Keeping an Eye on Hidden Markov Models in Gaze Data Classification

Malte Valentin Lueken[1]

[1] University of Amsterdam

Author Note

Add complete departmental affiliations for each author here. Each new line herein must be indented, like this line.

Enter author note here.

Abstract

One or two sentences providing a **basic introduction** to the field, comprehensible to a scientist in any discipline.

Two to three sentences of **more detailed background**, comprehensible to scientists in related disciplines.

One sentence clearly stating the **general problem** being addressed by this particular study.

One sentence summarizing the main result (with the words "**here we show**" or their equivalent).

Two or three sentences explaining what the **main result** reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.

One or two sentences to put the results into a more **general context**.

Two or three sentences to provide a **broader perspective**, readily comprehensible to a scientist in any discipline.

Keywords: keywords

Word count: X

Keeping an Eye on Hidden Markov Models in Gaze Data Classification

Algorithm Development

As part of answering my research questions, I developed an algorithm named gazeHMM to classify gaze data into discrete eye movement events. The following section describes the development of gazeHMM and its final version that has been used to obtain results. Technical details can be found in the Technical Appendix.

Eye movement metrics

Many different metrics can be used to describe gaze data and eye movement events (**???**). Here, the goal is to find those metrics that separate the gaze data samples belonging to different events the best. However, many metrics rely on thresholds or window ranges that have to be set by the user (e.g., the distance between the mean position in a 100 ms window before and after each sample, see Olsson, 2007, and Zemblys et al.). This can be problematic because these parameters are often set without theoretical justification and they differ substantially between metrics. Belonging to the most basic metrics, which do not require parameter settings, I used velocity, acceleration, and sample-to-sample angle (similar to relative or change in angle; **???**) in gazeHMM.

Theoretically, these three metrics should separate eye movement events clearly. Fixations typically inherit samples with low velocity and acceleration (**???**). Due to tremor, the angle between samples should not follow any direction but a random walk (**???**). In contrast, saccade samples usually have a high velocity and acceleration and follow the same direction. PSO samples tend to have moderate velocity and high acceleration since they occur after high velocity saccades (**???**). They are specifically distinguished by their change in direction clustered around 180° (**???**). Lastly, smooth pursuit samples have a moderate velocity but low acceleration (due to the smoothness) and like saccades they follow the same direction (**???**; **???**).

Fitering and smoothing

As with metrics, many methods can be used to filter or smooth gaze data before, while, or after computing eye movement metrics. Their purpose is to remove noise or artifacts from the gaze data that could distort the classification (**???**). Filtering (smoothing) methods that are applied before computing the eye movement metrics target the recorded gaze position. While removing noise, filters might also erase differences in the metrics between samples. For instance, tremor movements during fixations could be filtered out and in consequence, the samples would follow the same direction. This could complicate separating fixations from smooth pursuits. The same problem applies to PSOs and saccades where filtering could erase oscillations (see Figure 1).

Instead of filtering before computing the metrics, I decided to combine both in one step. Previous algorithms have used two methods that both filter and compute derivatives of the gaze position (i.e., velocity and acceleration). (**???**) compute the first and second discrete derivative (similar to a Sobel and Laplace filter, respectively) of the gaze position to be used by their algorithm. Similarly, (**???**) use a Savitzky-Golay (SG) filter (**???**) to estimate velocity and acceleration from a gaze signal. Figure 2 displays velocity and acceleration signals obtained by both methods. They follow a similar trend, but the SG velocity signal is less noisy and acceleration signal is consequently lower than the Laplace acceleration signal. Therefore, I chose to implement the SG filter to compute velocity and acceleration signals, because it filters out more noise than the discrete derivatives but still preserves the edges in the signal to distinguish between events. To preserve motor noise in the sample-to-sample angle signal, gazeHMM computes the absolute angle with the backward difference and the first discrete derivative as the backward difference (see Technical Appendix).

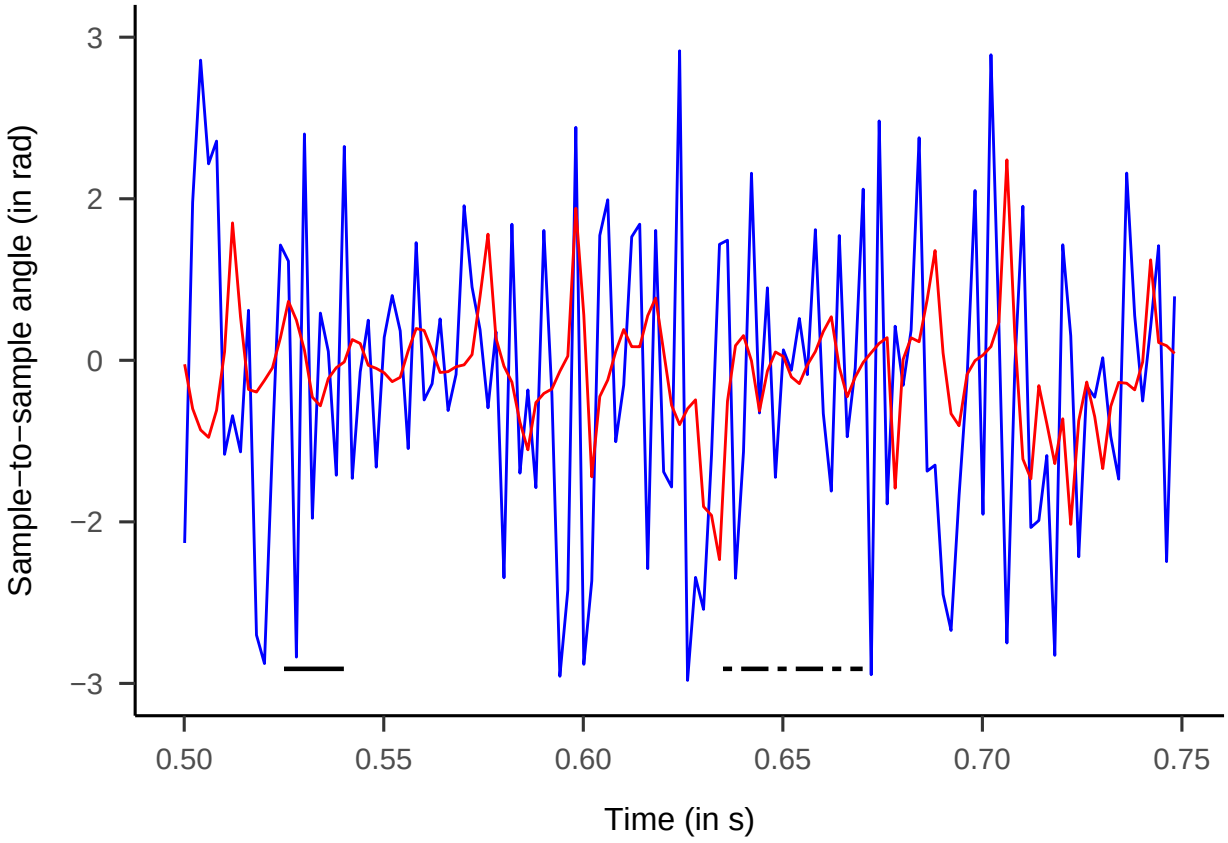

Figure 1. Example data from (**???**) displayed as sample-to-sample angle (in rad) over time (in s). Line color indicates whether the positional data has been filtered before computing the sample-to-sample angle (red) or not (blue). For this example, I used a Butterworth filter of order three and a normalized cutoff frequency of 0.3. Full line segment marks where potential oscillations are filtered out. Dashed line segment illustrates where potentially random change in angle is filtered out.

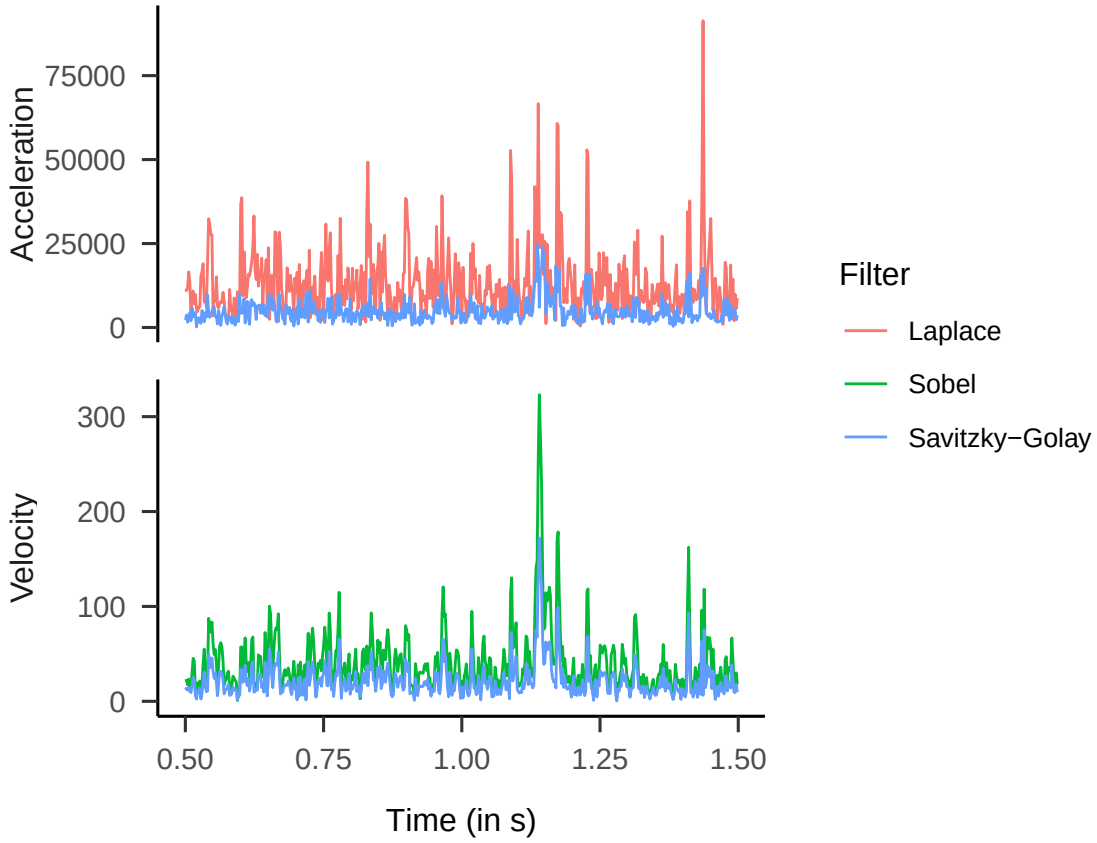

Figure 2. Example data from Andersson et al. (2017) displayed as velocity (deg/s) and acceleration (in deg/s^2) over time (in s). Line colors indicate which filter has been used to derive the signal. Savitzky-Golay filters were applied with order three and length five (corresponding to 10 ms).

The generative model

The generative model underlying gazeHMM is a multivariate hidden Markov model (HMM). It can contain between two and four states that correspond to different eye movement events: The first state always represents fixations, the second saccades, the third PSOs, and the fourth smooth pursuits. Thus, users can choose wether they would like to classify only fixations and saccades, or additionally PSOs and/or smooth puruits.

In general, HMMs consist of three submodels: An initial state model, a transition model, and a response model (**???**). In gazeHMM, the response model consists of three

response variables which are the velocity and acceleration signals obtained by the SG filter and the change in angle signal. The response variables are treated as conditionally independent on the states. I acknowledge that conditional independence might not accurately resemble the relationship between velocity and acceleration (which are naturally correlated). This step was merely taken to keep the HMM simple and identifyable.

Previous algorithms using HMMs have used Gaussian distributions to describe velocity and acceleration signals (sometimes after log-transforming them). However, several reasons speak against choosing the Gaussian: First, both signals are usually positive (depending on the computation) and due to noise above zero. Second, the distributions of both signals appear to be positively skewed conditionally on the states and third, to have variances increasing with their mean. Thus, instead of using the Gaussian, it could be more appropriate to describe velocity and acceleration with a distribution that follows these three properties. In gazeHMM, I use gamma distributions with a shape and scale parametrization for this purpose. It has to be noted that the gamma was choosen because of convenience and the best fitting distribution might be different between eye-trackers, subjects, and tasks.

To model the sample-to-sample angle, I pursued a novel approach in gazeHMM: Using a mixture of von-Mises distributions (with a mean and concentration parameter) and a uniform distribution. Both the distributions and the metric operate on the full unit circle (i.e., between 0 and 2π) which should lead to rather symmetric distributions. Moreover, the uniform distribution can distinguish fixations from the other events.

The initial state model and the transition model used multinomial distributions (with a category for each state). For all three submodels of the HMM, only parameter intercepts were estimated.

Missing data and blinks

HMMs can estimate parameters and hidden states despite missing data. They either treat data as missing at random or as missing dependent on states (**???**). In gazeHMM, I assume data to be missing at random. Most of the missing data in eye movement classification are due to blinks. The user can indicate in gazHMM which samples should be labeled as blinks (other missing samples are treated as noise). Often, eye-trackers record a few samples with unreasonably high velocity and acceleration before losing the pupil signal when a blink occurs. Since these samples could distort the classification of saccades in the HMM, I decided to remove them heuristically. Before classifying the samples, gazeHMM sets all samples within 50 ms before and after blink samples as missing. The window of 50 ms was rather motivated empirically than theoretically and can bet set to any value the user considers appropriate.

When evaluating the likelihood of the HMM given parameters and the joint density of response distributions for missing data, the model integrates over all possible values. The resulting density for missing samples is therefore set to one (**???**).

Optimization and classification

The paramters of the HMM are estimated through maximum likelihood using an expectation-maximization (EM) algorithm (**???**; **???**). The EM is generally suitable to estimate likelihoods with missing variables. For HMMs, it imputes missing with expected values and iteratively maximizes the joint likelihood of parameters conditional on the observed data (velocity, acceleration, and sample-to-sample angle) and the expected hidden states (eye movement events; **???**). The sequence of hidden states is estimated through the Viterbi algorithm (**???**; **???**) by maximizing the posterior state probability. Parameters of the response distributions (except for the uniform distribution) were optimized on the log-scale (except for the mean parameter of the von-Mises distribution) using a spectral

projected gradient method (**???**) and Barzilai-Borwein steplengths (**???**).

Postprocessing

After classifying gaze samples into states, gazeHMM applies a postprocessing routine to the estimated state sequence. I implemented this rountine because constraining the transition probabilities for PSOs to turn into non-saccade events to zero often caused PSOs not to appear in the state sequence at all. Moreover, gazeHMM did not explicitly control the duration of events in the HMM which ocassionally led to unreasonably short events. Thus, the postprocessing routine heuristically compensates for such violations. This routine relabels one-sample fixations and smooth pursuits, saccades with a duration below a minimum threshold, and PSOs that follow non-saccade events. Samples were relabeled as the state of the previous event. Finally, samples initially indicated as missing were labeled as noise (including blinks) and event metrics were computed (e.g., fixation durations).

Implementation

The algorithm is implemented in R (version: 3.6.3; **???**) and uses the packages signal (CITE) to compute velocity and acceleration signals, dplyr (CITE) to calculate sample-to-sample angles, depmixS4 (**???**) for the HMM, and BB (**???**) for Barzilai-Borwein spectral projected gradient optimization. The algorithm is available on GitHub (www.github.com/maltelueken/gazeHMM).

Simulation Study

To assess how well the HMM recovers parameters and state sequences, I 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). The major part of this

section is a direct copy of the preregistration. When appropriate, additional explanations were added and tenses adapted to make the design easier to read and understand.

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

Design

Parameter Variation. The simulation study was divided into four parts. In the first part, I varied the parameters of the HMM. For models with $k \in \{2, 3, 4\}$ states, $q \in \{10, 15, 20\}$ parameters were varied 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 was varied 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). I did not manipulate the initial state probabilities because these are usually irrelevant in the context of eye movement classification. For the transition probabilities, I only simultaneously varied the probabilities for staying in the same state (diagonals of the tranistion matrix) to reduce the complexity of the simulation. The left over probability mass was split evenly between the probabilites for switching to a different state (per row of the transition matrix). Moreover, I 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.

