The Diffusion Model in Stan: Implementing the Seven-Parameter, Truncated, and Censored Models With an Application to First-Person Shooter Task Data

Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der Wirtschafts- und Verhaltenswissenschaftlichen Fakultät
an der Albert-Ludwigs-Universität Freiburg i. Br.

vorgelegt von
Franziska Henrich
geboren am 21.06.1996 in Frankfurt am Main

WS 2025/26

Dekanin der Wirtschafts- und Verhaltenswissenschaftlichen Fakultät:
Prof. Dr. Eva Lütkebohmert-Holtz

Gutachter:
Prof. Dr. Karl Christioph Klauer
Prof. Dr. Andrea Kiesel

Datum des Promotionsbeschlusses: 09. Januar 2026

To my parents.

Für meine Eltern.

Contents

List of Abbreviations

CCDF	Complementary Cumulative Distribution Function
CDF	Cumulative Distribution Function
DM	Diffusion Model
FPST	First-Person Shooter Task
HDDM	Hierarchical Drift Diffusion Model
PDF	Probability Density Function
RT	Reaction Time
SBC	Simulation Based Calibration Study
WIT	Weapon Identification Task

Summary

This cumulative dissertation addresses the technical enhancement of the diffusion model within the probabilistic programming language Stan. Two core advancements are presented: first, the extension beyond the conventional four-parameter model variant, enabling the use of models incorporating up to seven parameters; and second, the adequate modeling of truncated and censored data.

These implementations are designed to overcome limitations inherent in the four-parameter model and to enable more nuanced analyses of substantive research questions. Specifically, the existing model is theoretically constrained, as it cannot account for certain patterns of response times - most notably fast and slow errors - which the seven-parameter formulation is able to capture. Furthermore, neither the standard four- nor seven-parameter models are equipped to accommodate truncated response time distributions, which arise for example in the context of response windows.

The aims of this work are threefold: (a) to elucidate the new functionalities and demonstrate their application through detailed code examples; (b) to rigorously test and validate the new implementations; and (c) to illustrate, via a practical example using real-world data, how censored and truncated data may be effectively modeled. In this context, research questions concerning model quality and applicability are systematically addressed.

A range of methodological approaches is employed to evaluate the implementations. Both new functionalities are assessed through recovery studies and simulation-based calibration studies. In addition, the seven-parameter functions are benchmarked against an established tool, HDDM. For the modeling of truncated and censored data, an empirical study utilizing existing datasets replicates previous findings while additionally demonstrating the novel analytical methods.

The simulation-based calibration and recovery studies confirm that the code has been accurately implemented, yielding satisfactory to excellent parameter recovery. The comparative analysis demonstrates that, in terms of parameter recovery, the new implementation performs competitively with HDDM. With respect to runtime, the four-parameter model is markedly faster with the new approach, whereas the seven-parameter model is substantially slower. The application study provides valuable insights into the behavior of various model specifications and illustrates that censored and truncated models can be successfully applied to real data.

As a result, researchers are now equipped with an extended analytical instrument that supports the practical implementation of models with up to seven parameters as well as the modeling of truncated and censored data. This dissertation thereby expands the methodological toolbox available for empirical research, furnishing researchers with enhanced techniques for the statistical analysis of complex decision processes.

Zusammenfassung

Diese kumulative Dissertation widmet sich der programmiertechnischen Erweiterung des Diffusionsmodells innerhalb der probabilistischen Programmiersprache Stan. Ziel ist die Bereitstellung zweier neuer Funktionalitäten: Erstens die Möglichkeit, neben der bisherigen vier parametrigen Modellvariante nun auch Modelle mit bis zu sieben Parametern zu verwenden, und zweitens die adäquate Modellierung trunkierter und zensierter Daten.

Die neuen Implementierungen dienen dazu, Schwächen des vier parametrigen Modells zu überwinden und inhaltliche Forschungsfragen präziser analysieren zu können. Konkret weist das bisherige Modell theoretische Grenzen auf, da es bestimmte Reaktionszeitmuster - insbesondere schnelle und langsame Fehler - nicht erklären kann, was das sieben parametrige Modell jedoch ermöglicht. Darüber hinaus sind die "klassischen" Modelle, seien es vier oder sieben parametrige, nicht in der Lage, abgeschnittenen Reaktionszeitverteilungen abzubilden, wie sie beispielsweise durch Antwortzeitfenster entstehen.

Die Ziele dieser Arbeit bestehen darin, (a) die neuen Funktionalitäten zu erklären und ihre Anwendung anhand zahlreicher Codebeipiele zu demonstrieren, (b) die Implementierungen zu testen und zu validieren sowie (c) anhand eines praxisnahen Anwendungsbeispiels zu zeigen, wie sich trunkierte und zensierte Daten modellieren lassen. Dabei werden Fragen zur Modellgüte und Anwendbarkeit eingehend behandelt.

Zur Überprüfung der Implementierungen werden verschiedene Methoden eingesetzt. Für beide neuen Funktionalitäten werden eine Wiederfindungsstudie sowie eine simulationsbasierte Kalibrierungsstudie durchgeführt. Für die sieben parametrige Modellvariante erfolgt zusätzlich ein Vergleich mit einem bestehenden ähnlichen Programm - HDDM. Für die Funktionen zur Modellierung trunkierter und zensierter Daten wird darüber hinaus eine Studie mit bereits vorhandenen Datensätzen durchgeführt, um einerseits bestehende Ergebnisse zu replizieren und andererseits die neuen Analysemethoden zu veranschaulichen.

Die Studien zur simulationsbasierten Kalibrierung und zur Wiederfindung zeigen, dass der Code korrekt implementiert ist und vorgegebene Parameterwerte zufriedenstellend bis sehr gut wiedergefunden werden. Die Vergleichsstudie verdeutlicht, dass die neue Implementierung hinsichtlich der Wiederfindung der Parameter mit HDDM konkurrieren kann. In Bezug auf die Laufzeit zeigt sich, dass das vier parametrige Modell mit der neuen Methode deutlich schneller berechnet werden kann, während das sieben parametrige Modell hingegen erheblich langsamer ist. Die Anwendungsstudie liefert aufschlussreiche Einblicke in das Verhalten der unterschiedlichen Modellspezifikationen und zeigt, dass die trunkierten und zensierten Modelle erfolgreich an realen Daten angewendet werden können.

Damit steht nun ein erweitertes analytisches Instrumentarium bereit, das die Anwendung sowohl von Modellen mit bis zu sieben Parametern als auch von trunkierten und zensierten Modellen in der Praxis ermöglicht. Diese Arbeit trägt somit dazu bei, den methodischen Werkzeugkasten der empirischen Forschung zu erweitern und Forschenden verbesserte Verfahren für die statistische Analyse komplexer Entscheidungsprozesse zur Verfügung zu stellen.

Chapter 1

Introduction

New York, 1999. It was a silent night when the 22-year old black Amadou Diallo went home to his apartment in the Wheeler Avenue. At the same time, four white police officers patrolled the streets of the Bronx in their civil car. They were searching for a serial criminal. When they saw Diallo, they confused him with the one they were looking for. They got out of their car with their guns raised. The officers asked Diallo to raise his hands. When they said they were police men, Diallo ran into the dark hall of his apartment house and put his hand into his jacket. Believing that he would get out a gun, one police officer fired a shot. But Diallo was unarmed. He probably wanted to get the ID from his wallet to identify himself. When one police officer stumbled and fell down the staircase in front of the house, the others thought that Diallo must have shot at him and they opened the shooting. Eventually, Amadou Diallo was shot with 41 bullets and died at the same place within minutes (Tagesspiegel, 2000).

The case of Amadou Diallo is not an individual case. Many other, similar incidents happened in the following years. Recent cases are for example the cases of George Floyd who died in a brutal police operation in Minneapolis in 2020, or Dexter Reed, who was shot with 96 shots in a police control in Chicago in 2024 (Tagesschau, 2024). Also in Germany, there are such cases known. In 2025, the 21-year old Lorenz A. was shot with three shots from behind in Oldenburg (Amadeu Antonio Stiftung, 2025). All cases have in common that the victims are of black skin color and the police officers are of white skin color.

These cases regularly revive the discussions on police violence. Weeks of protest and calls for civil disobedience followed the case of Amadou Diallo. But reactions were not only seen in the population. Also researchers, especially psychologists, showed interest in the case and started investigations on the so called *racial bias*. Questions like “Would the police officers also have shot so quickly if the other person were of white skin color?”, “Would the police officers also have shot if they were of black skin color themselves?”, or “Would the police officers have shot if they were better trained on the matters of skin color and prejudice?” moved into the focus of prejudice research in the early years of 2000.

Two main paradigms evolved in that time in the research field of cognitive and social psychology: The Weapon Identification Task (Payne, 2001), and the First-Person Shooter Task (Correll et al., 2002), also known as The Police Officer’s Dilemma. Both paradigms aim to investigate the underlying cognitive mechanisms in order to better understand the decision process and to be able to develop measures to prevent racial injustice.

Chapter 2

Theoretical Background

2.1 The Shooter Task Paradigms

The shooter task paradigms were developed in 2001 by Payne and 2002 by Correll et al.. Both paradigms aim to investigate the influence of the skin color on the reaction time and the error rate in the task of detecting weapons in an image. The paradigms are operationalized as computer experiments.

The *Weapon Identification Task* (WIT) is an experiment in the family of sequential priming procedures. Participants are first quickly shown an image of a face that depicts either a Black man or a White man (*prime*). Immediately afterwards, a second image is presented (*target*), showing either a weapon (*gun*) or a harmless object (*tool*). The participant has to press a left button or a right button to identify the object as gun or tool, respectively. This is to be done as fast as possible and independently from the prime that was shown only for a few milliseconds. The task measures the reaction time and the error rate.

In the *First-Person Shooter Task* (FPST), participants are shown images of a person that is Black or White (*prime*) and holds an object in its hand, either a gun or a tool (*target*). Participants are asked to press a "shoot"-button when there is a gun in the image and to press a "don't shoot"-button when there is a tool in the image. In contrast to the WIT, the prime is shown in parallel with the target in this experimental setup. Participants are also asked to respond as fast as possible. Reaction times and error rates are measured.

One central finding in both paradigms is that a harmless object is more often mistaken for a weapon when the person is Black than when the person is White. Moreover, participants are faster to correctly detect a weapon when the person is Black than when the person is White (e.g., Payne, 2001, 2006). That the skin color influences the results of the decision process is called *racial bias*. In both experiments, racial bias is measured in the error rates, also called *accuracy data*, as well as in the *reaction time data*.

There have been many research teams that used the WIT or the FPST to investigate different research questions and to evaluate the effect of skin color on error rates and reaction times (e.g., see meta-analysis by Rivers, 2017). Interesting findings in the WIT for widely differing research questions are, for example, that racial bias also occurs when the face-images (primes) do not depict men but boys, women, or girls (e.g., Thiem et al., 2019; Todd, Simpson, et al., 2016; Todd, Thiem, & Neel, 2016). Furthermore, the racial bias is found to be larger when

participants are additionally primed with alcohol advertisements (Stepanova et al., 2012), and smaller when the attention is actively directed away from the semantic nature of the prime (Ito & Tomelleri, 2017). Other research teams set their focus on the effects on brain activities and used neural models to find that racial bias also affects the error-related negativity wave recorded with an electroencephalograph (Amodio et al., 2004). Similar findings exist for the FPST. For example, when participants first read newspaper articles about Black (vs. White) criminals and then perform the FPST, racial bias was increased (Mayerl et al., 2019). Vice verca, when participants were trained before the experiment, or when police officers performed the task, less racial bias occurred than for untrained civilians (Johnson et al., 2018).

In some experimental setups, so called *response windows* are included. These force participants to respond as fast as possible within a certain time span. Response windows are used to reveal fast-acting, possibly implicit processes in stereotyping and prejudice. Typically, response windows in the WIT and FPST range from 500 ms to 850 ms (Johnson et al., 2017; Mayerl et al., 2019; Pleskac et al., 2018; Thiem et al., 2019; Todd et al., 2020).

In order to analyze reaction time and accuracy data, many different methods besides the standard ANOVA-analysis are commonly used. Among them are the process dissociation procedure (e.g., applied by Govorun & Payne, 2006), different neuroimaging techniques as the already mentioned neural model (e.g., applied by Amodio et al., 2004), the multinomial processing tree analysis (e.g., applied by Mayerl et al., 2019), or the diffusion model analysis (e.g., applied by Correll et al., 2015; Todd et al., 2020).

The diffusion model analysis is particularly interesting, as it allows one to model the reaction time data and the accuracy data simultaneously, in contrast to most of the other methods which either analyze the reaction time data or the accuracy data. Furthermore, there is a link between the diffusion model parameters and psychological processes underlying the decision process, which makes the interpretation of the results very interesting and tangible.

2.2 The Diffusion Model

The basic four-parameter diffusion model was first introduced by Ratcliff (1978). It belongs to the family of sequential sampling models that are also called information accumulation models. These models give a framework to understand how information is processed and used. They are often applied in cognitive psychology and are used to mathematically describe how a decision maker gathers information or evidence over time until a threshold is met and the decision is made. In the literature, the diffusion model is also referred to as Ratcliff diffusion model, drift diffusion model, or Wiener diffusion model.

Diffusion models are used to understand the decision process of a participant that performed a two-alternative forced-choice task. In such a task, the participant *has* to give an answer and there are only two response alternatives to choose. The standard experimental setup is the following: The participant sits in front of a screen and has two buttons on the desk - the left

button and the right button. A keyboard with two marked keys also works. Then the stimulus appears on the screen and the participant pushes one of the two buttons as fast as possible. Sometimes a fixation cross that centers the view or a prime that is meant to cause distraction appears right before the stimulus. Typical experiments in form of a two-alternative forced-choice task are the shooter task paradigms described above. A detailed review of the diffusion model and the many areas of research to which it has been applied is given by Ratcliff et al. (2016).

2.2.1 The model parameters and their interpretation

In diffusion modeling, it is assumed that participants accumulate evidence towards either of two response alternatives. Hence, the evidence scale is one-dimensional and for each response alternative one boundary is placed on the poles.

A diffusion model predicts the probability to choose one or the other response alternative and models the distributions of response times associated with each alternative. In the basic case, there are four parameters that characterize the diffusion model:

- The *boundary separation*, a, describes the distance between both response boundaries.
- The *relative starting point*, w, describes the position of the point between the two response boundaries where the decision process starts. In other words, it marks the state of evidence before evidence accumulation has started. This point does not necessarily have to lie exactly in the middle between the two boundaries. When the participant expects that one of the options is more likely than the other, the relative starting point shifts towards the associated response boundary.
- The *drift rate*, v, describes the evidence-accumulation rate or average rate of information uptake. During the decision process, participants accumulate decision-relevant information from the environment until one of the boundaries is reached. When a boundary is met, a decision for the associated response option is made.
- The *non-decision time*, t_0, summarizes all time costs for processes that do not belong to the decision process. Such processes comprise perception, encoding, or motor time, among others.

Since the evidence accumulation process itself is influenced by random noise, the process is approximated by a diffusion process. In Figure 2.1, a diffusion process with all four parameters is depicted.

One main advantage of the diffusion model is that the parameters can be interpreted in terms of cognitive processes (e.g., Arnold et al., 2015; Lerche & Voss, 2019; Ratcliff & Rouder, 1998; Voss et al., 2004). Here are some examples of parameter interpretations found by these and several other research teams:

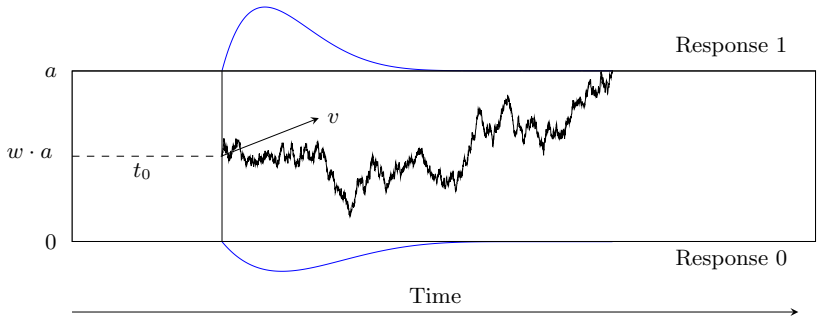


Figure 2.1: Realization of a Four-Parameter Diffusion Process Modeling the Binary Decision Process. *Note.* The parameters are the *boundary separation a* for two response alternatives, the *relative starting point w*, the *drift rate v*, and the *non-decision time* t_0. The decision process is illustrated as a jagged line between the two boundaries. The predicted distributions of the reaction times are depicted as curved lines below and above the response boundaries (blue).

The boundary separation can be interpreted as the participant's individual response style. On the one hand, a response style can be *conservative*. Then the participant has a tendency to prioritize accuracy over speed. This leads to slower responses and a lower rate of errors. A conservative response style influences the boundary separation in the way that it is larger. On the other hand, a response style can be *liberal*. Then the participant has a tendency to prioritize speed over accuracy. This leads to faster responses and a higher rate of errors. A liberal response style influences the boundary separation in the way that it is smaller. Such differences show up when the boundary separations of two different participants with different response styles are compared. Additionally, it is possible to influence the response style within one participant by instructing the participant to be careful and to make no mistakes if possible. Then the boundary separation becomes larger for that participant compared to trials where he was not instructed to be more careful.

The second parameter, the relative starting point can be interpreted as the participant's expectations. One way to influence the expectation is to inform about the proportion of stimuli in the stimulus set. When a participant for the FPST is told that there are more *guns* in the experiment than *tools*, the relative starting point moves towards the decision alternative which is associated to the *guns*-response. In this case it is the "shoot"-response.

Another way to influence the relative starting point is to introduce rewards for one of the responses. As there are four possible outcomes of a reaction (correct *shoot*, incorrect *shoot*, correct *dont't shoot*, incorrect *don't shoot*), the instructor could give different amounts of points for each of the four outcomes and tell the participant to gather as much points as possible - besides the original tasks of being as fast as possible while making as few errors as possible. When one of the four outcomes is rewarded with much more points than the other options, the relative starting point shifts to the response alternative for which the participant is rewarded.

The third parameter, the absolute value of the drift rate, can be influenced by the complexity of the stimuli. This can be seen in the case of the FPST: As described above, the stimulus in

this experiment is a picture that shows a person which holds an object. When the background of the picture is plain and monochrome and the object is relatively large and has high contrast to the background, information uptake is easy. This leads to a large absolute value of the drift rate. When, on the contrary, the background is multicolored and fully packed with information and the object can not be discriminated easily, then information uptake slows down and the absolute value of the drift rate becomes small.

The fourth parameter, the non-decision time, depends, among other things, on the forms of response. Another response form than the one described above may results from the combination of two tasks. One task could be to decide whether a number that will be shown on the screen is *even* or *odd*. The other task could be to decide whether that number is *greater than 42* or *less than 42*. Both tasks have to be done in the same experiment. But which of the two tasks the participant has to perform is only told just before the number appears. Thus, the task is set prior to each trial. These combined tasks are called task-switching tasks and influence the non-decision time in the way that it becomes larger.

Now that all four model parameters are introduced and interpretations and some dependencies are described, such that we have a feeling for the parameters and their characteristics, we continue with a finding from the literature that made it necessary to extend the basic diffusion model and to add three more parameters.

2.2.2 The seven-parameter extension

According to Ratcliff and Rouder (1998), the basic four-parameter model has problems to account for the full range of data in two-alternative forced-choice tasks. This means, there exist two pattern in the data that cannot be reproduced with the four-parameter model.

Before the two problematic patterns will be described in more detail, the notions of *response coding* and *accuracy coding* are introduced. These are two different ways to code the data that are inserted into the diffusion model. So far, we described the *response coding*. This means, the response boundaries correspond to the two response alternatives (*shoot* and *don't shoot* in the FPST). The other option to code the data is the *accuracy coding*. Here, the response boundaries correspond to *correct* and *error* responses. For the following two problems, we switch to Ratcliff and Rouder's perspective and assume that the data is accuracy coded.

Then, the first problematic pattern corresponds to *slow errors*. In this pattern, the weighted mean reaction time of responses that end in an error response is larger than the weighted mean reaction time of responses that end in a correct response. The reaction times are weighted with the probability to end at the according boundary (error/correct). But, one model property is that, when the relative starting point is centered between the response boundaries, the model predicts identical reaction time distributions for correct and error responses. This means, if one would simulate from such a model, one could never obtain a dataset that contains slow errors.

The authors solved this problem by introducing a new parameter: the inter-trial variability in

drift rate. In the four parameter model, all parameters are assumed to be the same for all trials. In this variant of the model, the drift rate parameter is allowed to vary from trial to trial within a certain range that is defined by its inter-trial variability. Then, for a large drift rate, reaction time is short and accuracy is high, whereas for a small drift rate, reaction time is slower and accuracy is lower. In sum, given variability in drift rate, the percentage of slow responses will increase among errors more than among correct responses.

The second problematic pattern corresponds to *fast errors*. In this pattern, the weighted mean reaction time of responses that end in an error response is smaller than the weighted mean reaction time of responses that end in a correct response. Like above, this pattern would never appear if one would simulate from the four-parameter diffusion model with centered starting point.

To solve this problem, the authors introduced another inter-trial variability: the inter-trial variability in the relative starting point. For a relative starting point near the correct response boundary, there will be few errors and they will be slow, because the way is much longer to the error bound than the way to the correct bound. On the contrary, for a starting point near the error response boundary, there will be more errors and they will be fast. In sum, given variability in the relative starting point, the percentage of fast responses will increase among errors more than among correct responses (Forstmann et al., 2016).

Later, Ratcliff and Tuerlinckx (2002) found that there is a third reaction time pattern that could not properly be modeled with the existing models. They found that the left part of the reaction time distributions, the fast reaction times, was not correctly accounted for. Therefore, they introduced a third inter-trial variability parameter, the inter-trial variability in the non-decision time. With that parameter included, the model allows for very short non-decision times. This lets the model predict very quick reaction times, without this being at the expense of other parameters or model fit.

For such reasons, the seven-parameter diffusion model was introduced, which extends the four-parameter model by adding inter-trial variabilities in the drift rate, the relative starting point, and the non-decision time. Variability in the drift rate is assumed to be normally distributed, and the variabilities in the relative starting point and the non-decision time are assumed to be uniformly distributed.

2.2.3 The notions of frequentist, Bayesian, and hierarchical modeling

When coming to the existing diffusion model implementations, it is necessary to elaborate on the two fundamentally different interpretations of “probability” that underlie the various implementations.

One main branch in statistics is the *frequentist statstics*. In the classical frequentist statistics, the probability of an event is seen as the limit of its relative frequency after a large number of trials (e.g., Kaplan, 2014). Hence, the central element in this approach is the idea of a repeated

experiment. The dataset is analyzed without a-priori knowledge. The results of a frequentist analysis are point estimates of the parameters. Typical examples are the coin toss experiment: To test whether a coin is fair one could toss a coin 100 times. When the coin shows 80 times heads and only 20 times tails, the estimate in a frequentist approach results in a biased coin with a probability of 0.8 for heads. This estimate becomes better, the larger the number of repetitions is.

The other main branch is the *Bayesian statistics*. In contrast to the frequentist approach, in Bayesian statistics an a-priori probability distribution (*prior*) is assumed (e.g., Kaplan, 2014). This means, knowledge and circumstances can be considered in the assessment of the experimental data. With each iteration of the experiment, the prior is updated with the observed data and an a-posteriori distribution, or an updated knowledge, results (*posterior*). This serves as the prior for the next round. Thus, from round to round, the parameter distribution becomes better. Instead of a point estimate, the Bayesian analysis results in a full distribution for the parameter.

Usually, the frequentist and the Bayesian approach lead to similar results. However, the advantage of the Bayesian approach is that it does not need an experiment that can be repeated (nearly) infinitely many times. A smaller database is sufficient. The only thing the Bayesian approach needs is a plausible choice for the a-priori distribution.

A funny application of Bayesian statistics is betting on a football match. You cannot let the same two teams play against each other for 100 times to obtain a point estimate for your bet. And you also do not *just bet something*. What you do is you use your knowledge about each of the teams and form an informed opinion on how many goals each team will shoot. You use all your a-priori knowledge to give a bet. After the match, you either see your assessment confirmed or not. You use this new, updated knowledge for your bet in the next match. This is Bayesian statistics in a simplified way.

Another real-world example of using knowledge in a Bayesian setting is puzzling. In this example, we can also introduce another important notion: *hierarchical modeling*. Puzzling always starts with a huge mountain of mixed up puzzle pieces. Each piece has several properties, like the color or the size and form. A single piece on its own is not yet very meaningful as it does not carry one key information: the position relative to the other pieces. Obviously, the aim of the puzzler is to find the position information for each piece. To do so, he uses information that come from the whole picture. For example, he puts pieces together that have the same color. This is due to his *prior*: He assumes that pieces with the same color fit most likely together. After testing the fit of several pieces, the puzzler updates his knowledge and obtains a *posterior* assumption: there might be two different objects in the picture that are of the same color but do not belong together. So, during the puzzle time, the puzzler uses information from another level to find the information on the position. This means, there are information on the piece-level (color, form), and information on the picture-level (position). These two levels form a hierarchy. This approach of using information from a higher level to solve a problem on the lower level is called *hierarchical*.

This finding can be transferred to psychological experiments. There are *hierarchical* and

non-hierarchical ways of modeling. In the non-hierarchical way, only data from the participant-level are analyzed, either each participant on its own or all participants as a whole as if they were one participant. In the hierarchical way, the researcher uses information from the group-behavior (*group level*) to infer to the participant level (*individual level*). This is very useful when not enough information is given on the participant level for single participants (*sparse data*) and directly leads to the problem researchers encountered in fitting data with the diffusion model.

The problem shows up in the process of finding the correct parameters (*parameter recovery*). For an accurate parameter recovery, very large trial numbers are required. Sometimes participants in a diffusion model study need to work on many trials (sometimes more than 2,000 trials per participant and condition; e.g., Ratcliff & Smith, 2004). As elaborated above, the problem can be addressed by using knowledge from other levels, this is by embedding the diffusion model in a Bayesian hierarchical framework (Vandekerckhove et al., 2011). This allows the researcher to calculate reliable and accurate estimates for the parameters of the decision process despite sparse data at the individual level by combining information from both levels, the individual and the group level. This *partial pooling* yields more robust parameter estimates than does fitting data for each individual separately (Rouder & Lu, 2005). Furthermore, this approach is helpful in integrating data across studies such that one can synthesize the evidence for the overall effects and can analyze how effects changed or did not change across studies (e.g., Pleskac et al., 2018).

Before a short overview of the diffusion model implementations is given, we have to introduce two other important concepts - truncated and censored data - as not all implementations are able to properly account for such restricted data.

2.2.4 Truncated and censored data

Both concepts are relevant in the context of modeling data from experiments which use response windows. As mentioned above, *response windows* are used to force participants to respond quickly. This is a technique to measure implicit processes that the participant cannot control actively. Such response windows change the reaction time distributions as no reaction times occur that are larger than the response deadline. Depending on how much information is still kept for trials that fall outside the response window, truncated or censored data are produced.

Data are called *censored* when the trials that fall outside the response window are counted. In the context of two-alternative forced-choice tasks where data is two dimensional (responses and response times), there are two cases of censored data. First, both bits of information are missing. The researcher knows how many trials fell outside the response window but the responses and the reaction times are unknown. Second, only the reaction time information is missing. The researcher knows (or somehow infers) the given responses but does not know the corresponding

reaction times. Hence, the trials that fell outside the response window are kept in the analysis and are treated specially.

In contrast, data are called *truncated* when no information is kept. This means there is neither information on the reaction time nor information on the given response. Additionally, there is no information on how many trials fell outside the response window. Hence, all trials that fell outside the response window are excluded from the analysis.

Both data lead to reaction time distributions that are cut at the limits of the response window. As can be seen in Figure 2.1, the tails of the reaction time distribution end smoothly in a standard diffusion model analysis. This means the standard approach cannot account for this type of data.

To model truncated or censored data properly, the probability density function (PDF), which defines the reaction time distribution, has to be corrected with the cumulative distribution function (CDF) or its complement (CCDF). This enables the researcher to cut the reaction time distribution at a certain point. Nevertheless, not all diffusion model implementations have the CDF and CCDF functions implemented and can therefore not offer a suitable way to model truncated or censored data.

To sum up, we described three drawbacks of the classical four-parameter model:

1. It is not able to cover all reaction time patterns in the data,
2. the non-hierarchical way of implementation needs too large datasets for accurate parameter recovery, and
3. it is not able to model truncated and censored data properly.

Therefore, in order to solve the first two limitations, it seems reasonable to extend the four-parameter model to the seven-parameter model and combine the larger model with the Bayesian hierarchical framework. In order to solve the third limitation, it seems necessary to find an implementation that is able to model truncated and censored data properly.

2.2.5 Common implementations

To estimate the diffusion model parameters, several user-friendly software packages have been developed. These implementations can be clustered into two main groups: non-Bayesian and Bayesian implementations. Within the Bayesian implementations, there is another division into implementations that can only be used in Python, or only in R, or that are independent of the programming environment as they have interfaces to several programming environments. However, note that this list can only be a selection as the research field is quickly evolving:

- Non-Bayesian
  - DMAT (Vandekerckhove & Tuerlinckx, 2007),
  - fast-dm (Voss & Voss, 2007),

  - EZ (Wagenmakers et al., 2007),

- Bayesian
  - Python-based
    - HDDM (Wiecki et al., 2013),
    - HSSM (Fengler et al., 2023),
  - R packages
    - DMC (Heathcote et al., 2019),
    - ggdmc (Lin & Strickland, 2020),
    - EMC2 (Stevenson et al., 2024),
  - With interfaces to several programming environments
    - WinBUGS (Vandekerckhove et al., 2011),
    - JAGS (Wabersich & Vandekerckhove, 2013),
    - Stan (Carpenter et al., 2017)

As described above, we are interested in Bayesian implementations to tackle the problems. Hence, the non-Bayesian implementations are listed here to show the diversity of existing implementations, but the focus lies on the Bayesian implementations.

We will have a short look on the other listed implementations:

First, some of the Bayesian implementations were developed and published after the work on this dissertation has started (EMC2 and HSSM).

Second, HDDM and HSSM are pure Python implementations. HDDM is a stand-alone Python-based toolbox which allows fast and flexible estimation of the diffusion model and other related models in a Bayesian and hierarchical manner. HSSM is the young successor of HDDM with state of the art implementations of the algorithms. Both packages, HDDM and HSSM, are limited in the choice of priors that users can specify[1]. This means, there are two sets of priors the user can choose from, but the user is not able to slightly change the priors or test different priors in the analysis.

Third, DMC and ggdmc are R-packages that allow to model the diffusion model in different variants. It is not explicitly possible to model truncated or censored data with both packages.

Fourth, WinBUGS, JAGS, and Stan are programming languages that have several interfaces to other programming environments that are commonly used among statisticians, like R, MATLAB, Python, or julia, among others. All three implementations are limited to the four-parameter model without inter-trial variabilities. In JAGS, a heuristic approach of modeling truncated and censored data is implemented, WinBUGS and Stan are lacking the possibility to model truncated or censored data.

[1] However, in HSSM it is planned for the future to enable a free choice of priors.

Vandekerckhove et al. (2011) proposed to implement inter-trial variabilities in a Bayesian framework by using the likelihood function of the basic (four-parameter) diffusion model and adding hyper-distributions for starting point, drift rate and non-decision time. In this framework, the specific parameter values for each trial of an experiment are drawn from the respective hyper-distributions. This procedure would be possible, for example, in JAGS or Stan. While this idea is theoretically interesting, this procedure led to convergence problems in our early applications and could therefore not be used.

Thus, the landscape of diffusion model implementations is quite large. However, an implementation that is able to meet the three limitations named above was still missing at the beginning of this dissertation project.

It should be an implementation

- which is able to model all seven parameters of the diffusion model,
- which allows for a Bayesian and hierarchical framework,
- which has the possibility to also model truncated or censored data,
- which is a flexible implementation in the sense of a free choice of the prior distributions,
- and which is not limited to a special programming language.

Therefore, we decided to implement the functions for the seven-parameter diffusion model and the truncated and censored diffusion model on our own. Eventually, we decided to implement the new functions in Stan as it has many advantages and meets all the above listed requirements:

(a) Stan is a user-friendly software which is freely available and structured as an open-source project. Thus, users are allowed to become a developer and actively participate in the programming process by adding own probability and distribution functions to the framework. Once, the new source-code is reviewed by the Stan-developers and officially accepted and included in the framework, other users can easily call the new distributions.

(b) Stan is a programming language for statistical modeling and high-performance statistical computation. Its Bayesian and hierarchical modeling framework is based on highly efficient state of the art algorithms.

(c) Stan provides the user with many functions for full Bayesian statistical inference and hierarchical modeling for many model families. Not only the diffusion model is now available in Stan but also many other distributions like the ones for Bernoulli, beta, binomial, exponential, normal, and Poisson distributions, to name just a few.

(d) Stan runs on all major platforms and has interfaces with the most popular data analysis languages, e.g., R, Python, shell, MATLAB, Julia, and Stata.

(e) Usage is easy and flexible. Users are allowed to define models and set priors freely.

(f) Stan allows for parallel processing on two levels: multiple processes can be run in parallel, and each process itself can be parallelized over as many cores as are available which makes computations more efficient and reduces computational time.

