. For PSOs and smooth pursuits, I found fair to moderate agreement. The performance for fixations was absolutely and relatively low.

### ***Disagreement and Confusion.***

At last, I computed the overall disagreement<sup>4</sup> as the percentage of all samples taken together that were classified differently by gazeHMM than the human coders. Figure 20 shows that in the image condition, gazeHMM with four events had a higher overall disagreement by between 10 and 20% than all but two of the other algorithms. In the moving dots and video conditions, the disagreement was between 20 and 30% lower than for all other algorithms. Notably, gazeHMM's disagreement was similar across conditions (around 45%). In summary, gazeHMM had a lower overall disagreement than all other algorithms in the moving dots and video conditions but higher disagreement in the image

<sup>4</sup> Even though I proposed to compute the *disagreement ratio* and Andersson et al. (2017) also used that term, the comparison technically involves percentages and not ratios. Thus, I decided to use only the term *disagreement* instead.

Table 8

*Cohen's Kappa Between Human Coders and Algorithms for Different Conditions and Events*

Algorithm	Fixations			Saccades			PSOs			Smooth pursuit		
	Image	Dots	Video	Image	Dots	Video	Image	Dots	Video	Image	Dots	Video
coderMN	0.92	0.81	0.83	0.95	0.91	0.94	0.88	0.82	0.83	-	-	-
coderRA	0.92	0.84	0.82	0.95	0.91	0.94	0.88	0.8	0.81	-	-	-
gazeHMM	0.28	0.24	0.18	0.68	0.58	0.72	0.33	0.14	0.4	0.01	0.56	0.21
CDT	0.38	0.06	0.11	0	0	0	0	0	0	-	-	-
EM	0	0	0	0.64	0.66	0.67	0	0	0	-	-	-
IDT	0.36	0	0.03	0.45	0.26	0.38	0	0	0	-	-	-
IKF	0.63	0.03	0.14	0.58	0.46	0.59	0	0	0	-	-	-
IMST	0.38	0	0.03	0.54	0.3	0.52	0	0	0	-	-	-
IHMM	0.67	0.03	0.13	0.69	0.6	0.71	0	0	0	-	-	-
IVT	0.67	0.03	0.13	0.75	0.63	0.76	0	0	0	-	-	-
NH	0.52	0	0.01	0.67	0.6	0.68	0.24	0.2	0.25	-	-	-
BIT	0.67	0.03	0.14	0	0	0	0	0	0	-	-	-
LNS	0	0	0	0.81	0.75	0.81	0.64	0.59	0.63	-	-	-

*Note.* Negative values were set to zero. gazeHMM classified four events.

Table adapted from Andersson et al. (2017).

condition.

Contrary to my second hypothesis, applying gazeHMM to a benchmark data set revealed that it did not outperform all other algorithms in terms of RMSD and sample-to-sample agreement. The two criteria rather indicated a moderate performance of gazeHMM. However, the overall disagreement was lower for gazeHMM than for all other algorithms in the dynamic conditions but the highest in the static image condition.

Looking at the confusion matrix in Table 9, it can be seen that gazeHMM confused fixations and smooth pursuit to a large extend. In the static image condition, it overclassified smooth pursuit events, while they were underclassified in the dynamic conditions.

**gazeHMM With Three States.** Even though it was not confirmed by the model comparison, applying gazeHMM with three states to static data seems theoretically more



Table 9

*Confusion Matrix Between gazeHMM (Rows) and Human Coders  
(Columns) for Different Conditions*

Event	Fixations	Saccades	PSOs	Pursuits	Blinks	Other
Image						
Fixations	51947	523	402	1400	324	2
Saccades	1107	8909	2758	179	877	94
PSOs	1055	562	1845	45	203	10
Pursuits	44353	1033	1560	1422	606	14
Blinks	632	260	79	41	5407	51
Moving dots						
Fixations	1718	7	5	6082	11	13
Saccades	34	946	314	743	36	21
PSOs	15	6	40	31	0	0
Pursuits	850	88	87	9843	31	9
Blinks	292	23	12	282	8776	8841
Video						
Fixations	11544	34	34	12448	1	0
Saccades	244	2727	810	367	149	19
PSOs	235	278	642	135	34	1
Pursuits	9795	51	261	17334	5	0
Blinks	85	8	0	22	810	2

*Note.* gazeHMM classified four events and blinks.