I defined approximate ranges for each response variable and chose true parameter intervals and default values so that they produced samples that roughly corresponded to

these ranges. Table 1 shows the assumed ranges for each event and Table X shows 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. I 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.

Sample Size and Noise Variation. In the second part, I varied 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 $N \in \{500, 2500, 10000\}$, I 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, I 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 drawn from moderately skewed distributions with modes equal to the original values. To angle, I added white noise from a von-Mises distribution with $\mu_{noise} = 0$ and $\kappa_{noise} \in 1/[0.1, 10]$ varying between data sets. τ_{noise} and κ_{noise} were varied simultaneously in equidistant steps in their intervals.

Variation of Starting Values. In the third part, I increased the variation in the starting values used for parameter estimation. The model parameters were set to their default values. For the shape, scale, and concentration parameters, I simultaneously increased the scale parameters of the starting value gamma distributions: For $k \in \{2, 3, 4\}$ states and $\beta_{start} = (\psi_{true}/2)\tau_{start}$ with $\tau_{start} \in \{1, 2, 3\}$, 100 data sets with $N = 2500$ samples were generated each ($100 \times 3 \times 3 = 900$ recoveries).

Missing data. In the last part, I set intervals of the generated data to be missing. The model parameters were set to their default values. For $k \in \{2, 3, 4\}$ states and $m \in \{1, 3, 5\}$ intervals, 100 data sets with $N = 2500$ samples were generated ($100 \times 3 \times 3 = 900$ recoveries). The length of the missing data interval $l \in [1, 200]$ samples varied in equidistant steps between the data sets.

Starting values

The HMM always started with a uniform distribution to estimate the initial state and state transition probabilities. To generate random starting values for the estimation of shape, scale, and concentration parameters, I used gamma distributions with a shape parameter of $\alpha_{start} = 3$ and $\beta_{start} = \psi_{true}/2$ with ψ_{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.

Data analysis

For each parameter, I calculated the root median square proportion deviation (RMdSPD; analogous to root median square percentage errors, see **???**) between the true and estimated parameter values[1]:

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

I treated $RMdSPD < 0.1$ as good, $0.1 \leq RMdSPD < 0.5$ as moderate, and $RMdSPD \geq 0.5$ as bad recovery of a parameter. By taking the median, I reduced the

[1] Note that the intial state probability ρ_i has also been estimated. Since the HMM only simulated one state sequence, this parameter is always either zero or one (leading to an RMdSPD of one). Therefore, I decided not to include it in the analysis.

influence of potential outliers in the estimation and using proportions enabled me to compare RMdSPD values across parameters and data sets.

Additionally, I 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 1. 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, I 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. Higher kappa values were interpreted as better model accuracy. I adopted the ranges proposed by (**???**) to interpret kappa values. Models that can not be fitted were excluded from the recovery.

Table X

HMM parameter values used to generate data

State	Parameter	Interval	Default	Description
1-4	ρ_i^*	-	$1/k$	Initial state probability for starting in state i
1-4	$a_{i=j}$	[.01,.99]	0.9	Transition probability for staying in the previous state i
1-4	$a_{i\neq j}$	$(1-a_{i=j})/(k-1)$	$0.1/(k-1)$	Transition probability for switching to from state i to a different state j

State	Parameter	Interval	Default	Description
1	α_{vel}	[1,5]	3	Shape parameter of the velocity gamma distribution
1	β_{vel}	[0.1,0.6]	0.35	Scale parameter of the velocity gamma distribution
1	α_{acc}	[1,5]	3	Shape parameter of the acceleration gamma distribution
1	β_{acc}	[0.05,0.25]	0.15	Scale parameter of the acceleration gamma distribution
1	min^*	-	0	Minimum of the uniform distribution
1	max^*	-	2π	Maximum of the uniform distribution
2	α_{vel}	[1,5]	3	
2	β_{vel}	[5,15]	10	
2	α_{acc}	[1,5]	3	
2	β_{acc}	[1,5]	3	
2	μ^*	-	0	Mean parameter of the von-Mises distribution
2	κ	$1/[0.1, 10]$	1	Concentration parameter of the von Mises distribution

State	Parameter	Interval	Default	Description
3	α_{vel}	[1,5]	3	
3	β_{vel}	[0.5,1.5]	1	
3	α_{acc}	[1,5]	3	
3	β_{acc}	[1,5]	3	
3	μ^*	-	π	
3	κ	$1/[0.1, 10]$	1	
4	α_{vel}	[1,5]	3	
4	β_{vel}	[0.5,1.5]	1	
4	α_{acc}	[1,5]	3	
4	β_{acc}	[0.05,0.25]	0.15	
4	μ^*	-	0	
4	κ	$1/[0.1, 10]$	1	

Note. Parameters marked with * will not be varied but always set to their default values. k is the number of states in the model.

Results

Simulation study

Parameter variation.

Two States.

In the first part of the simulation, I examined how varying the parameters in the HMM affects the deviation and accuracy between the true and estimated parameters and state sequence. Figure 3 displays the RMdSPD between true and estimated parameters depending on which parameter has been manipulated in the HMM. The RMdSPD was

Table 1

Approximate ranges of response variables used to generate parameter values

Event	Resp.variable	Range
Fixation	Velocity	0-50
Fixation	Acceleration	0-50
Fixation	Angle	uniform
Saccade	Velocity	50-1000
Saccade	Acceleration	50-500
Saccade	Angle	0
PSO	Velocity	20-100
PSO	Acceleration	10-90
PSO	Angle	pi
Smooth pursuit	Velocity	20-100
Smooth pursuit	Acceleration	0-30
Smooth pursuit	Angle	0

Note. Units are °/s (velocity), °/s^2 (acceleration), and radians (angle). ~ indicates that the distribution has a peak at this value. Velocity ranges are based on event velocities reported in (**???**). Since I would also like to test the algorithm on extreme data distributions, I extended the ranges beyond those found in typical eye movement data.

Table 2

State	Interval	Default	Description
1-4	-	1/k	Initial state probability for starting in state *i*
1-4	[.01,.99]	0.9	Transition probability for staying in the previous state *i*
1-4	1-a(i=j)/(k-1)	0.1/(k-1)	Transition probability for switching to from state *i* to a different st
1	[1,5]	3	Shape parameter of the velocity gamma distribution
1	[0.1,0.6]	0.35	Scale parameter of the velocity gamma distribution
1	[1,5]	3	Shape parameter of the acceleration gamma distribution
1	[0.05,0.25]	0.25	Scale parameter of the acceleration gamma distribution
1	-	0	Minimum of the uniform distribution
1	-	pi	Maximum of the uniform distribution
2	[1,5]	3	
2	[5,15]	10	
2	[1,5]	3	
2	[1,5]	3	
2	-	0	Mean parameter of the von-Mises distribution
2	1/[0.1,10]	1	Concentration parameter of the von Mises distribution
3	[1,5]	3	
3	[0.5,1.5]	1	
3	[1,5]	3	
3	[1,5]	3	
3	-	pi	
3	1/[0.1,10]	1	
4	[1,5]	3	
4	[0.5,1.5]	1	
4	[1,5]	3	
4	[0.05,0.25]	0.15	
4	-	0	

below 0.1 for all estimated and manipulated parameters, that is, the median deviation never exceeded 10% of the true parameters.[2] The regressions between manipulated true and estimated parameters are shown in Figures 4 and 5. With one outlier at parameter $\alpha_{acc;1}$, the estimated parameters matched the true parameters very closely. The deviation seemed to increase slighlty with parameter magnitude. Considering accuracy, Figure 6 displays Cohen's kappa between true and estimated hidden state sequences. With two exceptions, kappa values were almost one, indicating nearly perfect agreement. In sum, the HMM with two states recovered parameters and hidden states very well and outliers only occured rarely.

Three States.

For the simulation with three states, the RMdSPD is shown in Figure 7. When response parameters (other than $a_{i=j}$) were manipulated, the RMdSPDs for a_{12} and a_{31} were consistently below 0.5. Varying κ in states two and three led to RMdSPDs below 0.5 in the respective states. Otherwise, RMdSPDs were consistently lower than 0.1, indicating a median deviation between true and estimated parameters below 10% of the true parameter. Inspecting the regressions between manipulated true and estimated parameters (see Figures 8 and 9) revealed strong and unbiased linear relationships (intercepts close to zero and slopes close to one). Again, the deviation seemed to increase with true parameter magnitude (reverse for kappas). In contrast to the two-state HMM, larger deviatons and more outliers were observed. Cohen's kappa values for the three-state HMM are presented in Figure 10. For most estimated models, the kappas between true and estimated state sequences were above 0.95, indicating almost perfect agreement. However, for some models, I observed kappas clustered around zero or -0.33, suggesting that state labels switched. To summarize, the three-state HMM had a slightly worse parameter recovery and more

[2] Note that the intial state probability ρ_i has also been estimated. Since the HMM only simulated one state sequence, this parameter is always either zero or one (leading to an RMdSPD of one). Therefore, I decided not to include it in the analysis.

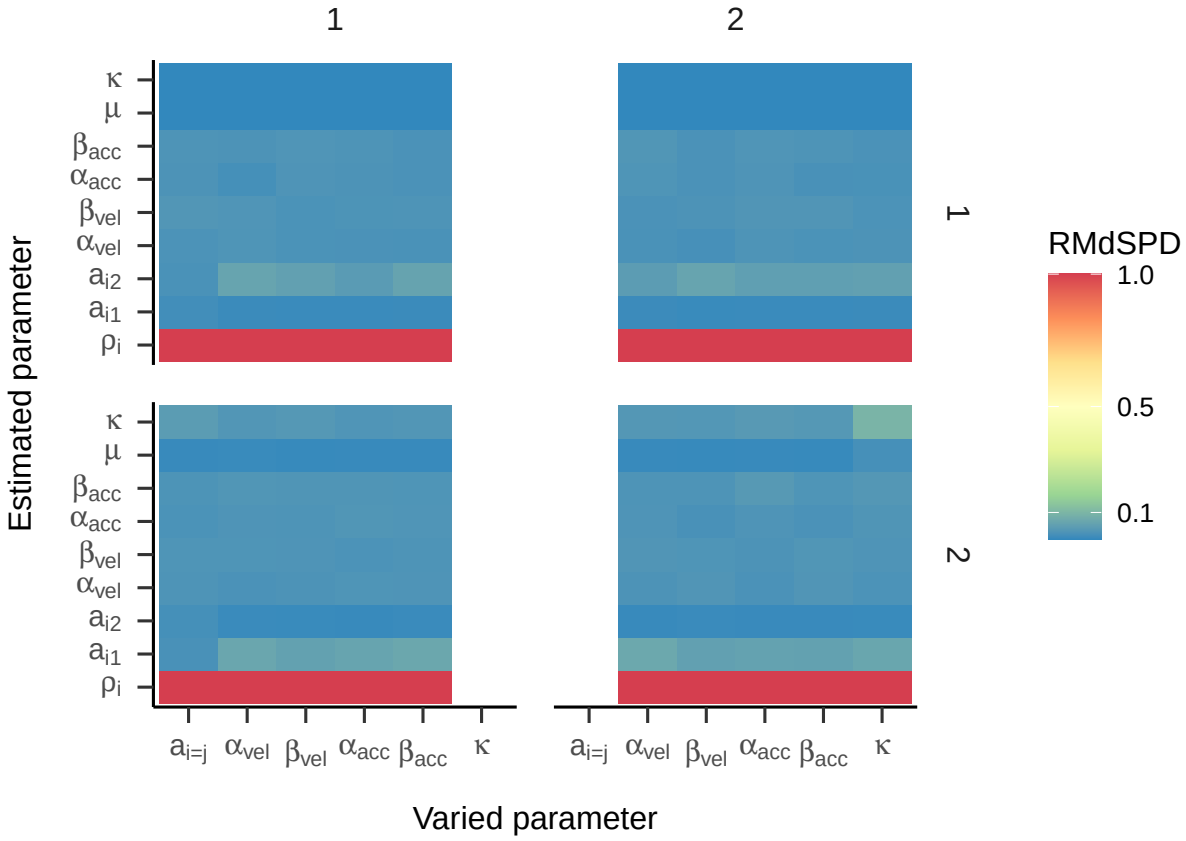

Figure 3. RMdSPD between true and estimated parameters of the two-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong.

outliers than the two-state model but shows a similar good accuracy.

Four States.

The RMdSPDs for the four-state HMM is illustrated in Figure 11. For estimated transition probabilities and α_{vel} and β_{vel} parameters in states one and four, RMdSPDs were below 0.5. Estimated kappa parameters in smooth pursuits were also often below 0.5 when paramteres in states two, three, and four were varied. Otherwise, RMdSPDs were below 0.1, indicating a median deviation below 10% of the true parameter. Looking at the

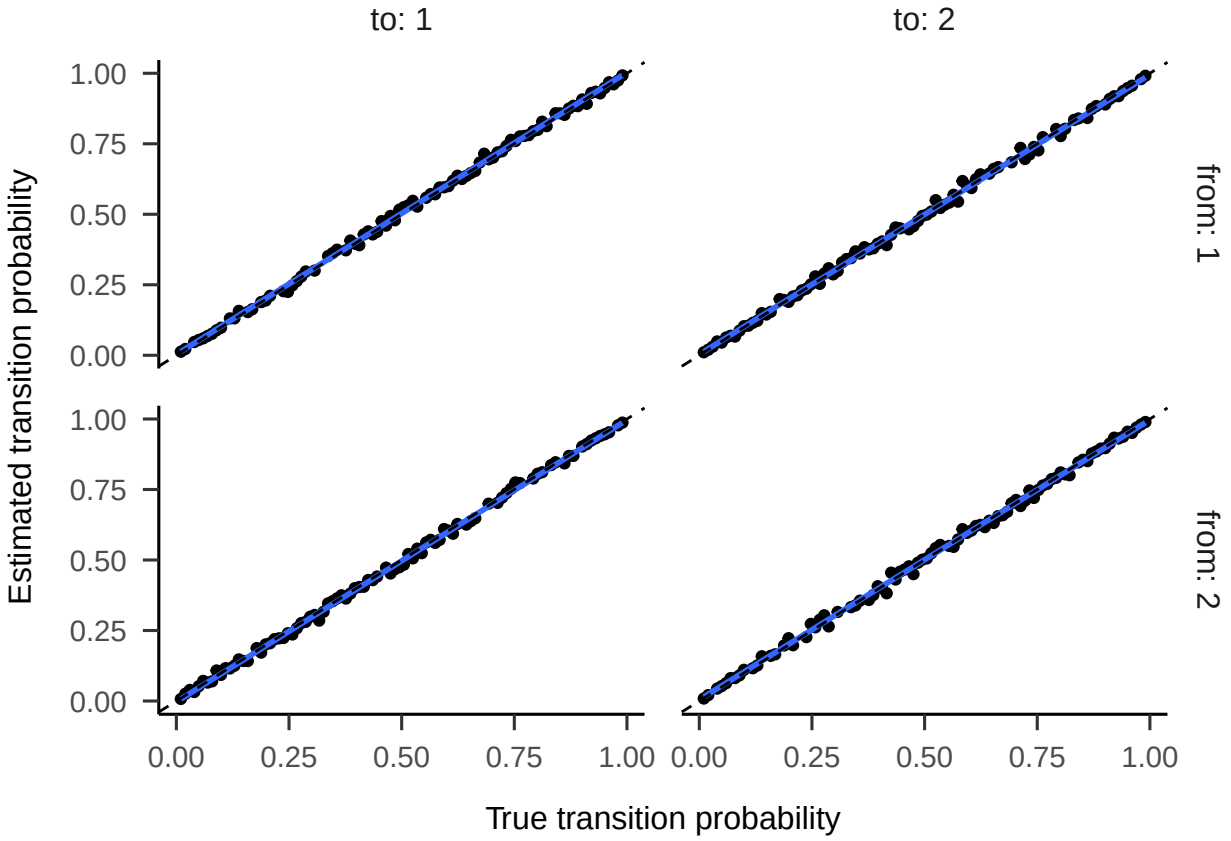

Figure 4. Regression lines between true and estimated transition probabilities for the two-state HMM in part one. Top facet labels show to and right facet labels show from which state the HMM is moving.

regressions between true and estimated paramters, Figures 12 and 13 show strong and unbiased relationships. However, there were larger deviations and more outliers than in the previous models, especially for states one and four. Accuracy measured by Cohen's kappa ranged between 0.6 and 0.9 for the majority of models, meaning moderate to almost perfect agreement between true and estimated state sequences (see Figure 14). Here, some kappa values clustered around 0.25 and zero, which again can be interpreted as the result of label switching. Summarizing, the parameter recovery for the four-state HMM was slightly worse with even more outliers compared to the three-state model. Especially the recovery of transition parameters in states corresponding to fixations and smooth pursuits decreased.