Chapter 3

The Aim of This Dissertation

As elaborated so far, the rich landscape of diffusion modeling implementations is missing a flexible implementation of the seven-parameter diffusion model to overcome some limitations of the four-parameter diffusion model. To furthermore model data that stems from experiments which have a response window included, an implementation is missing that properly accounts for such truncated or censored data

Therefore, the aim of this dissertation is to enrich the modeling landscape with an implementation of the diffusion model, in order to enable researchers, like the ones described above in prejudice research, (a) to apply a seven-parameter diffusion model analysis to their data, and (b) to properly model censored and truncated data.

To reach this aim, we split the project into three main steps, which form the three papers of this cumulative dissertation. In this course, we are going to implement the probability density function (PDF), cumulative distribution function (CDF), and the complementary cumulative distribution function (CCDF) in the programming language Stan. We are going to test each implementation extensively in order to show the correctness of the implemented algorithms. Furthermore, we want to give many hands-on code examples to demonstrate the usage of the new functionalities. Eventually, the new implementation will be applied to existing real-world censored data.

The implementation of the PDF and its partial derivatives of the seven-parameter diffusion model in Stan forms the first project (Paper 1). First, we will explain the mathematical and programming background. Then, we will perform an extensive recovery study and a simulation-based calibration study to validate the new implementation.

After these first two checks, where we want to show that the implementation does what we claim that it does, and where we test that the code itself is correct, we want to embed the new implementation in the landscape of diffusion model implementations. In order to do so, we are going to perform a comparison study to an already existing program that is also able to compute the seven-parameter diffusion model. As HDDM is a program that is most similar to our implementation, and as the authors of HDDM already performed a comparison study to other methods, we take HDDM as benchmark. We plan to rebuild the comparison study that was once published for HDDM and extend this one to fit our purposes. This comparison study will be the second project of this work (Paper 2).

When the seven-parameter model is tested thoroughly and ready to use, we want to tackle the second part of our aim: the implementation and validation of the CDF and CCDF functions

for truncated and censored models (Paper 3). Again, we want to start with theoretical background and the notions of truncated and censored data, and provide examples on how the new functionality can be used in Stan. Then, we plan analog validation tests as for the PDF implementation: a recovery study and a simulation-based calibration study. Finally, to come back to the application level, we want to apply the new implementation to a real-word data set from an FPST.

In the following, all three papers will be presented with their published (for Paper 2 it is the unpublished) content. The format has been adjusted to present a uniformly formatted overall text. The papers are followed by the General Discussion of this dissertation. The Acknowledgments, and information on funding, and code and data availability form the end of this work.

References

Amadeu Antonio Stiftung. (2025). Polizeiproblem: 21-jähriger Lorenz A. in Oldenburg von der Polizei erschossen. Retrieved September 23, 2025, from https://www.amadeu-antonio-stiftung.de/polizeiproblem-21-jaehriger-lorenz-in-oldenburg-von-der-polizei-erschossen-135595/

Amodio, D. M., Harmon-Jones, E., Devine, P. G., Curtin, J. J., Hartley, S. L., & Covert, A. E. (2004). Neural signals for the detection of unintentional race bias. *Psychological Science*, *15*(2), 88–93. https://doi.org/10.1111/j.0963-7214.2004.01502003.x

Arnold, N. R., Bröder, A., & Bayen, U. J. (2015). Empirical validation of the diffusion model for recognition memory and a comparison of parameter-estimation methods. *Psychological Research*, *79*(5), 882–898. https://doi.org/10.1007/s00426-014-0608-y

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, *76*(1). https://doi.org/10.18637/jss.v076.i01

Correll, J., Park, B., Judd, C. M., & Wittenbrink, B. (2002). The police officer's dilemma: Using ethnicity to disambiguate potentially threatening individuals. *Journal of Personality and Social Psychology*, *83*(6), 1314–1329. https://doi.org/10.1037//0022-3514.83.6.1314

Correll, J., Wittenbrink, B., Crawford, M. T., & Sadler, M. S. (2015). Stereotypic vision: How stereotypes disambiguate visual stimuli. *Journal of Personality and Social Psychology*, *108*(2), 219–233. https://doi.org/10.1037/pspa0000015

Fengler, A., Omar, A., Xu, P., Bera, K., & Frank, M. J. (2023). HSSM Documentation. Retrieved August 17, 2023, from https://lnccbrown.github.io/HSSM/

Forstmann, B. U., Ratcliff, R., & Wagenmakers, E.-J. (2016). Sequential Sampling Models in Cognitive Neuroscience: Advantages, Applications, and Extensions. *Annual Review of Psychology*, *67*, 641–666. https://doi.org/10.1146/annurev-psych-122414-033645

Govorun, O., & Payne, B. K. (2006). Ego-Depletion and Prejudice: Separating Automatic and Controlled Components. *Social Cognition*, *24*(2), 111–136.

Heathcote, A., Lin, Y.-S., Reynolds, A., Strickland, L., Gretton, M., & Matzke, D. (2019). Dynamic models of choice. *Behavior Research Methods*, *51*(2), 961–985. https://doi.org/10.3758/s13428-018-1067-y

Ito, T. A., & Tomelleri, S. (2017). Seeing is not stereotyping: The functional independence of categorization and stereotype activation. *Social Cognitive and Affective Neuroscience*, *12*(5), 758–764. https://doi.org/10.1093/scan/nsx009

Johnson, D. J., Cesario, J., & Pleskac, T. J. (2018). How prior information and police experience impact decisions to shoot. *Journal of Personality and Social Psychology*, *115*(4), 601–623. https://doi.org/10.1037/pspa0000130

Johnson, D. J., Hopwood, C. J., Cesario, J., & Pleskac, T. J. (2017). Advancing Research on Cognitive Processes in Social and Personality Psychology: A Hierarchical Drift Dif-

fusion Model Primer. *Social Psychological and Personality Science*, *8*(4), 413–423. https://doi.org/10.1177/1948550617703174

Kaplan, D. (2014). *Bayesian Statistics for the Social Sciences* (1st ed.). Guilford Publications.

Lerche, V., & Voss, A. (2019). Experimental validation of the diffusion model based on a slow response time paradigm. *Psychological Research*, *83*(6), 1194–1209. https://doi.org/10.1007/s00426-017-0945-8

Lin, Y.-S., & Strickland, L. (2020). Evidence accumulation models with R: A practical guide to hierarchical Bayesian methods. *The Quantitative Methods for Psychology*, *16*(2), 133–153. https://doi.org/10.20982/tqmp.16.2.p133

Mayerl, H., Rainer, A., & Gula, B. (2019). Modeling effects of newspaper articles on stereotype accessibility in the shooter task. *Social Cognition*, *37*(6), 571–595.

Payne, B. K. (2001). Prejudice and perception: The role of automatic and controlled processes in misperceiving a weapon. *Journal of Personality and Social Psychology*, *81*(2), 181–192. https://doi.org/10.1037/0022-3514.81.2.181

Payne, B. K. (2006). Weapon Bias: Split-second decisions and unintended stereotyping. *Current Directions in Psychological Science*, *15*(6), 287–291.

Pleskac, T. J., Cesario, J., & Johnson, D. J. (2018). How race affects evidence accumulation during the decision to shoot. *Psychonomic Bulletin & Review*, *25*(4), 1301–1330. https://doi.org/10.3758/s13423-017-1369-6

Ratcliff, R. (1978). A Theory of Memory Retrieval. *Psychological Review*, *85*(2), 59–108.

Ratcliff, R., & Rouder, J. N. (1998). Modelling response times for two-choice decisions. *Psychological Science*, *9*(5), 347–356.

Ratcliff, R., & Smith, P. L. (2004). A comparison of sequential sampling models for two-choice reaction time. *Psychological Review*, *111*(2), 333–367. https://doi.org/10.1037/0033-295X.111.2.333

Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion Decision Model: Current Issues and History. *Trends in Cognitive Sciences*, *20*(4), 260–281. https://doi.org/10.1016/j.tics.2016.01.007

Ratcliff, R., & Tuerlinckx, F. (2002). Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review*, *9*(3), 438–481. https://doi.org/10.3758/BF03196302

Rivers, A. M. (2017). The Weapons Identification Task: Recommendations for adequately powered research. *PloS one*, *12*(6), e0177857. https://doi.org/10.1371/journal.pone.0177857

Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review*, *12*(4), 573–604. https://doi.org/10.3758/BF03196750

Stepanova, E. V., Bartholow, B. D., Saults, J. S., & Friedman, R. S. (2012). Alcohol-related Cues Promote Automatic Racial Bias. *Journal of Experimental Social Psychology*, *48*(4), 905–911. https://doi.org/10.1016/j.jesp.2012.02.006

Stevenson, N., Donzallaz, M. C., Innes, R. J., Forstmann, B., Matzke, D., & Heathcote, A. (2024). EMC2: An R Package for cognitive models of choice. *Preprint*. https://doi.org/10.31234/osf.io/2e4dq

Tagesschau. (2024). Polizisten töten Schwarzen mit 96 Schüssen. Retrieved September 23, 2025, from https://www.tagesschau.de/ausland/amerika/usa-chicago-polizeigewalt-rassismus-schwarzer-getoetet-100.html

Tagesspiegel. (2000). Vier Polizisten müssen sich für Mord an einem unbewaffneten Schwarzen verantworten. Retrieved September 23, 2025, from https://www.tagesspiegel.de/gesellschaft/panorama/vier-polizisten-mussen-sich-fur-mord-an-einem-unbewaffneten-schwarzen-verantworten-654118.html

Thiem, K. C., Neel, R., Simpson, A. J., & Todd, A. R. (2019). Are Black Women and Girls Associated With Danger? Implicit Racial Bias at the Intersection of Target Age and Gender. *Personality & Social Psychology Bulletin*, *45*(10), 1427–1439. https://doi.org/10.1177/0146167219829182

Todd, A. R., Johnson, D. J., Lassetter, B., Neel, R., Simpson, A. J., & Cesario, J. (2020). Category salience and racial bias in weapon identification: A diffusion modeling approach. *Journal of Personality and Social Psychology*. https://doi.org/10.1037/pspi0000279

Todd, A. R., Simpson, A. J., Thiem, K. C., & Neel, R. (2016). The generalization of implicit racial bias to young black boys: Automatic stereotyping or automatic prejudice? *Social Cognition*, *34*(4), 306–323.

Todd, A. R., Thiem, K. C., & Neel, R. (2016). Does Seeing Faces of Young Black Boys Facilitate the Identification of Threatening Stimuli? *Psychological Science*, *27*(3), 384–393. https://doi.org/10.1177/0956797615624492

Vandekerckhove, J., & Tuerlinckx, F. (2007). Fitting the Ratcliff diffuion model to experimental data. *Psychonomic Bulletin & Review*, *14*(6), 1011–1026.

Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2011). Hierarchical diffusion models for two-choice response times. *Psychological Methods*, *16*(1), 44–62. https://doi.org/10.1037/a0021765

Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, *32*(7), 1206–1220.

Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, *39*(4), 767–775.

Wabersich, D., & Vandekerckhove, J. (2013). Extending JAGS: A tutorial on adding custom distributions to JAGS (with a diffusion model example). *Behavior Research Methods*, *46*(1), 15–28. https://doi.org/10.3758/s13428-013-0369-3

Wagenmakers, E.-J., van der Maas, H. L. J., & Grasman, R. P. P. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, *14*(1), 3–22.

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in Neuroinformatics*, *7*, 14. https://doi.org/10.3389/fninf.2013.00014

Chapter 4

Paper: The Seven-Parameter Diffusion Model in Stan

Henrich, F., Hartmann, R., Pratz, V., Voss, A., & Klauer, K.C. (2023). The Seven-parameter Diffusion Model: An Implementation in Stan for Bayesian Analyses. *Behavior Research Methods*.

Abstract

Diffusion models have been widely used to get information about cognitive processes from the analysis of responses and response-time data in two-alternative forced-choice tasks. We present an implementation of the seven-parameter diffusion model, incorporating inter-trial variabilities in drift rate, non-decision time, and relative starting point, in the probabilistic programming language Stan. Stan is a free, open-source software that gives the user much flexibility in defining model properties such as the choice of priors and the model structure in a Bayesian framework. We explain the implementation of the new function and how it is used in Stan. We then evaluate its performance in a simulation study that addresses both parameter recovery and simulation-based calibration. The recovery study shows generally good recovery of the model parameters in line with previous findings. The simulation-based calibration study validates the Bayesian algorithm as implemented in Stan.

Keywords: Ratcliff diffusion model · Bayesian inference · Stan function · model fitting

4.1 Introduction

Diffusion models (DMs) are among the most frequently used model families in modeling two-alternative forced-choice tasks (see Wagenmakers, 2009, for a review). DMs allow one to model response times and responses in two-alternative forced-choice tasks jointly. In this article, we focus on a seven-parameter version of the model that includes inter-trial variability in several of its components (Ratcliff & Rouder, 1998) as detailed below.

Since its introduction to psychological research, a number of user-friendly software tools have been developed to estimate the model parameters (Vandekerckhove & Tuerlinckx, 2007; Voss & Voss, 2007; Wagenmakers et al., 2007). Bayesian implementations have been proposed for use with WinBUGS (Vandekerckhove et al., 2011), JAGS (Wabersich & Vandekerckhove, 2013), Stan (Carpenter et al., 2017), and as a Python package called HDDM (Wiecki et al., 2013). The purpose of this article is to add to the existing Bayesian implementations, and to overcome limitations of the existing implementations. Specifically, the just-mentioned WinBUGS, JAGS and Stan implementations are limited to a more basic four-parameter version of the DM without inter-trial variabilities, whereas HDDM is limited in the choice of priors that users can specify.

Here, we provide an implementation of the seven-parameter model within the probabilistic programming language Stan (Carpenter et al., 2017). Stan is a free, open-source software that gives the user huge flexibility in defining and varying model properties such as the choice of priors. Stan runs on all major platforms and interfaces with the most popular data analysis languages (R, Python, shell, MATLAB, Julia, Stata).

In the following sections, we first briefly introduce the diffusion model. Following this, we provide details on our new Stan implementation. Finally, we present two sanity checks for our implementation: a simulation study showing good recovery on simulated data, and a simulation-based calibration study analyzing the same simulated data, providing a more rigorous test of the correctness of our algorithm.

4.2 The Diffusion Model

The basic four-parameter DM, first introduced by Ratcliff (1978), is a sequential sampling model used to explain data from two-alternative forced-choice tasks. It has been widely applied to tasks as, for example, the Eriksen flanker task (Assink et al., 2015; Eriksen & Erisken, 1974; White et al., 2010), among many others. In the Eriksen flanker task, participants decide whether a central target arrow among a set of distractor arrows points to the *left* or to the *right* (e.g., <<<><<<).

In diffusion modeling, it is assumed that participants accumulate evidence towards either of two response options on a unidimensional evidence scale on which two boundaries are placed, one for each response option. The distance between both boundaries is denoted as *boundary*

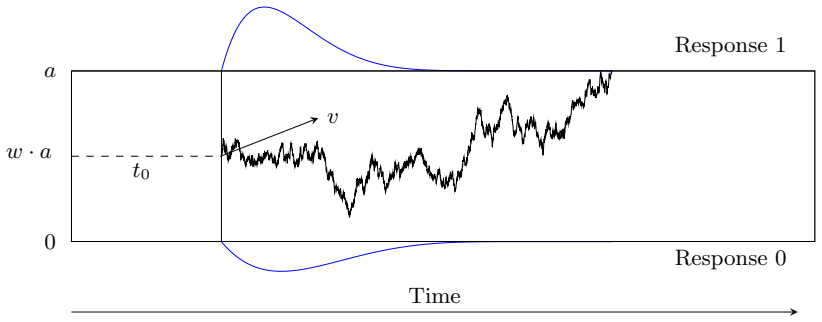


Figure 4.1: Realization of a four-parameter diffusion process modeling the binary decision process. *Note.* The parameters are the *boundary separation a* for two response alternatives, the *relative starting point w*, the *drift rate v*, and the *non-decision time* t_0. The decision process is illustrated as a jagged line between the two boundaries. The predicted distributions of the reaction times are depicted as curved lines below and above the response boundaries (blue).

separation, a. Participants start with a state of evidence placed between the two boundaries on the evidence scale. This point is denoted as *relative starting point*, w. Accommodating the possibility of prior bias, this starting point needs not to be equidistant from both boundaries. Participants then accumulate decision-relevant evidence from the environment until a boundary is reached. The evidence-accumulation rate is denoted as *drift rate*, v. The evidence accumulation process is noisy and is therefore approximated by a diffusion process. When a boundary is met, a decision for the associated response option is made. All time costs for processes that do not belong to the decision process are summarized in the *non-decision time*, t_0. Based on those four parameters (for the basic model), a DM predicts the probability to choose one or the other response alternative and models the distributions of response times associated with each alternative.

In Figure 4.1, a diffusion process is depicted. Since the evidence accumulation process is influenced by random noise, the process is drawn as a jagged line. One main advantage of the diffusion model is that the parameters can be interpreted in terms of cognitive processes. For example, the boundary separation is higher when the participant is focused on accuracy, the absolute value of the drift rate is smaller when stimuli are harder to discriminate, the non-decision time is higher for a more time-consuming form of response, and the relative starting point moves towards a decision alternative for which the participant is rewarded (e.g., Arnold et al., 2015; Lerche & Voss, 2019; Voss et al., 2004).

According to Ratcliff and Rouder (1998), the basic four-parameter model has problems to account for the full range of data in two-alternative forced-choice tasks. For example, the model predicts identical reaction time distributions for correct and error responses, if the relative starting point is centered between the boundaries. However, it may occur that, having a centered relative starting point, errors are slower than correct responses. Slow errors can be modeled with inter-trial variability in drift rate, because for a large drift rate, reaction time is short and accuracy is high, whereas for a small drift rate, reaction time is slower and accuracy is lower. In

sum, given variability in drift rate, the percentage of slow responses will increase among errors more than among correct responses. Another possibility is that errors are faster than correct responses. This reaction time pattern of fast errors can be modeled with inter-trial variability in starting point, because for a starting point near the correct response boundary, there will be few errors and they will be slow, whereas for a starting point near the error response boundary, there will be more errors and they will be fast. In sum, given variability in starting point, the percentage of fast responses will increase among errors more than among correct responses (Forstmann et al., 2016). For such reasons, Ratcliff and Rouder introduced the seven-parameter DM, which extends the four-parameter model by adding inter-trial variabilities in the drift rate, the non-decision time and the starting point. Variability in drift rate is assumed to be normally distributed, and the variabilities in non-decision time and starting point are assumed to be uniformly distributed.

Another problem regards parameter recovery. For an accurate parameter recovery large trial numbers are required. Therefore sometimes participants in a DM study need to work on many trials (sometimes more than 2,000 trials per participant and condition; e.g., Ratcliff & Smith, 2004). This problem can be mitigated by embedding the DM in a Bayesian hierarchical framework (Vandekerckhove et al., 2011), which allows one to calculate reliable and accurate estimates for the parameters of the decision process despite sparse data at the individual level by combining information from both levels, the individual and the group level[1]. This partial pooling yields more robust parameter estimates than does fitting data for each individual separately (Rouder & Lu, 2005). Furthermore, this approach is helpful in integrating data across studies such that one can synthesize the evidence for the overall effects and can analyze how effects changed or did not change across studies (e.g., Pleskac et al., 2018).

Therefore, the next logical step is to combine the seven-parameter model with the Bayesian hierarchical framework. An implementation of the highly efficient Hamiltonian algorithm for Markov chain Monte Carlo estimation (MCMC, Neal, 2011) in the form of the No-U-Turn Sampler (NUTS, Hoffman & Gelman, 2014) is given in Stan. Stan is a probabilistic programming language for statistical modeling and high-performance statistical computation. Stan, named after one of the pioneers in Monte Carlo methods, Stanislav Ulam, provides the user with tools for full Bayesian statistical inference and hierarchical modeling. The MCMC method draws samples from the joint posterior distribution of the parameters of a Bayesian model, which are used to draw inferences on the model parameters and the model fit. Stan is free and open-source and every user is invited to participate in the development of new features. Users can add new functions by specifying the logarithm of a density (log-density) in the C++ based Stan language (Stan Development Team, 2023a).

---

[1] Vandekerckhove et al. (2011) proposed to implement inter-trial variabilities in a Bayesian framework by using the likelihood function of the basic (four-parameter) diffusion model and adding hyper-distributions for starting point, drift rate and non-decision time. In this framework, the specific parameter values for each trial of an experiment are drawn from the respective hyper-distributions. While this idea is theoretically interesting, this procedure led to convergence problems in our applications and could therefore not be used.

4.3 The Stan Function `wiener_full_lpdf()`

We implemented the log-density of the first-passage time distribution of the seven-parameter DM to provide the new function `wiener_full_lpdf()` for Stan users. Since we added the function in Stan's math library (Stan Development Team, 2023b), the function can be used with every interface that supports Stan. The Hamiltonian Monte Carlo algorithm relies on partial derivatives of the log-likelihood function to sample the posterior distribution more efficiently (Neal, 2011). As deriving the derivatives for each model can be cumbersome, Stan automatically computes these partial derivatives using reverse-mode automatic differentiation and numerical approximation (Carpenter et al., 2015). For the underlying distributions that are used to build a model, it can make sense, however, to implement the partial derivatives manually. In the case of a very complex function with known partial derivatives, it is much more efficient and accurate to compute the values of the partial derivatives analytically instead of approximating them numerically. Therefore, we used the work by Hartmann and Klauer (2021), who derived the partial derivatives for the first-passage time distribution in diffusion models, to implement both the log-density of the seven-parameter model and its partial derivatives.

The new function `wiener_full_lpdf()` returns the logarithm of the first-passage time density function for a diffusion model with up to seven parameters for upper boundary responses. The same function can be used to obtain the log-density for the lower boundary as well (see below). Any combination of fixed and estimated parameters can be specified. In other words, with this implementation it is not only possible to estimate parameters of the full seven-parameter model, but also to estimate restricted models such as the basic four-parameter model, or a five- or six-parameter model, or even a one-parameter model when fixing the other six parameters. For example, it is possible to permit variability in just one or two parameters and to fix the other variabilities to 0, or even to estimate a three-parameter model when fixing more parameters (e.g., fixing the relative starting point at 0.5).

It is assumed that the reaction time data y are distributed according to `wiener_full()`:

$$y \sim \text{wiener\_full}(a, t_0, w, v, s_v, s_w, s_{t_0}). \tag{4.1}$$

Mathematically, the function consists of the reaction times, y, and the seven parameters, boundary separation, a, (lower bound of the) non-decision time, t_0, relative starting point, w, drift rate, v, inter-trial variability of the drift rate, s_v, inter-trial variability of the relative starting point, s_w, and inter-trial variability of the non-decision time, s_{t_0}. It can be stated in the following terms:

$$
\begin{aligned}
&\log\big[p(y \mid a, t_0, w, v, s_v, s_w, s_{t_0})\big] = \\
&\log\big[\frac{1}{s_{t_0}} \int_{t_0}^{t_0+s_{t_0}} \frac{1}{s_w} \int_{w-\frac{1}{2}s_w}^{w+\frac{1}{2}s_w} \int_{-\infty}^{\infty} p_3(y-\tau_0 \mid a, \nu, \omega) \\
&\times \frac{1}{\sqrt{2\pi s_v^2}} \exp\Big(-\frac{(\nu-v)^2}{2s_v^2}\Big)\, d\nu\, d\omega\, d\tau_0\big] = \\
&\log\big[\frac{1}{s_{t_0}} \int_{t_0}^{t_0+s_{t_0}} \frac{1}{s_w} \int_{w-\frac{1}{2}s_w}^{w+\frac{1}{2}s_w} M \times p_3(y-\tau_0 \mid a, \nu, \omega)\, d\omega\, d\tau_0\big],
\end{aligned}
\tag{4.2}
$$

where $p()$ denotes the density function, and M and $p_3()$ are defined, by using $t := y - \tau_0$, as

$$
M := \frac{1}{\sqrt{1+s_v^2 t}} \exp\Big(av\omega + \frac{v^2 t}{2} + \frac{s_v^2 a^2 \omega^2 - 2av\omega - v^2 t}{2(1+s_v^2 t)}\Big) \text{ and} \tag{4.3}
$$

$$
p_3(t \mid a, v, w) := \frac{1}{a^2} \exp\Big(-avw - \frac{v^2 t}{2}\Big) f\big(\frac{t}{a^2} \mid 0, 1, w\big), \tag{4.4}
$$

where $f(t^* = \frac{t}{a^2} \mid 0, 1, w)$ can be specified in two ways:

$$
f_l(t^* \mid 0, 1, w) = \sum_{k=1}^{\infty} k\pi \exp\Big(-\frac{k^2\pi^2 t^*}{2}\Big) \sin(k\pi w) \text{ and} \tag{4.5}
$$

$$
f_s(t^* \mid 0, 1, w) = \sum_{k=-\infty}^{\infty} \frac{1}{\sqrt{2\pi (t^*)^3}} (w+2k) \exp\Big(-\frac{(w+2k)^2}{2t^*}\Big). \tag{4.6}
$$

Which of these is used in the computations depends on which expression requires the smaller number of components k to guarantee a pre-specified precision (Blurton et al., 2017; Gondan et al., 2014; Hartmann & Klauer, 2021; Navarro & Fuss, 2009).

4.3.1 How to use the Function in Stan

After the mathematical formulation of the seven-parameter diffusion model, we now present a hands-on description of how to use the new function. In the declaration of a Stan model, `wiener_full` can be called in two different ways

```
y ~ wiener_full(a,t0,w,v,sv,sw,st0);
```

or

```
target += wiener_full_lpdf(y|a,t0,w,v,sv,sw,st0);
```

Since the function is not vectorized, it is called for each experimental trial in a for-loop for the reaction time and response observed in the trial with parameters appropriate to the condition (see Figure 4.2 for a template). Note that the function always returns the value for the upper response boundary. To compute the value for the lower response boundary the function has to be called with $-v$ instead of v, and $1-w$ instead of w. The model block shown in Figure 4.2 provides a template for calling the function for both, the upper and the lower response boundary.

As pointed out above, `wiener_full_lpdf()` also allows one to compute restricted models involving one, two, three, four, five, or six parameters by setting parameters to zero or fixing parameters to other given values. For example, s_v, s_w, and/or s_{t_0} can be set to zero, indicating, in order, no inter-trial variability in v, no inter-trial variability in w, and/or no inter-trial variability in t_0, respectively. Often it might also be useful to set the relative starting point to 0.5 (e.g., when assuming an unbiased decision maker). For example, if no inter-trial variabilities for the relative starting point and for the non-decision time are needed, the function call might look as follows:

```
target += wiener_full_lpdf(y|a,t0,w,v,sv,0,0);
```

For a very parsimonious three-parameter model, assuming no inter-trial variabilities at all and fixing the relative staring point at 0.5, the function call might look as follows:

```
target += wiener_full_lpdf(y|a,t0,0.5,v,0,0,0);
```

It is also possible to control the precision in the computation of the DM partial derivatives[2] by calling the function `wiener_full_prec_lpdf()`, analogously:

```
target += wiener_full_prec_lpdf(y|a,t0,w,v,
                                sv,sw,st0,precision);
```

The usage and behavior of the two functions are the same except for the added control over the precision parameter.

---

[2] The precision value controls the accuracy in the computation of the partial derivatives. The default value for the precision is 10^{-4}. The user can provide smaller values for increased precision. Note that this precision value only changes the precision in the computation of the partial derivatives, but not of the DM density itself. The precision value for the density is internally fixed to 10^{-6} and cannot be changed by the user. The partial derivatives determine the directions in which the parameter space is explored in MCMC sampling. Requesting more accurately computed derivatives may thereby help to increase the efficiency of the exploration of the parameter space, but it trades off against a time cost for computing more accurate derivatives. The validity of the algorithm as such is not influenced by this parameter.

```
data {
 int<lower=0> N;   // Number of trials
 array[N] real rt; // response times (in seconds)
 array[N] int<lower=0, upper=1> resp; // responses {0,1}
}

parameters {
 real<lower=0> a;            // boundary separation
 real v;                     // drift
 real<lower=0, upper=1> w;   // relative starting point
 real<lower=0> t0;           // non-decision time

 real<lower=0> sv;           // variability in drift
 real<lower=0, upper=1> sw; // variability in starting point
 real<lower=0> st0;          // variability in non-decision time
}

model {
 // prior distributions of parameters
 a  ~ normal(1,1);
 w  ~ normal(0.5,0.1);
 v  ~ normal(2,3);
 t0 ~ normal(0.435,0.12);

 sv  ~ normal(1,3);
 st0 ~ normal(0.183,0.09);
 sw  ~ beta(1,3);

 // diffusion model
 for (i in 1:N) {
  if (resp[i] == 1) { // upper boundary
   target += wiener_full_lpdf(rt[i] | a, t0, w, v, sv, sw, st0);
  } else { // lower boundary (mirror drift and starting point)
   target += wiener_full_lpdf(rt[i] |
               a, t0, 1-w, -v, sv, sw, st0);
  }
 }
}
```

Figure 4.2: Minimal example of a Stan script for a non-hierarchical seven-parameter DM. *Note.* See text for an explanation of the different components of this script.

4.3.2 Declaration of the Stan Model

To declare a Stan model the user should specify three blocks: the data block, the parameters block, and the model block. In the following, the blocks will be described in some detail (see Figure 4.2 for an example of a model declaration for a seven-parameter DM).

The Data Block

The data should consist of at least three variables:

1. The number of trials N,
2. the response, coded as $0 =$ "lower bound" and $1 =$ "upper bound" (in Figure 4.2), and
3. the reaction times in seconds (not milliseconds).

Note that two different ways of coding responses are commonly used: First, in *response coding*, the boundaries correspond to the two response alternatives. Second, in *accuracy coding*, the boundaries correspond to correct (upper bound) and wrong (lower bound) responses.

Depending on the experimental design, one would typically also provide the number of conditions and the condition associated with each trial as a vector. In a hierarchical setting, the data block would also specify the number of participants and the participant associated with each trial as a vector. It is also possible to hand over a precision value in the data block.

The Parameters Block

The model arguments of the `wiener_full_lpdf()` function that are not fixed to a certain value are defined as parameters in the parameters block. In this block, it is also possible to insert restrictions on the parameters. Note that the MCMC algorithm iteratively searches for the next parameter set. If the suggested sample falls outside the internally defined parameter ranges, the program will throw an error, which causes the algorithm to restart the current iteration. Since this slows down the sampling process, it is advisable to include the parameter ranges in the definition of the parameters in the parameters block to improve the sampling process (see Table 4.1 for the parameter ranges) as exemplified in Figure 4.2. In addition, the parameter space is further constrained by the following conditions:

1. The non-decision time has to be smaller or equal to the RT: $t_0 \leq y$.
2. The varying relative starting point has to be in the interval $(0,1)$ and thus,

$$\begin{aligned} &w + \frac{s_w}{2} < 1, \text{ and} \\ &0 < w - \frac{s_w}{2}. \end{aligned} \tag{4.7}$$

Parameter	Range	Parameter	Range
a	$(0,\infty)$	y [a]	$(0,\infty)$
v	$(-\infty,\infty)$	s_v	$[0,\infty)$
w	$(0,1)$	s_w	$[0,1)$
t_0	$[0,\infty)$	s_{t_0}	$[0,\infty)$

Table 4.1: Parameter Ranges.
Note.[a] The reaction time, y, is not a parameter.

The Model Block

In the model block, the priors are defined and the likelihood is called for the upper and the lower response boundary. Different kinds of priors can be specified here. When no prior is specified for a parameter, Stan uses default priors with specifications `uniform(-infinity, infinity)`. For further information see the Stan Development Team (2022b). Generally, mildly informative priors might help to get the full benefit of a Bayesian analysis.

In the second part of the model block, the likelihood function is applied to all responses. As explained above, this has to be done in a for-loop, and drift rate and relative starting point have to be mirrored for responses at the lower boundary.

4.4 Validating the new Function

In this section, we report results from two sanity checks: First, we present a simulation study to test whether our implementation of the full diffusion model is able to recover given parameters and, second, we perform a simulation-based calibration study (Talts et al., 2018) analyzing the same simulated data to test the adequacy of the resulting posterior distributions. For these studies, we chose prior distributions for all parameters as recommended in the literature, sampled different sets of parameters from these distributions, then simulated data from these parameters and ran the model on the data with the same distributions for the priors in order to analyze the results in two different ways.

4.4.1 Simulation Study

We conducted a simulation study to test, on the one hand, the precision of parameter recovery (recovery study), and, on the other hand, whether the new implementation is correct (simulation-based calibration study). For this purpose we simulated data once and then analyzed these data with respect to both aspects. Simulated datasets comprise trials from two conditions, representing two different stimulus types, where for Condition 1 and 2 positive and negative *drift rates*, respectively, are assumed. All other parameters are shared across condi-

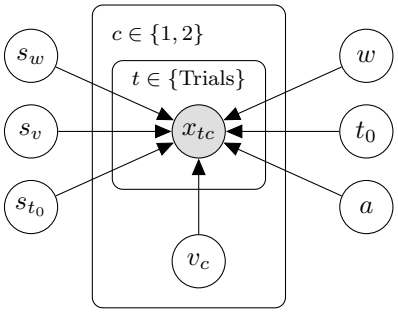


Figure 4.3: Graphical model representation in the simulation study. *Note.* Each data point x_{tc} (vector of reaction time and response) within trial t and condition c depends on the seven diffusion parameters, from which only the drift rate varies between conditions. This results in eight parameters to estimate.

tions as depicted in the graphical model representation in Figure 4.3. This is a common design in many reaction time experiments (e.g., see Arnold et al., 2015; Johnson et al., 2020; Ratcliff & Smith, 2004; Voss et al., 2004).