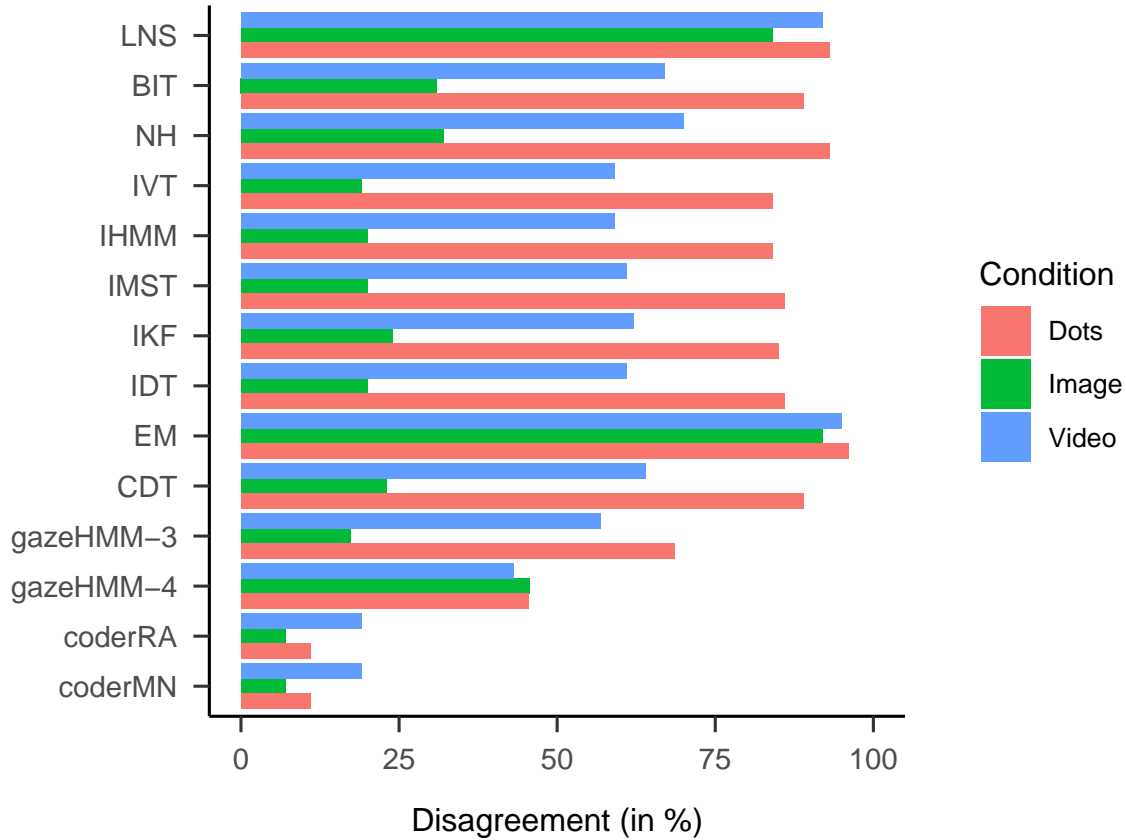


Figure 20. Disagreement between algorithms and human coders for different conditions (in %). gazeHMM-3 classified three and gazeHMM-4 classified four events. Figure adapted from Andersson et al. (2017).

appropriate. Including only three states could prevent gazeHMM from overclassifying smooth pursuits in the image data and yield better validation results. Therefore, I decided to apply gazeHMM with three states to the image data and explore the comparison to other algorithms.

Table 10 shows the RMSD between gazeHMM and human coders only for the image condition. For fixations, gazeHMM shows a comparably low RMSD, but it was relatively high for saccades and PSOs. The sample-to-sample agreement between gazeHMM with three states and the human coders measured by Cohen’s kappa is displayed in Table 11 (only for the image condition). For fixations, the absolute kappa indicated substantial

agreement and was relatively high compared to the other algorithms. The absolute kappa for saccades corresponded to moderate agreement. Compared to the other algorithms, it was rather low. For PSOs, the absolute agreement was fair, but relatively low compared to the other algorithms.

Figure 20 illustrates the overall disagreement between gazeHMM with three events and the human coders compared to the other algorithms. In all three conditions, the disagreement was slightly lower for gazeHMM than for all the other algorithms (5-10% for images and moving dots, around 20% for videos). Compared to gazeHMM with four states, the RMSD for fixations decreased while the sample-to-sample agreement increased. For saccades and PSOs, the opposite pattern was found. The overall disagreement substantially decreased. Overall, applying gazeHMM with three states to static data seems more appropriate, since the increase in fixation classification outweighs the decrease in classification of saccades and PSOs. There was support for my second hypothesis since gazeHMM had lower overall disagreement to human coders, ranging between 5 and 20% across conditions, than all other algorithms.