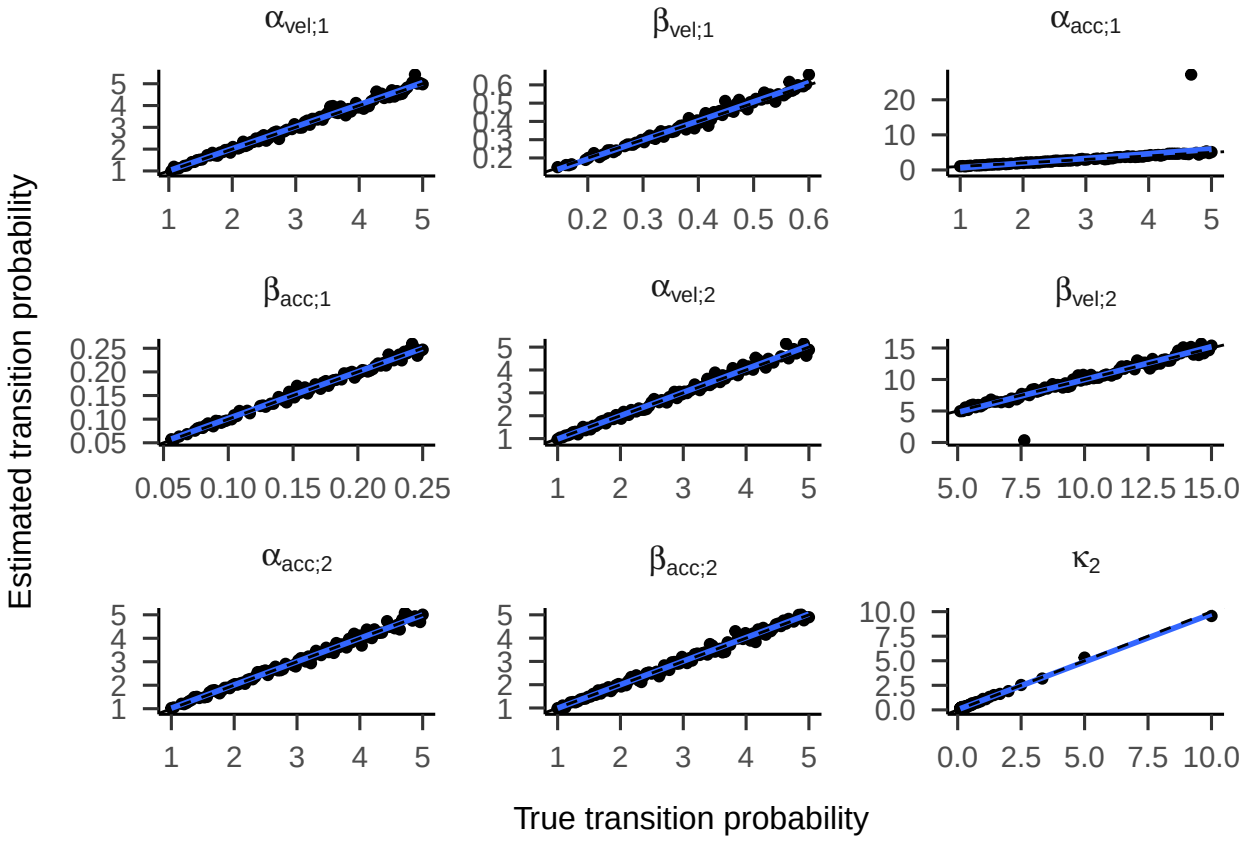

Figure 5. Regression lines between true and estimated transition response parameters of the two-state HMM in part one. Top facet labels indicate response parameters.

Overall, simulations in part one demonstrated that the HMM recovered parameters at least moderately well when parameters were manipulated (all RMdSPDs below 0.5). The HMM estimated state sequences very accurately. Adding states to the model decreased the accuracy and recovery slightly, leading to more outliers, especially when smooth pursuits were added.

Sample size and noise variation.

Two States.

In the second part, I 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 β_{vel} and β_{acc} in both states (see Figure 15). The other estimated parameters showed RMdSPDs close to or

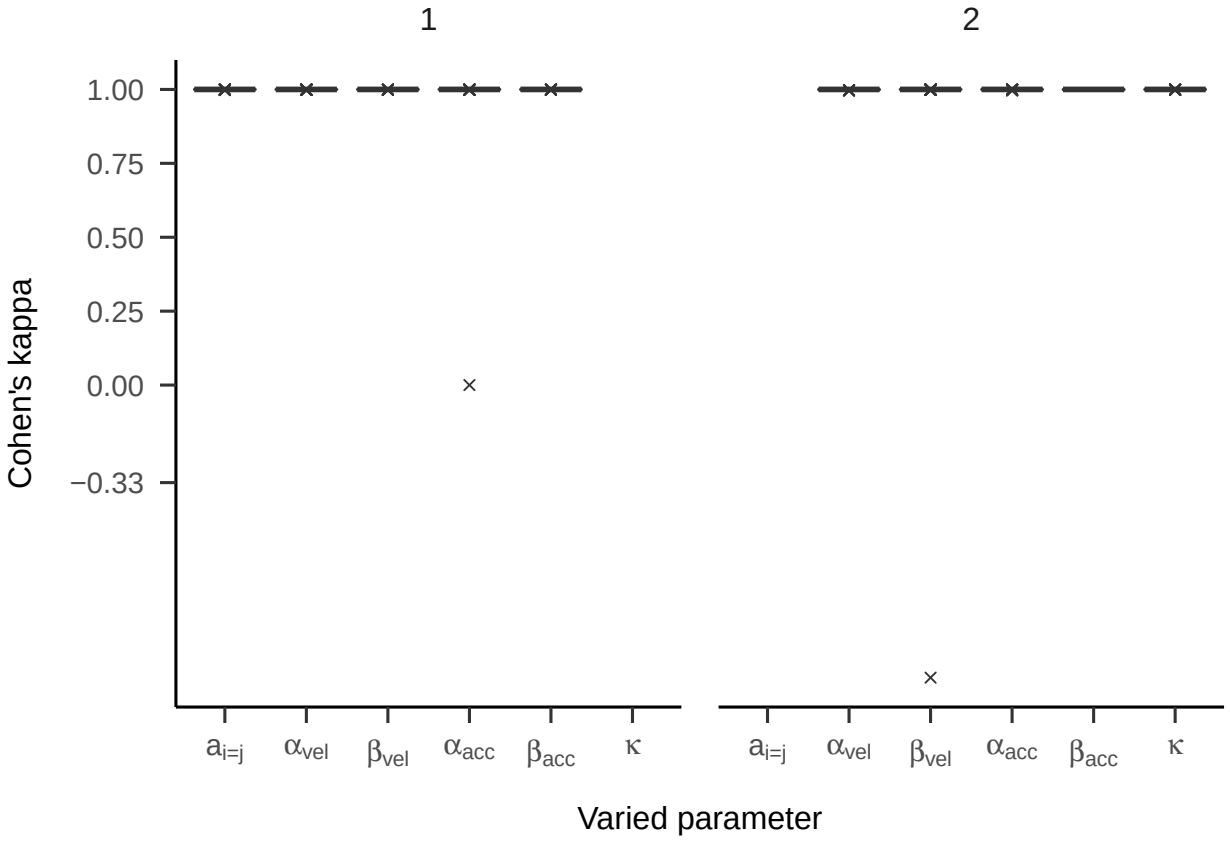

Figure 6. Boxplots displaying Cohen's kappa depending on which parameter of the two-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians. Crosses represent outliers (distance to first/third quartile higher than 1.5 times the inter-quartile range [IQR]).

below 0.1. Increasing the sample size seemed to improve RMdSPDs for most parameters slightly. For β_{vel} and β_{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 16, left plot). To conclude, adding noise did not affect the parameter recover or accuracy of the two-state HMM, whereas increasing the sample size improved the recovery slightly.

Three States.

The RMdSPDs for the β_{vel} and β_{acc} were above 0.5 in all three states (see Figure 17).

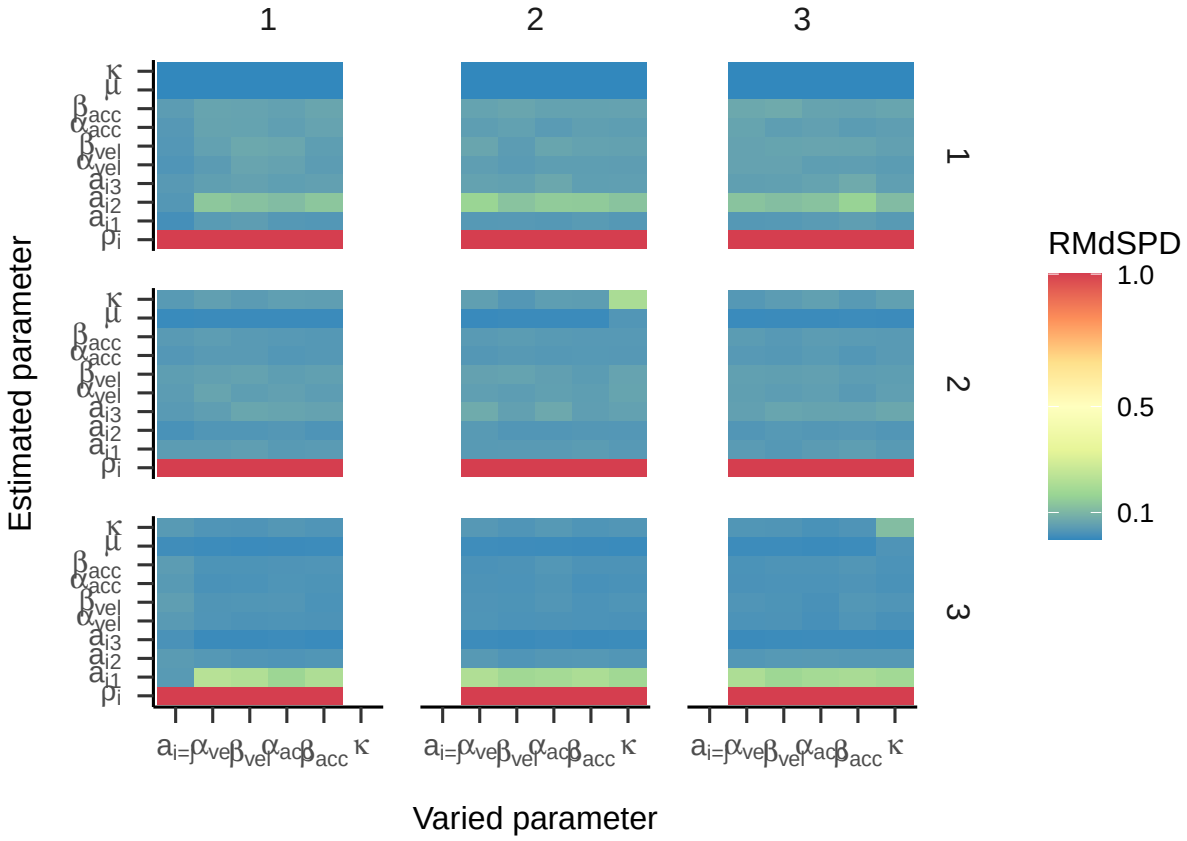

Figure 7. RMdSPD between true and estimated parameters of the three-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong.

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, higher sample sizes had lower RMdSPDs. The accuracy of the estimated models was almost perfect with most kappa values above 0.95 (see Figure 16, middle plot). Several outliers clustered around kappas of zero and -0.33, signaling label switching. For the three-state HMM, adding noise led to a slightly worse recovery and accuracy overall. However, scale parameters of gamma distributions in all three states were only badly recovered.

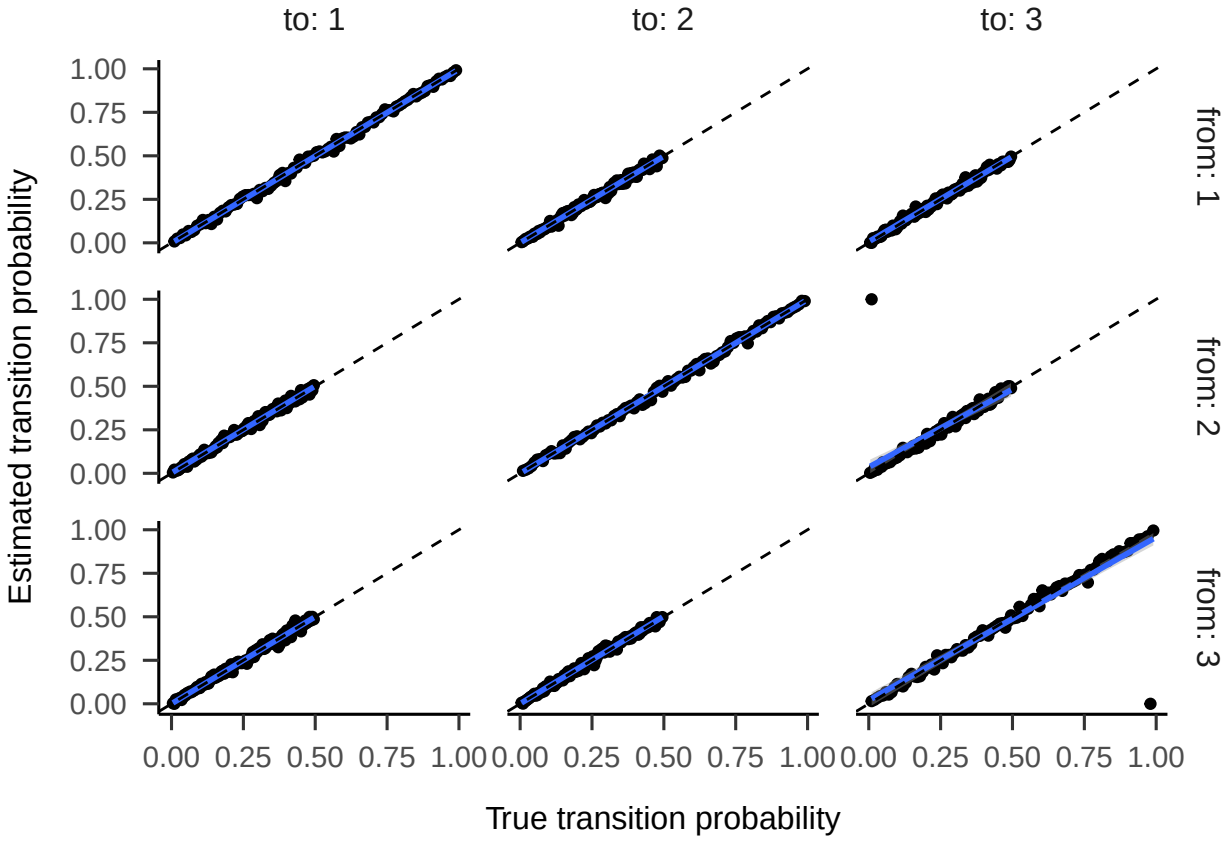

Figure 8. Regression lines between true and estimated transition probabilities for the three-state HMM in part one. Top facet labels show to and right facet labels show from which state the HMM is moving.

Four States.