The data were fitted with the full diffusion model, comprising a total of eight parameters (because of the 2 drift rates). Separate models were fitted for each simulated dataset.

Ground Truth and Priors

The parameters for the simulation, denoted as the *ground truth*, are randomly drawn from the prior distributions used in the model. This is a natural choice for informative priors in the case that the generating model is known, and a prerequisite for the simulation-based calibration.

The parameters are drawn from the distributions shown in Table 4.2, where $\mathscr{N}$ denotes the normal distribution, $\mathscr{B}$ the beta distribution, and $\mathrm{T}[.,.]$ denotes a truncation. At the same time, these distributions serve as priors in the Stan model. The prior distributions for a, w, and s_v are based on Wiecki et al. (2013, Figure 1 in the Supplements), the distributions for t_0 and s_{t_0} are based on Matzke and Wagenmakers (2009, Table 3) and the distributions for v and s_w are the ones used in Wiecki et al. (2013). To simulate the above-mentioned two conditions, v is drawn twice, and the second value is multiplied with the factor -1, such that in the first condition, v is directed to the upper boundary and in the second condition, v is directed to the lower boundary.

Datasets

For the choice of the number of datasets we follow the settings used in previous recovery analyses and simulation-based calibration analyses (using between $N = 500$ and $N = 10.000$ simulated datasets, see Hartmann et al., 2020; Heck et al., 2018; Klauer & Kellen, 2018; Lerche et al., 2017; Talts et al., 2018; Wabersich & Vandekerckhove, 2013), and consider computational

Parameter	Prior distribution
a	$\mathcal{N}(1,1)$ T$[0.5,3]$
v	$\mathcal{N}(2,3)$ T$[0,5]$
w	$\mathcal{N}(0.5,0.1)$ T$[0.3,0.7]$
t_0	$\mathcal{N}(0.435,0.12)$ T$[0.2,1]$
s_v	$\mathcal{N}(1,3)$ T$[0,3]$
s_w	$\mathcal{B}(1,3)$
s_{t_0}	$\mathcal{N}(0.183,0.09)$ T$[0,0.5]$

Table 4.2: Priors for simulation study.
Note. $\mathcal{N}$ = normal distribution; $\mathcal{B}$ = beta distribution; T$[.,.]$ = truncation.

time. Hence, we drew 2000 ground truths from the prior distributions shown in Table 4.2 and simulated two datasets for each ground truth, resulting in $2 \times N = 2 \times 2000$ datasets.

The first 2000 datasets each consist of 100 simulated trials (50 per condition), and the second 2000 datasets each consist of 500 simulated trials (250 per condition). Much more trials seem to be unrealistic in reaction time tasks and are very costly in terms of computation time. Much fewer trials are assumed to be too few for the successful estimation of inter-trial variabilities (Boehm et al., 2018). Therefore, using 500 trials seemed to be a good compromise, and 100 trials are chosen to see whether the method still performs reasonably well with fewer trials. Data were simulated with the `rdiffusion()`-function of the R-package `rtdists` (R Core Team, 2021; Singmann et al., 2022) with a precision of 4 and the *fastdm*-method (Voss & Voss, 2007).

Method Configuration

Analyses were run on the bwUniCluster within the framework program bwHPC with parallelization only in the Stan-model via the `reduce_sum()` routine. We chose to run 4 chains (as recommended by Vehtari et al., 2021, page 4). All chains were computed sequentially. In calibration studies we found that reducing the maximum treedepth to 5 speeds up the sampling process, while still resulting in good convergence and no divergent transitions.

We started computations with 150 warmup and 500 sampling iterations per chain and repeated computations up to seven times with increased warmup iterations for those datasets for which the model did not converge satisfactorily. For all other method parameters the Stan default values were taken.

Recovery Study

Convergence and Diagnostics. The Stan developers recommend that some diagnostics need to fulfill certain criteria before going deeper into the analysis (e.g., Vehtari et al., 2021). Among these diagnostics are the rank-normalized effective sample size, N_{eff}, the convergence parameter, $\hat{R}$, and the number of divergent transitions.

First, the effective sample size captures how many independent draws contain the same amount of information as the dependent draws obtained by the MCMC algorithm. It is recommended to check that the rank-normalized effective sample size is greater than 400, $N_{\text{eff}} > 400$ (Vehtari et al., 2021). A useful heuristic is to ensure that the relative effective sample size is large enough: $N_{\text{eff}}/N_{\text{samp}} > 0.1$, where N_{samp} is the number of samples drawn and retained from the posterior distribution (Stan Development Team, 2022a).

Second, the $\hat{R}$ value is a measure of convergence. This is recommended to be smaller than 1.01, $\hat{R} < 1.01$ (Vehtari et al., 2021), due to experience of the authors in practical use. This threshold is much tighter than the value of $\hat{R} < 1.1$ first recommended by Brooks and Gelman (1998).

Third, there should not be divergent transitions in the sampling process. Divergent transitions can bias the obtained estimates and are an indicator of convergence problems (Vehtari et al., 2021).

Therefore, we checked these diagnostics ($N_{\text{eff}}/N_{\text{samp}} > 0.1$, $\hat{R} < 1.01$, and divergent transitions), and reanalyzed datasets with insufficient diagnostics with more warmup iterations to ensure the chains have converged at the start of sampling. We started analyses with 150 warmup iterations per chain. As this quickly turned out to be too low to reach the strict convergence criteria, we continued the analyses with higher warmup and sampling iterations per chain. In the end, most of the datasets met the criteria with 1000 warmup and 1000 sampling iterations per chain (about 99%). There were only a few datasets that needed up to 3000 warmup and 1000 sampling iterations (about 1%).

For the retained 4000 MCMC samples, all effective sample sizes are above 400, all relative effective sample sizes are greater than 0.1, nearly all $\hat{R}$ values are smaller than 1.01 (2 out of 32000 $\hat{R}$ were bigger than 1.01) and no divergent transitions occurred. There is one dataset for which the $\hat{R}$ equals 1.012 for the a parameter and $\hat{R}$ equals 1.013 for the s_v parameter. But, as these values are still below 1.05 we stopped reanalyzing and included this dataset into further analyses.

Recovery. Next, we compute some typical measures to test recovery in the Bayesian context. We present *correlations* between the true values and the posterior medians, *coverage* via the percentage of times across the datasets that the true value lies in the 50% and 95% highest density interval (HDI), respectively, as another measure of recovery, and the mean of the *Monte Carlo standard errors* (mMCSE) as a quantitative suggestion of how big the estimation noise in Markov chains is. The MCSE indicates the estimated SD of the posterior mean in the chain, where SD is the standard deviation of the posterior samples, and is interpreted on the scale of

Par.	r	50% [a]	95% [a]	mMCSE [b]
		— 100 Trials —		
a	.96	51	95	0.00486
v_1	.91	48	95	0.01796
v_2	.91	50	95	0.01821
t_0	.98	49	94	0.00063
w	.87	50	95	0.00120
s_v	.68	48	94	0.01964
s_w	.40	51	95	0.00564
s_{t_0}	.83	47	96	0.00133
		— 500 Trials —		
a	.99	49	94	0.00280
v_1	.97	49	95	0.01170
v_2	.97	49	94	0.01170
t_0	.99	49	94	0.00037
w	.97	50	94	0.00063
s_v	.84	50	94	0.01486
s_w	.62	50	95	0.00543
s_{t_0}	.95	49	95	0.00071

Table 4.3: Parameter recovery study: Evaluation criteria (Correlations, Coverage, mMCSE), for parameters estimated from 100, and 500 simulated trials, respectively.
Note. Par.=Parameters; r=Correlations (between true parameter values and posterior medians)
[a] Percent of simulated datasets with true value in the HDI of this percentage
[b] Mean of Monte Carlo standard error (mMCSE) across simulated datasets.

the parameter value (Kruschke, 2015; Vehtari et al., 2021). The MCSE is basically defined as $\mathrm{SD}/\sqrt{N_{\mathrm{eff}}}$. Results are shown in Table 4.3. Furthermore, we display the *bias* in terms of the difference between the posterior median and the true value in violin plots in Figure 4.4 and in Figure 4.5 for the datasets with 100 and 500 trials, respectively. Note the different scaling of the y-axes in the two figures. In Appendix 4.6.1, we present a *runtime analysis*.

The correlations show a similar pattern as already found in literature (e.g., Boehm et al., 2018): the three inter-trial variabilities show smaller correlations with the true values than the other model parameters. Nonetheless, in the analyses with 500 trials, correlations of .62 for s_w, .84 for s_v, and even .95 for s_{t_0} were obtained.

The coverage values meet the expectations in this setup with values between 49% and 50% for 500 trials in 50% HDI and between 94% and 95% for 500 trials in 95% HDI. The MCSE values show that the parameters are estimated with small standard errors that decrease with the number of trials.

MCSE quantifies the variability of parameter estimates calculated from the sample of the posterior distribution, whereas bias assesses systematic deviation of such estimates from the

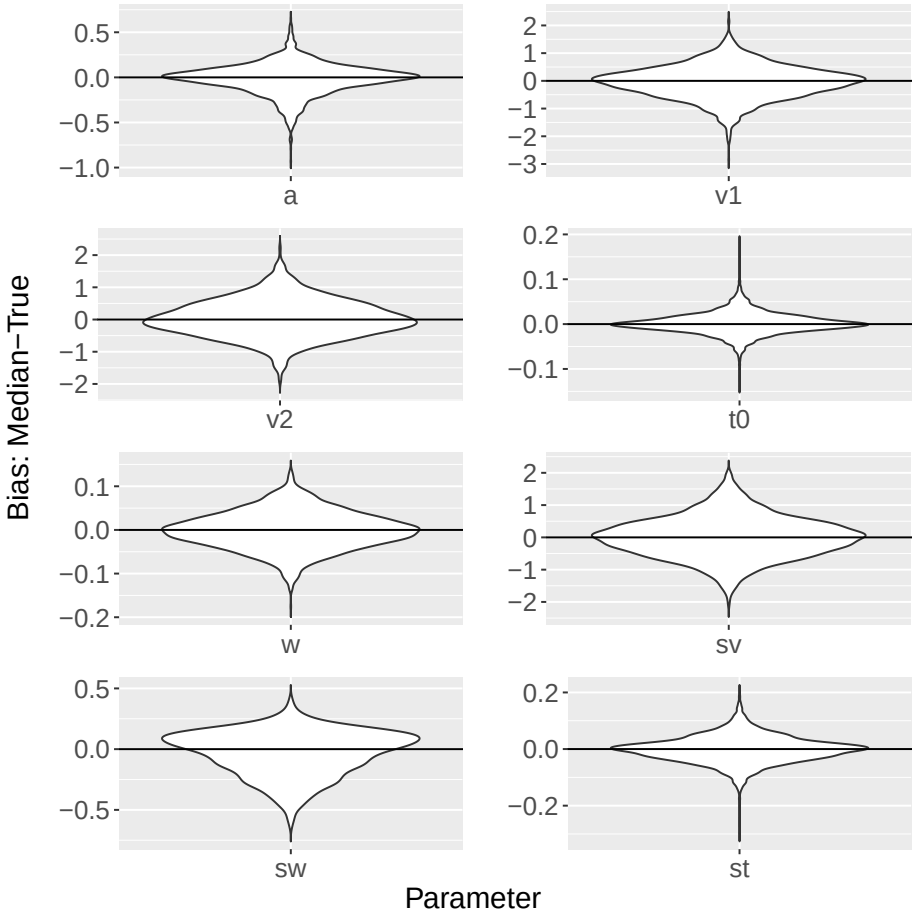


Figure 4.4: Violin plots of bias between posterior median and true value for 100 trials.

ground truth. The violin plots for 500 trials (Figure 4.5) show smaller biases than the violin plots for 100 trials (Figure 4.4). All plots except the plot for s_w have most of their mass at 0 and are quite symmetric, meaning that there is no sign of systematic over- or underestimation. The plots for s_w show small overestimation and a non-symmetric distribution of bias. This may reflect the non-symmetric prior distribution for s_w. The plots for v_1, v_2, and s_v show a relatively wide spread, whereas the plots for w, t_0, and s_{t_0} suggest that these parameters can be recovered with small absolute biases.

In summary, the results of the recovery study are in line with findings in the literature (e.g., Boehm et al., 2018). Specifically, the estimation of the inter-trial variability in relative starting point seems to be tricky in this setup. Nevertheless, results suggest that the new implementation is able to recover the parameters of the seven-parameter diffusion model. As expected, the parameter recoveries based on 500 trials are better than those based on 100 trials, that is, correlations are higher, coverage is better, and mean MCSEs and biases are smaller.

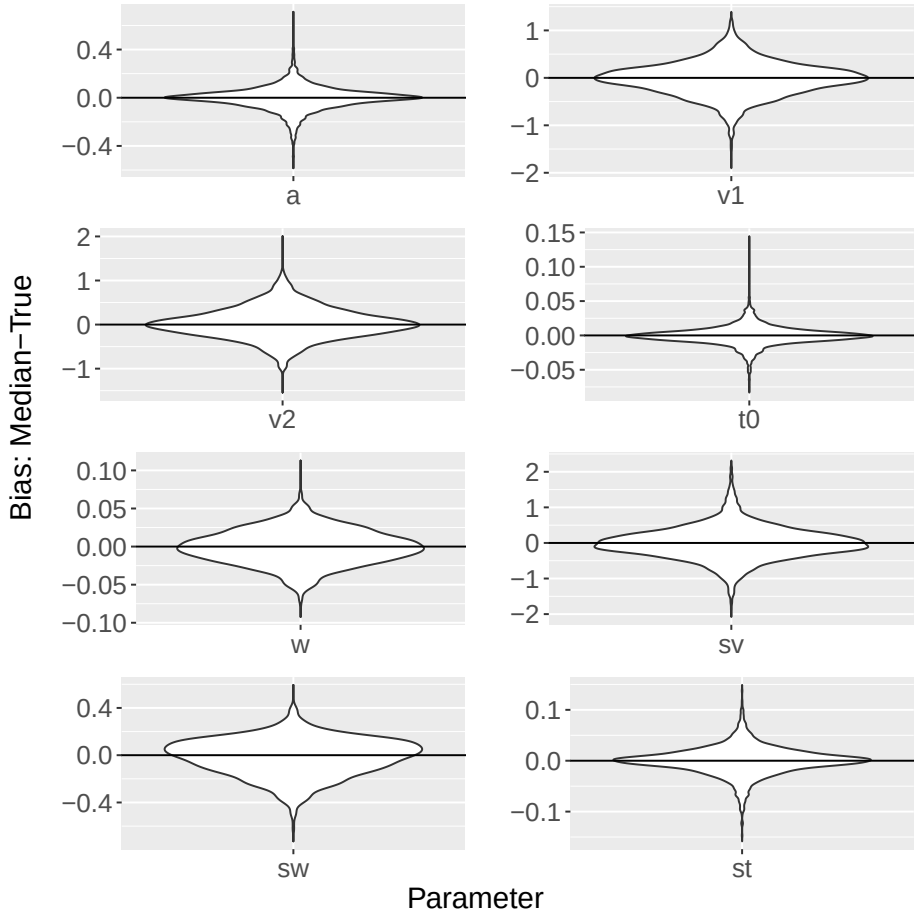


Figure 4.5: Violin plots of bias between posterior median and true value for 500 trials.

Simulation-Based Calibration Study

Recovery studies in Bayesian contexts are limited by the facts that it is difficult to conclude that a Bayesian algorithm is validly implemented from successful recovery and conversely that it is difficult to conclude that it is invalid from the occurrence of systematic bias (Talts et al., 2018). This is not true for simulation-based calibration (SBC, Talts et al., 2018), which tests whether the algorithm satisfies a consistency condition that a valid algorithm must respect. SBC can therefore reveal conclusive evidence for the invalidity of an invalid algorithm. To further illustrate the correct functioning of the new implementation of the seven-parameter diffusion model, we therefore performed a simulation-based calibration study.

SBC is a method to validate inferences from Bayesian algorithms that generate posterior samples. The method identifies inaccurate computations and inconsistencies in the implementation of the model. According to Talts et al. (2018), the only assumption for SBC is that there exists a generative model for the data. The procedure is to repeatedly sample parameters from the prior distributions, simulate data from these parameters, and fit the model to these datasets using the same priors from which the parameters were sampled. The analysis, if implemented

correctly, must satisfy the following *self consistency condition*:

$$\pi(\theta) = \int\int \pi(\theta \mid \tilde{y})\pi(\tilde{y} \mid \tilde{\theta})\pi(\tilde{\theta})\, d\tilde{y}\, d\tilde{\theta}, \tag{4.8}$$

where $\tilde{\theta} \sim \pi(\theta)$ are the parameters - denoted as the *ground truth* - sampled from the prior distribution, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$ are the data generated from the model using the ground truth, and $\theta \sim \pi(\theta \mid \tilde{y})$ the posterior samples. This condition implies that the prior sample $\tilde{\theta}$ and the posterior sample θ follow the same distribution. Modrak et al. (2022) proposed an extension of the SBC check such that the implication not only holds for the parameter space but also includes the data space. From this extension follows that the *rank statistic* r_{total} of the prior sample relative to the posterior sample, defined for any one-dimensional random variable with domain parameter and data space, $f : \Theta \times Y \to \mathbb{R}$,

$$\begin{aligned} r_{\text{less}}\big(\{f(\theta_1, y), \ldots, f(\theta_L, y)\}\big) &:= \sum_{l=1}^{L} \mathbb{I}\left[f(\theta_l, y) < f(\tilde{\theta}, y)\right] \in [0, L] \\ r_{\text{equals}}\big(\{f(\theta_1, y), \ldots, f(\theta_L, y)\}\big) &:= \sum_{l=1}^{L} \mathbb{I}\left[f(\theta_l, y) = f(\tilde{\theta}, y)\right] \in [0, L] \\ K &\sim \text{uniform}(0, r_{\text{equals}}) \\ r_{\text{total}} &:= r_{\text{less}} + K, \end{aligned} \tag{4.9}$$

should be uniformly distributed over the natural numbers in $[0, L]$, where L is the number of samples of the posterior distribution, and $\mathbb{I}$ is the indicator function taking the value 1 if the condition in the parentheses holds and the value 0 otherwise.

Our simulation study was designed to allow us to test this expectation. That is, given a correct implementation of the function `wiener_full_lpdf()`, the SBC should result in uniformly distributed rank values. Specifically, for each model parameter and for the log-density, we compute the rank statistic of the ground truth in the posterior sample, and the histogram of rank statistics as proposed by Modrak et al. (2022). The histogram should reveal a uniform distribution of the rank statistic if the algorithm is valid, whereas systematic deviations from the uniform distribution allow one to diagnose specific problems of the algorithm (Talts et al., 2018).

Since the MCMC-algorithm used in Stan produces autocorrelated samples, we have to thin our posterior samples to obtain (a smaller number of) independent draws from the posterior distribution. As mentioned above, we ensured that all effective sample sizes are above 400. Therefore, we uniformly thin the posterior samples to $L = 399$ high-quality draws according to Algorithm 2 by Talts et al. (2018), and compute the rank statistic as defined in Equation (6.19) for each of the N datasets and analyze the resulting histogram for uniformity. For the histograms, we set the number of bins to 100, so that across the 2000 simulated datasets, there are 20 observations expected per bin. In Figures 4.6 and 4.7, we add a gray band to the histogram

that covers 99% of the variation expected for each frequency in a histogram of a uniform distribution. Specifically, the band covers the interval from the 0.005 percentile to the 0.995 percentile of the Binomial(N; 100^{-1}) distribution.

Additionally, we calculate the χ^2-statistic for the differences between observed and expected frequencies of observations per bin for each parameter and for the log-density with expected frequencies given by the expected uniform distribution (i.e., 20 per bin). In this way, it is possible to check whether the SBC assumption is significantly violated, which would indicate that the posterior distributions are flawed. For each parameter, the observed χ^2 value is compared to the critical χ^2-value of 123.23, for $p = .95$ with $df = 99$ (number of bins minus 1). In sum, we calculate 18 χ^2-statistics, 9 for the 100 trials study and 9 for the 500 trials study. In order to check whether the aggregate of the individual χ^2-tests follows the hypothesis of uniformity, we also aggregated the resulting 18 p-values by means of Fisher's combined probability test (Fisher, 1950)[3].

Results and Discussion. The results of the SBC study for 100 trials are displayed in Figure 4.6 and for 500 trials in Figure 4.7. Visual inspection suggests that none of the histograms shows systematic deviation from the uniform distribution. This means that there is no clear pattern in the histograms that indicates some kind of bias as described in Modrak et al. (2022) and Talts et al. (2018).

Furthermore, the χ^2-statistic testing for uniformity is significant at the 5% level for only one out of 18 calculated statistics: the χ^2 value for s_w in the simulation with 500 trials was $\chi^2(99) = 128.6$ with $p = .024$. Note that with 18 tests at the 5% level, one significant result is well within the range of expectations. This is confirmed by Fisher's combined probability test. Across all 18 p-values, the combined test yielded a $\chi^2(36)$ of 34.65 with $p = .467$, indicating that the set of p-values is consistent with the composite hypothesis that all histograms follow a uniform distribution. Taken together, we conclude that there is little indication in these analyses that our DM algorithm is implemented incorrectly.

4.5 General Discussion

The purpose of this paper was to introduce a new implementation of the seven-parameter diffusion model in a Bayesian framework – the probabilistic programming language Stan. As mentioned in the introduction, this implementation overcomes a number of shortcomings of previous implementations of the DM. Unlike previous implementations in WinBUGS, JAGS, and Stan, the current implementation enables users to incorporate the variability parameters that define the seven-parameter version of the DM. Unlike the implementation via HDDM, the Stan framework provides the user with great flexibility in choosing priors and in implement-

[3] Thus, we test whether the test statistic $-2\sum_{i=1}^{18}\log(p_i)$ indicates a violation of the composite hypothesis that the p-values themselves stem from a uniform distribution as they should under the H_0 of uniformly distributed histograms. Under the H_0, the test statistic should follow a χ^2-distribution with 36 degrees of freedom.

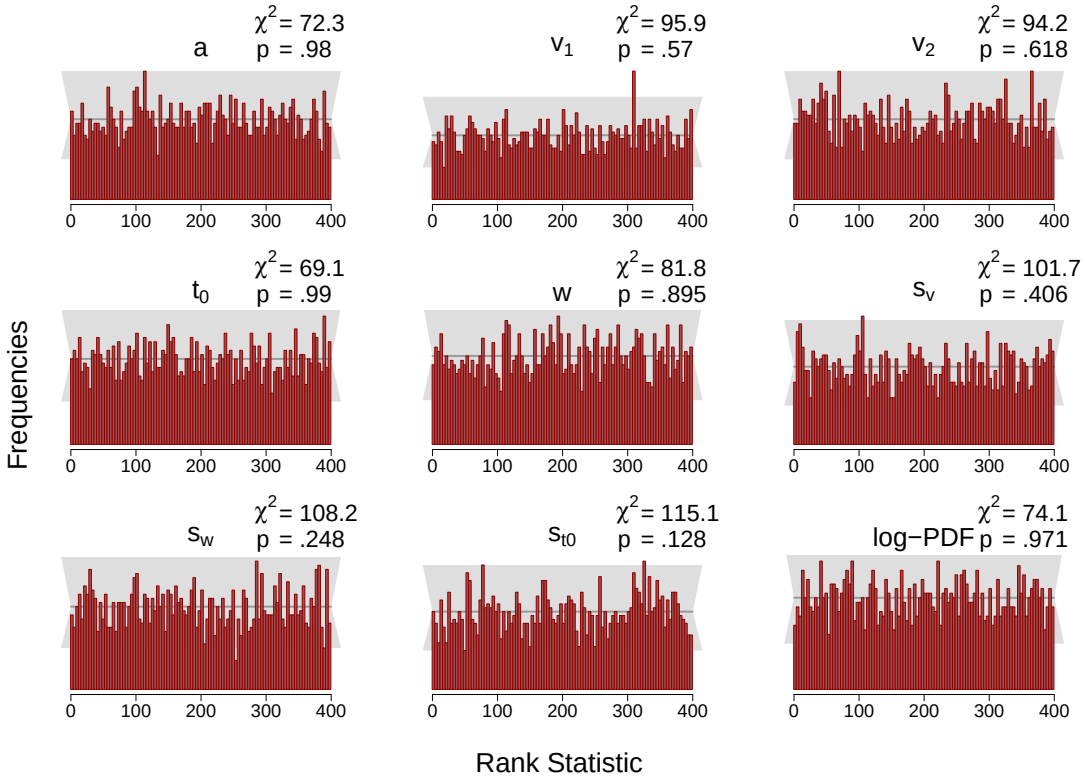


Figure 4.6: Histograms of the rank statistic for 100 trials. *Note.* The histograms indicate no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar).

ing complex hierarchically structured models. Additionally, the new implementation within the Stan framework allows users to benefit from all resources that are available for this platform, including libraries for model comparison (e.g., using loo), or (graphical) analyses of MCMC convergence. In the present paper, we described how to use the newly implemented Stan function and presented simulation studies that examined the recovery of the parameters and a correctness check of the implemented algorithm.

In summary, the results of the recovery study are in line with findings in the literature. We found satisfactory to good parameter recovery in terms of correlations, bias and coverage, with better recovery for the basic model parameters than for the variability parameters as previously observed (e.g., Boehm et al., 2018). Specifically, recovery of the inter-trial variability in relative starting point seems to be tricky in this setup. Nevertheless, simulation-based calibration does not show any systematic errors, suggesting that the implementation is correct and that bias in the estimation reflects the influence of the chosen prior. Furthermore, the results of the simulation-based calibration study suggest that the new algorithm is implemented correctly, and Stan is suitable for fitting DMs with its Hamiltonian MCMC algorithm.

The design of our simulation studies was constrained by the goals that we pursued in this brief article, namely to implement a number of validity checks of our algorithm. For this reason, the scope of this simulation study is limited to the case with informative priors in a simple non-

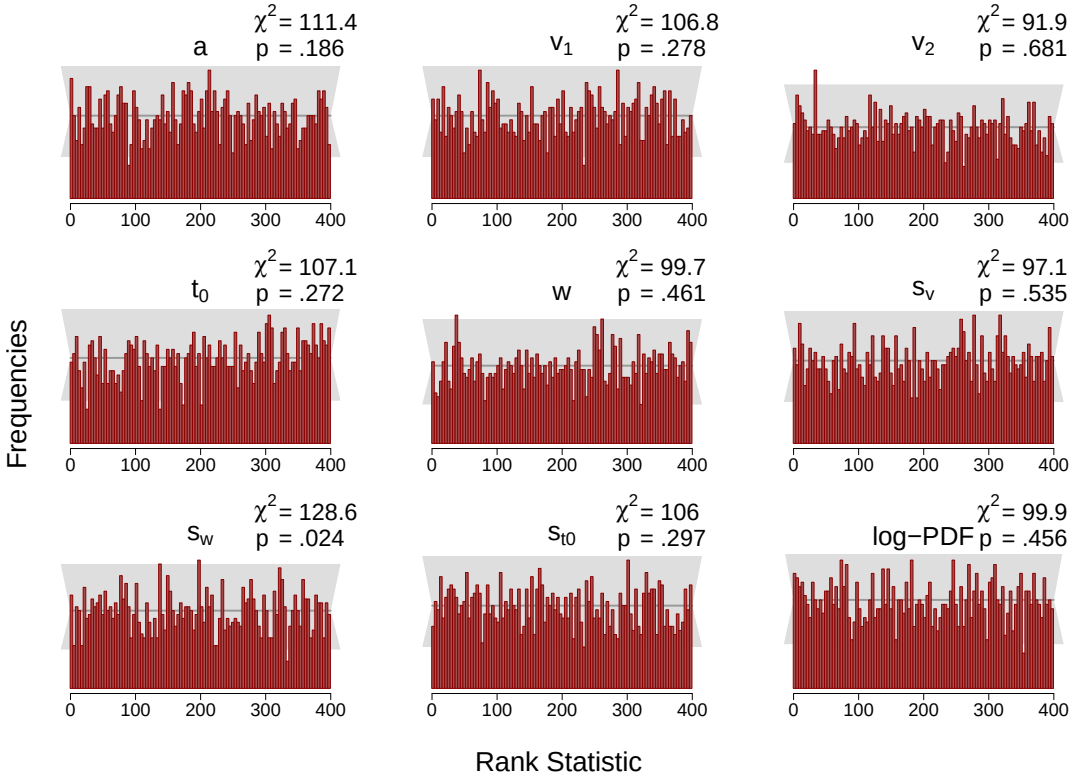


Figure 4.7: Histograms of the rank statistic for 500 trials. *Note.* The histograms indicate no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar).

hierarchical model with data that were generated from the DM model without contaminants as occur in real data. It is thus up to future research to examine the performance of the new Stan implementation in other settings; for example with less informative or even uninformative priors, with real data, with a hierarchical approach, or in comparison to other methods.

In conclusion, the implementation offers new opportunities to simultaneously examine response time and responses. We hope that it will prove to be a useful enrichment to the current modeling landscape.

4.6 Appendix

4.6.1 Runtime Analysis

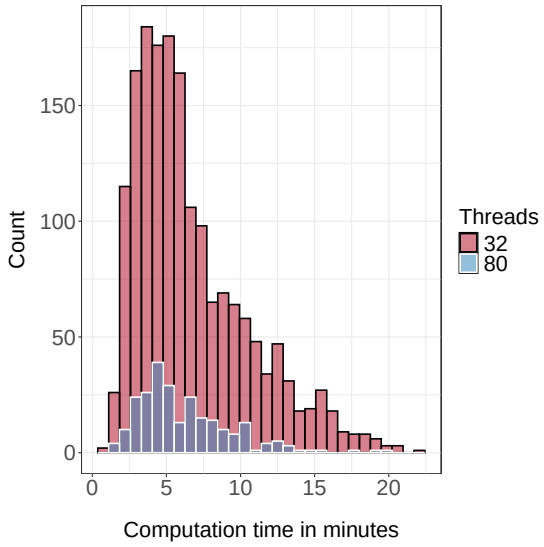


Figure 4.8: Histograms of runtimes for 100 trials. *Note.* Analyses were parallelized on 32, and 80 threads per chain, respectively. Mean runtime on 32 threads per chain is 6.7 minutes. Mean runtime on 80 threads per chain is 6.2 minutes.

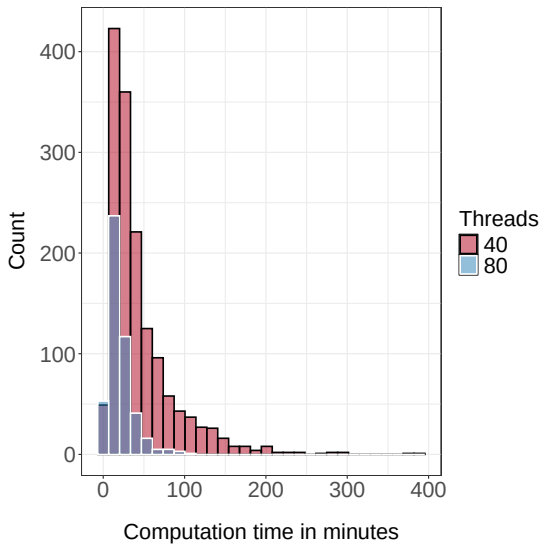


Figure 4.9: Histograms of runtimes for 500 trials. *Note.* Analyses were parallelized on 40, and 80 threads per chain, respectively. Mean runtime on 40 threads per chain is 44 minutes. Mean runtime on 80 threads per chain is 20 minutes.

References

Arnold, N. R., Bröder, A., & Bayen, U. J. (2015). Empirical validation of the diffusion model for recognition memory and a comparison of parameter-estimation methods. *Psychological Research*, *79*(5), 882–898. https://doi.org/10.1007/s00426-014-0608-y

Assink, N., van der Lubbe, R. H. J., & Fox, J.-P. (2015). Does time pressure induce tunnel vision? An examination with the eriksen flanker task by applying the hierarchical drift diffusion model. In *Proceedings of the International Conference on Neural Networks-Fuzzy Systems (NN-FS 2015)* (pp. 30–40).

Blurton, S. P., Kesselmeier, M., & Gondan, M. (2017). The first-passage time distribution for the diffusion model with variable drift. *Journal of Mathematical Psychology*, *76*, 7–12. https://doi.org/10.1016/j.jmp.2016.11.003

Boehm, U., Annis, J., Frank, M. J., Hawkins, G. E., Heathcote, A., Kellen, D., Krypotos, A.-M., Lerche, V., Logan, G. D., Palmeri, T. J., van Ravenzwaaij, D., Servant, M., Singmann, H., Starns, J. J., Voss, A., Wiecki, T. V., Matzke, D., & Wagenmakers, E.-J. (2018). Estimating across-trial variability parameters of the Diffusion Decision Model: Expert advice and recommendations. *Journal of Mathematical Psychology*, *87*(4), 46–75. https://doi.org/10.1016/j.jmp.2018.09.004

Brooks, S., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*, 434–455.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, *76*(1). https://doi.org/10.18637/jss.v076.i01

Carpenter, B., Hoffman, M. D., Brubaker, M., Lee, D., Li, P., & Betancourt, M. (2015). The stan math library: Reverse-mode automatic differentiation in C++. *arXiv*, *arXiv:1509.07164v1*.

Eriksen, B. A., & Erisken, C. W. (1974). Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception & Psychophysics*, *16*(1), 143–149.