Table 10

*Event Duration Descriptives and RMSD Between Algorithms and Human Coders for the Image Condition*

Algorithm	Fixations				Saccades				PSOs			
	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD	Mean	SD	Events	RMSD
MN	0.28	0.29	403	0.04	0.03	0.02	377	0.1	0.03	0.01	313	0.24
RA	0.27	0.29	391	0.04	0.03	0.01	374	0.1	0.03	0.01	310	0.24
gazeHMM	0.2	0.28	503	0.45	0.05	0.04	511	1.45	0.02	0.02	280	1.6
CDT	0.4	0.56	251	1.33	-	-	-	-	-	-	-	-
EM	-	-	-	-	0.02	0.02	787	1.19	-	-	-	-
IDT	0.4	0.33	242	0.81	0.02	0.02	258	0.41	-	-	-	-
IKF	0.17	0.24	513	0.66	0.06	0.04	353	1.43	-	-	-	-
IMST	0.3	0.29	333	0.23	0.02	0.01	335	0.61	-	-	-	-
IHMM	0.13	0.22	701	1.18	0.05	0.03	368	0.72	-	-	-	-
IVT	0.11	0.2	827	1.49	0.04	0.02	373	0.41	-	-	-	-
NH	0.26	0.3	292	0.26	0.05	0.02	344	0.6	0.03	0.01	237	1.35
BIT	0.21	0.14	423	0.62	-	-	-	-	-	-	-	-
LNS	-	-	-	-	0.03	0.01	390	0.23	0.02	0.01	319	0.48

*Note.* Durations are displayed in seconds. gazeHMM classified three events.

RMSD = root mean square deviation. Table adapted from Andersson et al. (2017).

## References

- Andersson, R., Larsson, L., Holmqvist, K., Stridh, M., & Nyström, M. (2017). One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods*, *49*, 616–637.  
<https://doi.org/10.3758/s13428-016-0738-9>
- Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(7), 719–726.  
<https://doi.org/10.1109/34.865189>
- Ehinger, B. V., Groß, K., Ibs, I., & Peter, K. (2019). A new comprehensive eye-tracking test battery concurrently evaluating the Pupil Labs glasses and the

Table 11

*Cohen's Kappa Between Human Coders and Algorithms for Different Events in the Image Condition*

Algorithm	Fixations	Saccades	PSOs
coderMN	0.92	0.95	0.88
coderRA	0.92	0.95	0.88
gazeHMM-4	0.28	0.68	0.33
gazeHMM-3	0.67	0.5	0.26
CDT	0.38	0	0
EM	0	0.64	0
IDT	0.36	0.45	0
IKF	0.63	0.58	0
IMST	0.38	0.54	0
IHMM	0.67	0.69	0
IVT	0.67	0.75	0
NH	0.52	0.67	0.24
BIT	0.67	0	0
LNS	0	0.81	0.64

*Note.* Negative values were set to zero.

gazeHMM-3 classified three and gazeHMM-4 classified four events. Table adapted from Andersson et al. (2017).

- EyeLink 1000. *bioRxiv*. <https://doi.org/10.1101/536243>
- Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688.  
<https://doi.org/10.1016/j.ijforecast.2006.03.001>
- Landis, R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 159–174.
- Larsson, L., Nystrom, M., & Stridh, M. (2013). Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *IEEE Transactions on Biomedical Engineering*, 60(9), 2484–2493.  
<https://doi.org/10.1109/TBME.2013.2258918>
- Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42(1), 188–204. <https://doi.org/10.3758/BRM.42.1.188>
- Pekkanen, J., & Lappi, O. (2017). A new and general approach to signal denoising and eye movement classification based on segmented linear regression. *Scientific Reports*, 1–13. <https://doi.org/10.1038/s41598-017-17983-x>
- Pohle, J., Langrock, R., Beest, F. M. van, & Schmidt, N. M. (2017). Selecting the number of states in hidden Markov models: Pragmatic solutions illustrated using animal movement. *Journal of Agricultural, Biological, and Environmental Statistics*, 22(3), 270–293. <https://doi.org/10.1007/s13253-017-0283-8>
- Wagenmakers, E. J., & Farrell, S. (2004). AIC model selection using Akaike weights. *Psychonomic Bulletin and Review*, 11(1), 192–196.  
<https://doi.org/10.3758/BF03206482>