RMdSPDs regarding the four-state HMM are displayed in Figure 18. For states one and four, values for most parameters (including all transition probabilites) were above 0.5. Similarly, RMdSPDs for β_{vel} and β_{acc} in states two and three were above 0.5. 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 16, right plot). Multiple kappa values clustered around 0.25 or zero, which can be explained by label switching. In summary, adding noise to data generated by the four-state HMM heavily

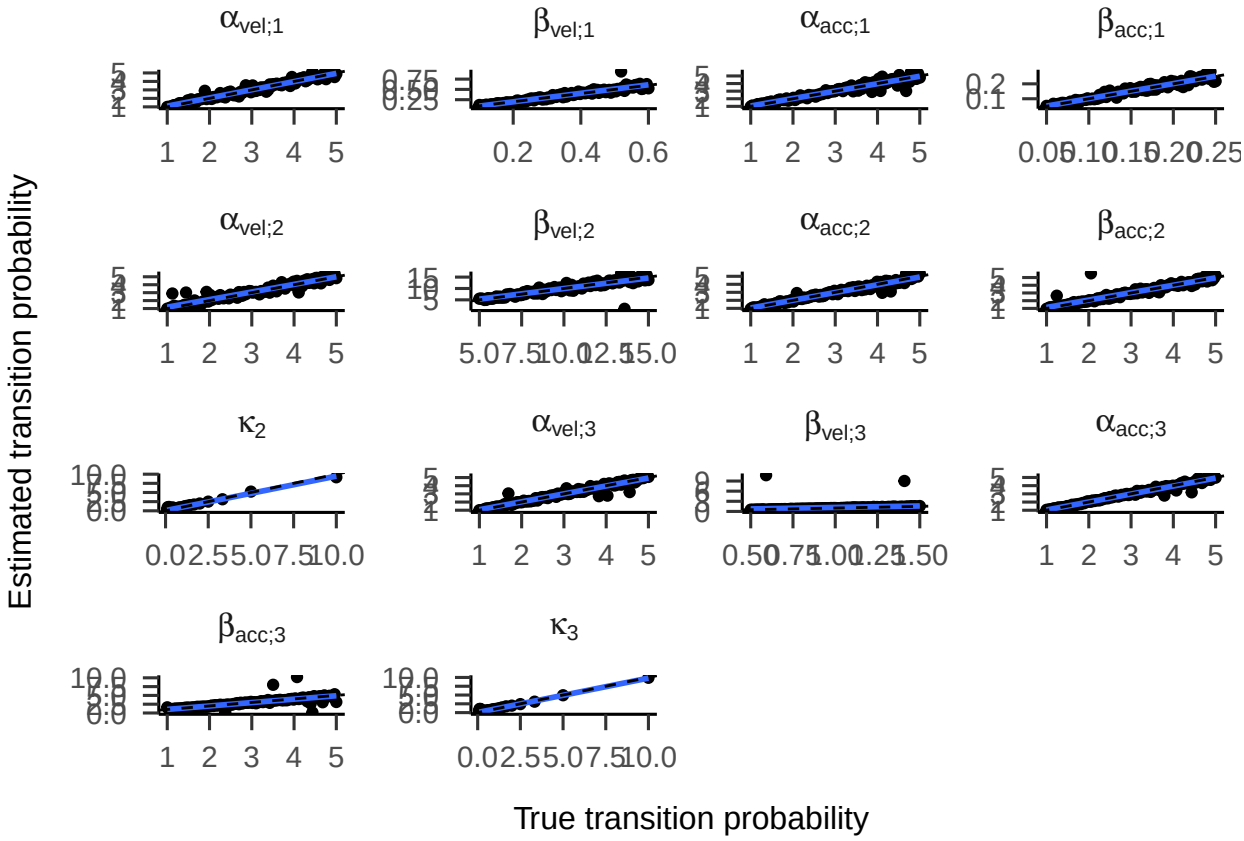

Figure 9. Regression lines between true and estimated transition response parameters of the three-state HMM in part one. Top facet labels indicate response parameters.

decreased the recovery for most parameters in states corresponding to fixations and smooth pursuits. For saccade and PSO states, only scale parameters of gamma distributions were badly recovered, but increasing the sample size slightly improved the recovery.

In general, the HMM recovered parameters well despite noise being added to the data. However, when smooth pursuits were added to the model, the parameter recovery for fixation and smooth pursuit states substantially decreased. When PSOs or smooth pursuits were included, scale parameters of gamma distributions were badly recovered. Increasing the samples in the HMM slightly improved the recovery of most parameters. The accuracy of the model was slightly lowered when including more states, but it was neither affected by the noise variability τ_{noise} nor the sample size.

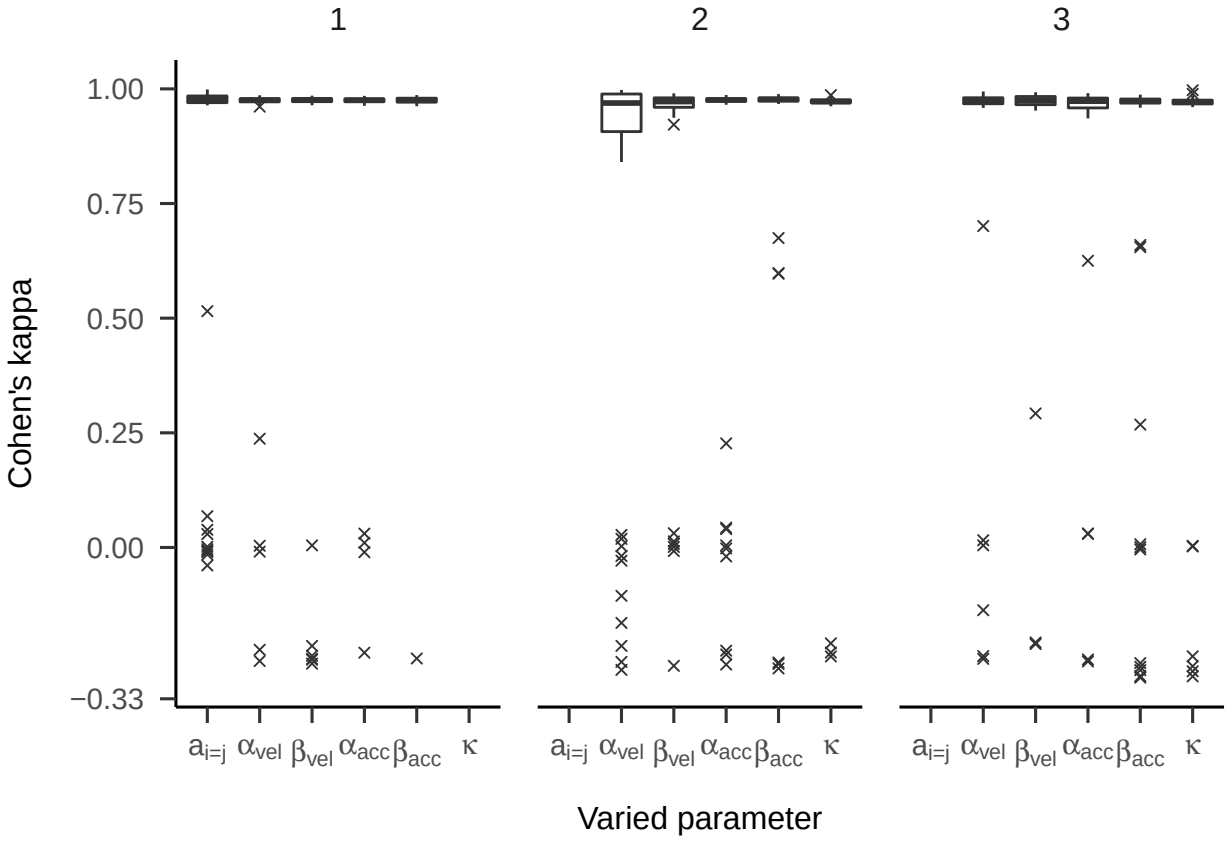

Figure 10. Boxplots displaying Cohen's kappa depending on which parameter of the three-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

Variation of starting values. The third part of the simulation investigated how increasing the variation in generating starting values affected parameter recover and accuracy of the HMM. For the two-state HMM, all parameters displayed RMdSPDs lower than 0.1 (see Figure 19). Cohen's kappa values were slightly lower than 1, indicating almost perfect accuracy (see Figure 20). For the three-state HMM, RMdSPDs were lower than 0.1 except for a_{12} and a_{31}, which were below 0.5 (see Figure 21). As in previous parts, Cohen's kappa values were mostly above 0.95 (almost perfect accuracy) with exceptions

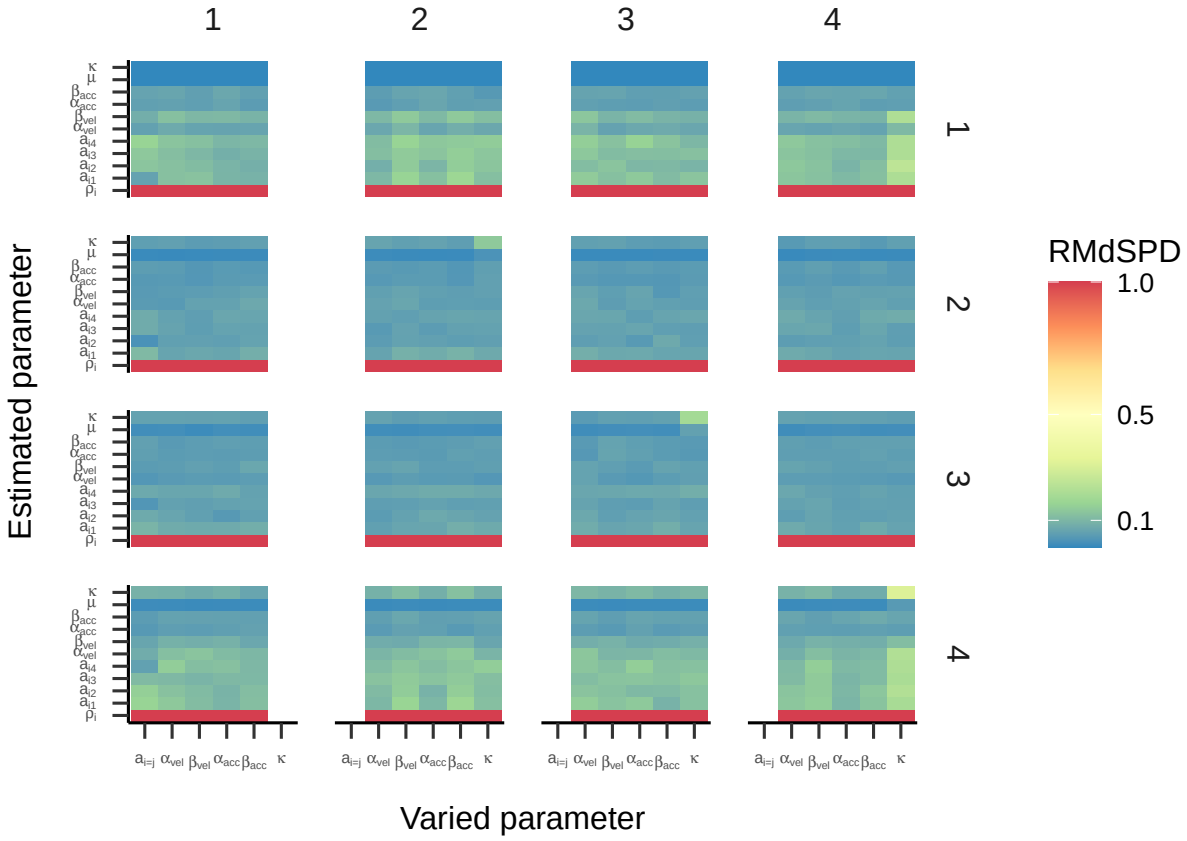

Figure 11. RMdSPD between true and estimated parameters of the four-state HMM in part one of the simulation. Labels on the x-axis indicate which true parameters have been manipulated and labels on the y-axis show for which estimated parameter the RMdSPD is displayed. Top facet labels specify in which state the parameters have been varied and right facet labels denote to which state estimated parameters belong.

clustering around zero and -0.33 (see Figure 20). Regarding the four state-HMM, RMdSPS of transition probabilities for states one and four were clustered above 0.1, other estimated parameters in the model showed values below 0.1 (see Figure 22). The HMM showed substantial to almost perfect accuracy, as Cohen's kappa values were mostly above 0.8 with a few values clustering around 0.6, 0.25, and zero (see Figure 20). In conclusion, the parameter recovery of the model worked very well. State sequences were accurately estimated and more states being included in the model slightly decreased the accuracy.

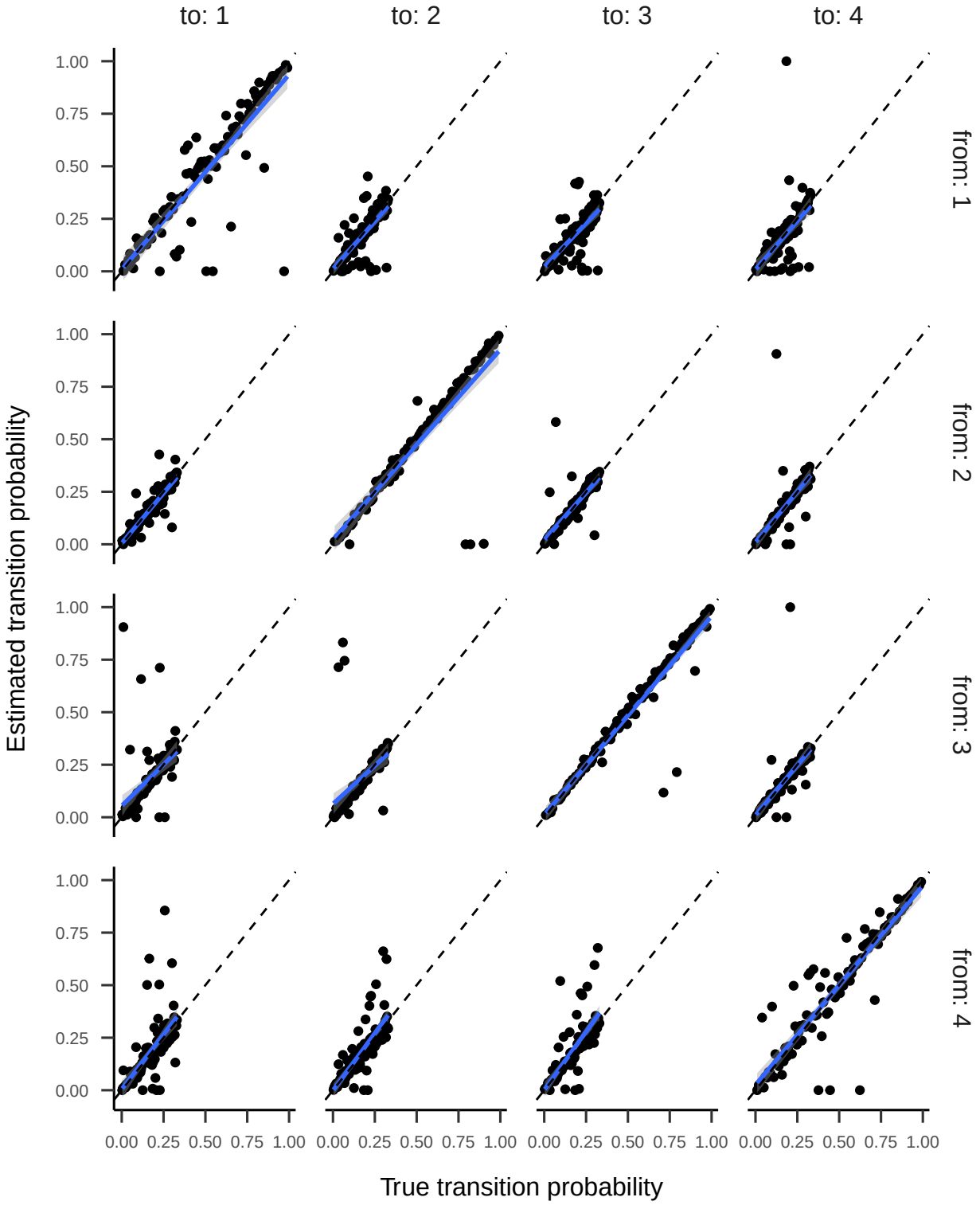

Figure 12. Regression lines between true and estimated transition probabilities for the four-state HMM in part one. Top facet labels show to and right facet labels show from which state the HMM is moving.