Fisher, R. A. (1950). *Statistical methods for research workers* (11th ed.). Oliver and Boyd.

Forstmann, B. U., Ratcliff, R., & Wagenmakers, E.-J. (2016). Sequential Sampling Models in Cognitive Neuroscience: Advantages, Applications, and Extensions. *Annual Review of Psychology*, *67*, 641–666. https://doi.org/10.1146/annurev-psych-122414-033645

Gondan, M., Blurton, S. P., & Kesselmeier, M. (2014). Even faster and even more accurate first-passage time densities and distributions for the Wiener diffusion model. *Journal of Mathematical Psychology*, *60*, 20–22. https://doi.org/10.1016/j.jmp.2014.05.002

Hartmann, R., Johannsen, L., & Klauer, K. C. (2020). rtmpt: An R package for fitting response-time extended multinomial processing tree models. *Behavior Research Methods*, *52*(3), 1313–1338. https://doi.org/10.3758/s13428-019-01318-x

Hartmann, R., & Klauer, K. C. (2021). Partial derivatives for the first-passage time distribution in Wiener diffusion models. *Journal of Mathematical Psychology*, *103*, 102550. https://doi.org/10.1016/j.jmp.2021.102550

Heck, D. W., Arnold, N. R., & Arnold, D. (2018). TreeBUGS: An R package for hierarchical multinomial-processing-tree modeling. *Behavior Research Methods*, *50*(1), 264–284. https://doi.org/10.3758/s13428-017-0869-7

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623.

Johnson, D. J., Stepan, M. E., Cesario, J., & Fenn, K. M. (2020). Sleep Deprivation and Racial Bias in the Decision to Shoot: A Diffusion Model Analysis. *Social Psychological and Personality Science*, *17*(3), 194855062093272. https://doi.org/10.1177/1948550620932723

Klauer, K. C., & Kellen, D. (2018). RT-MPTs: Process models for response-time distributions based on multinomial processing trees with applications to recognition memory. *Journal of Mathematical Psychology*, *82*, 111–130. https://doi.org/10.1016/j.jmp.2017.12.003

Kruschke, J. K. (2015). *Doing bayesian data analysis: A tutorial with R, JAGS, and Stan* (2nd ed.). Elsevier.

Lerche, V., & Voss, A. (2019). Experimental validation of the diffusion model based on a slow response time paradigm. *Psychological Research*, *83*(6), 1194–1209. https://doi.org/10.1007/s00426-017-0945-8

Lerche, V., Voss, A., & Nagler, M. (2017). How many trials are required for parameter estimation in diffusion modeling? A comparison of different optimization criteria. *Behavior Research Methods*, *49*(2), 513–537. https://doi.org/10.3758/s13428-016-0740-2

Matzke, D., & Wagenmakers, E.-J. (2009). Psychological interpretation of the ex-Gaussian and shifted Wald parameters: A diffusion model analysis. *Psychonomic Bulletin & Review*, *16*(5), 798–817. https://doi.org/10.3758/PBR.16.5.798

Modrak, M., Moon, A. H., Kim, S., Bürkner, P., Huurre, N., Faltejskova, K., Gelman, A., & Vehtari, A. (2022). Simulation-Based Calibration Checking for Bayesian Computation: The Choice of Test Quantities Shapes Sensitivity. *arXiv*, *arXiv:2211.02383v1*.

Navarro, D. J., & Fuss, I. G. (2009). Fast and accurate calculations for first-passage times in Wiener diffusion models. *Journal of Mathematical Psychology*, *53*(4), 222–230. https://doi.org/10.1016/j.jmp.2009.02.003

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 113–162). Chapman & Hall/CRC.

Pleskac, T. J., Cesario, J., & Johnson, D. J. (2018). How race affects evidence accumulation during the decision to shoot. *Psychonomic Bulletin & Review*, *25*(4), 1301–1330. https://doi.org/10.3758/s13423-017-1369-6

R Core Team. (2021). *R: A Language and Environment for Statistical Computing*. https://www.R-project.org/

Ratcliff, R. (1978). A Theory of Memory Retrieval. *Psychological Review*, *85*(2), 59–108.

Ratcliff, R., & Rouder, J. N. (1998). Modelling response times for two-choice decisions. *Psychological Science*, *9*(5), 347–356.

Ratcliff, R., & Smith, P. L. (2004). A comparison of sequential sampling models for two-choice reaction time. *Psychological Review*, *111*(2), 333–367. https://doi.org/10.1037/0033-295X.111.2.333

Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review*, *12*(4), 573–604. https://doi.org/10.3758/BF03196750

Singmann, H., Brown, S., Gretton, M., & Heathcote, A. (2022). *rtdists: Response Time Distributions*. https://CRAN.R-project.org/package=rtdists

Stan Development Team. (2022a). Bayesplot: Visual MCMC Diagnostics (version 2.31). Retrieved December 1, 2022, from https://mc-stan.org/bayesplot/articles/visual-mcmc-diagnostics.html

Stan Development Team. (2022b). Stan Modeling Language Users Guide and Reference Manual (version 2.31). Retrieved November 11, 2022, from https://mc-stan.org

Stan Development Team. (2023a). Github Stan. Retrieved April 5, 2023, from https://github.com/stan-dev

Stan Development Team. (2023b). Github Stan Math. Retrieved April 5, 2023, from https://github.com/stan-dev/math

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating Bayesian Inference Algorithms with Simulation Based Calibration. *arXiv*, *arXiv:1804.06788v2*.

Vandekerckhove, J., & Tuerlinckx, F. (2007). Fitting the Ratcliff diffuion model to experimental data. *Psychonomic Bulletin & Review*, *14*(6), 1011–1026.

Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2011). Hierarchical diffusion models for two-choice response times. *Psychological Methods*, *16*(1), 44–62. https://doi.org/10.1037/a0021765

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *arXiv*, *arXiv:1903.08008v5*.

Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, *32*(7), 1206–1220.

Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, *39*(4), 767–775.

Wabersich, D., & Vandekerckhove, J. (2013). Extending JAGS: A tutorial on adding custom distributions to JAGS (with a diffusion model example). *Behavior Research Methods*, *46*(1), 15–28. https://doi.org/10.3758/s13428-013-0369-3

Wagenmakers, E.-J. (2009). Methodological and empirical developments for the Ratcliff diffusion model of response times and accuracy. *European Journal of Cognitive Psychology*, *21*(5), 641–671. https://doi.org/10.1080/09541440802205067

Wagenmakers, E.-J., van der Maas, H. L. J., & Grasman, R. P. P. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, *14*(1), 3–22.

White, C. N., Ratcliff, R., Vasey, M. W., & McKoon, G. (2010). Using diffusion models to understand clinical disorders. *Journal of Mathematical Psychology*, *54*(1), 39–52. https://doi.org/10.1016/j.jmp.2010.01.004

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in Neuroinformatics*, *7*, 14. https://doi.org/10.3389/fninf.2013.00014

Chapter 5

Paper: Comparing two Seven-Parameter Diffusion Model Implementations, `wiener()` in Stan, and HDDM

Henrich, F., Hartmann, R., Pratz, V., & Klauer, K.C. (2024). Comparing two Seven-Parameter Diffusion Model Implementations, `wiener()` in Stan, and HDDM. *Manuscript*.

Abstract

Diffusion models have been widely used to model the decision process based on data from two-alternative forced-choice tasks. In this article, we compare the newly implemented Stan-function `wiener()` with the already established toolbox HDDM. Both methods allow one to estimate up to seven parameters of the diffusion model in a Bayesian hierarchical framework. Four simulation experiments vary the number of participants, number of trials, and complexity of the model to compare how the methods perform in different contexts and to validate `wiener()` against HDDM as a benchmark. Overall, `wiener()` produces similar results as HDDM. The Stan-based algorithm, however, requires more time than HDDM in sampling for complex models and less time for simpler models. Finally, we present a simulation-based calibration study for HDDM.

Keywords: Ratcliff diffusion model · Bayesian hierarchical modeling · Stan function · HDDM · reproducibility · model fitting

5.1 Introduction

In this article, we compare the newly implemented Stan-function `wiener()` (Henrich et al., 2023) with the already established toolbox HDDM (hierarchical drift diffusion model, Wiecki et al., 2013). Both are methods to estimate up to seven parameters of the diffusion model (Ratcliff, 1978) in a Bayesian hierarchical framework. The new method based on the function `wiener()` is implemented in the probabilistic programming language Stan (Stan Development Team, 2022). For `wiener()`, we already conducted (a) a recovery study that showed satisfactory to good recovery for all seven model parameters, and (b) a simulation-based calibration (Talts et al., 2018) study that attested to the correctness of the implemented algorithm (Henrich et al., 2023). This third part aims to embed the new implementation into the landscape of existing tools for estimating the diffusion model parameters. There are many implementations of the diffusion model that are listed below. As HDDM is one of the most popular ways for fitting the seven-parameter diffusion model and frequently used in the literature (more than 1000 citations), we chose HDDM as a benchmark method for our comparison. Furthermore, HDDM is the method that is most similar in functionality to `wiener()` and therefore its closest competitor. There already exists a simulation study that shows that "HDDM can recover parameters better than the commonly used alternatives (i.e. maximum likelihood and χ^2-Quantile estimation)" (p.9, Wiecki et al., 2013). In order to compare `wiener()` with HDDM, we closely stick to the setup of the aforementioned simulation study.

This paper is structured as follows. First, the diffusion model is briefly introduced. Second, the two software packages, Stan and HDDM, are described. Third, the aim and design of the numerical experiments presented here are described and results are reported. Fourth, a simulation-based calibration study for HDDM is reported.

5.2 The Diffusion Model

The diffusion model (Ratcliff, 1978) is a cognitive model in the family of information accumulation models. It is widely used to model two-alternative forced-choice tasks by simultaneously modeling reaction times and responses observed in these tasks. Examples comprise the Eriksen flanker task (Eriksen & Erisken, 1974) and the lexical decision task (Rubenstein et al., 1970), among many others in behavioral research. A detailed review of the diffusion model and the many areas of research to which it has been applied is given by Ratcliff et al. (2016).

The diffusion model is based on the assumption that evidence is accumulated over time until the decision maker has enough evidence to make a decision. The basic model consists of four parameters that describe the accumulation process: The *boundary separation*, a, the *relative starting point*, w, the *drift rate*, v, and the *non-decision time*, t_0. As shown in Figure 5.1, the decision process can be depicted as a jagged line that is limited by two boundaries which represent the two response alternatives. The boundary separation, a, describes the distance between

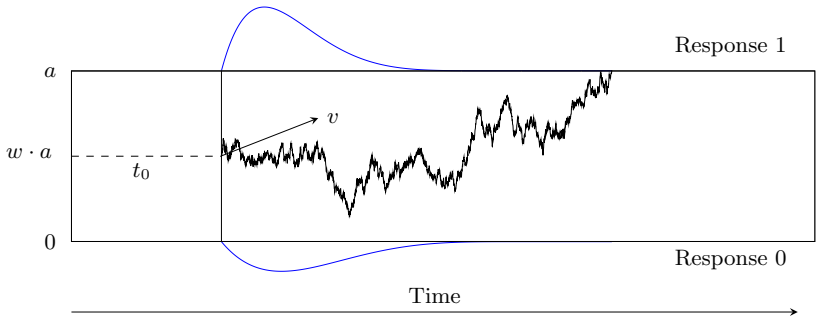


Figure 5.1: Realization of a four-parameter diffusion process modeling the binary decision process. *Note.* The parameters are the *boundary separation a* for two response alternatives, the *relative starting point w*, the *drift rate v*, and the *non-decision time* t_0. The evidence accumulation process is illustrated as a jagged line between the two boundaries (example for one trial). The predicted distributions of the reaction times are depicted as curved lines below and above the response boundaries (blue).

both boundaries. The process starts with a bias to one or the other boundary, represented by the relative starting point, w. If $w = 0.5$, there is no bias and the process starts in the middle between both boundaries. The direction and speed of the decision process is determined by the drift rate v, which denotes the average rate of information uptake. Once a threshold is reached, the decision process is completed and the response associated with the threshold is executed. Processes that do not belong to the decision process, like encoding or motor time, are summed up in the non-decision time parameter t_0. The model predicts the reaction time distributions for both response alternatives and the probabilities with which either response is made. Several studies have shown that the model parameters can be interpreted in terms of different cognitive processes (e.g., Arnold et al., 2015; Lerche & Voss, 2019; Voss et al., 2004).

Nevertheless, there are several patterns in reaction time that cannot be modeled by the four-parameter diffusion model. For example, the model cannot explain crossovers in reaction time patterns which yield fast errors at high levels of accuracy, and slow errors at low levels of accuracy. For such reasons, Ratcliff and Rouder (1998) extended the four-parameter model by three inter-trial variabilities: *inter-trial variability in drift rate*, s_v, *inter-trial variability in relative starting point*, s_w, and *inter-trial variability in non-decision time*, s_{t_0}. The seven-parameter diffusion model is able to overcome the limitations of the four-parameter diffusion model and can explain the aforementioned reaction time patterns.

To estimate the model parameters several user-friendly software packages have been developed. Early, non-Bayesian implementations are given in DMAT (Vandekerckhove & Tuerlinckx, 2007), fast-dm (Voss & Voss, 2007), or EZ (Wagenmakers et al., 2007). Bayesian implementations are given in WinBUGS (Vandekerckhove et al., 2011), JAGS (Wabersich & Vandekerckhove, 2013), Stan (Carpenter et al., 2017), HDDM (Wiecki et al., 2013), HSSM (Fengler et al., 2023), or R-packages as EMC2 (Stevenson et al., 2024) DMC (Heathcote et al., 2019), and ggdmc (Lin & Strickland, 2020), among others.

Unfortunately, the implementations in WinBUGS, JAGS, and Stan were limited to the four-

parameter model without inter-trial variabilities, whereas HDDM and HSSM are pure Python implementations. HSSM is the young successor of HDDM with state of the art implementations of the algorithms. Both packages are limited in the choice of priors that users can specify. However, in HSSM it is planned for the future to enable a free choice of priors.

In the R-package EMC2, the diffusion model is implemented with nine parameter types. Besides the four basic parameters and the three inter-trial variabilities there are also parameters for the within-trial standard deviation of the drift rate and for the relative difference in non-decision time between responses.

All the hierarchical implementations can fit each participant's data independently, fully hierarchically or partially hierarchically. So, the programs not only allow the basic model parameters to be hierarchical, but also the inter-trial variabilities, or all combinations of hierarchical and non-hierarchical parameters. Nevertheless, it is quite uncommon to define a model with all seven parameters being hierarchical. Often, there is not enough information in the participant's data to meaningfully estimate the variabilities on a per-subject basis. Furthermore, defining the variabilities hierarchically makes the sampling process very slow as computations are very complex.

Thus, the landscape of diffusion model implementations is quite large. However, a flexible implementation of the seven-parameter diffusion model in the Bayesian hierarchical framework Stan that can be used with different programming languages was still missing. In order to overcome this caveat we chose to implement the seven-parameter diffusion model in Stan (Henrich et al., 2023), filling the gap with the function `wiener()`. With this implementation it is possible to define a model in a way that all seven parameters are hierarchical. The user decides in the model specification which parameters are included in the model and which of them are hierarchical and which non-hierarchical.

As Stan is an open-source programming language it also gives the opportunity to build on this implementation and add the distribution function of the seven-parameter diffusion model in the future in order to enable modeling truncated and censored data.

Henrich et al. (2023) performed two validation checks of the new implementation. The first check documented good recovery in a basic setup (we used a non-hierarchical model with informative priors). The second check showed the correctness of the implementation via a simulation-based calibration study.

The next interesting question is how this new implementation performs in comparison to existing methods. A close alternative that is well-tested and widely established is HDDM, which is also based on Bayesian algorithms that allow for hierarchical estimation of all seven diffusion model parameters. The main differences between HDDM and Stan are (a) that Stan has many interfaces to different computing environments, whereas HDDM is Python-based, and (b) that in Stan it is quite easy for the user to set the prior distributions for the model parameters, whereas HDDM limits the user to a small choice of predefined prior distributions.

The aim of the present article is to compare the new Stan function, `wiener()`, with the existing Python-based toolbox HDDM. The focus lies on the validation of the functionality of

`wiener()` and to show that it produces reliable results, comparable to the well-tested HDDM (see e.g., Ratcliff & Childers, 2015). For this purpose, we conducted four simulation studies fitting simulated data with `wiener()` and with HDDM.

5.3 The Software Packages

5.3.1 Stan

We implemented the seven-parameter diffusion model in the probabilistic programming language Stan, which is user-friendly, open-source, free, and is based on a highly efficient algorithm for Markov chain Monte Carlo estimation (MCMC, Neal, 2011), the No-U-Turn Sampler (NUTS, Hoffman & Gelman, 2014). Stan is a programming language for statistical modeling and high-performance statistical computation and provides the user with many functions for full Bayesian statistical inference and hierarchical modeling for many model families (not only the diffusion model). It runs on all major platforms and has interfaces with the most popular data analysis languages (e.g., R, Python, shell, MATLAB, Julia, Stata). Since `wiener()` is embedded in the Stan libraries, it can be used with any of the interfaces. Usage is easy and flexible. Users are allowed to define models and set priors freely (see Appendix 5.9.1 for an example model). Furthermore, Stan allows for parallel processing on two levels: (a) multiple chains can be run in parallel, and (b) each chain itself can be parallelized over as many cores as are available via the internal `reduce_sum` routine.

5.3.2 HDDM

HDDM stands for *hierarchical drift diffusion model* and is also free and open-source. It is a software package written in Python that comprises several Bayesian hierarchical model formulations for the diffusion model and the linear ballistic accumulator model (Brown & Heathcote, 2008). Here, we are interested in its diffusion model implementations. In HDDM, the Markov chain Monte Carlo slice sampling method is implemented via PyMC (Patil et al., 2010). To assess model fit some commonly used statistics and plotting functionalities are available in the package.

The user has several options to define dependencies, different conditions, or linear regressions in the model. Furthermore, the user can decide whether they wants to use either informative priors or non-informative priors. Both sets of priors are internally fixed in HDDM. The implemented informative priors are specified in Table 5.1, and are based on findings in the literature as summarized by Matzke and Wagenmakers (2009).

Finally, there exists an embedding of HDDM into docker together with the Bayesian modeling Python package ArviZ (Kumar et al., 2019), called dockerHDDM (Chuan-Peng et al.,

2022), that circumvents compatibility issues during installation and provides richer data analysis functions and data visualization tools. Additionally, unlike HDDM itself, dockerHDDM allows one to run multiple chains in parallel. Note that a single chain cannot be parallelized as is possible in Stan. We will use dockerHDDM in our simulation study for the HDDM analyses.

5.4 Comparison Study

The aim of the comparison study is to test whether `wiener()` is on a par with HDDM. Our focus is therefore on the performance of `wiener()` relative to HDDM assuming that HDDM gives good results for the analyses (as suggested by Wiecki et al., 2013). For this comparison study, it is thus of secondary importance how well the methods perform in recovering underlying parameter values in absolute terms. For example, if HDDM does not perform very well in recovering underlying parameters in our simulation study, but `wiener()` produces similar results as HDDM, then the outcome is positive, as it means that `wiener()` is on par with HDDM. We closely follow procedures used by Wiecki et al. (2013) for validating HDDM and thus, focus on hierarchical models (in contrast to the above-mentioned recovery study, where the focus was on non-hierarchical models). For this purpose, we will present four simulation studies referred to as experiments in what follows. For Experiments 1 and 2 we use the same design as in the first two experiments reported by Wiecki et al. (2013) and thus, a small version of the diffusion model with less than seven parameters (see Figure 5.2). Since `wiener()` is a seven-parameter implementation, we are also interested in the performance of the full model. Therefore, for Experiments 3 and 4, we extend the design and run analogous studies for the full diffusion model (see Figure 5.3). In order to obtain comparable results, we choose the priors and the model structures in Stan to match those of HDDM (see Wiecki et al., 2022).

Note that the aim is *not* to show that one method outperforms the other. Rather, the aim is to validate the Stan implementation against the established benchmark method HDDM.

5.4.1 Model structures

The first two experiments use the same model. Experiment 1 examines differing trial numbers and Experiment 2 differing participant numbers. The design of both experiments consists of two conditions with one stimulus each.

In Condition 1, the stimulus provides medium evidence for the upper response (i.e., its *drift rate* is positive and medium-sized). In Condition 2, the stimulus provides strong evidence for the upper response (i.e., is an *easier* stimulus with *doubled drift rate*). All other parameters are shared across conditions.

Following Wiecki et al. (2013), we generate and fit data with both methods hierarchically with a small version of the diffusion model consisting of four model parameters: *Boundary separation*, a, *drift rates*, v_1, and v_2, *non-decision time*, t_0, and *inter-trial variability in drift*

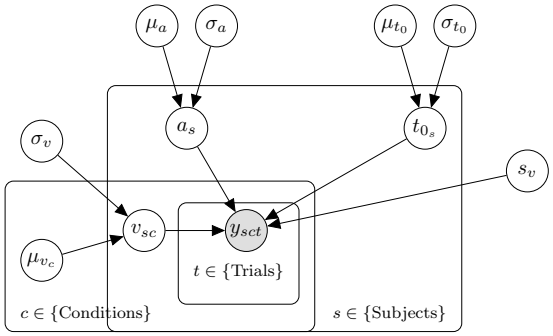


Figure 5.2: Graphical model representation of the small model used in Experiments 1 and 2 in Wiecki et al. (2013). *Note.* $a =$ boundary separation; $t_0 =$ non-decision time; $v =$ drift rate; $s_v =$ inter-trial variability in drift rate. The data y consist of response and response time for each trial.

rate, s_v, which is defined as a group-level parameter (as depicted in Figure 5.2). The *relative starting point w* is fixed at 0.5 and the two remaining *inter-trial variabilities in relative starting point*, s_w, and *non-decision time*, s_{t_0}, are set to zero.

In Experiments 3 and 4, we perform analogous analyses for the full diffusion model with all seven model parameters. Hence, we augment the above small model by parameters for the *relative starting point*, w, the *inter-trial variability in relative starting point*, s_w, and the *inter-trial variability in non-decision time*, s_{t_0}. Again, all three inter-trial variabilities are defined as group-level parameters (see Figure 5.3). It is a natural choice to define the inter-trial variabilities as group-level parameters, meaning non-hierarchically, although the implementation in Stan also allows to define the inter-trial variabilities per person (hierarchically) in the model declaration. One reason for this is that often times there is too less information in the data on participant level to estimate the variabilities per person meaningfully. Another, quite practical reason is that the computation time increases to an unrealistic amount. Therefore, the full diffusion model is defined partially hierarchical in the following analyses.

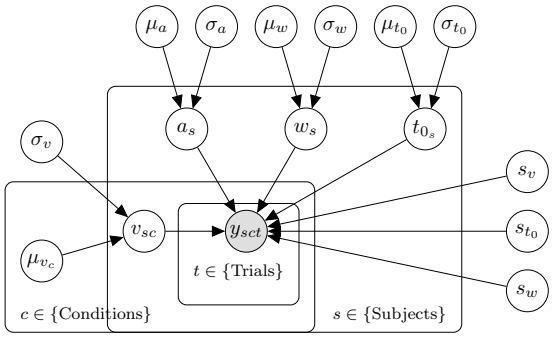


Figure 5.3: Graphical model representation of the full model used in Experiments 3 and 4. *Note.* $a =$ boundary separation; $t_0 =$ non-decision time; $v =$ drift rate; s_v, s_w, s_{t_0} are inter-trial variabilities in drift rate, starting point, and non-decision time, respectively. The data y consist of response and response time for each trial.

Group Means	Group Variances	Individual Parameters	Variabilities
$\mu_a \sim \mathscr{G}(1.5, 0.75)$	$\sigma_a \sim \mathscr{HN}(0.1)$	$a_s \sim \mathscr{G}(\mu_a, \sigma_a^2)$	-
$\mu_v \sim \mathscr{N}(2, 3)$	$\sigma_v \sim \mathscr{HN}(2)$	$v_s \sim \mathscr{N}(\mu_v, \sigma_v^2)$	$s_v \sim \mathscr{HN}(2)$
$\mu_w \sim \mathscr{N}(0.5, 0.5)$	$\sigma_w \sim \mathscr{HN}(0.05)$	$w_s \sim \text{invlogit}(\mathscr{N}(\mu_w, \sigma_w^2))$	$s_w \sim \mathscr{B}(1, 3)$
$\mu_{t_0} \sim \mathscr{G}(0.4, 0.2)$	$\sigma_{t_0} \sim \mathscr{HN}(1)$	$t_{0_s} \sim \mathscr{N}(\mu_{t_0}, \sigma_{t_0}^2)$	$s_{t_0} \sim \mathscr{HN}(0.3)$

Table 5.1: Informative Priors in the Model Declaration of HDDM. *Note.* $\mathscr{N}$ = normal distribution parameterized by mean and standard deviation, $\mathscr{HN}$ = half-normal distribution parameterized by standard deviation, $\mathscr{G}$ = Gamma distribution with mean μ_{gamma} and variance σ^2_{gamma}, $\mathscr{B}$ = Beta distribution parameterized by two shape parameters.

5.4.2 Priors

In HDDM, the user can choose between two sets of priors for the analyses: informative priors and non-informative priors. The priors are internally fixed and users cannot easily change them. Following Wiecki et al. (2013), we focus on the informative priors.

The informative priors that are given in the HDDM documentation for the full diffusion model are shown in Table 5.1. For the small model all priors in Table 5.1 that are not needed are ignored. We implement the models and priors in Stan following the HDDM documentation as closely as possible. The Stan model code is provided in Appendix 5.9.1, and the HDDM model specifications in Appendix 5.9.2.

In order to make the Stan-algorithm itself more efficient, the two model parameters *drift rate* and *relative starting point* are reparameterized via a non-centered parameterization (Stan Development Team, 2024). That is, a parameter, say θ, is reparameterized in the following way:

$$\theta = \theta_{\text{raw}} \cdot \sigma_\theta + \mu_\theta, \tag{5.1}$$

where θ_{raw}, the new parameter, is standard normally distributed, and μ_θ and σ_θ are the parameters of the prior of θ. This procedure shifts the data's correlation with the parameters to the hyper-parameters. Note finally, that for the *non-decision time*, we hand over the minimal reaction time per person and let the starting value depend on this value (see Appendix 5.9.1).

5.4.3 Ground Truths and Datasets

In Experiments 1 and 2, we simulate data in the way Wiecki et al. (2013) do using the same prior distributions as these authors.

Experiment 1 - Trial conditions, small model.

The simulated datasets generated in Experiment 1 stem from 30 different sets of (group-level) parameters. From each parameter set, seven datasets that differ in the number of trials per participant are generated . The number of participants per dataset is fixed to twelve. This results in 30×7 datasets generated as follows:

1. Draw 30 group-level parameter sets from the priors in Table 5.2 (upper part, left hand side).

2. For each of the 30 group-level parameter sets, generate twelve participant ground truths (see Table 5.2, upper part, right hand side).

3. Use the ground-truth values to generate datasets with 12 participants and trial numbers $\in \{20, 30, 40, 50, 76, 100, 150\}$.[1]

Note that following Wiecki et al. (2013), datasets in Experiments 1 and 2 are simulated such that participants have different individual inter-trial variability parameters in drift rate. This means that there exist true values for the inter-trial variability in drift rate per person, which is a realistic assumption for real data. But, following Wiecki et al. (2013), the hierarchical model only estimates the group-level parameter for the inter-trial variability in drift rate, s_v, and not an individual s_{v_s} for each person. This is due to the trade-off between model complexity and amount of information in the data (see also Boehm et al., 2018).

Experiment 2 - Participant conditions, small model.

The simulated datasets for Experiment 2 also stem from 30 different sets of (group-level) parameters. The simulated datasets now differ in the number of participants per hierarchical analysis (in six steps), whereas the number of trials per participant is fixed to 76 in each condition.[2] This results in 30×6 datasets generated as follows:

1. Draw 30 group parameter sets from the priors in Table 5.2 (upper part, left hand side).

2. For each of the 30 group parameter sets, generate ground truths for different participant numbers $\in \{8, 12, 16, 20, 24, 28\}$ (see Table 5.2, upper part, right hand side).

3. Use these ground-truth values to generate datasets with 76 trials per participant and participant numbers $\in \{8, 12, 16, 20, 24, 28\}$.

---

[1] Wiecki et al. (2013) used 75 trials. We decided to take 76 trials such that each drift rate condition comprises equal numbers of trials.

[2] See Footnote 1.

Experiments 3 and 4 go beyond Wiecki et al. (2013) in that the seven-parameter version of the diffusion model is used to generate and fit the data. For this purpose, we adjust the prior distributions for the ground truths from Experiments 1 and 2 (since some priors produced unrealistic parameter values) and add informative prior distributions for the three missing model parameters (based on Lerche et al., 2017; Matzke & Wagenmakers, 2009) in order to fit the full diffusion model. In these two experiments, only 4 parameter sets instead of 30 are used due to the higher model complexity implying much longer computation time.

Experiment 3 - Trial conditions, full model.

Experiment 3 uses four group-level parameter sets and like in Experiment 1, there are seven conditions varying by trial number; participant number is fixed to twelve. This results in 4×7 datasets generated as in Experiment 1:

1. Draw four group-level parameter sets from the priors in Table 5.2 (left hand side).

2. For each of the four group parameter sets, generate twelve participant ground truths (see Table 5.2, right hand side).

3. Use the ground-truth values to generate datasets for different trial numbers (see Experiment 1).

Note that like for s_v in Experiments 1 and 2, each participant has different inter-trial variabilities, s_v, s_w, and s_{t_0}, whereas in the model only group-level variabilities are estimated.

Experiment 4 - Participant conditions, full model.

Experiment 4 also uses four group-level parameter sets, and like in Experiment 2, there are six conditions varying by participants; trial number is fixed to 76. This results in 4×6 datasets generated as in Experiment 2:

1. Draw four group-level parameter sets from the priors in Table 5.2 (left hand side).

2. For each of the four group parameter sets, generate participant ground truths as in Experiment 2.

3. Use these ground-truth values to generate datasets with 76 trials per participant and participant numbers $\in \{8, 12, 16, 20, 24, 28\}$.

We simulated data with the sampling method `sampWiener()` of the R-package WienR (Hartmann & Klauer, 2021).

GP	Prior	PP	Prior
v_1	$\mathscr{U}(0.1,0.5)$	v_{js} [b]	$\mathscr{N}(v_j,0.2)$
a	$\mathscr{U}(0.5,2)$ [a]	a_s [c]	$\mathscr{N}(a,0.2)\mathrm{T}[0,]$
t_0	$\mathscr{U}(0.2,0.5)$	t_{0_s} [d]	$\mathscr{N}(t_0,0.1)\mathrm{T}[0,]$
s_v	$\mathscr{U}(0,2.5)$	s_{v_s}	$\mathscr{N}(s_v,0.1)\mathrm{T}[0,]$
v_2	equals $2 \cdot v_1$		
w	$\mathscr{U}(0.4,0.6)$	w_s	$\mathscr{N}(w,0.1)\mathrm{T}[0.2,0.8]$
s_w	$\mathscr{U}(0,\min(0.5,2w,2(1-w)))$	s_{w_s}	$\mathscr{N}(s_w,0.1)\mathrm{T}[0,\min(2w_s,2(1-w_s),0.5)]$
s_{t_0}	$\mathscr{U}(0,0.2)$	$s_{t_{0_s}}$	$\mathscr{N}(s_{t_0},0.1)\mathrm{T}[0,]$

Table 5.2: Prior Distributions for the Simulation of the Data in all Experiments for Group Parameters (GP) and Participant Parameters (PP). *Note.* T[,] denotes a truncation. Priors in the upper part of the table are taken from Wiecki et al. (2013), priors in the lower part of the table are based on Lerche et al. (2017), and Matzke and Wagenmakers (2009). For Experiments 1 and 2, the parameters s_w, and s_t are set to zero, and w is set to 0.5. For Experiments 3 and 4 the priors for these three parameters are chosen according to the lower part of the table.
[a] In their paper, Wiecki et al. (2013) report $a \sim \mathscr{U}(0.5,0.2)$, but they probably mean $a \sim \mathscr{U}(0.5,2)$.
[b] For drift-rate condition $j \in \{1,2\}$, and participant $s \in \{1,\ldots,n\}$, where n is the number of participants.
[c] For Experiments 3 and 4, this parameter is truncated from below at 0.1.
[d] For Experiments 3 and 4, this parameter is truncated from below at 0.15.

5.5 Criteria for the Comparison

MCMC Criteria

MCMC chains were required to meet two criteria (e.g., Vehtari et al., 2021): The first criterion is the effective sample size N_{eff}. This diagnostic gives the number of independent samples from the posterior distribution that would be expected to yield the same standard error of the posterior mean as is obtained from the dependent samples returned by the MCMC algorithm (Bürkner, 2017). To obtain reliable and stable estimates, the effective sample size should be larger than 400, $N_{\text{eff}} > 400$ (Vehtari et al., 2021).

The second criterion is the $\hat{R}$ value as a measure of convergence. This value is recommended to be smaller than 1.01, $\hat{R} < 1.01$ (Vehtari et al., 2021). Note that this recommended value is much tighter than the value of $\hat{R} < 1.1$ first recommended by Brooks and Gelman (1998).

We check these criteria for both methods. Analyses with posterior samples which do not meet the above criteria were repeated with an increased value for the method parameter specifying the desired *sampling iterations*. In Stan, we started with 1000 warmup iterations and 1000 sampling iterations. Each time a dataset had to be restarted we increased the sampling iterations by 1000. In HDDM, we started with 1500 burnin iterations and 1000 sampling iterations. Dif-

ferent samplers are implemented in both methods, and HDDM usually needs more iterations to converge. Therefore, each time an analysis had to be restarted in HDDM, we increased the sampling iterations by 2500.

Comparison Criteria

Like Wiecki et al. (2013), we report a measure for the error - the *mean error* - and the *detection rate*. The *mean error* is computed as the difference between the true value and the posterior median, averaged over all datasets and participants within datasets for the participant-level parameters, and over all datasets for the group-level parameters, per (trial number or participant number) condition. The *detection rate* is the proportion of times that the difference between the group-level drift rates ($v_1 - v_2$) is detected (remember: group-level drift rate v_2 is twice as large as group-level drift rate v_1). For this purpose, we count how often zero lies outside the 95% highest density interval (HDI) of the difference between both group-level drift rate parameters and divide by the total number of analyses per condition (that is by 30 in Experiments 1 and 2, and by 4 in Experiments 3 and 4). Furthermore, we present *correlations* between the true value and the posterior median, a *runtime analysis* comprising runtimes and sampling iterations, and an effectivity analysis via the *relative effective sample sizes*.

We performed these analyses on the high-performance computing cluster in Karlsruhe, Germany, BwUniCluster2.0,[3] with 4 cores per model fit. Due to very long runtimes in 10 of the Stan-analyses from Experiments 3 and 4, we ran these analyses on up to 25 cores per chain on another linux computer. As mentioned above, we used the program dockerHDDM for the HDDM-analyses, which embeds HDDM into a docker container to enable the user to run parallel chains and do more advanced data analyses than it is possible in stand-alone HDDM (Chuan-Peng et al., 2022).

5.6 Results and Discussion

MCMC Criteria

All analyses eventually reached the thresholds for the diagnostic criteria: $N_{\text{eff}} > 400$, and $\hat{R} < 1.01$. For a few datasets, both methods had troubles finding suitable starting values. For HDDM, the omission of the function `find_starting_values()` solved the problem and analyses started properly. To solve the problem for Stan, we first ran a fast-dm analysis (Voss & Voss, 2007) on the problematic datasets. We added some noise to the fast-dm estimates and used these values as starting values for our analyses.

[3] Retrieved September 23, 2025 from https://wiki.bwhpc.de/e/Registration/bwUniCluster

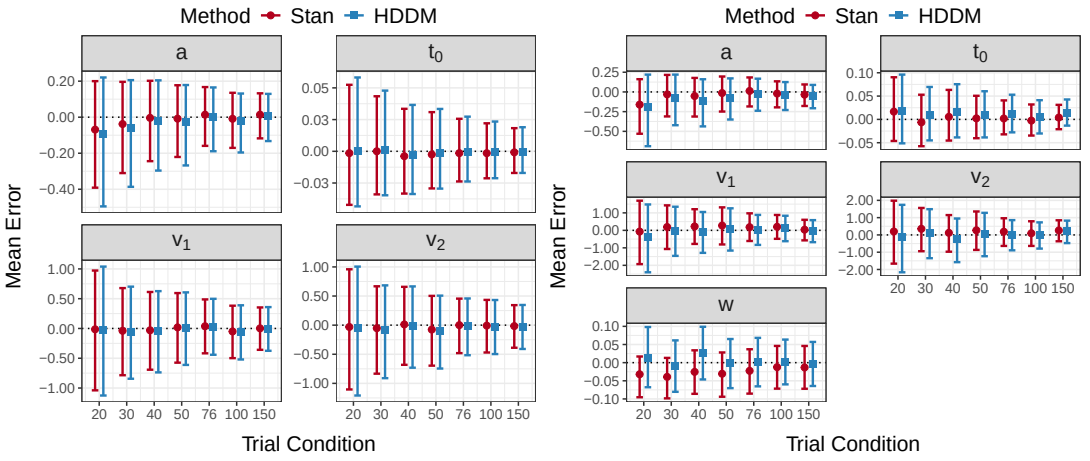
Comparison Criteria

In the following, we report the results for the above defined comparison criteria. The report is split into results for participant-level parameters and group-level parameters. Major results will be reported here in the body of the paper; supplemental results are placed in the appendix. For *mean errors* and *correlations*, results for the group-level parameters for Experiments 3 and 4 are not very informative since the criteria rely on only four quite variable observations per condition. Therefore, and for reasons of space, these results will not be reported in this article, but can be found on https://osf.io/vb4ex/ (Retrieved September 23, 2025).

Mean errors for the participant-level parameters in Experiments 1 and 3 are shown in Figure 5.4. Both methods, Stan and HDDM, produce similar errors over all conditions and parameters (see dot positions and lengths of error bars). As trial numbers increase, estimates become more accurate and error bars smaller. This is also the case as the number of participants increases (Experiments 2 and 4). It is noticeable that *mean errors* for the relative starting point seem to underestimate the true values in Stan compared to HDDM. This might be attributed to the relative starting point alone or rather to the complex entanglement of all model parameters. With a lower relative starting point the decision process for the correct decision lasts longer. But as the model tries to fit the data and the reaction time is given, the other parameter estimates have to be adjusted. Hence, a lower relative starting point can be compensated with a smaller boundary separation, a higher drift rate or a lower non-decision time. Since results for Experiments 2 and 4 are very similar to results of Experiments 1 and 3, respectively, they are shown in Appendix 5.9.3. In the appendix, *mean error* for the group-level parameters for Experiments 1 and 2 can also be seen. Since we used the same experimental setup for Experiments 1 and 2 as Wiecki et al. (2013) did in their Experiments 1 and 2, and both studies yield similar values for the point estimates, our study replicates Wiecki's findings in terms of the *mean error*.

Detection rates are shown in Figure 5.5. In 19 of the 26 conditions Stan and HDDM detect the same percentage of differences. In a few conditions Stan detects more differences than HDDM (4 out of 26) and vice versa (3 out of 26). Thus, regarding *detection rate*, there is no preference for one method over the other. Furthermore, as for the *mean error*, our results replicate Wiecki's findings in terms of the *detection rate*. In general, *detection rates* are rather small. This may be due to the choice of the distributions for the true parameters and the choice of μ_{v_2}. The true μ_{v_2} is twice as big as true μ_{v_1} and μ_{v_1} stems from a $\mathscr{U}(0.1, 0.5)$ distribution. This means that μ_{v_2} is from a $\mathscr{U}(0.2, 1.0)$ distribution, which overlaps the distribution of μ_{v_1} substantially. Therefore, the differences between the true parameters could often be too small to be detectable for both methods, `wiener()` and HDDM.

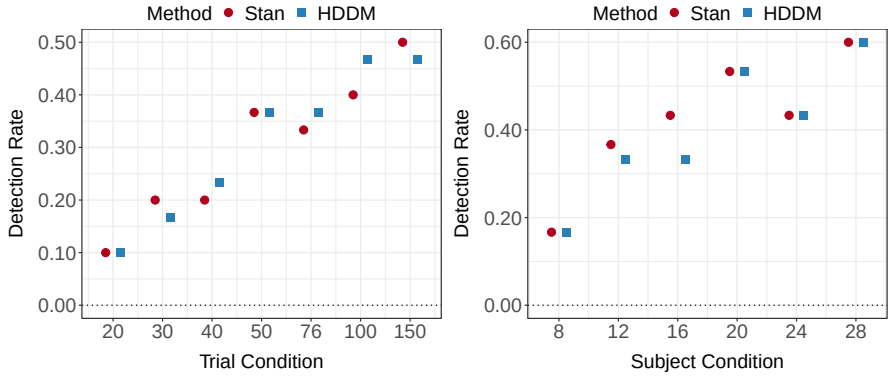
Correlations between estimated and true participant-level parameters are shown in Figure 5.6. As can be seen, both methods produce very similar correlations. In Experiments 1 and 2, values for correlation do not deviate between the methods. In Experiments 3 and 4, there is a slight difference between Stan and HDDM for the parameters v_1, v_2, and w. Results for *correlations* for the group-level parameter for Experiments 1 and 2 are shown in the Ap-



(a) Mean Errors for Experiment 1.

(b) Mean Errors for Experiment 3.

Figure 5.4: Mean error (true-median) for participant-level parameters for Experiments 1 and 3. *Note.* Dots/squares represent the mean of the errors, averaged across participants and groups for each condition. Bars are the 95% highest density intervals for the error.



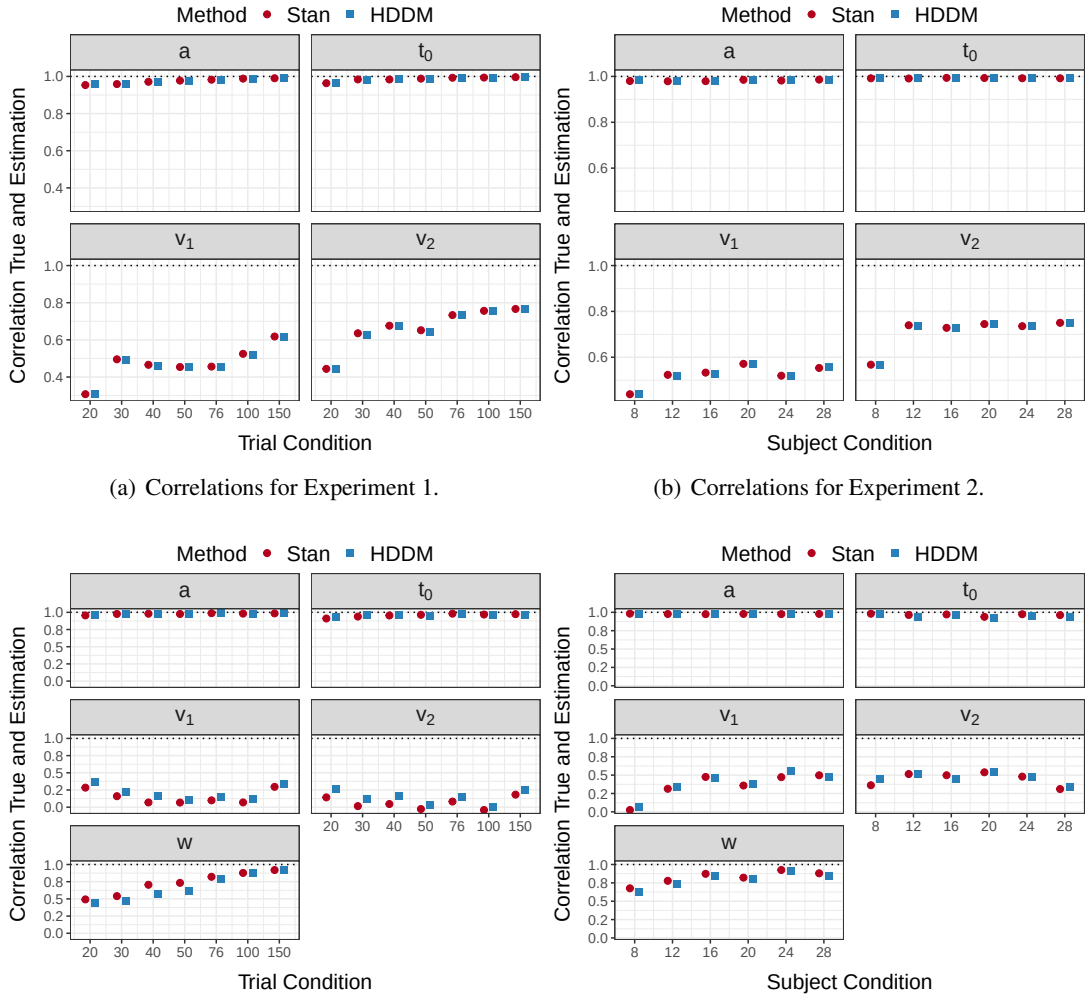
(a) Detection Rates for Experiment 1.

(b) Detection Rates for Experiment 2.

(c) Detection Rates for Experiment 3.

(d) Detection Rates for Experiment 4.

Figure 5.5: Detection rates for all experiments. *Note.* Percentage of times that the difference in both group-level drift rates was detected. In Experiments 1 and 2: 30 observations. In Experiments 3 and 4: 4 observations.



(a) Correlations for Experiment 1.

(b) Correlations for Experiment 2.

(c) Correlations for Experiment 3.

(d) Correlations for Experiment 4.

Figure 5.6: Correlations between true value and posterior median for participant-level parameters for all experiments.

pendix 5.9.4. Group-level results for Experiments 3 and 4 can be found on https://osf.io/vb4ex/ (Retrieved September 23, 2025). In general, *correlations* for v_1 and v_2 are rather small for the full model: The participant-level drift rate parameters are thus not estimated very precisely for the range of trial numbers considered.

As described above, we increased sampling iterations each time an analysis did not meet the two diagnostic criteria. Mean values of *sampling iterations* and *runtimes* needed to meet the diagnostic criteria are displayed in Table 5.3. Results show (a) that Stan needs much fewer iterations than HDDM in all experiments, (b) that Stan is much faster than HDDM for the smaller model (in Experiments 1 and 2), and (c) that Stan is much slower than HDDM for the full model (in Experiments 3 and 4). These patterns likely reflect the different samplers implemented in Stan and HDDM and the different numerical integration routines needed for cases with non-zero inter-trial variabilities in relative starting point, s_w, and non-decision time,

Experiment	Mean Sampling Iterations		Mean Runtime (in minutes)	
	Stan	HDDM	Stan	HDDM
1	1920	12989	11	60
2	1489	14792	18	124
3	2679	16983	1376	98
4	2396	12563	1635	140

Table 5.3: Mean Number of Sampling Iterations and Mean Runtime in Minutes Needed to Meet the MCMC Criteria for Stan and HDDM.

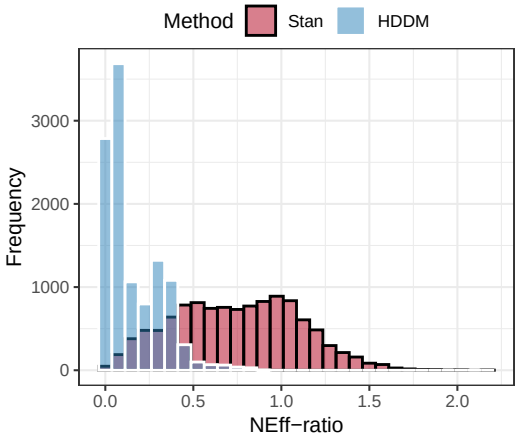
s_{t_0}. In Stan, the No-U-Turn Sampler (NUTS, Hoffman & Gelman, 2014) is implemented and in HDDM a slice sampling algorithm is implemented. For `wiener()`, a multidimensional integration rule is implemented and in HDDM, unidimensional integration rules are nested if integration across more than one dimension is required. Note that in Stan, unlike in HDDM, it is possible to parallelize each chain over several cores, which speeds up hierarchical analyses considerably. Thus, if there is much computational power available, the user can easily implement the parallelization via the `reduce_sum` routine in Stan and use as many cores as are available. For an example how to use `reduce_sum`, see Appendix 5.9.4.

To get a better grasp of the runtime and sampling iteration results, we report a measure of efficiency, the *relative effective sample size*. Results are displayed in Figure 5.7. The effective sample size is the number of independent samples that contain the same information as the total number of autocorrelated samples (Vehtari et al., 2021). Since both methods differ in their total numbers of sampling iterations, we compute the ratio of effective sample size to total number of samples, N_{eff}-ratio. Results show that Stan samples more effectively than HDDM in all four experiments. Note that the No-U-turn sampler used in Stan can produce negative autocorrelation within the samples to reduce variance. Then, the effective sample size may become greater than the total number of samples, $N_{\text{eff}} > N$. This is often the case for parameters which have close to Gaussian posterior distributions and little dependency on other parameters (Stan Development Team, 2023). This can be seen in Figure 5.7, where N_{eff}-ratio > 1.[4]

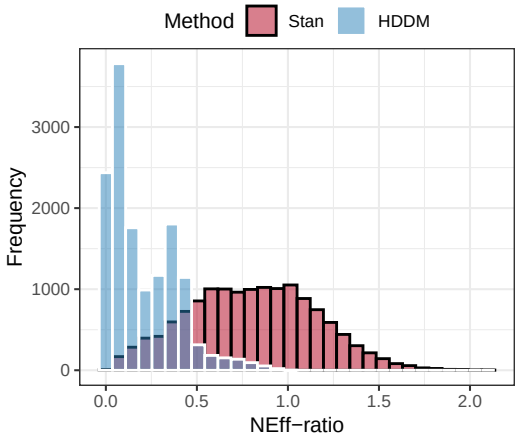
5.7 Simulation-based Calibration Study for HDDM

Given the small but possibly systematic deviations in correlations, we ran a simulation-based calibration study (SBC, Talts et al., 2018) for HDDM that was closely fashioned after the SBC

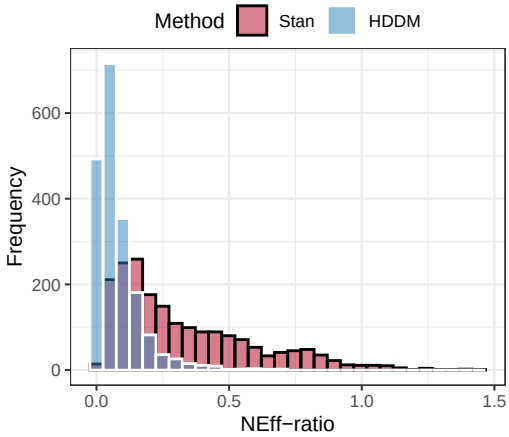
[4] Note that effective sample size in dockerHDDM and in Stan is estimated based on the same formula (see ArviZ Development Team, 2023).



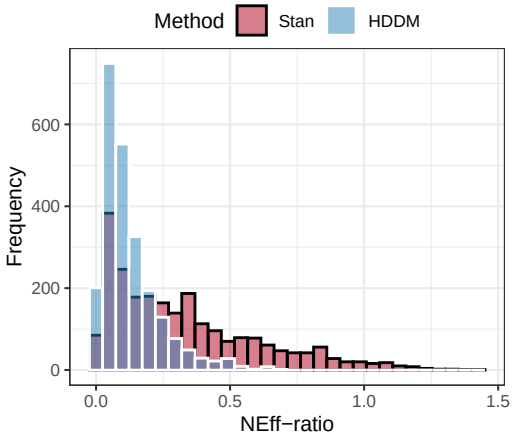
(a) Relative effective sample sizes for Experiment 1.



(b) Relative effective sample sizes for Experiment 2.



(c) Relative effective sample sizes for Experiment 3.



(d) Relative effective sample sizes for Experiment 4.

Figure 5.7: Relative effective sample sizes for all experiments.

study for our Stan module `wiener()` reported in Henrich et al. (2023).[5]

A simulation-based calibration study (SBC, Talts et al., 2018) is used to test whether the underlying algorithm of a Bayesian implementation is implemented correctly. For an SBC, a simulation study is performed for which the true parameters stem from the distributions used as priors in the model. A valid Bayesian algorithm satisfies the *self consistency condition*:

$$\pi(\theta) = \int\int \pi(\theta \mid \tilde{y})\pi(\tilde{y} \mid \tilde{\theta})\pi(\tilde{\theta})\, d\tilde{y}\, d\tilde{\theta}, \tag{5.2}$$

[5] We also tested some datasets on a random basis with the Kolmogorov-Smirnov test, using the R-function `ks.test()`, to check whether the posterior samples from Stan and HDDM stem from the same distribution. In four datasets, we tested 144 posterior distributions. When data were not thinned, only 11 of 144 tests yielded a non-significant p-value. When data were thinned to 399 high quality samples 34 of 144 tests yielded a non-significant p-value, meaning that for most of the tested parameters the posterior distributions obtained in Stan and in HDDM differ significantly. We chose to thin to 399 samples since the effective sample sizes of all parameters are at least 400.

where $\tilde{\theta} \sim \pi(\theta)$ are the true parameters - denoted as the *ground truth* - sampled from the prior distribution, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$ are the data generated from the model using the ground truth, and $\theta \sim \pi(\theta \mid \tilde{y})$ the posterior samples. From this condition, it follows that the prior sample $\tilde{\theta}$ and the posterior sample θ follow the same distribution. Consequently, the *rank statistic* r of the prior sample relative to the posterior sample of size L, defined for any one-dimensional random variable, $f : \Theta \to \mathbb{R}$,

$$r\big(\{f(\theta_1), \ldots, f(\theta_L)\}, f(\tilde{\theta})\big) := \sum_{l=1}^{L} \mathbb{I}\big[f(\theta_l) < f(\tilde{\theta})\big] \in [0, L] \tag{5.3}$$

should be uniformly distributed over the natural numbers in $[0, L]$, where L is the number of samples of the posterior distribution, and $\mathbb{I}$ is the indicator function taking the value 1 if the condition in the parentheses holds and the value 0 otherwise. Uniformity is evaluated with histograms of the *rank statistics*. Systematic deviations from the uniformity allow one to diagnose specific problems of the algorithm (Talts et al., 2018).

5.7.1 Priors and Datasets

Using R, we sampled 2000 ground truths from the informative priors that are implemented in HDDM (see Appendix 5.13). From these ground truths, we simulated two times 2000 datasets to perform two SBC studies - one for 100, and one for 500 trials per dataset using the R function `sampWiener()` from the WienR R-package (Hartmann & Klauer, 2021; R Core Team, 2021) with precision set to 1e-12. In HDDM, we fitted the full diffusion model to these data, non-hierarchically, with one condition, and no outliers (see Appendix 5.14).

For the analyses, we used the same convergence criteria as in the simulation study above, such that for each analysis all effective sample sizes are at least 400 and all $\hat{R}$-values are less than 1.01. As the MCMC-samples in HDDM are autocorrelated, we used a subset of the samples to compute the *rank statistics* for each model parameter and thinned the posterior samples according to Algorithm 2 in Talts et al. (2018) to $L = 399$ high-quality samples. We set the number of bins in the histogram to 100, such that there are 20 observations expected per bin, across the 2000 simulated datasets.

We also calculated the χ^2-statistic for the differences between observed and expected frequencies of observations per bin for each parameter with expected frequencies given by the expected uniform distribution (i.e., 20 per bin). For each parameter, the observed χ^2 value is compared to the critical χ^2 value of 123.23, for $\alpha = .05$ with $df = 99$ (number of bins minus 1).

5.7.2 Results and Discussion

The results for the SBC for 100 trials are shown in Figure 5.8 and for 500 trials in Figure 5.9. Visual inspection of the histograms for the 100 trials SBC indicates no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar). Also, none of the χ^2 statistics is significant, except the one for the boundary separation parameter, a. Since the corresponding histogram does not show systematic deviation from a uniform distribution, this may just be due to chance.

Visual inspection of the histograms for the 500 trials SBC shows that most histograms slightly deviate from the uniform distribution. They have a peak at one or both sides. Also six out of seven χ^2 statistics indicate significant deviations from uniformity. Following Talts et al. (2018, e.g., Figure 12), a peak on the left (right) hand side of the histogram indicates that the algorithm overestimates (underestimates) the true parameter and peaks on both sides of the histogram can indicate that there might be autocorrelation in the samples (although this seems an unlikely explanation giving the use of thinning as per Algorithm 2 in Talts et al., 2018).

These deviances can stem from issues in different sources: a) the MCMC-sampler, b) the priors, or c) the likelihood and different precisions. In the following, we discuss these three possibilities.

MCMC-sampler

If there were issues in the MCMC-sampler, we would expect the histograms of *both* SBC studies, for 100 and for 500 trials, to show systematic deviances from uniformity. But since the histograms from the 100 trials SBC are inconspicuous, it seems unlikely that the MCMC-sampler itself is implemented incorrectly.

Prior

It may be the case that the distributions we used to sample the ground truths differ from the priors defined in HDDM. Indeed, initial troubles to obtain uniform histograms prompted us to take a closer look beyond the documentation (Wiecki et al., 2022) into the code on Github (HDDM Development Team, 2023a, 2023b). It turned out that three priors are differently implemented than documented: the priors for the relative starting point, the group-level mean of the relative starting point, and the group-level standard deviation of the boundary separation (for a detailed explanation on the differences, see Appendix 5.13). All three priors are implemented more informatively than documented.

As an aside note, this means that in the comparison study above, the priors we defined in the Stan model are based on the *documented* priors and, hence, are less informative than the ones used in HDDM. This is not in line with our initial intention to rebuild the HDDM-priors

in our Stan-model *as closely as possible*. But this is not a problem as results in the study above show that Stan already meets the benchmark results (which was the aim of the study). Taken together, our results (a) show that Bayesian results are quite stable against minor differences in the priors, and (b) at the same time illustrate one major advantage of Stan over HDDM, namely the capability to freely vary the choice of the priors. Moreover, we saw major differences in the runtimes and effectivity of the sampling processes. These will probably not change much for slightly different priors, since they are mainly due to the differing MCMC-samplers and integration-routines. Nevertheless, we reran the Stan analyses in the above comparison study with corrected priors, to see how the comparison criteria (*mean errors* and *correlations*) change. Results are shown in Appendix 5.9.6. For Experiments 1 and 2, nearly all minor differences we saw between HDDM and Stan have been levelled in this setup. For Experiments 3 and 4 we also observe that results align with each other to a slightly higher degree than before. This means that Stan is indeed able to obtain the benchmark results in terms of recovery when the same priors are implemented. Thus, the above conclusions are underlined by this corrected analysis.

Coming back to the SBC, as an SBC *needs* to have exactly the same priors in the simulation process as in the fitting process, we sampled the ground truths again, this time from the distributions we found on Github (HDDM Development Team, 2023a, 2023b) and ran the analyses again (see Figure 5.13 for the priors we used to draw the ground truths from). The results from the SBC reported above stem from these analyses. As can be seen, this did not eliminate all problems. However, it is still possible that we misspecified the priors given the difficulty of discerning the priors from the complex source code on Github and given the fact that we failed to get confirmation as to their correctness from the HDDM forum (see Google Groups HDDM Users, 2023) as of the time of writing this paper. We set the priors to the best of our knowledge and belief.

Likelihood and Precision

It may also be the case that the likelihood of the seven-parameter diffusion model is not computed correctly or at least with a lower precision than the simulated data. To check this, we compared the values of the log-likelihood computed in HDDM with the ones computed with `WienerPDF()` in R (Hartmann & Klauer, 2021) for each of the datasets generated for the SBC study. Most of the likelihood values were closely comparable, but there were also noticeable discrepancies that were somewhat more pronounced for the larger datasets: Table 5.4 shows the first five log-likelihood values computed in HDDM and `WienerPDF` with the precision set to 1e-12, split into the values for 100 trials and 500 trials, and sorted by the difference between the values in HDDM and `WienerPDF`, largest differences at the top. After checking these datasets in more detail, we found substantially more extreme values of the rank statistics in the parameters than for a random dataset with lower differences in the likelihoods. As of now, these

Dataset	HDDM	WienerPDF
	— 100 Trials —	
0556	-4556.96287	24.84079
1111	51.00406	51.34622
1927	-79.06271	-78.97416
1839	3.05994	3.08993
0670	-27.61269	-27.58889
	— 500 Trials —	
1696	-Inf	811.73597
0556	-23692.12590	128.64674
0302	52.79599	53.25085
0885	-216.07525	-215.76768
1111	156.63400	156.86148

Table 5.4: Log-likelihoods for the 100 and 500 trials SBC datasets. *Note.* Values sorted by the differences between HDDM and WienerPDF, largest differences on the top. Precision for WienerPDF = 1e-12.

differences are our best bet as to the reason for the lack of calibration in our SBC study.[6]

5.8 General Discussion

The aim of this paper was to further evaluate the new implementation of the seven-parameter diffusion model, `wiener()`, in the probabilistic programming language Stan and to place it in the context of existing methods. For a non-hierarchical setup, we already showed that `wiener()` achieves good recovery and that the algorithm as implemented passes a validity check in terms of a simulation-based calibration study (Henrich et al., 2023). Here, we focused on hierarchical models and compared results with another method that also implements the seven-parameter diffusion model in a Bayesian, hierarchical framework - the stand-alone python-based toolbox HDDM. Unlike previous implementations of the diffusion model, the implementation in Stan provides the user with huge flexibility in defining priors.

In the present paper, we reported results of a simulation study replicating and extending a

[6] We also reran the Kolmogorov-Smirnov test from Footnote 5 on the same datasets. Indeed, with the corrected priors, the number of significant p-values decreased and more parameters are attested to stem from the same distribution than before (52 of 144 p-values are non-significant for the non-thinned data and 86 of 144 for the thinned data). Nevertheless, due to the differences in the precision of the likelihoods we cannot expect all discrepancies to be eliminated.

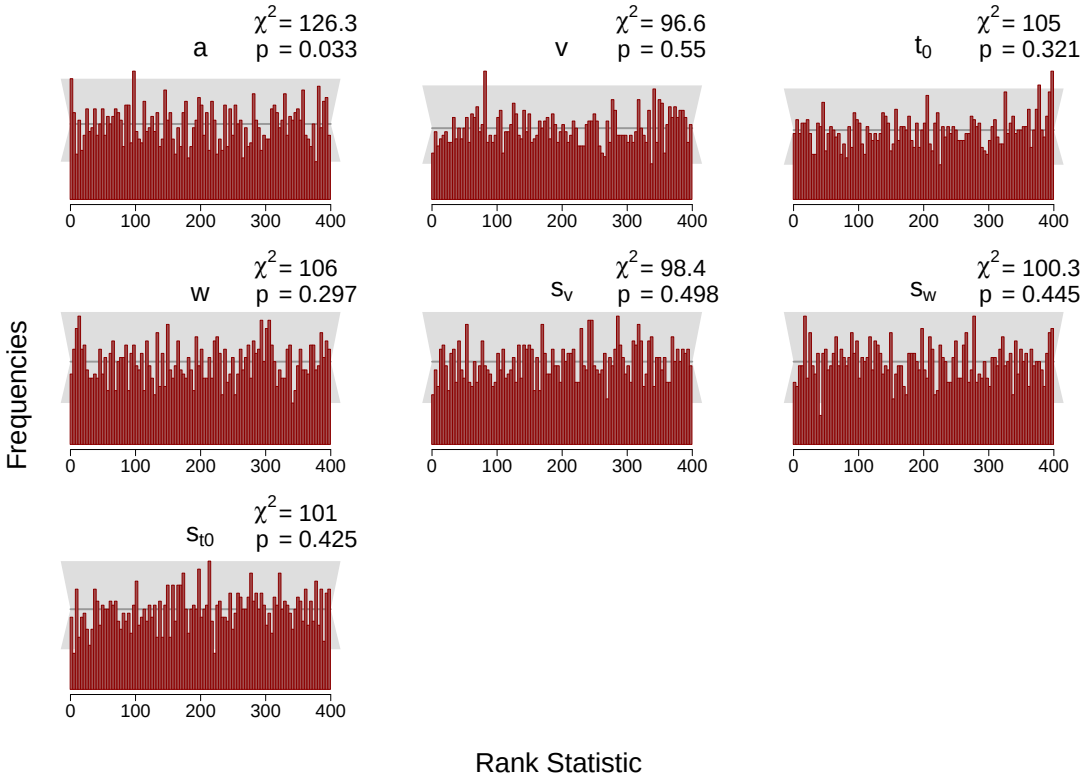


Figure 5.8: Histograms of the rank statistic for 100 trials. *Note.* The histograms indicate no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar).

study by Wiecki et al. (2013) comparing the `wiener()`-implementation in Stan to the already established and well-tested HDDM toolbox. We wanted to see whether the Stan implementation is up to the implementation in HDDM and could serve as an alternative to HDDM. For this purpose, we conducted four experiments - two for a smaller diffusion model and two for the full diffusion model. Furthermore, to elucidate minor discrepancies between the two methods, we performed a simulation-based calibration study on HDDM fashioned after the one we reported for `wiener()` in Henrich et al. (2023).

In summary, results of the comparison study show that `wiener()` is on a par with HDDM. Errors, detection rate, and correlations do not differ substantially between both methods and none of the methods outperforms the other. A runtime analysis shows that Stan is much faster than HDDM for the smaller model and much slower than HDDM for the full model. This likely reflects the different samplers and integration routines implemented. Given the possibility to parallelize a single chain over many cores in Stan, the runtime disadavantage seen for the full model in Stan can be compensated for. The efficiency analysis reveals that Stan samples more effectively than HDDM. Results for the full model also show that correlations with true parameters are very small for the drift-rate parameters under both methods. This could indicate that trial numbers were too small for the full model analyses.