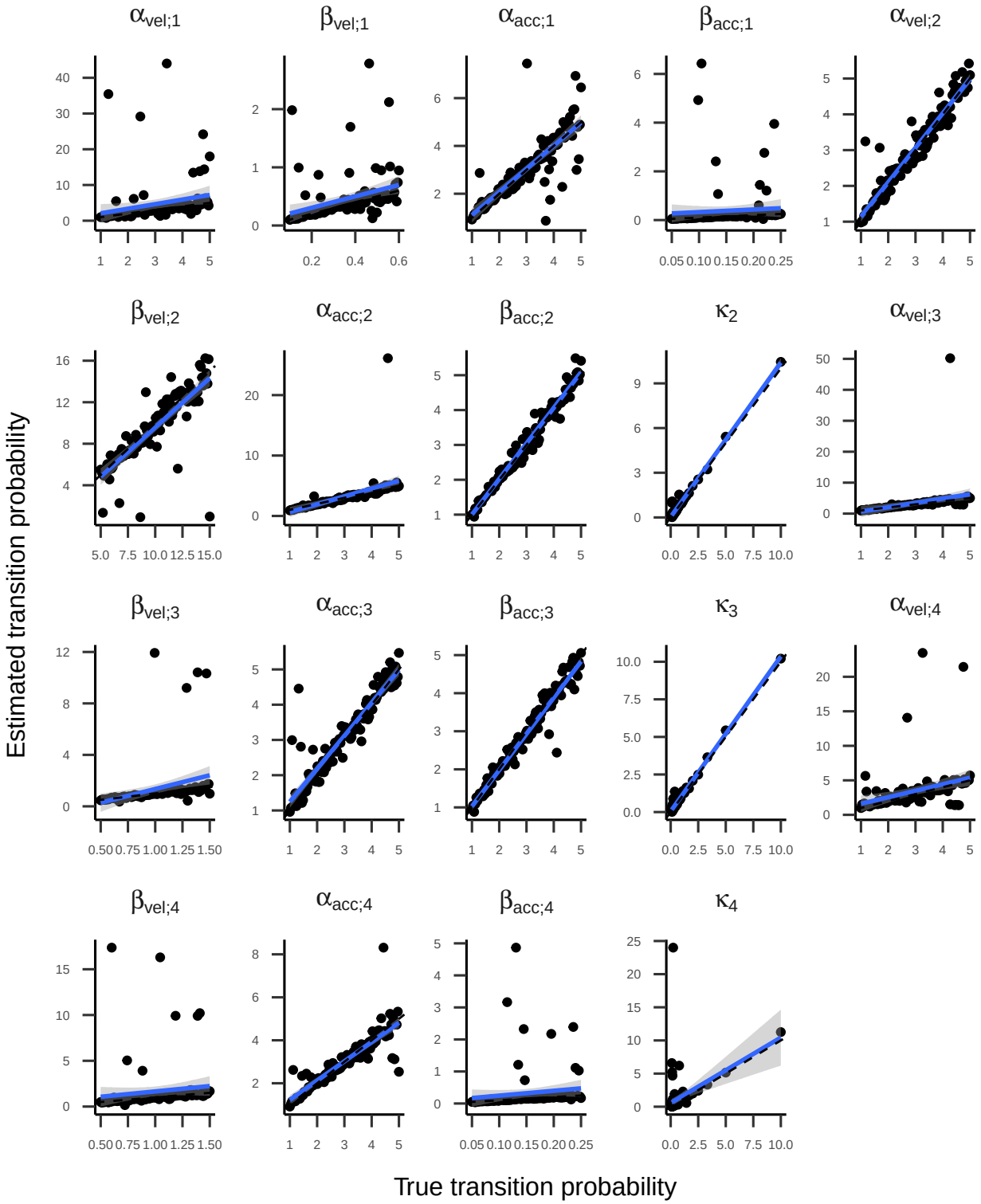

Figure 13. Regression lines between true and estimated transition response parameters of the four-state HMM in part one. Top facet labels indicate response parameters.

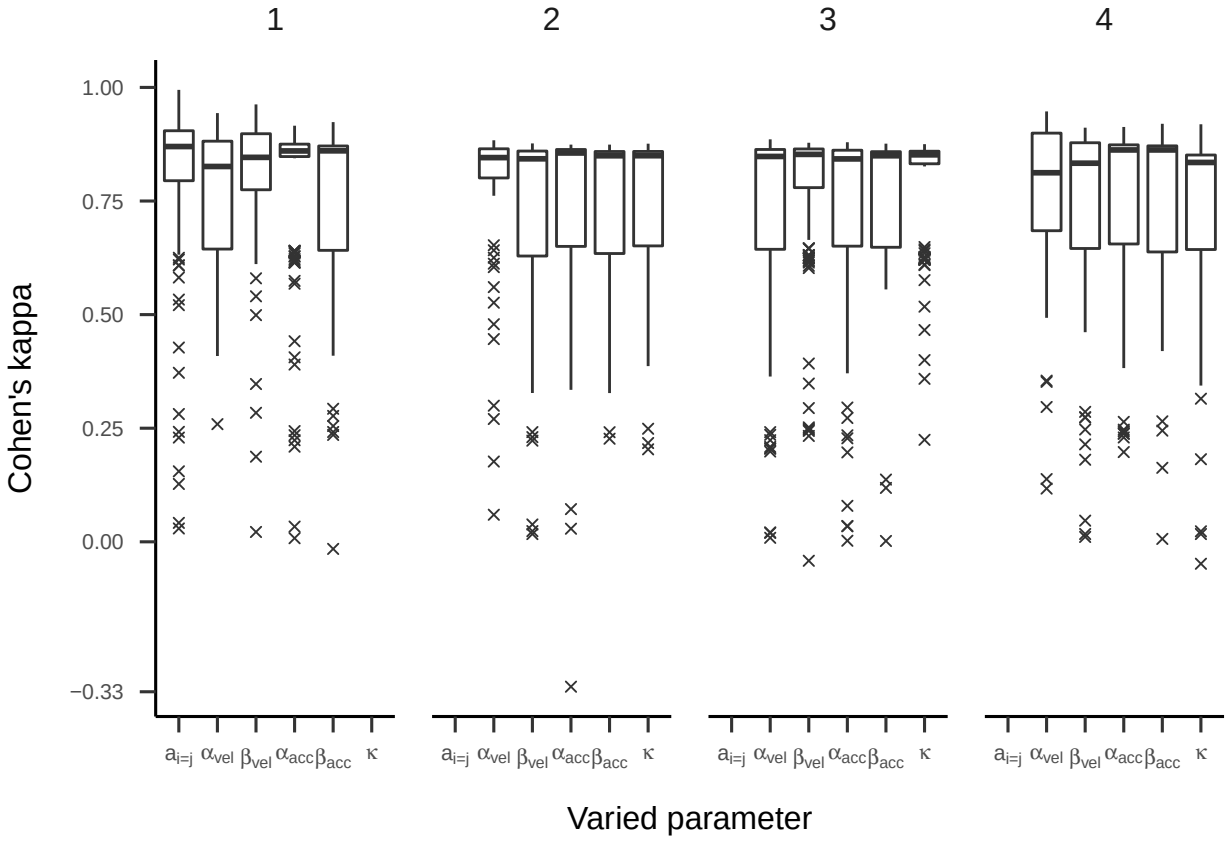

Figure 14. Boxplots displaying Cohen's kappa depending on which parameter of the four-state HMM has been manipulated in part one. Top facet labels indicate for which state parameters have been manipulated. Black solid lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

Increasing the variation in starting values for parameter estimation did neither affect the parameter recovery nor accuracy.

In this part, I explored post-hoc whether clusters of low Cohen's kappa values were due to label switching. According to Visser and Speekenbrink (**???**), label switching occurs when exchanging the order of the states leads to equally likely models. Thus, the model is accurate but has oppositely estimated states compared to the true states. In this case, an HMM with three states that is perfectly accurate but has one label swtiched would have a

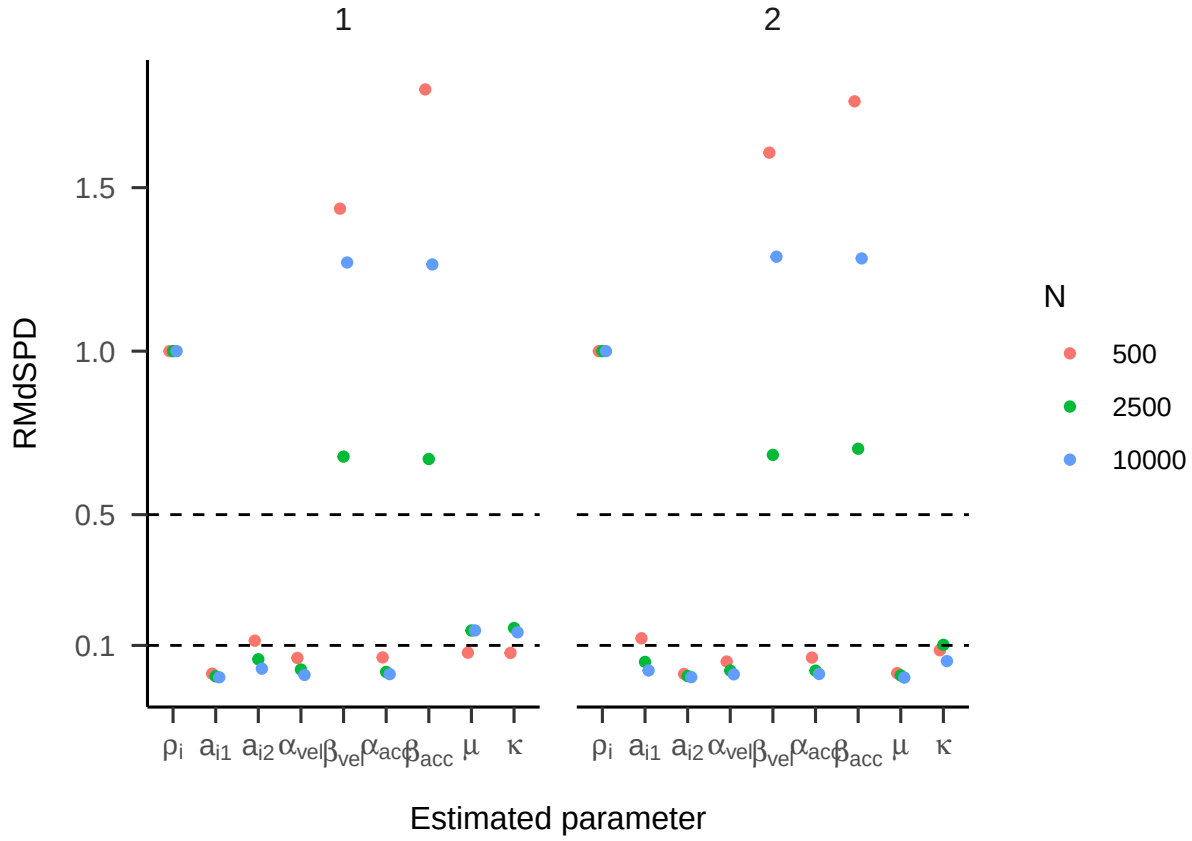

Figure 15. RMdSPD between true and estimated parameters of the two-state HMM in part two of the simulation. Colours indicate different sizes of generated data. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters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

Cohen's kappa of 0. To test if label switching occured, I manually switched one or two labels post-hoc (see Figure 23). It can be seen that this approach resolved low accuracy clusters for three and four states.

Missing data. In the last part, data intervals of varying length were set to be missing. Regarding parameter recovery, RMdSPDs were almost exactly mirroring those in the previous part for two, three, and four states (see Figures 24, 25, and 26, respectively). Looking at accuracies, Figure 27 illustrates an interaction between the length of and amount of missing data intervals. Cohen's kappa values linearly decreased with the length

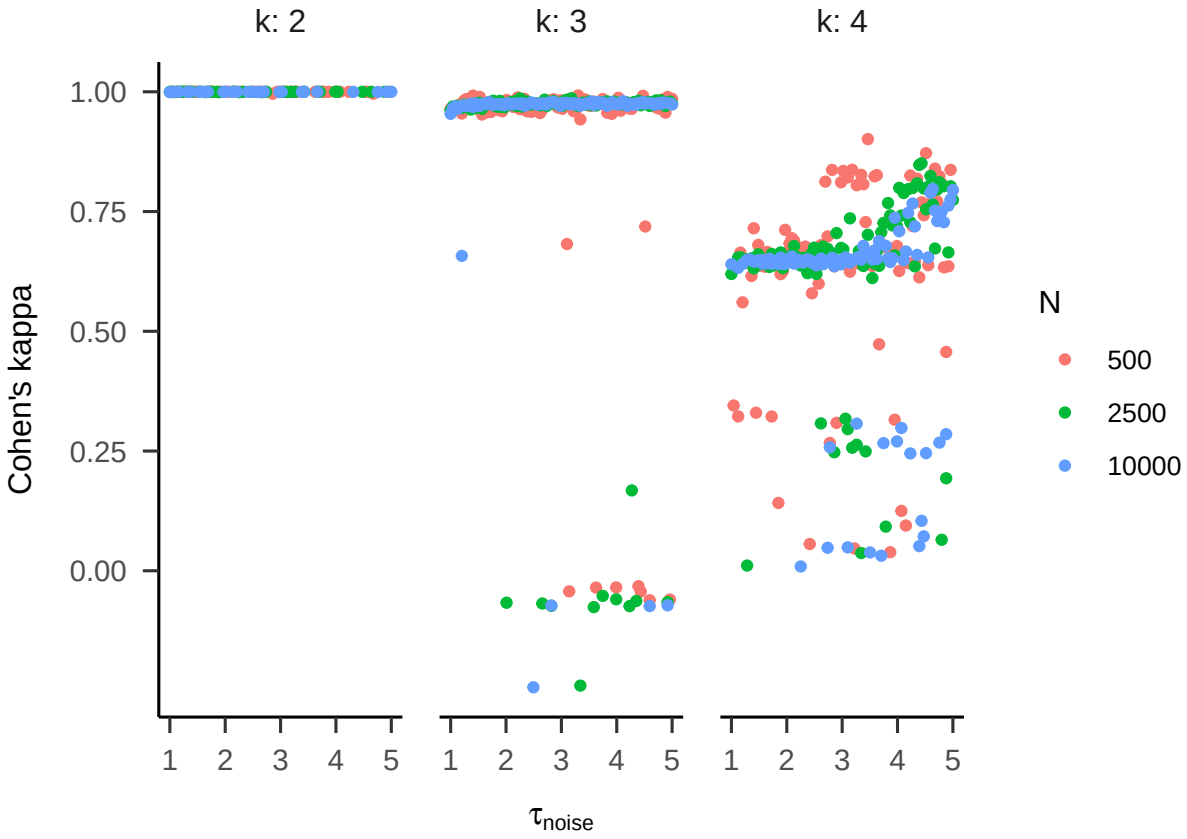

Figure 16. Cohen's kappa depending on the variation of noise added to the data generated by the HMM. Colours indicate different sizes of generated data. Top facet labels indicate the number of states in the HMM.

of missing data intervals and the decrease became linearly steeper when more intervals were included. For two and three state models, kappas started at almost one and decreased to 0.6 for five missing intervals, indicating substantial to almost perfect accuracy. For the four-state model, kappas started around 0.85 and decreased to 0.5 for five missing intervals, suggesting moderate to substantial accuracy. As in previous parts, a several kappa values clustered around zero and -0.33 for the three-state model and around 0.6, 0.25, and zero for the four-state model. In total, missing data did not affect the parameter recovery but linearly decreased the accuracy of the model from an nearly perfect to a moderate extent.