From these findings, the question arises whether a fully hierarchically defined diffusion

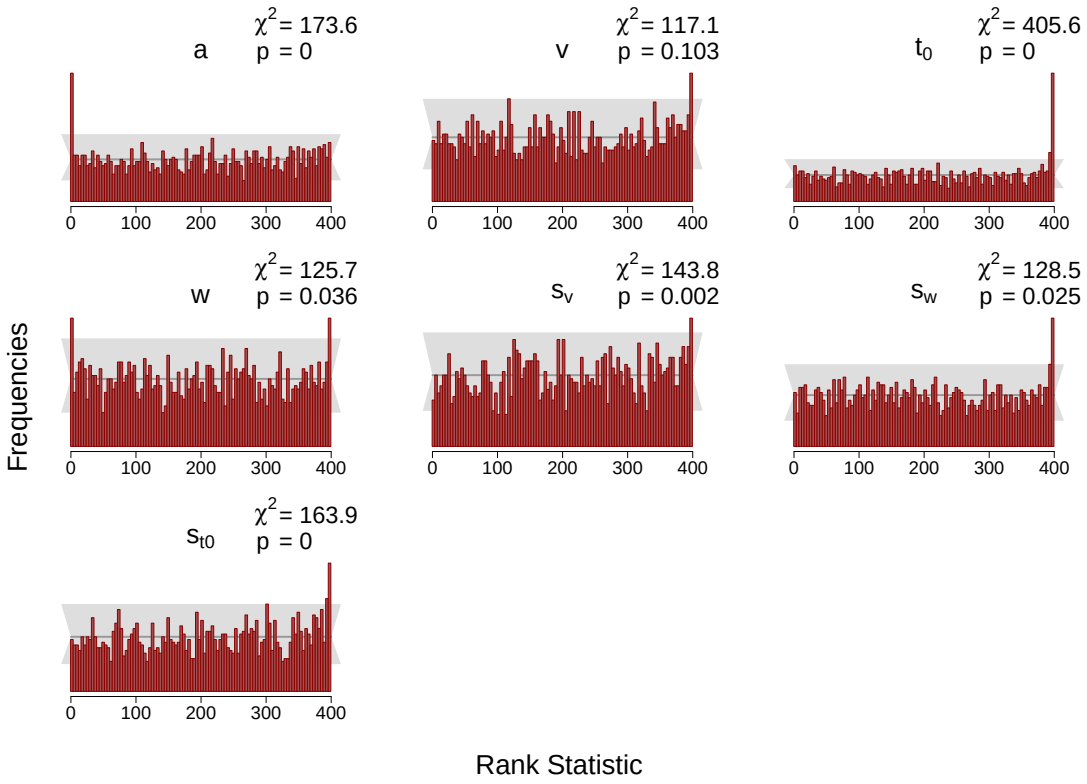


Figure 5.9: Histograms of the rank statistic for 500 trials. *Note.* Some histograms indicate issues as the empirical rank statistics (red vertical bars) deviate from the variation expected of a uniform histogram (gray horizontal bar).

model, with all three inter-trial variabilities defined per person (instead of as group parameters) would produce better results in terms of the given criteria. In order to test this, we also tried to fit data with a fully hierarchical diffusion model. Unfortunately, we faced two issues that did not allow us to successfully run that model: First, for several datasets the program was not able to find suitable starting values, such that the analysis did not even start. Second, when the analysis started, it was so painfully slow that it would not end in reasonable time.

One possibility to work around this problem is to use a smaller model. When running different model specifications, we found out that the inter-trial variability in the relative starting point is the most troublesome parameter. A model with inter-trial variabilities in drift rate and non-decision time defined hierarchically and inter-trial variability in relative starting point defined non-hierarchically is able to find starting values and finish the sampling process in acceptable time. Though, in this special use case, the convergence criteria and trace plots were not satisfying. Another option would be to leave the inter-trial variability in relative starting point out of the model and fit a six-parameter model to the data. This also drastically accelerates the sampling process.

Results of the simulation-based calibration study (a) revealed that three of the informative priors in HDDM are implemented differently than documented and (b) revealed issues with the HDDM algorithm that may reflect issues with the likelihood implemented in HDDM. As al-

ready mentioned, we got no feedback from the HDDM-community, we can only guess that our findings are due to a trade-off between precision of the calculations and runtime. This is an issue that needs to be addressed by the HDDM community. But as we only see minor differences between the methods in our comparison study (especially after reanalysing all datasets with the corrected priors in Stan), those deviations may not be relevant in practical applications, where other sources of uncertainty like model misspecification or measurement error are in place as well. This means, despite the lack of transparency in the priors of HDDM, our findings suggest that HDDM is a proper benchmark method, particularly for estimating the full diffusion model for which it is the only other Bayesian method available. As HDDM is already 10 years old, a successor package, HSSM (Fengler et al., 2023), was recently released. It comprises many of the functions that are implemented in HDDM. But, up to now, the full diffusion model in a hierarchical form is not yet available in this package, although it is planned to be implemented later.

The present study has a number of limitations. First, like all simulation studies, our data-generating process covers only a subset of potential parameter settings and model specifications. Second, the design of our comparison study was constrained by the goal to stick closely to Experiments 1 and 2 by Wiecki et al. (2013). Therefore, we started with two experiments for a smaller diffusion model before we went on to the full diffusion model. Moreover, with the simulation settings chosen by Wiecki et al. (2013), both HDDM and Stan showed at best mediocre performance in some aspects such as detection rate and correlations. This should not shadow the actual outcome of interest, namely that `wiener()` performed quite similarly to HDDM across the board. It is important to acknowledge, nevertheless, that both methods had problems in estimating some of the model parameters precisely. Furthermore, we simulated data without outliers and defined the models in Stan and HDDM without modeling outliers. Note that it is possible to model outliers. In HDDM, a mixture model is implemented for this purpose, and in Stan the same method among others can be implemented by the user.

It should be mentioned that the full diffusion model has to be used carefully in a hierarchical way as it needs much information to fit the data and missing information lead to poor convergence and model fit. In most cases it is not senseful to also model the inter-trial variabilites hierarchically as information on that is mostly scarce. This is also a limitation in our analyses as in our data-generating process the inter-trial variabilities were defined per person and the model was declared with the inter-trial variabilites on group-level. This is probably the main reason for some problems in model fit we discussed above. Therefore, when defining a model, one should try to approach the data-generating process and fit models with various complexities to get a feeling on which information are contained in the data and which not.

In conclusion, the new implementation, `wiener()` in Stan, produces competitive results to HDDM and adds flexibility to Bayesian estimation of the diffusion model. It thereby enriches the landscape of diffusion modeling approaches. We hope that it serves users as a valuable tool for their data analyses.

5.9 Appendix

5.9.1 Stan Model

This is the Stan model definition for the full model that is shown in the graphical model representation in Figure 5.3. For the smaller model it is basically the same setup: you hand over the values for the three fixed parameters in the data block ($s_{t_0} = 0$, $s_w = 0$, $w = 0.5$) and leave out the corresponding lines in the code.

```
// Stan Model for the comparison study
functions { // function to parallelize each chain
 real partial_sum_fullddm(array[] real rt_slice, int start,
 int end, array[] real a, vector t0_m, real t0_s, vector w_m,
 real w_s, matrix v_m, real v_s, array[] int resp,
 array[] int cnd, array[] int subj) {
  real ans = 0;
  for (i in start:end) {
   if (resp[i] == 1) { // upper threshold
    ans += wiener_full_lpdf(rt_slice[i+1-start] | a[subj[i]],
     v_m[subj[i],cnd[i]], w_m[subj[i]], t0_m[subj[i]],
     v_s,  w_s, t0_s);
   } else { // lower threshold (mirror drift and
            //starting point!)
    ans += wiener_full_lpdf(rt_slice[i+1-start] | a[subj[i]],
     -v_m[subj[i],cnd[i]], 1 - w_m[subj[i]], t0_m[subj[i]],
     v_s,  w_s, t0_s);
   }
  }
  return ans;
 } } // end partial_sum_fullddm, end functions

data {
 int <lower=1> Nsubj;  // No participants
 int<lower=1> Ncnds;   // No conditions
 int<lower=0> N;       // Length of data
 array[N] real<lower=0> rt;  // response times (seconds)
 array[N] int<lower=1, upper=Ncnds> cnd;
  // stimulus type/condition
 array[N] int<lower=0, upper=1> resp;     // responses (0,1)
 array[N] int<lower=0, upper=Nsubj> subj;
  // participants (1, ..., Nsubj)
 vector<lower=0>[Nsubj] min_rt_subj;
  // minimal response time for each participant
}
```

```
36
37 parameters {
38  // group parameters
39  real<lower=0> mu_a;
40  real<lower=0> sigma_a;
41  real<lower=0, upper=1> mu_w;
42  real<lower=0> sigma_w;
43  real mu_v_1;
44  real mu_v_2;
45  real<lower=0> sigma_v;
46  real<lower=0> mu_t0;
47  real<lower=0> sigma_t0;
48
49  // individual parameters
50  real<lower=0> v_s;    // sd of drift
51  real<lower=0> t0_s;   // sd of non-decision-time
52  real<lower=0> w_s;   // sd of starting point
53  array[Nsubj] real<lower=0> a;  // threshold separation
54  // helper for starting point: non-centered parameterization
55  // w_helper = w_helper_raw * sigma_w + mu_w
56  vector[Nsubj] w_helper_raw;
57  // mean drift for 2 stimulus types:
58  // non-centered parameterization:
59  // v_m = v_m_raw * sigma_v + mu_v (see generated quantities)
60  matrix[Nsubj, 2] v_m_raw;
61  // non-decision time as fraction of minimal response time
62  // of each participant, ensures t_0 < min(rt),
63  // as the likelihood is 0 otherwise
64  vector<lower=0, upper=1>[Nsubj] t0_m_rel;
65 }
66
67 transformed parameters {
68  vector<lower=0, upper=1>[Nsubj] w_m;   // rel. starting point
69  real<lower=0> sigma_square_a;
70  vector[Nsubj] w_helper;
71  w_helper = w_helper_raw * sigma_w + mu_w;
72  w_m = inv_logit(w_helper);
73  sigma_square_a = sigma_a * sigma_a;
74  matrix[Nsubj, 2] v_m;
75  v_m[,1] = v_m_raw[,1] * sigma_v + mu_v_1;
76  v_m[,2] = v_m_raw[,2] * sigma_v + mu_v_2;
77  vector<lower=0>[Nsubj] t0_m;
78  t0_m = t0_m_rel .* min_rt_subj;
79 }
80
```

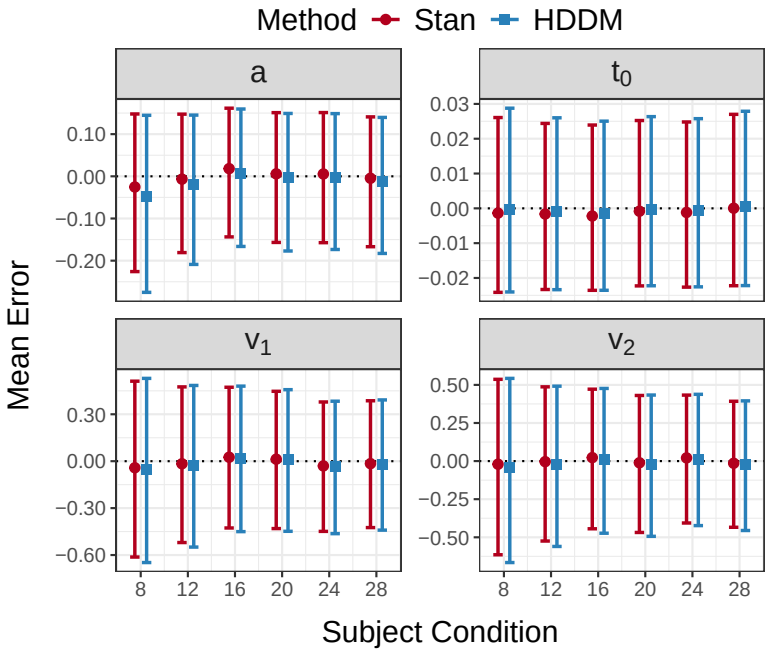
```
model {
//Note: HDDM uses Gamma-distribution with mean mu, rate sigma
// Stan with scale alpha and rate beta
//calculate alpha=(mu*mu)/(sigma*sigma), beta=mu/(sigma*sigma)
 mu_a ~ gamma(4, 4/1.5); //HDDM: gamma(1.5, 0.75)
 mu_v_1 ~ normal (2, 3);
 mu_v_2 ~ normal (2, 3);
 mu_w ~ normal(0.5, 0.5);
 mu_t0 ~ gamma(4, 10); //HDDM: gamma(0.4, 0.2)
 sigma_a ~ normal(0, 0.1) T[0,];
 sigma_v ~ normal(0, 2) T[0,];
 sigma_w ~ normal(0, 0.05) T[0,];
 sigma_t0 ~ normal(0, 1) T[0,];
 a ~ gamma(mu_a*mu_a/sigma_square_a, mu_a/sigma_square_a);
  //HDDM: gamma(mu_a, sigma_square_a)
 w_helper_raw ~ std_normal();
  //implies: w_helper ~ normal(mu_w, sigma_w);
 v_m_raw[,1] ~ std_normal();
  // implies: v_m[,1] ~ normal(mu_v_1, sigma_v);
 v_m_raw[,2] ~ std_normal();
  // implies: v_m[,2] ~ normal(mu_v_2, sigma_v);
 t0_m ~ normal(mu_t0, sigma_t0);
 t0_s ~ normal(0, .3) T[0,];
 w_s ~ beta(1, 3);
 v_s ~ normal(0, 2) T[0,];

 // using reduce_sum to allow for parallel processing
 target += reduce_sum(partial_sum_fullddm, rt, 1,
   a, t0_m, t0_s, w_m, w_s, v_m, v_s, resp, cnd, subj);
}

generated quantities {
// for the detection of the difference in the drift rates:
 real diff_drift;
 diff_drift = mu_v_2 - mu_v_1;
}
```

5.9.2 HDDM Model

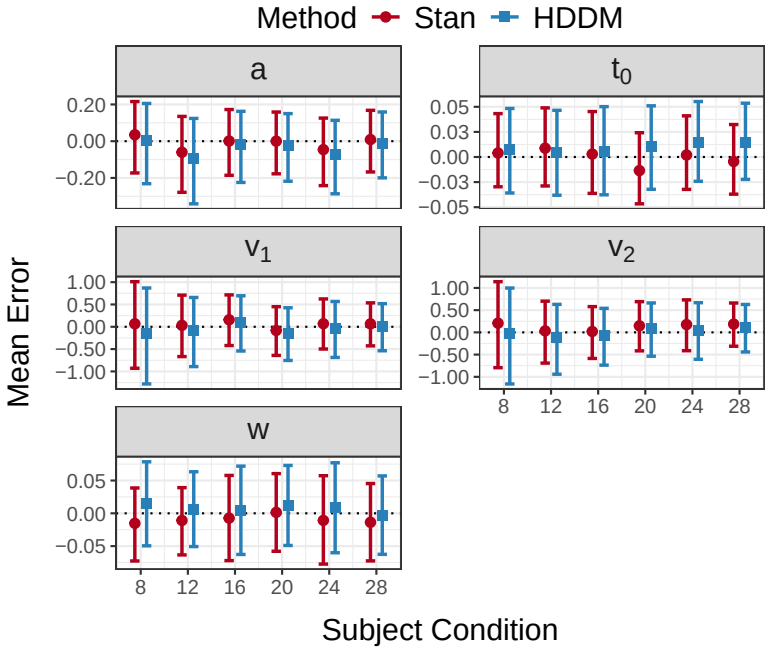
These are the HDDM model definitions for the small model that is shown in Figure 5.2 and the full model that is shown in Figure 5.3.

```
small_model = hddm.HDDM(df,
                        p_outlier=0.0,
                        include=['sv'],
                        group_only_nodes=['sv'],
                        depends_on={'v': 'cnd'})



full_model = hddm.HDDM(df,
                       p_outlier=0.0,
                       include=['z', 'sv', 'sz', 'st'],
                       group_only_nodes=['sv', 'sz', 'st'],
                       depends_on={'v': 'cnd'})
```

5.9.3 Results Mean Error

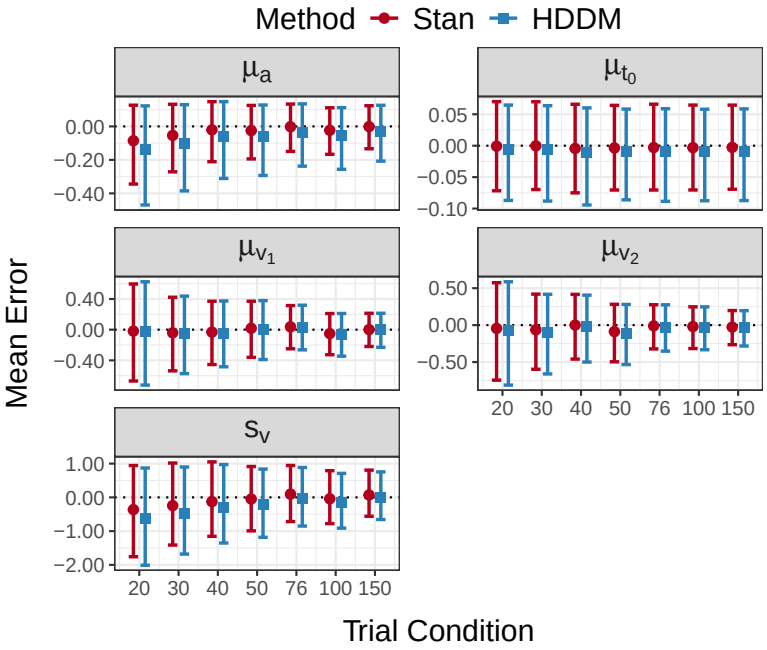


(a) Mean Errors for Experiment 2.

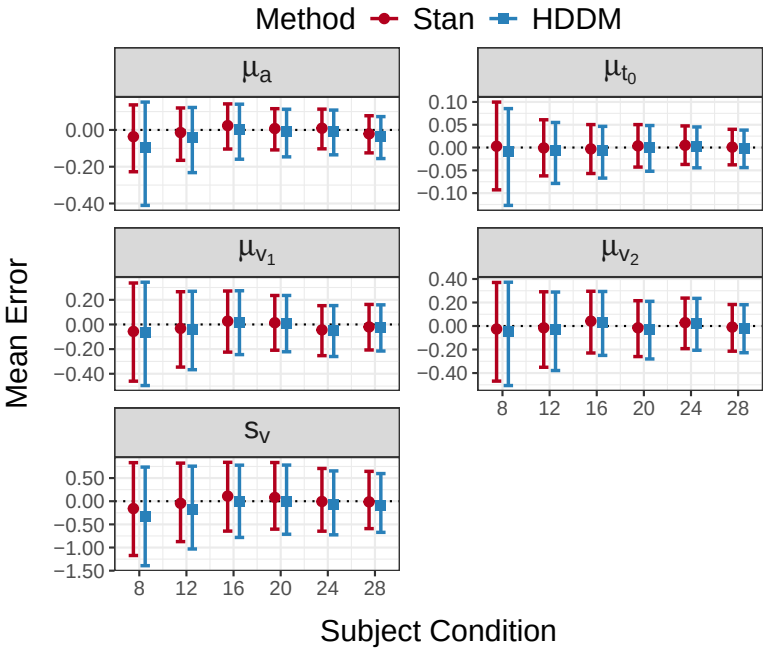


(b) Mean Errors for Experiment 4.

Figure 5.10: Mean Error (true-median) for participant-level parameters for Experiments 2 and 4. *Note.* Dots/squares represent the mean of the errors, averaged across datasets and participants within datasets for each trial-number condition. Bars are the 95% highest density intervals for the error.



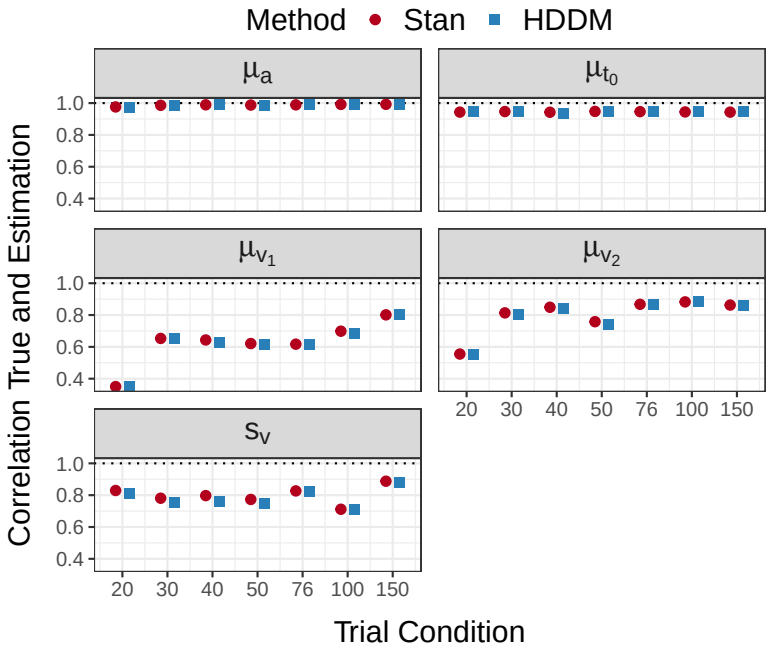
(a) Mean Errors for Experiment 1.



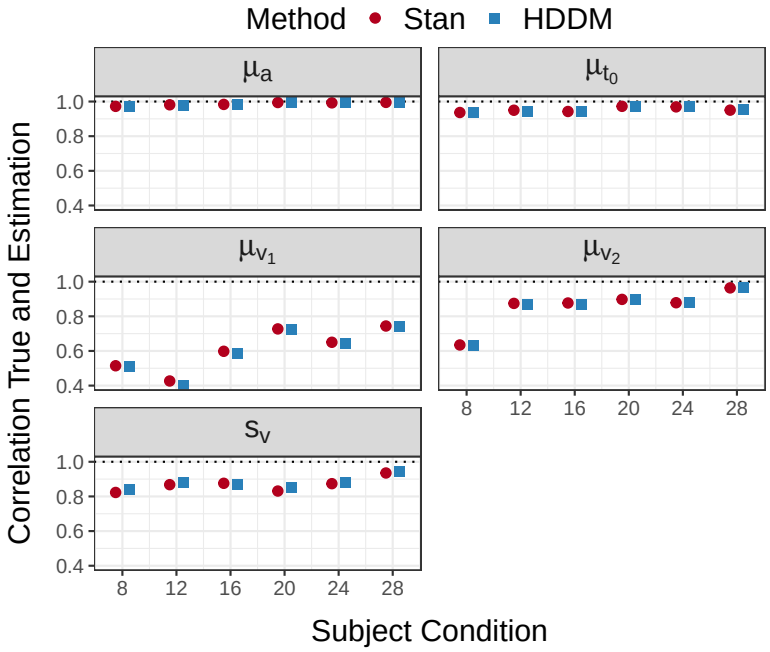
(b) Mean Errors for Experiment 2.

Figure 5.11: Mean Error (true-median) for group-level parameters for Experiment 1 and 2. *Note.* Dots/squares represent the mean of the errors, averaged across participants and groups for each condition. Bars are the 95% highest density intervals for the error.

5.9.4 Results Correlations



(a) Correlations for Experiment 1.



(b) Correlations for Experiment 2.

Figure 5.12: Correlations between true value and posterior median for group-level parameters for Experiments 1 and 2.

5.9.5 SBC HDDM

We found three deviations in the code on Github from the HDDM-documentation Wiecki et al. (2022): (a) The relative starting point w is implemented as $w \sim \textbf{invlogit}(\mathcal{N}(\textbf{logit}(\mu_w), \sigma_w^2))$ instead of $w \sim \textbf{invlogit}(\mathcal{N}(\mu_w, \sigma_w^2))$, (b) the group-level mean of w is implemented as $\mu_w \sim \textbf{invlogit}(\mathcal{N}(\textbf{logit}(0.5), 0.5))$ instead of $\mu_w \sim \mathcal{N}(0.5, 0.5)$, and (c) the group-level variance of a is implemented as $\sigma_a \sim \mathcal{HN}(2)$ instead of $\sigma_a \sim \mathcal{HN}(0.1)$.

This means the implemented priors are more informative than the documented priors.

```
a   = rgamma(1, shape=4, rate=4/1.5),  #HDDM: gamma(1.5, 0.75)
v   = rnorm(1, mean=2, sd=3),
sv  = rtruncnorm(1, a=0, mean=0, sd=2)
w   = inv.logit(rnorm(1, logit(0.5), 0.5))
   //(w as implemented in the code on Github)
sw  = rbeta(1, 1, 3)
while (w-.5*sw<0 || w+.5*sw>1) {
    sw  = rbeta(1, 1, 3)
}
t0  = rgamma(1, shape=4, rate=10)  #HDDM: gamma(0.4, 0.2)
st0 = rtruncnorm(1, a=0, mean=0, sd=0.3)
while (t0-.5*st0<0) {
    st0 = rtruncnorm(1, a=0, mean=0, sd=0.3)
}
```

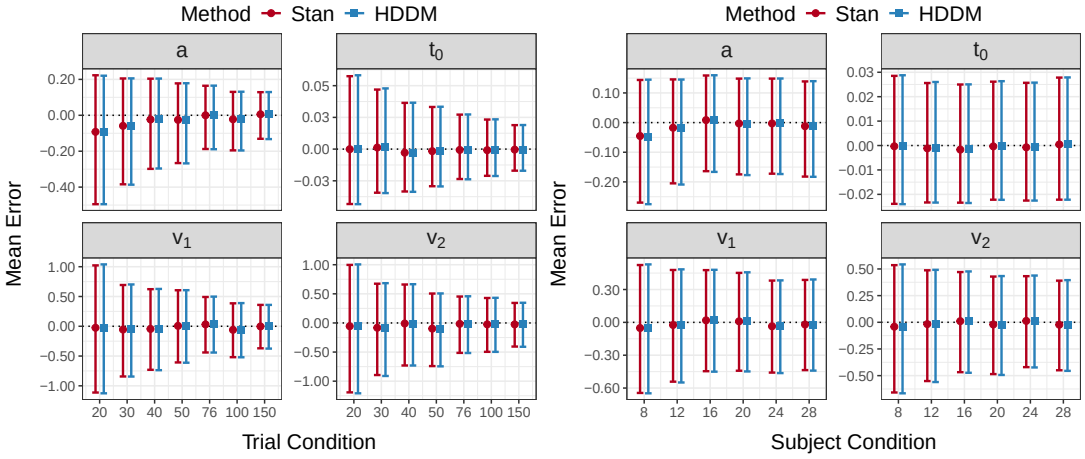
Figure 5.13: R code to draw the ground truths from the priors as defined in the HDDM code. *Note.* The HDDM-code can be found on the Github page of HDDM (HDDM Development Team, 2023a, 2023b).

```
SBC_model = hddm.HDDM(df,
                p_outlier=0.0,
                include=['z', 'sv', 'sz', 'st'])
```

Figure 5.14: HDDM-model for the SBC. Non-hierarchical, with all 7 parameters, no outliers.

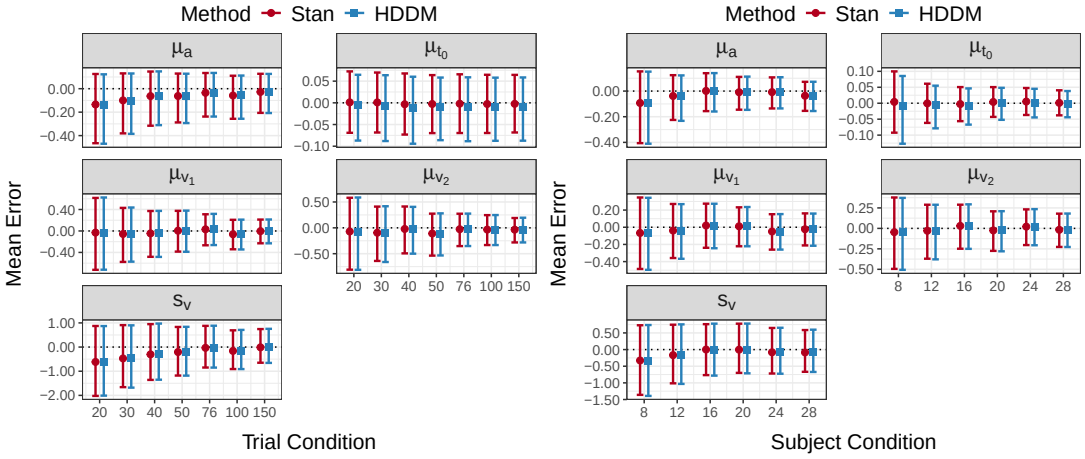
5.9.6 Results for the comparison study with corrected priors

As elaborated in Appendix 5.9.5, there are three priors that are differently implemented than documented in HDDM. Therefore, we reran the Stan-analyses in the comparison study and investigated the comparison criteria *mean errors* and *correlations*. All results align.



(a) Experiment 1, participant-level parameters.

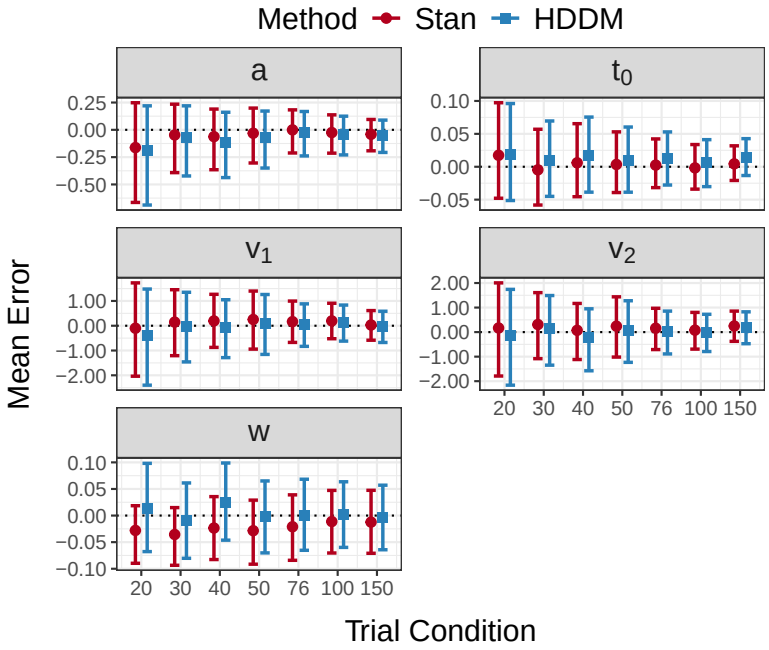
(b) Experiment 2, participant-level parameters.



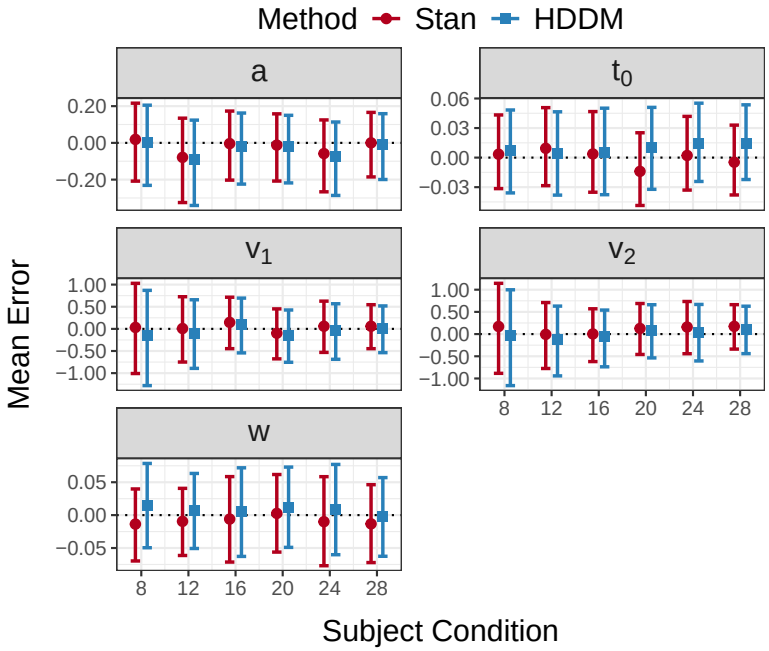
(c) Experiment 1, group-level parameters.

(d) Experiment 2, group-level parameters.

Figure 5.15: Mean errors for results with corrected priors in Stan for Experiments 1 and 2.

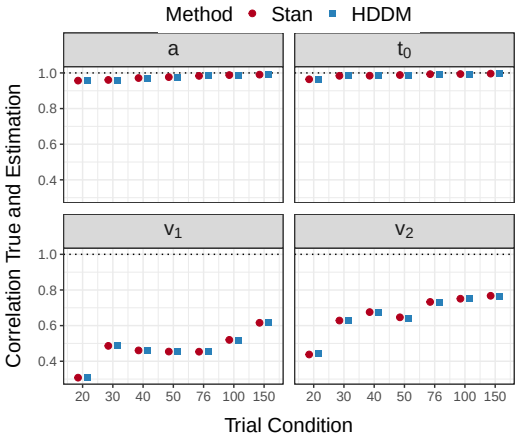


(a) Experiment 3, participant-level parameters.

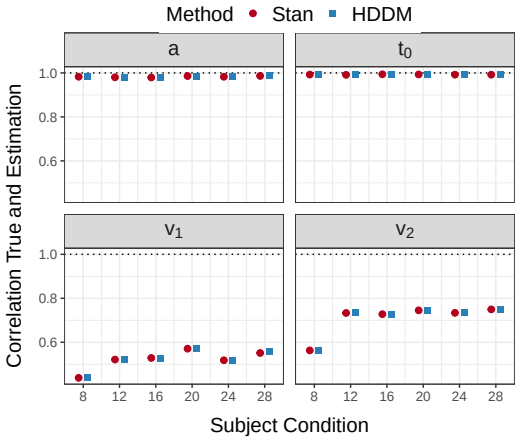


(b) Experiment 4, participant-level parameters.