With a few exceptions, the simulation study revealed that the HMM recovered

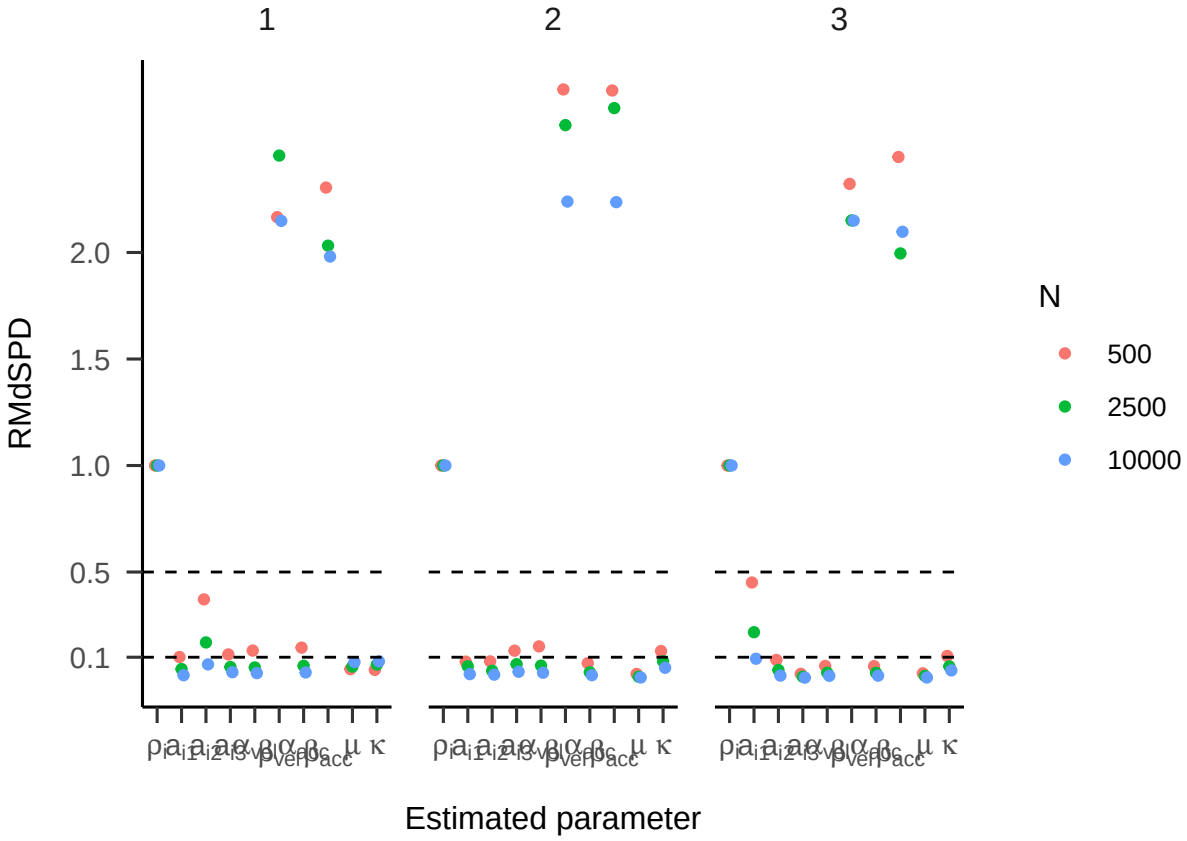

Figure 17. RMdSPD between true and estimated parameters of the three-state HMM in part two of the simulation. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

parameters well and accurately estimated the true state sequences. The most critical decrease in recovery occured when noise was added to the data generated by models including smooth pursuit. In the noise condition, scale parameters of gamma distributions were often badly recovered. Higher sample sizes slightly improved parameter recovery, but neither variation in starting values nor missing data affected it. In contrast, the accuracy of the model was linearly descreasing with more data missing but not influenced by manipulating parameters, noise, or starting value variation. Adding more states to the HMM generally decreased the parameter recovery and the accuracy.

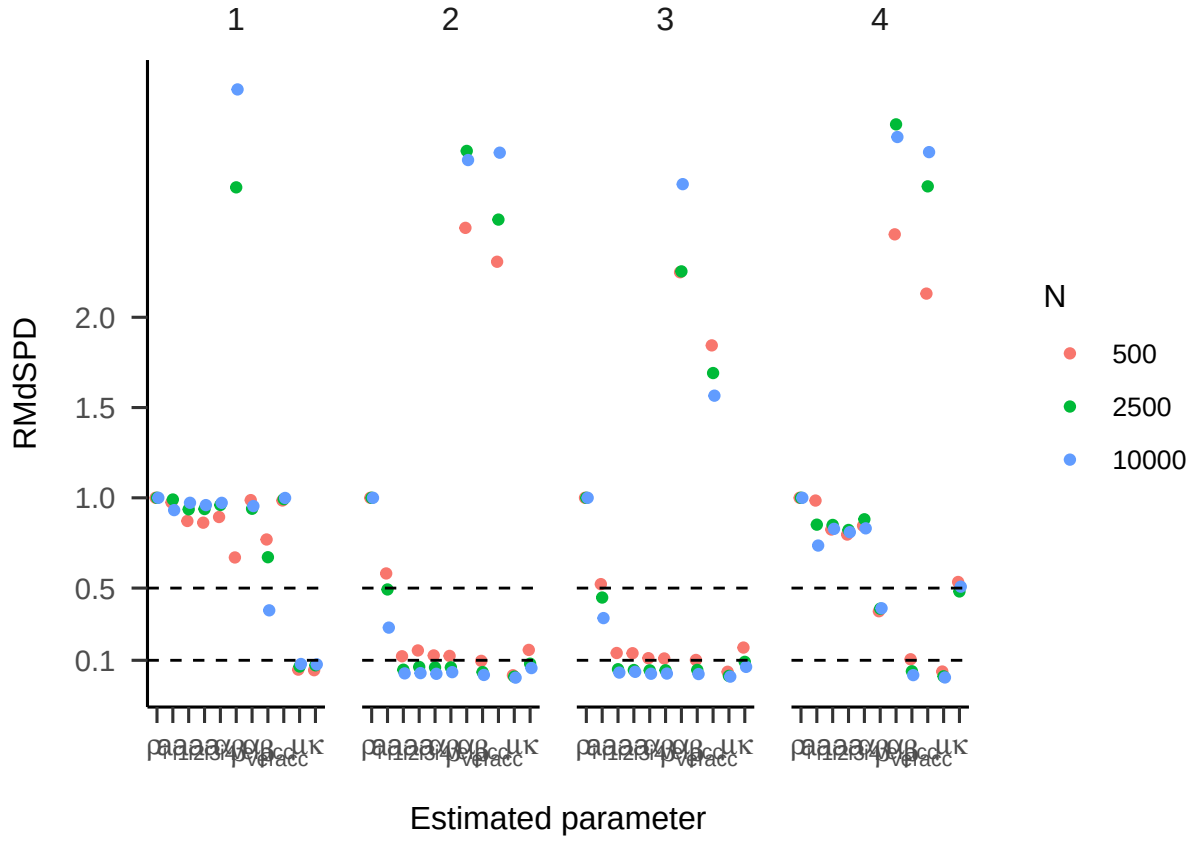

Figure 18. RMdSPD between true and estimated parameters of the three-state HMM in part two of the simulation. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

Validation

Model comparison. To test whether HMMs are accurately describing eye movements, I applied them with one, two, three, four, and five states to two benchmark data sets and compared their Schwarz weights. 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 condition, I predicted the highest weights for the four-state models. Figure X shows the Schwarz weights for different subjects and models in the image condition. For all subjects, the five-state model displays the highest weight, suggesting that it is the most likely to have generated the data. In the moving dots condition, the

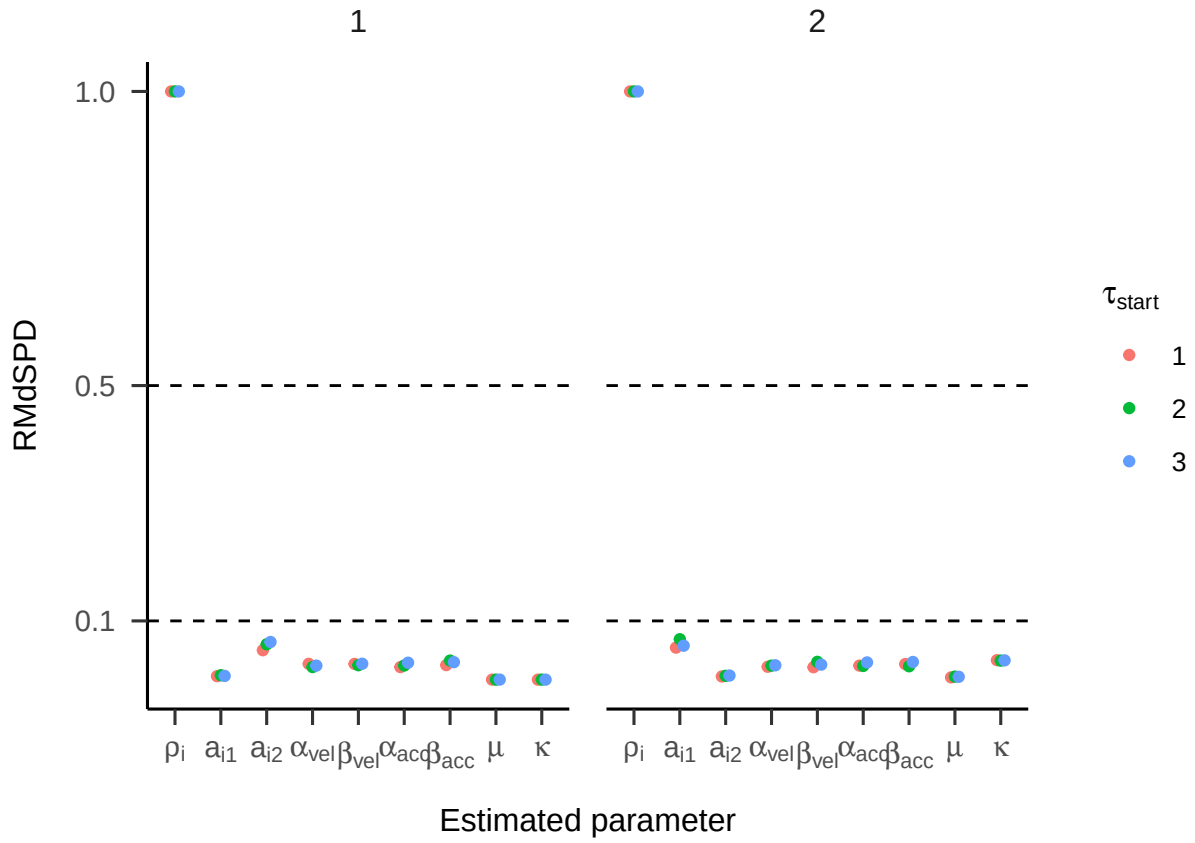

Figure 19. RMdSPD between true and estimated parameters of the two-state HMM in part three of the simulation. Colours indicate the variation in starting values used to estimate parameters. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

five-state model has the highest weights for most subjects, but the one-, three, and four-state models exhibit the highest weights for two subjects each (see Figure X). The video condition has the same pattern as the image condition, as the five-state model consistently has the highest weights (see Figure X). In contrast to my hypothesis, applying different HMMs to the Andersson et al. 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 likelihoods of the models.

Ehinger data set

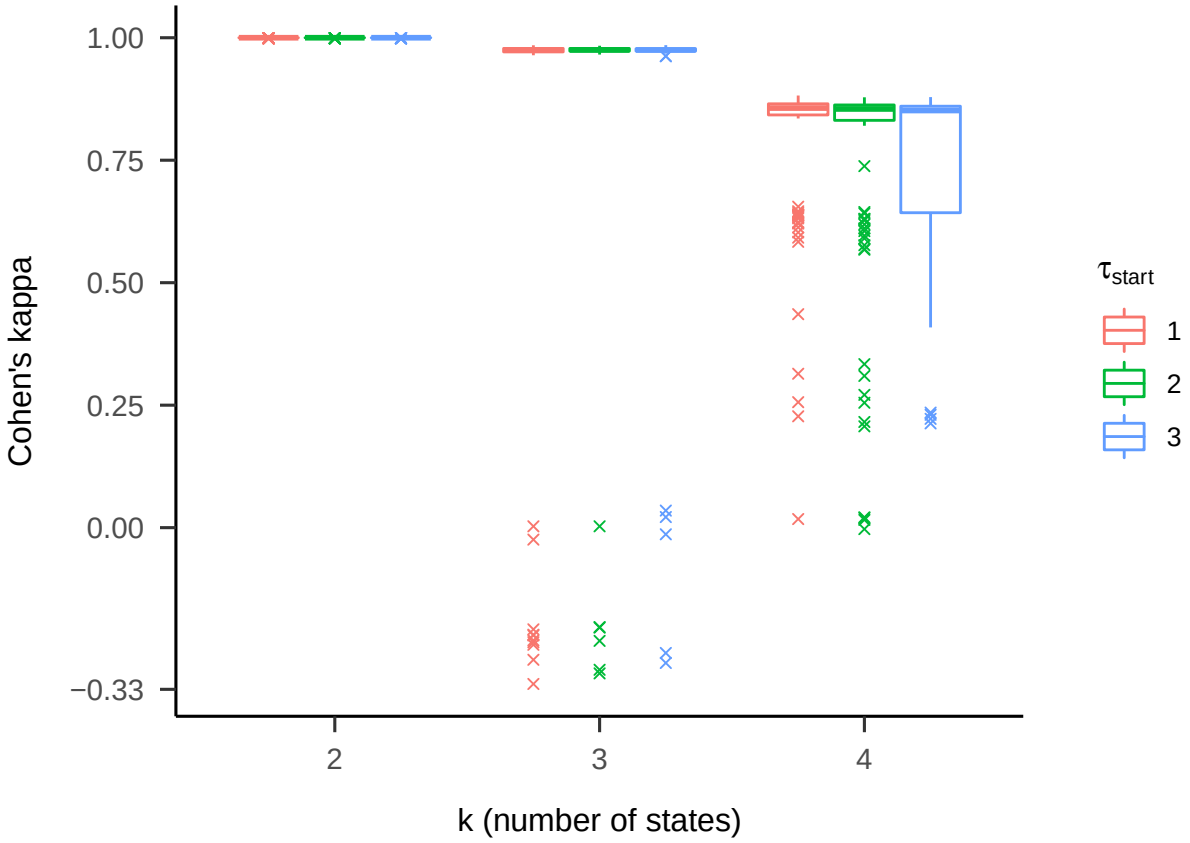

Figure 20. Boxplots displaying Cohen's kappa depending on the number of states in the HMM in part three. Colours indicate the variation in starting values used to estimate parameters. Solid vertical lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

Comparison to other algorithms. The hypothesis that HMMs improve the classification performance of eye movement events was investigated by comparing gazeHMM against other algorithms. Therefore, I applied gazeHMM with four events to the Andersson et al. (2017) data set. First, I calculated the RMSD[3] as described in the original

[3] In my proposal, I declared to compute the RMSE instead of RMSD. However, since I conducted an in sample comparison, the term deviation is more appropriate. Andersson et al. (2017) also referred to it as RMSD.

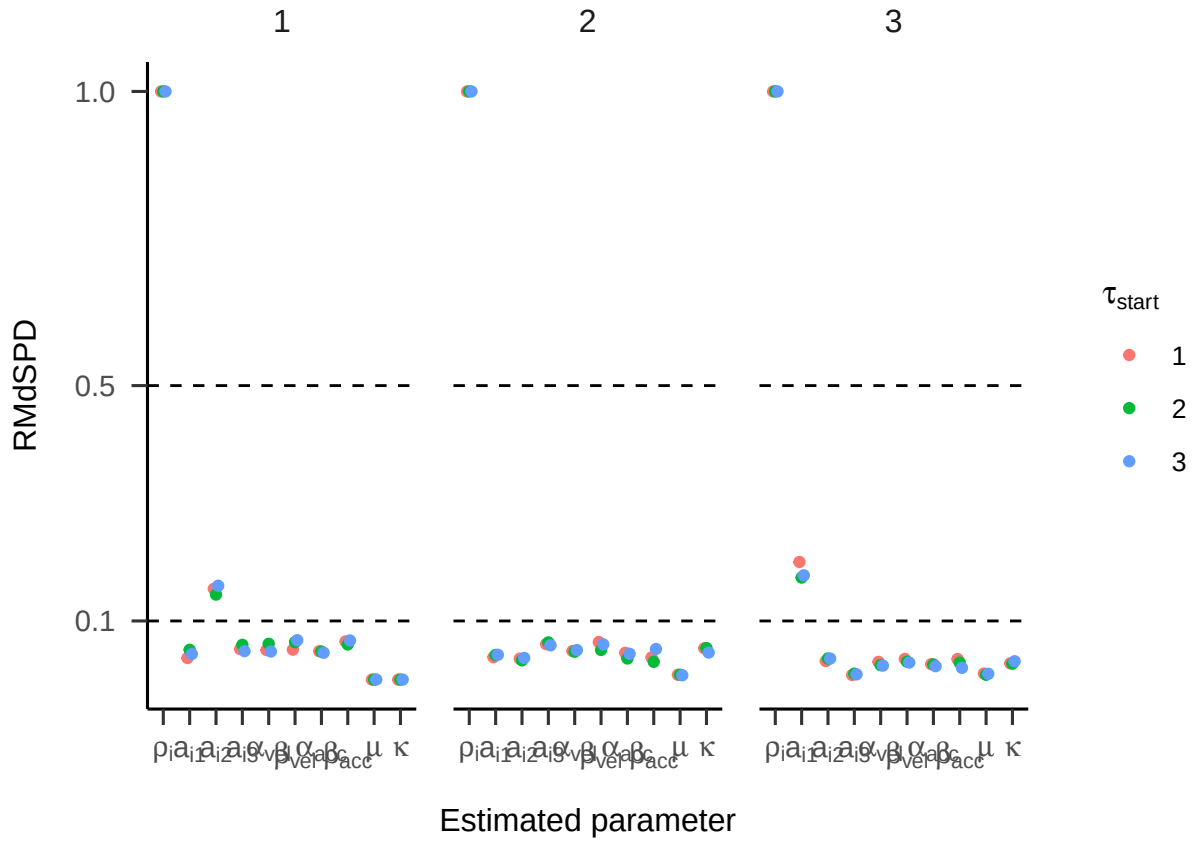

Figure 21. RMdSPD between true and estimated parameters of the three-state HMM in part three of the simulation. Colours indicate the variation in starting values used to estimate parameters. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

article by Andersson et al. for all algorithms included in the study plus gazeHMM. Table 4 shows that in all three conditions, gazeHMM has a higher RMSD than all the the other algorithms for classified fixations due to a large amount of very short fixations. For saccades, gazeHMM has the fourth lowest RMSD of the algorithms in the image and video conditions but the second highest in the moving dots condition (see Table 5). Here, gazeHMM overestimated the number and duration of saccades (either slightly or substantially). Moreover, gazeHMM underestimated the number and duration of PSOs in the image and moving dots conditions, leading to the highest and intermediate RMSD

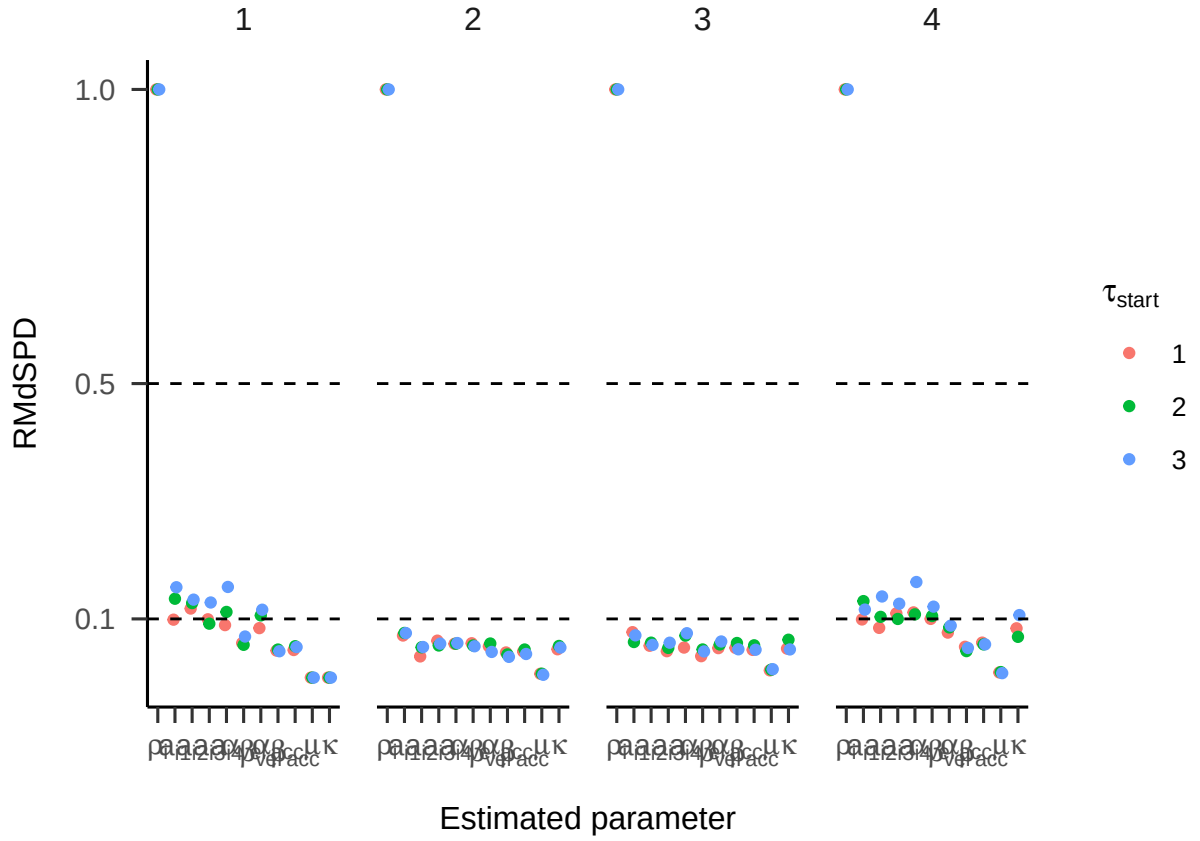

Figure 22. RMdSPD between true and estimated parameters of the four-state HMM in part three of the simulation. Colours indicate the variation in starting values used to estimate parameters. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

among the competitors, repsectively (see Table 6). In the video condition the duration was slightly overestimated but the duration was underestimated with the highest RMSD. No other algorithm was designed to classify smooth pursuits, but gazeHMM classified a large number of short smooth pursuits, causing a substantially higher RMSD than among the human coders (see Figure 7). In sum, except for saccades and PSOs, gazeHMM performed worse with regard to the duration of classified events compared to the other algorithms.

Second, I compared gazeHMM to the other algorithms by using Cohen's kappa as a measure of agreement between algorithms and human coders (see Table 8). For fixations,

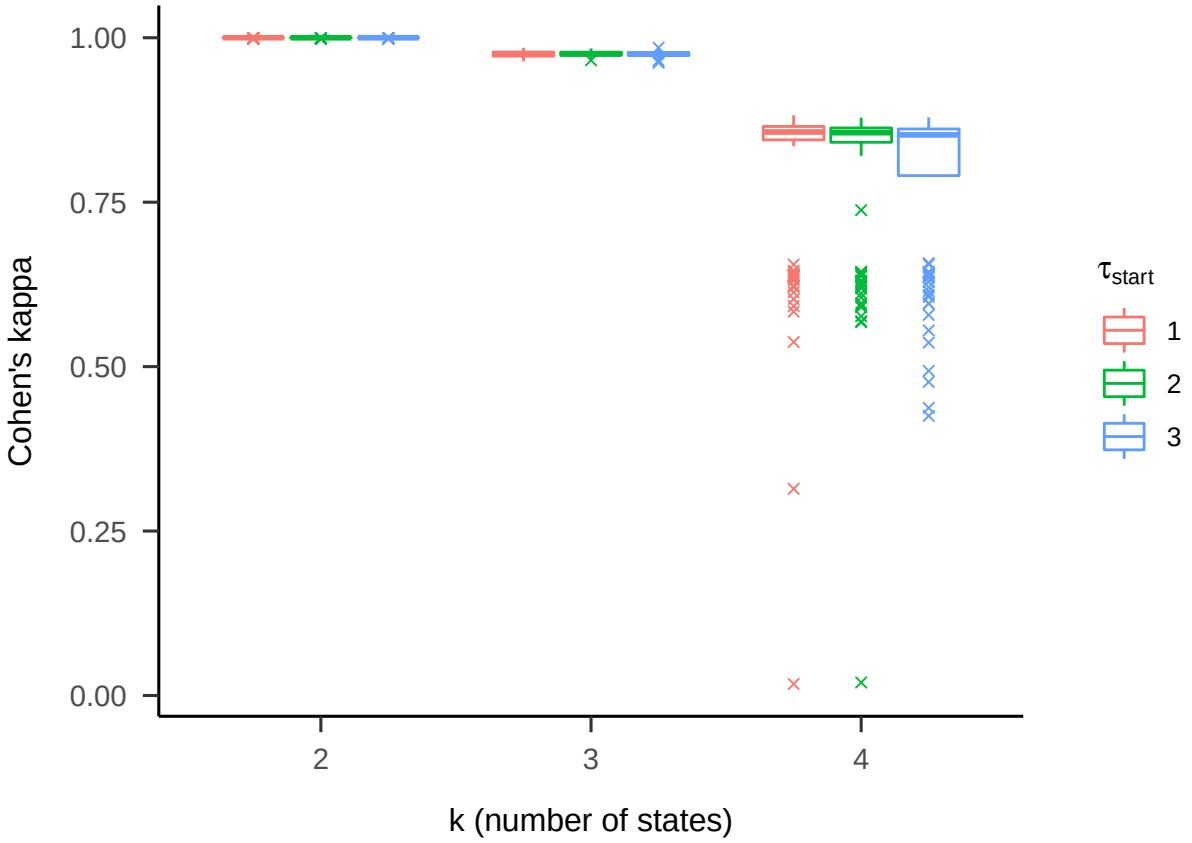

Figure 23. Boxplots displaying Cohen's kappa depending on the number of states in the HMM in part three after state labels were switched post-hoc (exploratory analysis). Colours indicate the variation in starting values used to estimate parameters. Solid vertical lines symbolize medians and hinges the first and third quartile. Whiskers range from hinges to lowest/highest value within 1.5 times the IQR. Crosses represent outliers.

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 hightest in the moving dots and video conditions. In contrast, gazeHMM showed absolute kappa values for saccades that correspond to a moderate to substantial agreement. In all three conditions, the agreement was intermediate in the moving dots condition and higher than for most other algorithms in the image and video conditions. For PSOs, absolute kappa values showed slight to fair

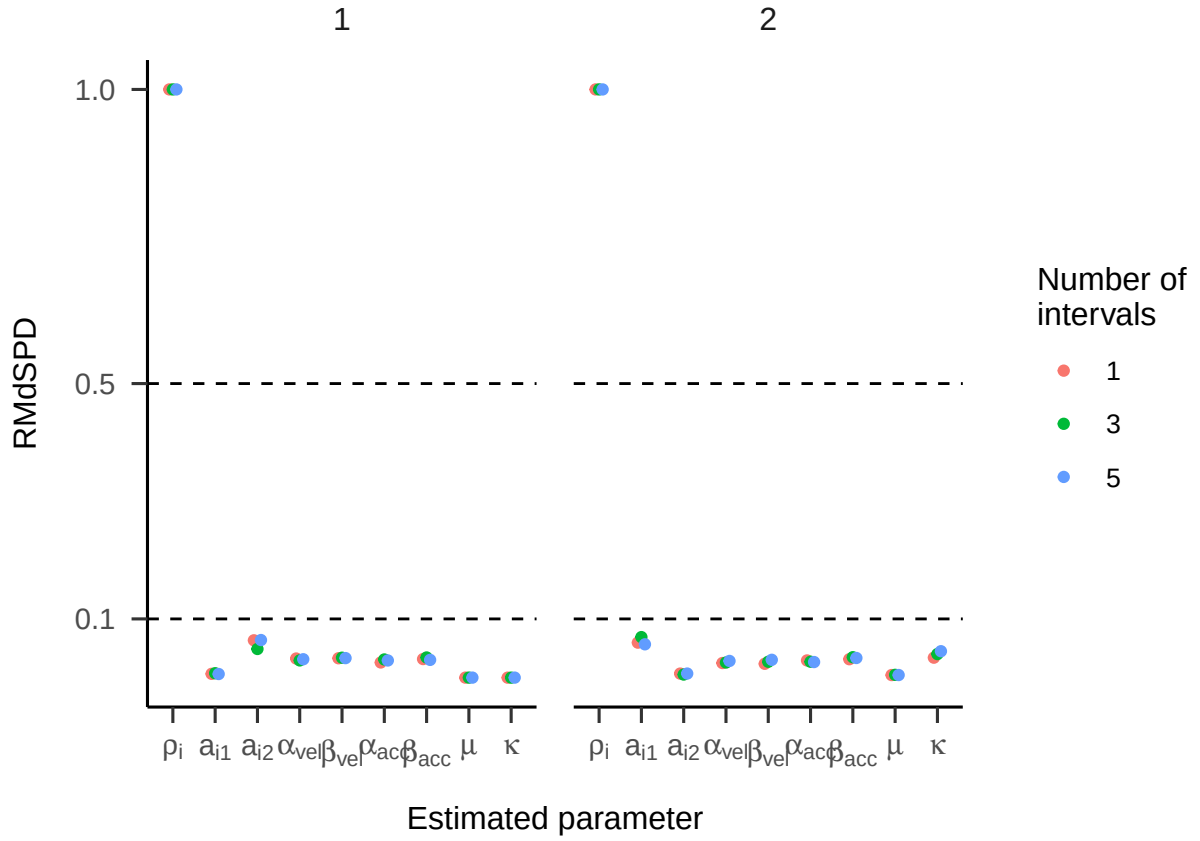

Figure 24. RMdSPD between true and estimated parameters of the two-state HMM in part four of the simulation. Colours indicate the number of missing data intervals. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

agreement agreement to human coders. It was lower in the moving dots condition but intermediate in the image and video conditions compared to the other algorithms. The agreement for smooth puruits was moderate in the moving dots condition but slight in the image and video conditions. No other algorithm in the study was designed to detect smooth pursuits. In sum, gazeHMM classified samples as saccades more similar to human coders than most other algorithms. For the other events, the classification differed from human coding more than for most other algorithms.

Third, I computed the disagreement ratio as the percentage of samples that were

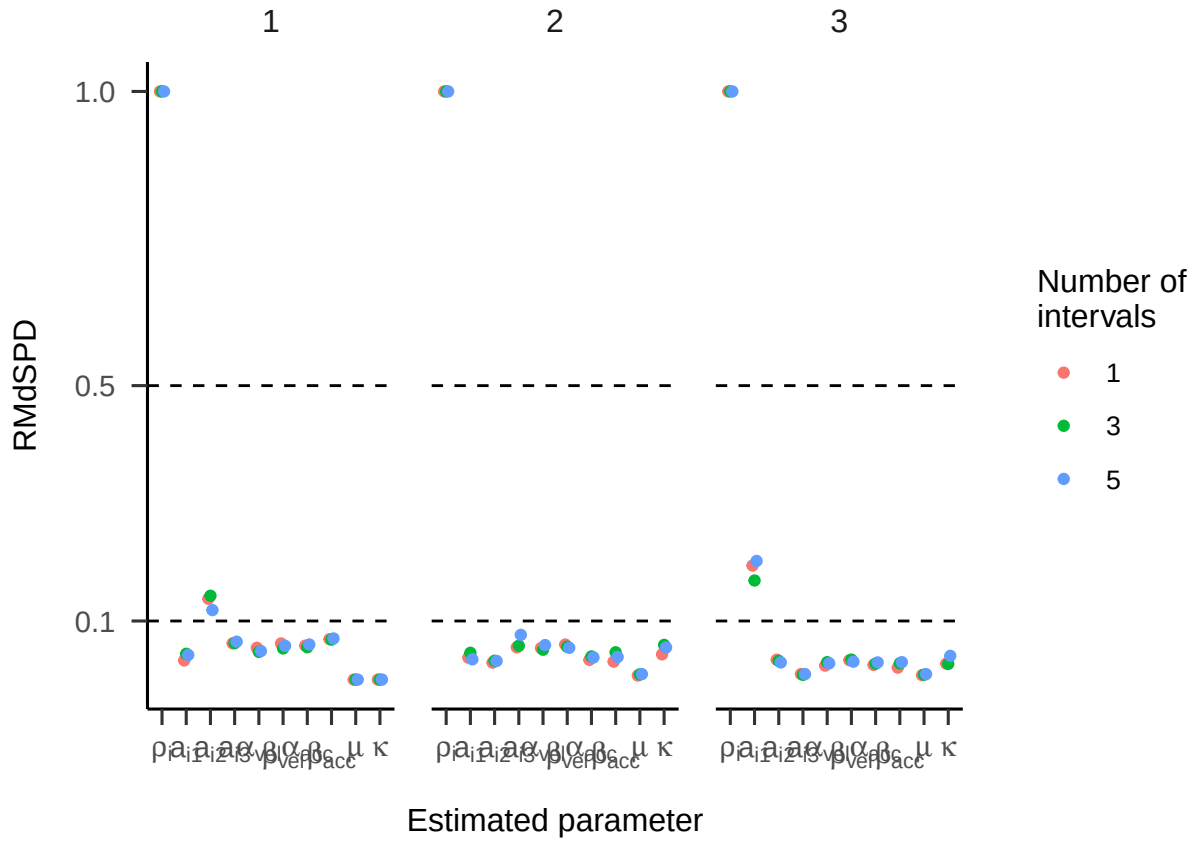

Figure 25. RMdSPD between true and estimated parameters of the three-state HMM in part four of the simulation. Colours indicate the number of missing data intervals. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

classified differently by gazeHMM compared to the human coders. Figure 28 shows that in the image condition, gazeHMM with four events has a higher disagreement ratio than all but two of the other algorithms. In the moving dots and video conditions, the disagreement ratios are considerably lower than for all the other algorithms. Notably, gazeHMM's disagreement ratios were similar across conditions (around 45%).