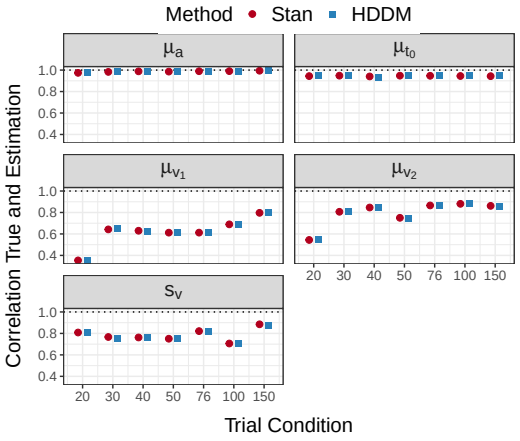
Figure 5.16: Mean errors for results with corrected priors in Stan for Experiments 3 and 4.



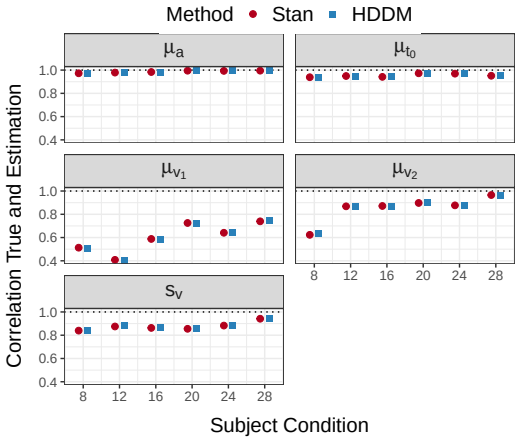
(a) Experiment 1, participant-level parameters.



(b) Experiment 2, participant-level parameters.

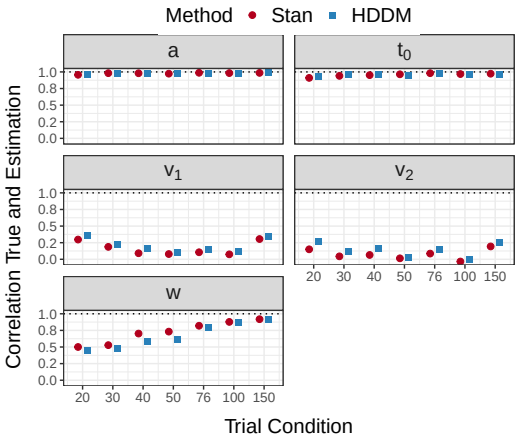


(c) Experiment 1, group-level parameters.

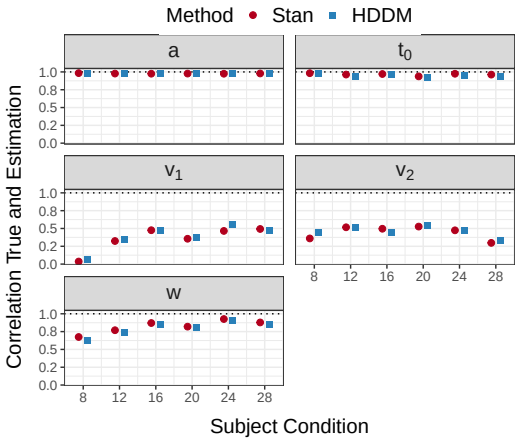


(d) Experiment 2, group-level parameters.

Figure 5.17: Correlations for results with corrected priors in Stan for Experiments 1 and 2.



(a) Experiment 3, participant-level parameters.



(b) Experiment 4, participant-level parameters.

Figure 5.18: Correlations for results with corrected priors in Stan for Experiments 3 and 4.

References

Arnold, N. R., Bröder, A., & Bayen, U. J. (2015). Empirical validation of the diffusion model for recognition memory and a comparison of parameter-estimation methods. *Psychological Research*, *79*(5), 882–898. https://doi.org/10.1007/s00426-014-0608-y

ArviZ Development Team. (2023). Effective Sample Size (0.16.1). Retrieved July 21, 2023, from https://python.arviz.org/en/stable/api/generated/arviz.ess.html#arviz.ess

Boehm, U., Annis, J., Frank, M. J., Hawkins, G. E., Heathcote, A., Kellen, D., Krypotos, A.-M., Lerche, V., Logan, G. D., Palmeri, T. J., van Ravenzwaaij, D., Servant, M., Singmann, H., Starns, J. J., Voss, A., Wiecki, T. V., Matzke, D., & Wagenmakers, E.-J. (2018). Estimating across-trial variability parameters of the Diffusion Decision Model: Expert advice and recommendations. *Journal of Mathematical Psychology*, *87*(4), 46–75. https://doi.org/10.1016/j.jmp.2018.09.004

Brooks, S., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*, 434–455.

Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: linear ballistic accumulation. *Cognitive Psychology*, *57*(3), 153–178. https://doi.org/10.1016/j.cogpsych.2007.12.002

Bürkner, P.-C. (2017). brms : An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software*, *80*(1). https://doi.org/10.18637/jss.v080.i01

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, *76*(1). https://doi.org/10.18637/jss.v076.i01

Chuan-Peng, H., Geng, H., Zhang, L., Fengler, A., Frank, M. J., & Zhang, R.-Y. (2022). *A hitchhiker's guide to bayesian hierarchical drift-diffusion modeling with dockerHDDM*.

Eriksen, B. A., & Erisken, C. W. (1974). Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception & Psychophysics*, *16*(1), 143–149.

Fengler, A., Omar, A., Xu, P., Bera, K., & Frank, M. J. (2023). HSSM Documentation. Retrieved August 17, 2023, from https://lnccbrown.github.io/HSSM/

Google Groups HDDM Users. (2023). Does the source code differ from documentation? [Online Forum Post]. Google Groups. Retrieved June 9, 2023, from https://groups.google.com/g/hddm-users/c/hBqxXld26rI

Hartmann, R., & Klauer, K. C. (2021). Partial derivatives for the first-passage time distribution in Wiener diffusion models. *Journal of Mathematical Psychology*, *103*, 102550. https://doi.org/10.1016/j.jmp.2021.102550

HDDM Development Team. (2023a). HDDM Github base.py. Retrieved June 7, 2023, from https://github.com/hddm-devs/hddm/blob/master/hddm/models/base.py

HDDM Development Team. (2023b). HDDM Github hddm_info.py. Retrieved June 7, 2023, from https://github.com/hddm-devs/hddm/blob/master/hddm/models/hddm_info.py

Heathcote, A., Lin, Y.-S., Reynolds, A., Strickland, L., Gretton, M., & Matzke, D. (2019). Dynamic models of choice. *Behavior Research Methods*, *51*(2), 961–985. https://doi.org/10.3758/s13428-018-1067-y

Henrich, F., Hartmann, R., Pratz, V., Voss, A., & Klauer, K. C. (2023). The Seven-parameter Diffusion Model: An Implementation in Stan for Bayesian Analyses. *Behavior Research Methods*. https://doi.org/10.3758/s13428-023-02179-1

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623.

Kumar, R., Carroll, C., Hartikainen, A., & Martin, O. (2019). ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, *4*(33), 1143. https://doi.org/10.21105/joss.01143

Lerche, V., & Voss, A. (2019). Experimental validation of the diffusion model based on a slow response time paradigm. *Psychological Research*, *83*(6), 1194–1209. https://doi.org/10.1007/s00426-017-0945-8

Lerche, V., Voss, A., & Nagler, M. (2017). How many trials are required for parameter estimation in diffusion modeling? A comparison of different optimization criteria. *Behavior Research Methods*, *49*(2), 513–537. https://doi.org/10.3758/s13428-016-0740-2

Lin, Y.-S., & Strickland, L. (2020). Evidence accumulation models with R: A practical guide to hierarchical Bayesian methods. *The Quantitative Methods for Psychology*, *16*(2), 133–153. https://doi.org/10.20982/tqmp.16.2.p133

Matzke, D., & Wagenmakers, E.-J. (2009). Psychological interpretation of the ex-Gaussian and shifted Wald parameters: A diffusion model analysis. *Psychonomic Bulletin & Review*, *16*(5), 798–817. https://doi.org/10.3758/PBR.16.5.798

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 113–162). Chapman & Hall/CRC.

Patil, A., Huard, D., & Fonnesbeck, C. (2010). PyMC : Bayesian Stochastic Modelling in Python. *Journal of Statistical Software*, *35*(4). https://doi.org/10.18637/jss.v035.i04

R Core Team. (2021). *R: A Language and Environment for Statistical Computing*. https://www.R-project.org/

Ratcliff, R. (1978). A Theory of Memory Retrieval. *Psychological Review*, *85*(2), 59–108.

Ratcliff, R., & Childers, R. (2015). Individual Differences and Fitting Methods for the Two-Choice Diffusion Model of Decision Making. *Decision (Washington, D.C.)*, *2015*. https://doi.org/10.1037/dec0000030

Ratcliff, R., & Rouder, J. N. (1998). Modelling response times for two-choice decisions. *Psychological Science*, *9*(5), 347–356.

Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion Decision Model: Current Issues and History. *Trends in Cognitive Sciences*, *20*(4), 260–281. https://doi.org/10.1016/j.tics.2016.01.007

Rubenstein, H., Garfield, L., & Millikan, J. A. (1970). Homographic entries in the internal lexicon. *Journal of Verbal Learning and Verbal Behavior*, *9*(5), 487–494. https://doi.org/10.1016/S0022-5371(70)80091-3

Stan Development Team. (2022). Stan Modeling Language Users Guide and Reference Manual (version 2.31). Retrieved November 11, 2022, from https://mc-stan.org

Stan Development Team. (2023). Effective Sample Size (version 2.32). Retrieved July 21, 2023, from https://mc-stan.org/docs/reference-manual/effective-sample-size.html

Stan Development Team. (2024). Reparameterization (version 2.33). Retrieved January 16, 2024, from https://mc-stan.org/docs/stan-users-guide/reparameterization.html

Stevenson, N., Donzallaz, M. C., Innes, R. J., Forstmann, B., Matzke, D., & Heathcote, A. (2024). EMC2: An R Package for cognitive models of choice. *Preprint*. https://doi.org/10.31234/osf.io/2e4dq

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating Bayesian Inference Algorithms with Simulation Based Calibration. *arXiv*, *arXiv:1804.06788v2*.

Vandekerckhove, J., & Tuerlinckx, F. (2007). Fitting the Ratcliff diffuion model to experimental data. *Psychonomic Bulletin & Review*, *14*(6), 1011–1026.

Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2011). Hierarchical diffusion models for two-choice response times. *Psychological Methods*, *16*(1), 44–62. https://doi.org/10.1037/a0021765

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *arXiv*, *arXiv:1903.08008v5*.

Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, *32*(7), 1206–1220.

Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, *39*(4), 767–775.

Wabersich, D., & Vandekerckhove, J. (2013). Extending JAGS: A tutorial on adding custom distributions to JAGS (with a diffusion model example). *Behavior Research Methods*, *46*(1), 15–28. https://doi.org/10.3758/s13428-013-0369-3

Wagenmakers, E.-J., van der Maas, H. L. J., & Grasman, R. P. P. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, *14*(1), 3–22.

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in Neuroinformatics*, *7*, 14. https://doi.org/10.3389/fninf.2013.00014

Wiecki, T. V., Sofer, I., Pedersen, M. L., Fengler, A., Govindarajan, L., Bera, K., & Frank, M. J. (2022). HDDM Documentation. Retrieved August 17, 2023, from https://hddm.readthedocs.io/en/latest/index.html

Chapter 6

Paper: Modeling Truncated and Censored Data With the Diffusion Model in Stan

Henrich, F., and Klauer, K.C. (2025). Modeling Truncated and Censored Data With the Diffusion Model in Stan. *Behavior Research Methods*.

Abstract

Reaction time data in psychology are frequently censored or truncated. For example, two-alternative forced-choice tasks that are implemented with a response window or response deadline give rise to censored or truncated data. This must be accounted for in the data analysis as important characteristics of the data, such as the mean, standard deviation, skewness, and correlations, can be strongly affected by censoring or truncation. In this paper, we use the probabilistic programming language Stan to analyze such data with Bayesian diffusion models. For this purpose, we added the functionality to model truncated and censored data with the diffusion model by adding the cumulative distribution function for reaction times generated from the diffusion model and its complement to the source code of Stan. We describe the usage of the truncated and censored models in Stan, test their performance in recovery and simulation-based calibration, and reanalyze existing datasets with the new method. The results of the recovery studies are satisfactory in terms of correlations ($r = .93 - 1.00$), coverage ($93 - 95\%$ of true values lie in the 95% highest density interval), and bias. Simulation-based calibration studies suggest that the new functionality is implemented without errors. The reanalysis of existing datasets further validates the new method.

Keywords: Ratcliff diffusion model · Stan · truncation · censoring

6.1 Introduction

Truncation and censoring frequently occur in psychological data collection (Barchard & Russell, 2024; Ulrich & Miller, 1994). For reaction time data, truncated and censored data regularly arise in psychological studies as a consequence of using response windows or deadlines. These are sometimes introduced in the analysis of data to exclude reaction times that appear too short or too long, but they are also sometimes already built into the study procedures to push participants to respond within a specific temporal window. For example, in the area of social cognition (Carlston et al., 2024), experiments frequently use two-alternative forced-choice tasks to measure implicit mechanisms in stereotyping and prejudice. To reveal fast-acting, possibly implicit processes in stereotyping and prejudice, researchers focus on the outcomes of fast automatic processing at the expense of slow controlled processes. One way to facilitate this is to introduce a response window, forcing participants to respond quickly.

Below, we reanalyze such data stemming from the First-Person Shooter Task (FPST, Correll et al., 2002). In this and the similar Weapon Identification Task (WIT, Payne, 2001), participants are to discriminate between a weapon and a harmless object, independent of the skin color of a person shown before (WIT) or in parallel (FPST) with the target object. One central finding is that a harmless object is more often mistaken for a weapon when the person is Black than when the person is White. Moreover, participants are faster to correctly detect a weapon when the person is Black than when the person is White (e.g., Payne, 2001, 2006). Thus, there is *racial bias* in the accuracy data as well as in the reaction time data.

To make sure that participants respond as fast as possible, response deadlines are often implemented in the task. If participants do not respond prior to a certain response deadline, the trial is terminated and excluded from subsequent analyses (e.g., Lambert et al., 2003; Payne, 2001). Typical response deadlines in such tasks range from 500 ms (e.g., Todd et al., 2016) to 850 ms (e.g., Johnson et al., 2017).

Another use of response windows relies on the fact that the effects that are of interest are often considerably more pronounced in the accuracy data when a response window is in place. This has been found to increase the size and reliability of effects in some paradigms (Draine & Greenwald, 1998; Krause et al., 2012). Finally, as already mentioned, response windows are regularly imposed post-hoc in outlier analyses to exclude responses with implausibly short or long reaction times.

Depending on the implementation of the response window, two different types of data arise: truncated data or censored data. Since the effects of truncation or censoring on summary statistics such as mean, median, standard deviation, and skewness is regularly too large to ignore (Ulrich & Miller, 1994), data analysts are well advised to account for these effects. Here, we focus on analyses using diffusion models, which are frequently applied to data from two-alternative forced-choice tasks. For example, in the context of the FPST and the WIT, diffusion modeling has been employed by Correll et al. (2002), Payne (2001), Pleskac et al. (2018), Rivers (2017), Thiem et al. (2019), and Todd et al. (2020).

Diffusion models model response times and responses simultaneously, thereby maximizing the use of the available data. The basic diffusion model incorporates four parameters (Ratcliff, 1978), which can be interpreted in terms of psychological processes; an extended version of the diffusion model uses seven model parameters (Ratcliff & Rouder, 1998). A number of software packages allow one to estimate the parameters of the diffusion model such as dedicated modules implemented in Stan (Carpenter et al., 2017), JAGS (Wabersich & Vandekerckhove, 2013), WinBUGS (Vandekerckhove & Tuerlinckx, 2007), stand-alone software such as fast-dm (Voss & Voss, 2007), HDDM (Wiecki et al., 2013), HSSM (Fengler et al., 2023), or R-packages such as EMC2 (Stevenson et al., 2024), DMC (Heathcote et al., 2019), and ggdmc (Lin & Strickland, 2020), among others. However, only a few of these data-analytic solutions are able to directly model truncated or censored response time data. For example, the probabilistic programming language Stan does not have a built-in method to handle censoring or truncation with the diffusion model. To fill this gap, we added the functionality to deal with truncated and censored data in diffusion model analyses to Stan. This requires implementing the cumulative distribution function (CDF) of response-time distributions that arise under the diffusion model and its complement (CCDF) in Stan.

We chose Stan because of its usefulness and popularity as a free and open-source software package that provides users with many functions for Bayesian statistical inference and hierarchical modeling for a huge range of model families. Besides the diffusion model, numerous other models can be estimated in Stan, as many probability density functions like the ones for Bernoulli, beta, binomial, exponential, normal, and Poisson distributions, to name just a few, are implemented in Stan. These functions enable the user to choose the priors for the model parameters in a flexible manner. Furthermore, the probability density function (PDF) of the seven-parameter diffusion model is already available in this programming language (Henrich et al., 2023).

The goal of the present paper is to document and validate the new functionality to model truncated or censored data with the diffusion model in Stan using the CDF and CCDF functions. In the following sections, we provide a brief introduction to the diffusion model. Next, we elaborate on the notions of truncated and censored data and how they can be modeled in Stan. Following this, we conduct two validity checks for the new functionality: a simulation study showing good recovery for truncated and censored data, and a simulation-based calibration study. Finally, we present a reanalysei of existing datasets from the First-Person Shooter Task with a basic, a censored, and a truncated diffusion model.

6.2 The Diffusion Model

The diffusion model (Ratcliff, 1978) is a member of the family of information accumulation models. It is widely used to model two-alternative forced-choice tasks by simultaneously modeling response time and responses (for a review, see Ratcliff et al., 2016).

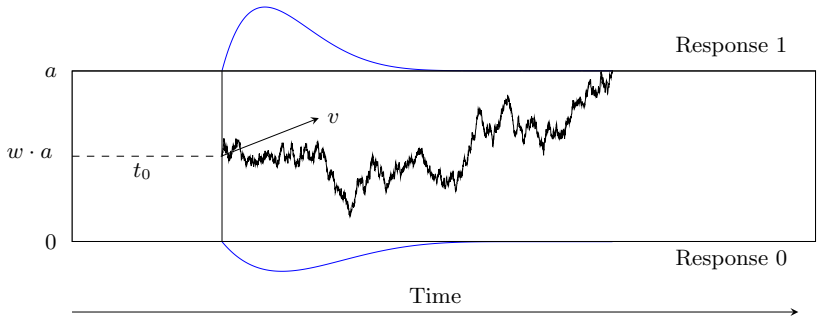


Figure 6.1: Realization of a Four-Parameter Diffusion Process Modeling the Binary Decision Process. *Note.* The parameters are the *boundary separation a* for two response alternatives, the *relative starting point w*, the *drift rate v*, and the *non-decision time* t_0. The decision process is illustrated as a jagged line between the two boundaries. The predicted distributions of the reaction times are depicted as curved lines below and above the response boundaries (blue).

In the basic model, four parameters describe the decision process (see Figure 6.1): The process starts at a *relative starting point*, *w*, between the two response boundaries. Bits of information are noisily accumulated until one of the boundaries is reached, in which case the response associated with that boundary is initiated. The distance between both response boundaries is the *boundary separation*, *a*. The direction of the accumulation process is described by the *drift rate*, *v*, which corresponds to the average rate of information uptake. And finally, all processes that do not belong to the decision process itself, for example, the time taken for early perceptual encoding or production of the motor response, are summed in the *non-decision time*, t_0. The model predicts the reaction time distributions for the response associated with each boundary and the probabilities with which either response is made. Ratcliff and Rouder (1998) later extended the four-parameter diffusion model by adding three inter-trial variabilities (in *relative starting point*, *drift rate*, and *non-decision time*) to account for several reaction time patterns that could not be handled by the basic diffusion model. For example, when the relative starting point is set to 0.5, as is a priori plausible in many discrimination tasks when responses are coded as false versus correct, the basic diffusion model predicts the same response-time distributions for false and correct responses. In contrast, as explained by Ratcliff and Rouder (1998), the seven-parameter diffusion model allows one to account for error responses that are systematically faster (or slower) than correct responses.

Existing implementations of the diffusion model enable the estimation of four or seven parameters in both non-hierarchical and hierarchical settings, as well as in non-Bayesian and Bayesian contexts. However, only a few of the existing implementations of the diffusion model are able to directly model censored or truncated data arising from the use of response windows or response deadlines.[1] Instead, many researchers use models that treat data as if they were not

[1] In Bayesian analyses, it is possible to model censored data indirectly via a data augmentation step in which the missing reaction times are parameters of an extended model with values imputed in each step of a Markov-chain Monte Carlo algorithm (see Kruschke, 2015, Chap. 25.4). Note, however, that this approach requires further modification in diffusion modeling when not only the reaction time is missing, but also the response itself.

truncated or censored (e.g., Correll et al., 2015; Todd et al., 2020). In the following, we will elaborate on the notions of truncated and censored data and on how such data can be modeled with the diffusion model in Stan.

6.3 Truncated and Censored Data

6.3.1 Truncated Data

Data are called *truncated* when there is no information available for analysis from trials with values larger (or smaller) than a right (or left) boundary. In our example of reaction time experiments, reaction time data are truncated if trials with reaction times outside the response window are excluded from the analysis. Not even a count of those omitted trials is kept.

Mathematically, truncation can be defined as follows: First, the notion of *cumulative distribution functions* is needed. A cumulative distribution function (CDF) of a real-valued random variable X evaluated at x is defined as the probability P that X takes a value less than or equal to x: $\text{CDF}(x) = P(X \leq x)$.

Let X be a random variable, $\text{PDF}(x)$ its probability density function, and $\text{CDF}(x)$ its cumulative distribution function. Then, the PDF of X after truncating to the interval $(L, U]$, such that $L < X \leq U$, is defined as follows:

$$\text{PDF}(x \mid L < X \leq U) = \frac{\text{PDF}(x) \cdot \mathbb{I}_{\{L<x\leq U\}}}{\text{CDF}(U) - \text{CDF}(L)}, \tag{6.1}$$

where $\mathbb{I}$ is the indicator function taking the value 1 if the condition in the parentheses holds and the value 0 otherwise.

In the case that X is truncated at only one side, its PDF is defined for left truncation as

$$\text{PDF}(x \mid L < X) = \frac{\text{PDF}(x) \cdot \mathbb{I}_{\{L<x\}}}{1 - \text{CDF}(L)}, \tag{6.2}$$

and for right truncation as

$$\text{PDF}(x \mid X \leq U) = \frac{\text{PDF}(x) \cdot \mathbb{I}_{\{x\leq U\}}}{\text{CDF}(U)} \tag{6.3}$$

6.3.2 Censored Data

Data are *censored* when observations that are above or below a right or left boundary value are reported as occurrences of the event $(x > U)$, for U the right bound, or as occurrences of the event $(x \leq L)$, for L the left bound, respectively. Like for truncated data, the range of the possible values is restricted, but the number of observations that fall outside the boundaries is

kept, whereas in truncation, no count would be kept.

Let X be a random variable, $\text{PDF}_X(x)$, and $\text{CDF}_X(x)$ be the probability density function and the cumulative distribution function of X. Let Z be a second random variable that is censored in the interval $(L,U]$. Let Z take the value of X if a realization of X is within the boundaries, and the value $l \leq L$ if it is smaller than the lower bound, and the value $u > U$ if it is larger than the upper bound:

$$z = \begin{cases} x, & \text{for } L < x \leq U \\ l, & \text{for } x \leq L \\ u, & \text{for } x > U \end{cases} \tag{6.4}$$

The probability density function $\text{PDF}_Z(z)$ of the censored variable Z is then given by:

$$\text{PDF}_Z(z) := \begin{cases} \text{PDF}_X(x), & \text{for } L < z \leq U \\ \text{CDF}_X(L), & \text{for } z = l \\ 1 - \text{CDF}_X(U), & \text{for } z = u \end{cases} \tag{6.5}$$

In the case that Z is censored at only one side, its probability function is defined for left censoring as

$$\text{PDF}_Z(z) := \begin{cases} \text{PDF}_X(x), & \text{for } z > L \\ \text{CDF}_X(L), & \text{for } z = l \end{cases} \tag{6.6}$$

and for right censoring as

$$\text{PDF}_Z(z) := \begin{cases} \text{PDF}_X(x), & \text{for } z \leq U \\ 1 - \text{CDF}_X(U), & \text{for } z = u \end{cases} \tag{6.7}$$

6.4 Modeling Truncated Data in Stan

As the CDF and the CCDF are needed to model truncated or censored data, we recently extended the diffusion model family in Stan by these functions. Remember that the cumulative distribution function is defined as the probability P that X takes a value less than or equal to an evaluated value x: $\text{CDF}(x) = P(X \leq x)$. Furthermore, the *complementary cumulative distribution function* is defined as the complement of the CDF: $\text{CCDF}(x) = 1 - \text{CDF}(x)$. CDF and CCDF for reaction time distributions under the diffusion model are, however, traditionally defined slightly differently in terms of so-called *defective* distribution functions as explained in the following.

For this purpose, we discriminate between the terms *(left/right) rt-bound* to refer to the (left-

/right) response-time bound in the response window, and the terms *response-0-* and *response-1 boundary* for the lower and upper response boundary, respectively, of the diffusion process.

Consider first the basic four-parameter diffusion model. Let a, w, v, and t_0 be the diffusion model parameters as introduced above, and let x be an observed reaction time. It is important to highlight that usually, the PDF of a random variable sums up or integrates to 1. This also means that the CDF converges to 1 as x increases. In the diffusion model, we see a split for the data belonging to the response-0 boundary and the response-1 boundary. This means that we can define the probability density function and the cumulative distribution function for the response-0 boundary, PDF_0 and CDF_0, and for the response-1 boundary, PDF_1 and CDF_1. One possibility to implement the functions is as *defective* functions. That is, not the individual PDFs and CDFs but the sum $\text{PDF}_0 + \text{PDF}_1$, or $\text{CDF}_0 + \text{CDF}_1$ integrates to 1 or converges to 1, respectively. In this case, the cumulative distribution functions converge to the probability to hit the response-boundary: $\text{CDF}_1(\infty \mid a,w,v) = P(a,w,v)$ for the response-1 boundary and $\text{CDF}_0(\infty \mid a,w,v) = \text{CDF}_1(\infty \mid a,1-w,-v) = P(a,1-w,-v)$ for the response-0 boundary, where $P(a,w,v)$ is the probability that the diffusion process terminates at the responsee-1-boundary (see Eq. 6.9). It also follows that the defective complementary cumulative distribution function can be written as $\text{CCDF}_1(x \mid a,w,v) = P(a,w,v) - \text{CDF}_1(x \mid a,w,v)$ for the response-1 boundary and $\text{CCDF}_0(x \mid a,w,v) = P(a,1-w,-v) - \text{CDF}_0(x \mid a,w,v)$ for the response-0 boundary.

In the following, we introduce the definition of the cumulative distribution function for the response-1 boundary. There are two expressions of the CDF of decision times: one that supports efficient computation of its values for relatively large times, and the other one is more attuned to small times. The formula for the large-time CDF of decision times (excluding the additive reaction time components summarized in t_0 for the time being) at the response-1-boundary is stated as follows (adapted from response-0 boundary definition in Hartmann & Klauer, 2021):

$$\text{CDF}_1(x \mid a,w,v) \coloneqq P(a,w,v) - \exp(va(1-w) - \frac{v^2x}{2})\text{CDF}_l(x \mid a,w,v), \tag{6.8}$$

where $P(a,w,v)$ is the probability to hit the response-1-boundary, defined as

$$P(a,w,v) = \begin{cases} \frac{1-\exp(2vaw)}{\exp(-2va(1-w))-\exp(2vaw)}, & \text{for } v \neq 0 \\ w, & \text{for } v = 0, \end{cases} \tag{6.9}$$

and

$$\text{CDF}_l(x \mid a,w,v) = \frac{2\pi}{a^2}\sum_{k=1}^{\infty}\frac{k\sin(k\pi(1-w))}{v^2+(k\pi)^2/a^2}\exp(-\frac{k^2\pi^2x}{2a^2}). \tag{6.10}$$

The formula for the small-time CDF at the response-1-boundary is stated as follows:

$$\mathrm{CDF}_1(x \mid a,w,v) \coloneqq \exp(va(1-w) - \frac{v^2x}{2})\mathrm{CDF}_s(x \mid a,w,v), \tag{6.11}$$

where

$$\begin{aligned}\mathrm{CDF}_s(x \mid a,w,v) = \sum_{k=0}^{\infty}(-1)^k\phi\big(\frac{a(k+w_k^*)}{\sqrt{x}}\big)\times \\ \big(R\big(\frac{a(k+w_k^*)+vx}{\sqrt{x}}\big)+R\big(\frac{a(k+w_k^*)-vx}{\sqrt{x}}\big)\big),\end{aligned} \tag{6.12}$$

where $w_k^* = (1-w)$ for k even, $w_k^* = w$ for k odd, and R is Mill's ratio (see section 1 in the supplementary materials of Hartmann & Klauer, 2021; Mitrinović, 1970). The CDF for the response-0-boundary is $\mathrm{CDF}_0(x \mid a,w,v) = \mathrm{CDF}_1(x \mid a,1-w,-v)$.

From here, it is possible to compute the CDF and CCDF taking into account additive reaction time components t_0 as well as the CDF and CCDF for the seven-parameter diffusion model, which also includes the intertrial variabilities for t_0, v and w, where needed. The latter step requires numerical integration in some cases.

For these reasons, the Eqs. (6.1) to (6.3) for the density of the truncated data also have to be adapted for the diffusion model. Let L denote the left rt-bound and U denote the right rt-bound of a response window.

Then, the density of truncated data from response boundary resp $\in \{0,1\}$ can be formulated as follows:

$$\begin{aligned}&\mathrm{PDF}_{\mathrm{resp}}(x \mid L < X \leq U,a,w,v) = \\ &\frac{\mathrm{PDF}_{\mathrm{resp}}(x \mid a,w,v)\cdot\mathbb{I}_{\{L<x\leq U\}}}{\big(\mathrm{CDF}_0(U \mid a,w,v)+\mathrm{CDF}_1(U \mid a,w,v)\big)-\big(\mathrm{CDF}_0(L \mid a,w,v)+\mathrm{CDF}_1(L \mid a,w,v)\big)}\end{aligned} \tag{6.13}$$

The density of left truncated data can be formulated as follows:

$$\mathrm{PDF}_{\mathrm{resp}}(x \mid L < X,a,w,v) = \frac{\mathrm{PDF}_{\mathrm{resp}}(x \mid a,w,v)\cdot\mathbb{I}_{\{L<x\}}}{1-\big(\mathrm{CDF}_0(L \mid a,w,v)+\mathrm{CDF}_1(L \mid a,w,v)\big)}, \tag{6.14}$$

and the density of right truncated data can be formulated as follows:

$$\mathrm{PDF}_{\mathrm{resp}}(x \mid X \leq U,a,w,v) = \frac{\mathrm{PDF}_{\mathrm{resp}}(x \mid a,w,v)\cdot\mathbb{I}_{\{x\leq U\}}}{\mathrm{CDF}_0(U \mid a,w,v)+\mathrm{CDF}_1(U \mid a,w,v)} \tag{6.15}$$

Next, we describe how to define a truncated model in the model block of a .stan-file. For a detailed description of the other .stan-file blocks (data-, and parameters-block) see Henrich et al. (2023).

As of Stan version 2.35.0, the seven-parameter version of the diffusion model is available

in Stan as described in Henrich et al. (2023).[2] The three additional parameters in the seven-parameter diffusion model comprise the inter-trial variability in the relative starting point, called s_w, in the non-decision time, called s_{t_0}, and in the drift rate, called s_v (see Henrich et al., 2023, for more information). For a reaction time x at the response-1-boundary, this full model can be called with the following command:

```
x ~ wiener(a,t0,w,v,sv,sw,st0);
```

or

```
target += wiener_lpdf(x|a,t0,w,v,sv,sw,st0);
```

For a reaction time at the response-0-boundary, replace w by $1-w$ and v by $-v$.

All smaller models can be called by fixing one or more parameters to 0. For example, a model without the inter-trial variability in the relative starting point looks as follows:

```
x ~ wiener(a,t0,w,v,sv,0,st0);
```

or

```
target += wiener_lpdf(x|a,t0,w,v,sv,0,st0);
```

The four-parameter model can be called by setting all inter-trial variabilities to 0:

```
x ~ wiener(a,t0,w,v,0,0,0);
```

or

```
target += wiener_lpdf(x|a,t0,w,v,0,0,0);
```

As the functions are implemented defectively, a truncated diffusion model cannot be calculated with the truncation functor $T[,]$ (see Stan Development Team, 2023b). This means the function call: `x ~ wiener(...)T[L,U]` does not work the way it is supposed to. When the truncation functor is called in Stan, Stan searches for a CDF implementation internally. In the case of the diffusion model, Stan would find the CDF, but is not aware of its defective implementation and calculates the computations as if it were a non-defective CDF. This causes misleading and incorrect results. Therefore, to implement the truncated model, write out Eq. 6.13 on the log-scale with left_bound = L and right_bound = U, where `wiener_lcdf()` calls the logarithmized CDF of the diffusion model at the response-1-boundary:

---

[2] Note, however, that its name changed from `wiener_full()` to `wiener()` in the course of its recent release in Stan.

```
model { // compute the denominator of Formula 12 on log scale
 real denom = log_diff_exp(
  log_sum_exp(
   wiener_lcdf(right_bound | a, t0, w, v, sv, sw, st),
   wiener_lcdf(right_bound | a, t0, 1-w, -v, sv, sw, st)),
  log_sum_exp(
   wiener_lcdf(left_bound | a, t0, w, v, sv, sw, st),
   wiener_lcdf(left_bound | a, t0, 1-w, -v, sv, sw, st))
  ); // parenthesis log_diff_exp
 // compute log-likelihood
 for (i in 1:N) {
  if (resp[i] == 1) { // response-1 boundary
   target += wiener_lpdf(rt[i] | a, t0, w, v, sv, sw, st);
  } else { // response-0 boundary (mirror v and w)
   target += wiener_lpdf(rt[i] | a, t0, 1-w, -v, sv, sw, st);
  }
  target += -denom;
 } // end for
}
```

How to call a truncated model within the parallelization routine of `reduce_sum` or with truncation to only one side (in line with Eqs. (6.14) and (6.15)) is described in Appendix 6.9.1.