(ref:plot.disag) Disagreement ratio between algorithms and human coders for different conditions. gazeHMM-3 classified three and gazeHMM-4 classified four events. Figure adapted from (**???**).

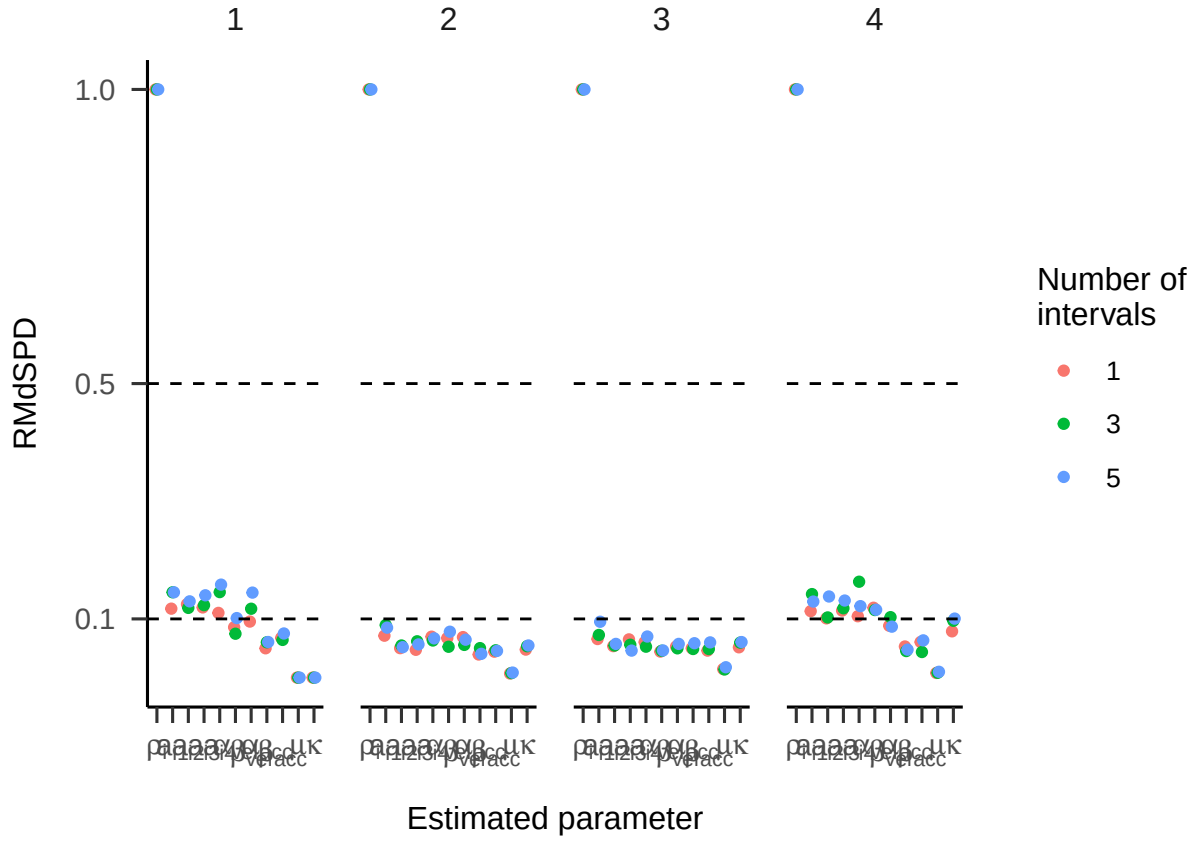

Figure 26. RMdSPD between true and estimated parameters of the four-state HMM in part four of the simulation. Colours indicate the number of missing data intervals. Labels on the x-axis indicate for which estimated parameter the RMdSPD is displayed. Top facet labels denote to which state estimated parameters belong.

Contrary to my second hypothesis, applying gazeHMM to a benchmark data set revealed that it did not outperform all other algorithms with regard to RMSD, sample-to-sample agreement, or disagreement ratio. It rather fared better or worse than most algorithms for different events and conditions. Especially for saccades, the agreement was relatively high and improved upon the IHMM algorithm included in the study.

Even though it was not confirmed by the model comparison, applying gazeHMM with three states to static data might be more appropriate and yield better validation results. Table X shows the RMSD between gazeHMM and human coders only for the image

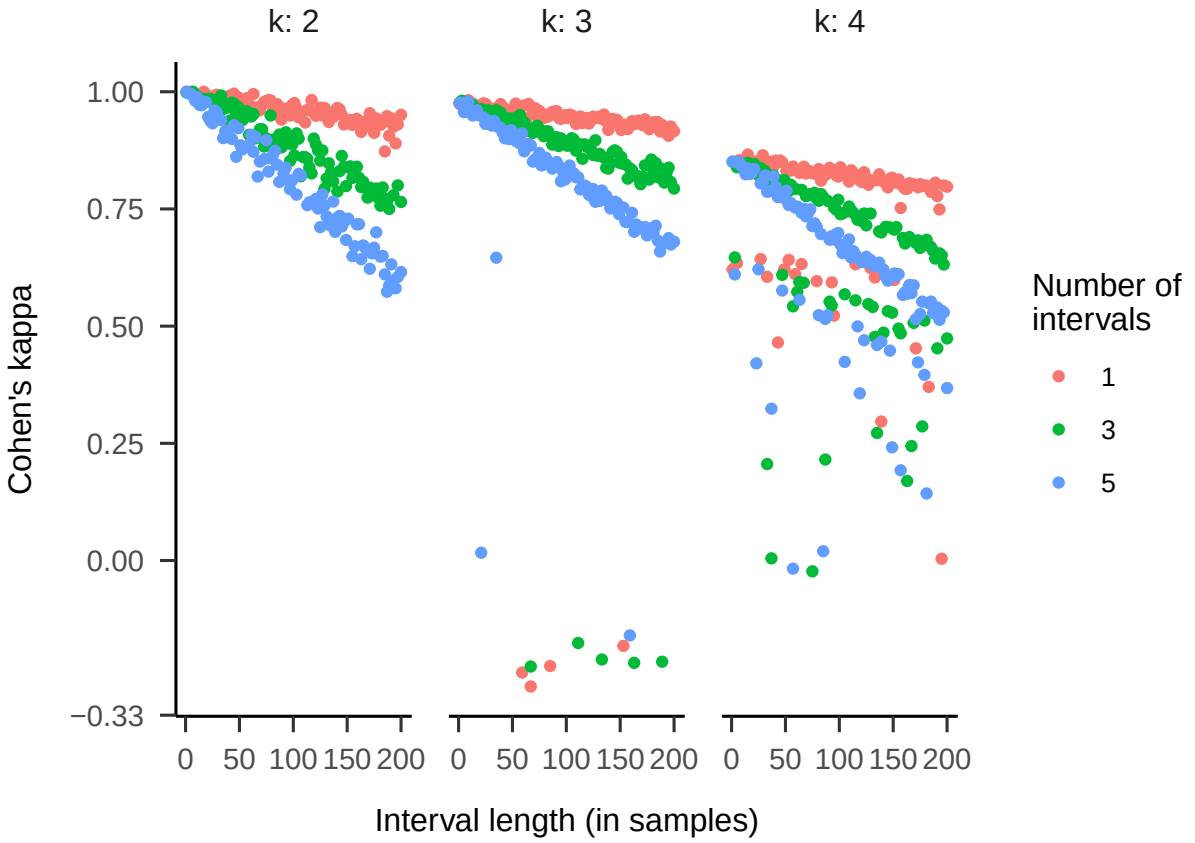

Figure 27. Cohen's kappa depending on the length of missing data intervals. Colours indicate the number of missing data intervals. Top facet labels indicate the number of states in the HMM.

condition. For fixations, gazeHMM shows a comparably low RMSD and only two other algorithms have lower values. Oppositely, saccades had the highest RMSD among the compared algorithms. The same result was found for PSOs.

The agreement between gazeHMM with three states and the human coders measured by Cohen's kappa is displayed in Table **??** (only for the image condition). For fixations, the absolute kappa indicated substantial agreement and was 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 intermediate with regard to the other algorithms.

Table 4

Mean and standard deviations of fixation durations, number of fixations, and RMSD between fixation durations classified by algorithms and human coders

	Moving dots				Image				Video			
Algorithm	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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

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

Figure 28 illustrates the disagreement ratio between gazeHMM with three events and the human coders compared to the other algorithms. In the image condition, the disagreement ratio was slightly lower for gazeHMM than for all the other algorithms. Summing up, using gazeHMM with three states increased the agreement to human coding with regard to fixation durations and samples classified as fixations. However, for saccades and PSOs the agreement decreased.

Technical Appendix

Table 5

Mean and standard deviations of saccade durations, number of saccades, and RMSD between saccade durations classified by algorithms and human coders

	Moving dots				Image				Video			
Algorithm	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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

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

Table 6

Mean and standard deviations of PSO durations, number of PSOs, and RMSD between PSO durations classified by algorithms and human coders

	Moving dots				Image				Video			
Algorithm	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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

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

Table 7

Mean and standard deviations of smooth pursuit durations, number of smooth pursuits, and RMSD between smooth pursuit durations classified by algorithms and human coders

	Moving dots				Image				Video			
Algorithm	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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
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	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

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

Table 8

Cohen's kappa between human coders and algorithms for different conditions and events

	Fixations			Saccades			PSOs			Smooth pursuit		
Algorithm	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	0.00	0.00	0.00
coderRA	0.92	0.84	0.82	0.95	0.91	0.94	0.88	0.80	0.81	0.00	0.00	0.00
gazeHMM	0.28	0.24	0.18	0.68	0.58	0.72	0.33	0.14	0.40	0.01	0.56	0.21
CDT	0.38	0.06	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
EM	0.00	0.00	0.00	0.64	0.66	0.67	0.00	0.00	0.00	0.00	0.00	0.00
IDT	0.36	0.00	0.03	0.45	0.26	0.38	0.00	0.00	0.00	0.00	0.00	0.00
IKF	0.63	0.03	0.14	0.58	0.46	0.59	0.00	0.00	0.00	0.00	0.00	0.00
IMST	0.38	0.00	0.03	0.54	0.30	0.52	0.00	0.00	0.00	0.00	0.00	0.00
IHMM	0.67	0.03	0.13	0.69	0.60	0.71	0.00	0.00	0.00	0.00	0.00	0.00
IVT	0.67	0.03	0.13	0.75	0.63	0.76	0.00	0.00	0.00	0.00	0.00	0.00
NH	0.52	0.00	0.01	0.67	0.60	0.68	0.24	0.20	0.25	0.00	0.00	0.00
BIT	0.67	0.03	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LNS	0.00	0.00	0.00	0.81	0.75	0.81	0.64	0.59	0.63	0.00	0.00	0.00

Note. Negative values were set to zero. gazeHMM classified four events. Table adapted from (**???**).

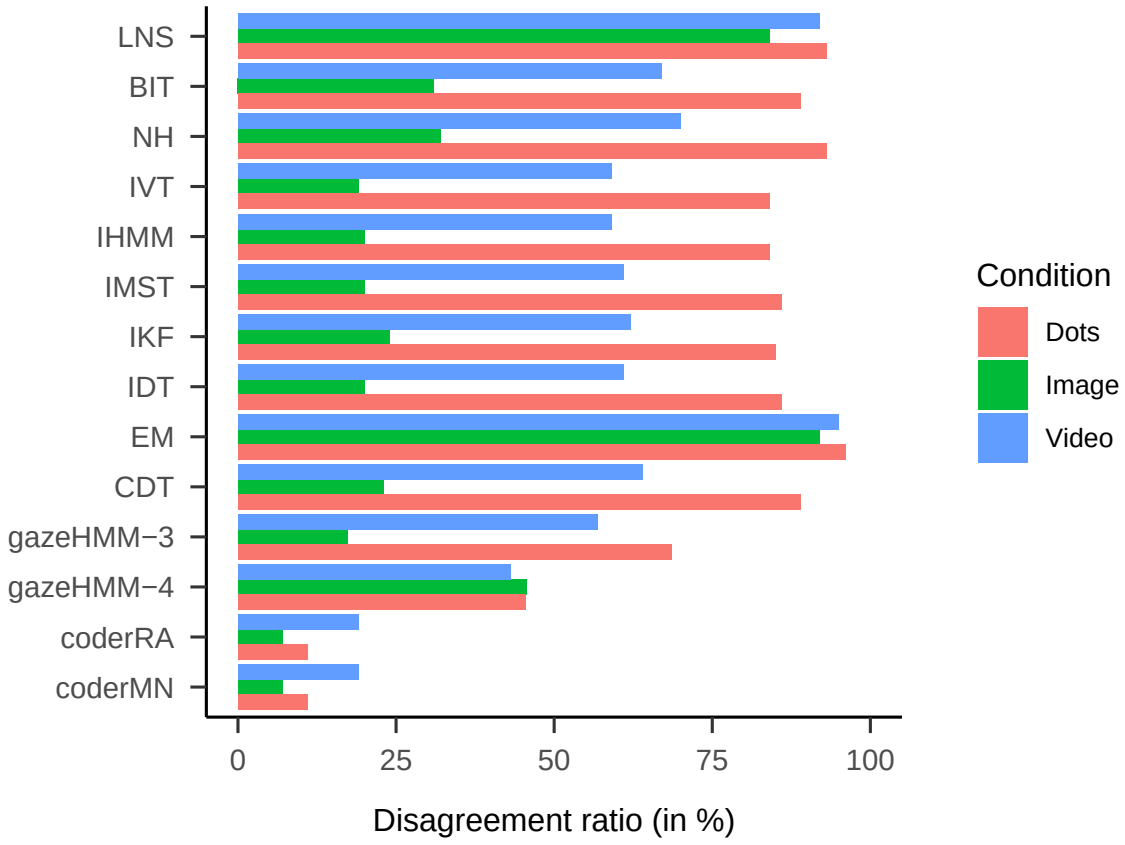

Figure 28. (ref:plot.disag)

Table 9

Confusion matrix between gazeHMM (rows) and human coders (columns) for different events and 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.

Table 10

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	0.50	0.26
CDT	0.38	0.00	0.00
EM	0.00	0.64	0.00
IDT	0.36	0.45	0.00
IKF	0.63	0.58	0.00
IMST	0.38	0.54	0.00
IHMM	0.67	0.69	0.00
IVT	0.67	0.75	0.00
NH	0.52	0.67	0.24
BIT	0.67	0.00	0.00
LNS	0.00	0.81	0.64

Note. Negative values were set to zero. gazeHMM-3 classified three and gazeHMM-4 classified four events. Table adapted from (**???**).