6.5 Modeling Censored Data in Stan

For the censored model, we distinguish two cases: In the first case, the responses of the censored trials are known, but the reaction times are not known. In the second case, neither the responses nor the reaction times of the censored trials are known. Note that the second case differs from a truncated model in the fact that the number of censored trials is still known. Consider first the case where the response is known even for censored data.

To model such data in Stan, the left and right rt-bounds, `left_bound` and `right_bound`, respectively, are handed over in the **data block**, as well as a vector `censored` that tracks whether a trial is censored $(= 1)$ or not $(= 0)$, and counts of trials censored at the left rt-bound and counts of trials censored at the right rt-bound for each response in $\{0, 1\}$. There are four such count variables: `N_cens_left_0`, `N_cens_left_1`, `N_cens_right_0`, `N_cens_right_1`:

```
model{ // ... // definition of priors for all model parameters
 for (i in 1:N) {
  if (resp[i] == 1) { // response-1 boundary
   if (censored[i] == 0) {
    x[i] ~ wiener(a, t0, w, v, sv, sw, st0);
```

```
    }
   } else if (resp[i] == 0) { // response-0 boundary
    if (censored[i] == 0) {
     x[i] ~ wiener(a, t0, 1-w, -v, sv, sw, st0);
    } } } // end if, end else if, end for
  // summands for response = 0
  target += N_cens_left_0 * wiener_lcdf(left_bound |
                            a, t0, 1-w, -v, sv, sw, st0);
  target += N_cens_right_0 * wiener_lccdf(right_bound |
                            a, t0, 1-w, -v, sv, sw, st0);
  // summands for response = 1
  target += N_cens_left_1 * wiener_lcdf(left_bound |
                            a, t0, w, v, sv, sw, st0);
  target += N_cens_right_1 * wiener_lccdf(right_bound |
                            a, t0, w, v, sv, sw, st0);
}
```

When data are censored at only one side, omit the lines for the other side in the code.

When data consist of many conditions, it is sometimes more convenient to loop over all trials instead of using count variables as described above, using the following notation and code. A vector containing the information whether a trial is censored or not, here `censored`, needs to be handed over in the **data block**. This vector splits the data into three bins: all trials `i` with `censored[i]=0` are censored below the left rt-bound, all trials `i` with `censored[i]=1` fall between the rt-bounds, and all trials `i` with `censored[i]=2` are censored above the right rt-bound. For non-censored trials, the log-PDF is computed, for left censored trials, the log-CDF is computed, and for right censored trials, the log-CCDF is computed:

```
model{ // ... // definition of priors for all model parameters
 for (i in 1:N) { //right censored at right_bound
  if (resp[i] == 1) { // upper response boundary
   if (censored[i] == 1) {
    target += wiener_lpdf(x[i] | a, t0, w, v, sv, sw, st0);
   } else if (censored[i] == 0) {
    target +=
     wiener_lcdf(left_bound | a, t0, w, v, sv, sw, st0);
   } else if (censored[i] == 2) {
    target +=
     wiener_lccdf(right_bound | a, t0, w, v, sv, sw, st0);
   }
  } else { // lower response boundary (mirror drift and
           // starting point!)
   if (censored[i] == 1) {
    target +=
```

```
17        wiener_lpdf(x[i] | a, t0, 1-w, -v, sv, sw, st0);
18      } else if (censored[i] == 0) {
19       target +=
20        wiener_lcdf(left_bound | a, t0, 1-w, -v, sv, sw, st0);
21      } else if (censored[i] == 2) {
22       target +=
23        wiener_lccdf(right_bound | a, t0, 1-w, -v, sv, sw, st0);
24      }
25     }
26    }
27  }
```

When the data are censored to only one side, omit the case that is not needed. Note that this block can be inserted in the definition of the parallelization function, `partial_sum_fullddm()`, as defined in Appendix 6.9.1. [3]

Censoring sometimes includes the response (i.e., it is known that the reaction time in a trial fell outside the response window, but which response was given is unknown). One method that has been used to model such data has involved inferring the numbers of missing responses of either kind from the observed relative frequencies of the two responses (e.g., Pleskac et al., 2018). This approach has the problem that quite specific assumptions on the missing data have to be made (namely, that the proportions of the two kinds of responses are the same for responses within and outside the response window).

We recommend a principled approach that uses the cumulative distribution functions and their complements to provide the likelihood of censored data. As before, let L be the left rt-bound, and U the right rt-bound, and consider decision times without inter-trial variabilities for the sake of simplicity. It follows that the likelihood p_l of observing a left-censored data point is given by

$$p_l(a,w,v) = \mathrm{CDF}_0(\mathrm{L} \mid a,w,v) + \mathrm{CDF}_1(\mathrm{L} \mid a,w,v), \tag{6.16}$$

whereas the likelihood p_r of a right-censored data point is given by

$$p_r(a,w,v) = \mathrm{CCDF}_0(U \mid a,w,v) + \mathrm{CCDF}_1(U \mid a,w,v). \tag{6.17}$$

See the following code for an example of Stan code implementing this second case of censoring. This model call deals with the problem of unknown responses by computing the probability of choosing the response-1- or response-0 boundary outside the response window. Here, the CDF and/or the CCDF are required, depending upon whether there is only left-censoring, right-censoring, or censoring both to the left and to the right. The following code shows the **func-**

[3] Also note that no NA values can be handed over to Stan. Hence, fill the missing values with some information, e.g., `censored[i]` $= 42$, for coding NA in the i-th field. Every value is fine, except NA, since the vector `censored` splits cases, and the censored cases are independent of the concrete reaction time values.

tions block for a model that is right-censored using the function `partial_sum_fullddm()` for parallel computations. Combine this block with the **model block** in the example in Appendix 6.9.1:

```
functions { // parallelization function
 real partial_sum_fullddm(array[] real rt_slice, int start,
 int end, real a, real t0, real w, real v, real sv, real sw,
 real st, array[] int resp, real right_bound,
 array[] int censored) {
  real ans = 0;
  for (i in start:end) {
   if (censored[i] == 1) { // not censored data
    if (resp[i] == 1) { // upper boundary
     ans += wiener_lpdf(rt_slice[i+1-start] |
             a, t0, w, v, sv, sw, st);
    } else { // lower boundary (mirror v and w)
     ans += wiener_lpdf(rt_slice[i+1-start] |
             a, t0, 1 - w, -v, sv, sw, st);
    }
   } else { // censored data
    ans += log_sum_exp(
     wiener_lccdf(right_bound | a, t0, w, v, sv, sw, st),
     wiener_lccdf(right_bound | a, t0, 1-w, -v, sv, sw, st);
   }
  } // end for
  return ans;
 } // end partial_sum_fullddm
} // end functions
```

6.6 Validating the new Implementation

In this section, we present two consistency checks for the new methods for analyzing truncated and censored data: First, a simulation study to test parameter recovery, and second, a simulation-based calibration study (SBC, Talts et al., 2018) to show the correctness of the implemented algorithm. Both studies have an analogous design as the consistency checks for the implementation of the (non-truncated, non-censored) diffusion model in Stan (see Henrich et al., 2023).

We chose a typical experimental design and priors based on findings in the literature, drew the true parameters from distributions that coincide with these priors, and simulated data using the true parameters. We simulated data that are truncated with a right rt-bound as well as data that are censored with a right rt-bound (in the following referred to as *truncated analysis* and

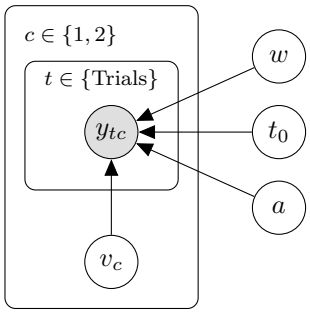


Figure 6.2: Graphical Model Representation in the Simulation Study. *Note.* Each data point y_{tc} (comprising of reaction time and response) within trial *t* and condition *c* depends on the four diffusion parameters. The drift rate varies between conditions. This results in five parameters to estimate.

censored analysis, respectively). These both correspond to a task with a response deadline in reaction time experiments. We then fitted the data with the appropriate (truncated or censored) model using the parameter distributions underlying data generation in the simulation process as priors. Finally, we analyzed results with respect to recovery and with respect to simulation-based calibration.

6.6.1 Design

The simulated datasets comprise trials from two conditions, representing two different stimuli, where Condition 1 has positive, and Condition 2 negative *drift rate*. All other parameters are shared across conditions. For reasons of feasible computational time, we simulated data from a non-hierarchical four-parameter model, instead of a seven-parameter model. This is a common design in reaction time experiments (e.g., Arnold et al., 2015; Johnson et al., 2020; Ratcliff & Smith, 2004; Voss et al., 2004). A graphical model representation is given in Figure 6.2.

6.6.2 Ground Truths, Priors, and Parameter Distributions Underlying Data Generation

The true parameters for the simulation study, denoted as the *ground truths*, are randomly drawn from parameter distributions which coincide with the priors used in the model and are shown in Table 6.1.

The choice of the priors and therefore also of the parameter distributions underlying the data generation are based on typical ranges of parameter values as reported in the literature. Specifically, the distributions for *a* and *w* are based on Wiecki et al. (2013, Fig. 1 in the Supplements), the parameter distribution for t_0 is based on Matzke and Wagenmakers (2009, Table 3), and

Parameter	Prior / Data-generating parameter distribution
a	$\mathcal{N}(1,1)$ T$[0.5,3]$
v	$\mathcal{N}(2,3)$ T$[0,5]$
w	$\mathcal{N}(0.5,0.1)$ T$[0.3,0.7]$
t_0	$\mathcal{N}(0.435,0.12)$ T$[0.2,1]$

Table 6.1: Parameter Distribution for Data Generation in the Simulation Study. *Note.* $\mathcal{N}$ = normal distribution; T$[.,.]$ = truncation.

the parameter distribution for v is the one used in Wiecki et al. (2013)[4]. To simulate the two conditions with different drift rates, we drew two values from the drift rate distribution and multiplied the second value with -1, such that in Condition 1, the drift rate is directed to the response-1 boundary and in Condition 2 to the response-0 boundary.

6.6.3 Datasets

Following Henrich et al. (2023), we drew 2000 ground truths from the data-generating parameter distributions for the truncated analysis and another 2000 ground truths for the censored analysis. Then, for each analysis, we simulated two datasets for each ground truth: one comprising 100 trials (50 per condition) and one comprising 500 trials (250 per condition). This results in four simulation studies, each comprising 2000 datasets (truncated: 100 and 500 trials, and censored: 100 and 500 trials).

Data were simulated in R (R Core Team, 2021) using the sampling method `sampWiener()` of the package WienR (Hartmann & Klauer, 2021), which allows one to sample responses and reaction times from truncated diffusion model response time distributions with a right rt-bound. All three inter-trial variabilities were set to 0.[5] For the truncated analysis, the rt-bound was set to 0.91s. To obtain this value, we first simulated 2000 datasets without a right rt-bound. Next, we determined for each dataset the 80% quantile, that is, an individual right-bound rt-value that splits the specific dataset into 80% less than that value and 20% greater than that value. Finally, we took the mean of all these individual right-bound rt-values to obtain a general right rt-bound for this simulation study, meaning that all datasets in the two truncated studies are truncated above 0.91. This results in 100 and 500 trials, respectively, where each trial has an rt-value less

[4] Wiecki et al. (2013) based their choice of prior distributions for the diffusion model parameters on values reported in the literature and collected by Matzke and Wagenmakers (2009).

[5] A simulation study of the present size would hardly be feasible with the seven-parameter diffusion model, as the required computational time would be very large. As the critical new parts of the code occur in the four-parameter core model, we performed our analyses on the four-parameter diffusion model.

than 0.91s. Note that there is no information on the actual number of truncated trials.

For the censored analysis, the information on the number of trials that are above the rt-bound is included in the model. Here, we first simulated data without any rt-bound. In a second step, we labeled each trial according to whether it had a reaction time below or above the right rt-bound of 0.91, then discarded the reaction time for reaction times above the rt-bound and counted for each of the two drift rate conditions how many of these censored trials had response 0 and 1, respectively.

6.6.4 Method Configuration

Analyses were run on the high-performance computing cluster in Karlsruhe, Germany, BwUniCluster2.0[6], within the framework program bwHPC. For each analysis, we ran four chains (as recommended by Vehtari et al., 2021). Chains were computed in parallel, and each chain was parallelized on up to 15 cores via the Stan internal parallelization routine `reduce_sum()`. The method parameter `max_treedepth` was set to 5 to speed up the sampling process, while still preserving good convergence.

We started computations with 500 warmup and 250 sampling iterations per chain. When results did not converge satisfactorily with this setting, we repeated the analysis of this dataset with increased sampling iterations until all convergence criteria (see below) were met.

6.6.5 Recovery Study

Convergence and Diagnostics

It is recommended to check some convergence criteria before analyzing the results of the estimation process (e.g., Vehtari et al., 2021). Among these criteria are the *effective sample size*, N_{eff}, and a convergence measure, $\hat{R}$.

The *effective sample size* is a measure of how many independent samples contain the same amount of information as the dependent samples obtained by the sampling process. It is recommended that the rank-normalized effective sample size is greater than 400, $N_{\text{eff}} > 400$, for each model parameter (Vehtari et al., 2021). The $\hat{R}$ value is a measure of convergence and should be less than 1.01, $\hat{R} < 1.01$ (Vehtari et al., 2021).

We checked these two criteria for each dataset and reanalyzed those datasets that did not meet the criteria with more *sampling iterations* until all datasets met the criteria. Thus, all effective sample sizes are above 400 and all $\hat{R}$ values are below 1.01.

---

[6] Retrieved September 23, 2025 from https://wiki.bwhpc.de/e/Registration/bwUniCluster

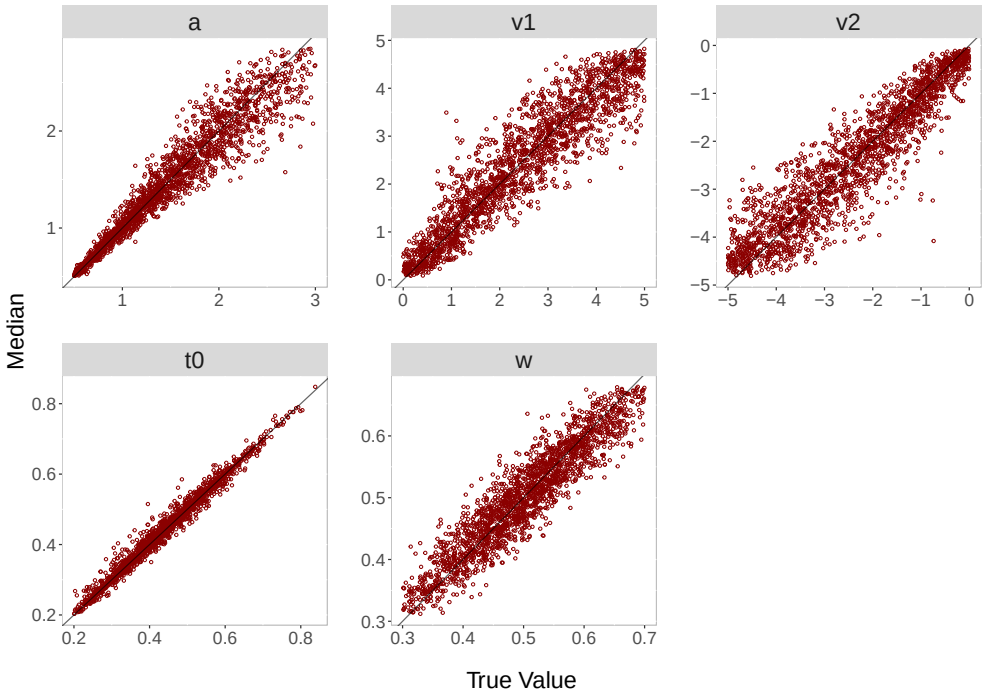


Figure 6.3: Diagonal Plot Between Posterior Median and True Value for 100 Trials for the Truncated Analysis.

Recovery

To assess recovery, we present three measures: *correlations* between the true values and the posterior median, *coverage*, meaning the percentage of times across the datasets that the true value lies in the 50% and 95% highest density interval (HDI), respectively, and a graphical representation of the *bias* via diagonal plots of the true values against the posterior medians. Results for *correlations* and *coverage* are shown in Table 6.2 and results for the bias are shown in Figures 6.3 to 6.6.

As can be seen, in both analyses, correlations for all parameters are close to 1 (all greater than .93) and increase in size for datasets with more trials. Moreover, the coverage values closely match the nominal 50% and 95% values for the two HDIs that we monitored.

The diagonal plots for both analyses for 500 trials (Figures 6.4 and 6.6) show smaller biases than the diagonal plots for 100 trials (Figures 6.3 and 6.5). The diagonal plots for the censored analysis (Figures 6.5 and 6.6) are more narrow than the diagonal plots for the truncated analysis (Figures 6.3 and 6.4) for the same trial numbers.

In summary, the results for the recovery study are satisfactory. As expected, the parameter recoveries based on 500 trials are better than those based on 100 trials; that is, correlations are higher, coverage is better, and biases are smaller. Furthermore, recovery results for the censored analysis are slightly better than recovery results for the truncated analysis, suggesting that the information on the upper tails of the diffusion model reaction time distributions present in the censored data is especially helpful in pinning down parameter estimates.

Par.	r	50% [a]	95% [a]	Par.	r	50% [a]	95% [a]
— 100 Trials, truncated —				— 100 Trials, censored —			
a	.96	50	94	a	.97	50	95
v_1	.93	50	95	v_1	.96	47	95
v_2	.93	50	93	v_2	.96	47	94
t_0	.99	49	95	t_0	.99	47	95
w	.93	49	95	w	.93	48	95
— 500 Trials, truncated —				— 500 Trials, censored —			
a	.99	50	95	a	.99	49	95
v_1	.98	48	95	v_1	.99	50	95
v_2	.98	49	95	v_2	.99	50	95
t_0	1.00	49	95	t_0	1.00	49	95
w	.98	49	94	w	.98	52	94

Table 6.2: Parameter Recovery Study: Evaluation Criteria (Correlations, Coverage) for Parameters Estimated from 100 and 500 Simulated Trials, Respectively, for the Truncated and the Censored Analysis. *Note.* Par.=Parameters; r=Correlations (between true parameter values and posterior medians)
[a] Percent of simulated datasets with true value in the HDI of this percentage.

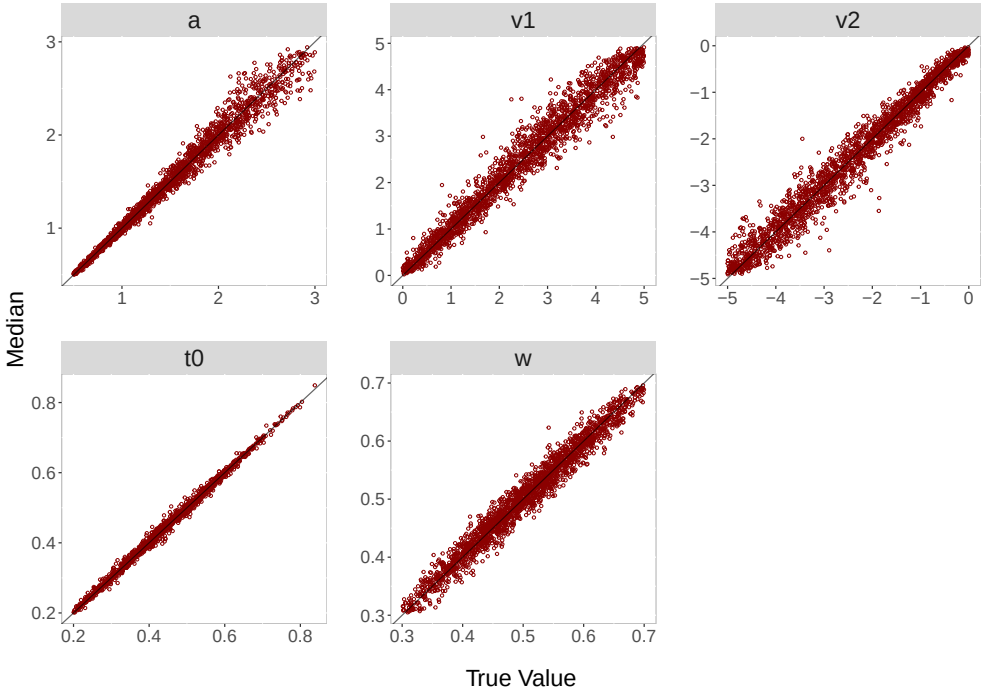


Figure 6.4: Diagonal Plot Between Posterior Median and True Value for 500 Trials for the Truncated Analysis.

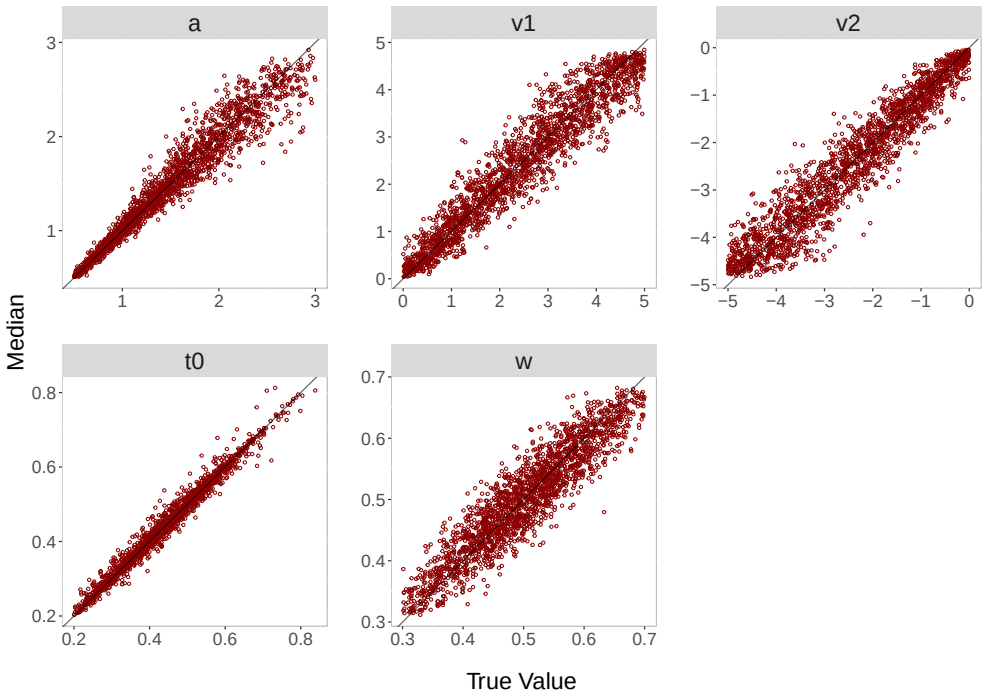


Figure 6.5: Diagonal Plot Between Posterior Median and True Value for 100 Trials for the Censored Analysis.

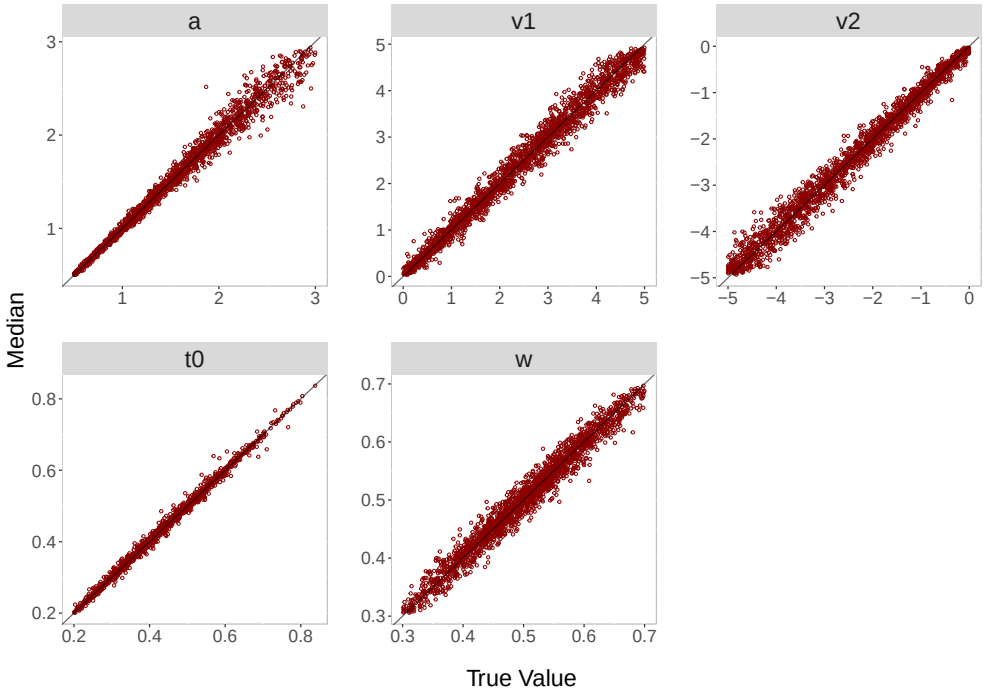


Figure 6.6: Diagonal Plot Between Posterior Median and True Value for 500 Trials for the Censored Analysis.

6.6.6 Simulation-Based Calibration Study

In the Bayesian context, good recovery is neither sufficient nor necessary to demonstrate the validity of a Bayesian algorithm. A more rigorous test is provided by testing simulation-based calibration (SBC, Modrak et al., 2022; Talts et al., 2018). The purpose of an SBC is to show that the implemented algorithm is implemented correctly without errors in the code. This is done by testing whether an algorithm satisfies a consistency condition that it must satisfy if implemented correctly. If this consistency condition is not satisfied, it must be concluded that there are errors in the implementation.

The consistency condition can be stated as follows: If the algorithm is implemented correctly, then the *self-consistency condition* holds:

$$\pi(\theta) = \int\int \pi(\theta \mid \tilde{y})\pi(\tilde{y} \mid \tilde{\theta})\pi(\tilde{\theta})\, d\tilde{y} d\tilde{\theta}, \tag{6.18}$$

where $\tilde{\theta} \sim \pi(\theta)$ are the parameters - referred to as the *ground truth* - sampled from the prior distribution,[7] $\tilde{y} \sim \pi(y \mid \tilde{\theta})$ are the data generated from the model using the ground truth, and $\theta \sim \pi(\theta \mid \tilde{y})$ the posterior samples.

Thus, if sets of parameters $\tilde{\theta} \sim \pi(\theta)$ are repeatedly sampled from the priors, datasets $\tilde{y}$ generated from them, and samples θ drawn from the posterior distribution given these data, then these samples should follow the same distribution as the samples drawn directly from the prior. This can be tested by computing the *rank statistic* r of the prior sample relative to the posterior sample, defined for any one-dimensional function f mapping parameters on the real numbers as

$$r\big(f(\theta_1), \ldots, f(\theta_L), f(\tilde{\theta})\big) \coloneqq \sum_{l=1}^{L} \mathbb{I}\left[f(\theta_l) < f(\tilde{\theta})\right] \in [0, L], \tag{6.19}$$

where L is the number of samples of the posterior distribution, and $\mathbb{I}$ is the indicator function taking the value 1 if the condition in the parentheses holds and the value 0 otherwise. If self-consistency holds, the rank statistic should be uniformly distributed on the set of numbers from 0 to L.

The simulation study was designed to test this condition. As the MCMC-samples in Stan are autocorrelated, we use a subset of the samples to compute the *rank statistic* for each model parameter and thin the posterior samples according to Algorithm 2 in Talts et al. (2018) to $L = 399$ high-quality samples. We set the number of bins in the histogram to 100, such that there are 20 observations expected per bin, across the 2000 simulated datasets. We computed the rank statistic for each model parameter using Equation (6.19). Following recommendations by Modrak et al. (2022), we also compute the rank statistic for the model's log-likelihood. The resulting distributions of the rank statistic can be depicted by means of histograms to assess

[7] For simulation-based calibration, it is important that the parameter distribution underlying the data generation is the same as the prior distribution in the Bayesian model.

deviations from the uniform distribution. We add a gray band to the histograms that covers 99% of the variation expected for each frequency in a histogram of a uniform distribution, where the 99% expected range of the uniform distribution is determined using the quantile function of the binomial distribution, as the frequency of each bin of the histogram is binomially distributed.

We also calculated the χ^2-statistics for the differences between expected and observed frequencies of observations per bin for each parameter with expected frequencies given by the expected uniform distribution (i.e., 20 per bin). For each parameter, the observed χ^2 value is compared to the critical χ^2 value of 123.23, for $\alpha = .05$ with $df = 99$ (number of bins minus 1).

Results and Discussion

We present results from the SBCs for 100 and 500 trials for the truncated and the censored analyses, respectively, via histograms of the rank statistics (see Figures 6.7 to 6.10). Visual inspection yields that none of the histograms shows systematic variation from the uniform distribution. This means that there is no clear pattern in the histograms that would indicate a bias in the implemented algorithm as described by Modrak et al. (2022) and Talts et al. (2018). Furthermore, all χ^2-statistics testing for uniformity are non-significant at the 5% level for both analyses.

To sum up, we conclude that there is little indication in these analyses suggesting that the new implementation might be implemented incorrectly.

6.7 Application with First-Person Shooter Task Data

As mentioned in the beginning, reaction time experiments with response windows are typical experiments in which truncated and censored data are produced. Ulrich and Miller (1994) advise to include truncation or censoring in the model if data are truncated or censored, respectively. For example, the effects of truncation can alter mean and median reaction times by 10% or more, independent of the exact distribution, and are therefore as large as those of many common experimental manipulations (Ulrich & Miller, 1994)[8]. In the case of diffusion modeling, if right-censoring or truncation is not accounted for, response times appear faster than they truly are, which will in turn impact the parameter estimates; for example, by increasing the absolute magnitude of the estimated drift rates (Pleskac et al., 2018).

To demonstrate functionality of our new implementation, we reanalyze real data from an experiment operationalizing the First-Person Shooter Task (FPST, Correll et al., 2002). We chose to reanalyze data of Study 1 and Study 2 by Pleskac et al. (2018). These datasets suit our purposes due to the following reasons: (a) data and information about the model are freely available

[8] Ulrich and Miller (1994) examined distributions like the lognormal, Erlangian, and Ex-Gaussian distribution.

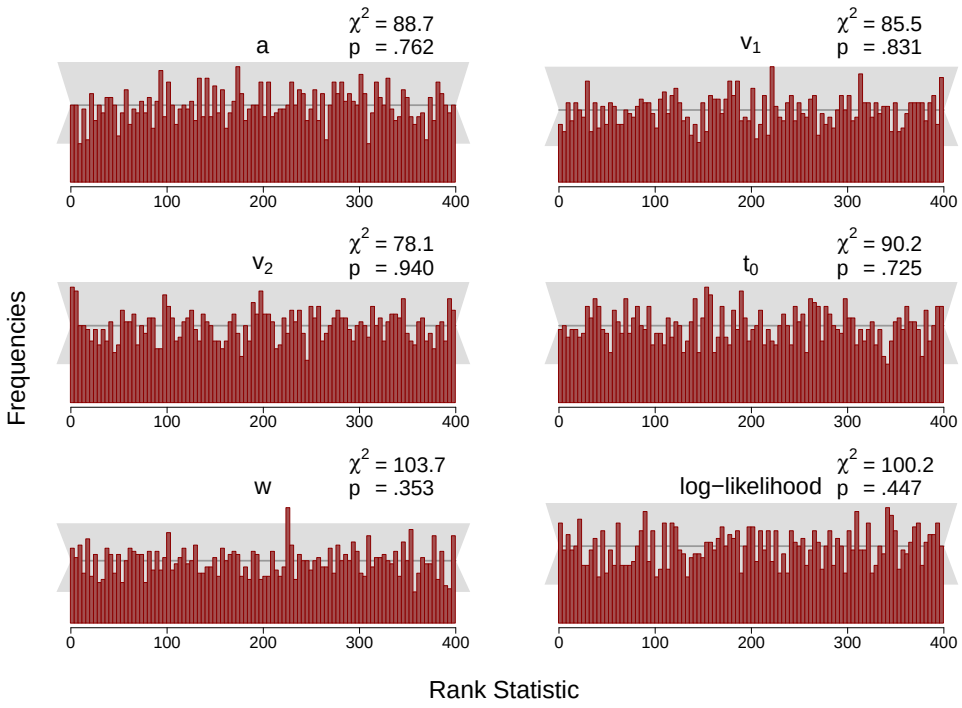


Figure 6.7: Histograms of the Rank Statistic for 100 Trials for the Truncated Model.
Note. The histograms indicate no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar).

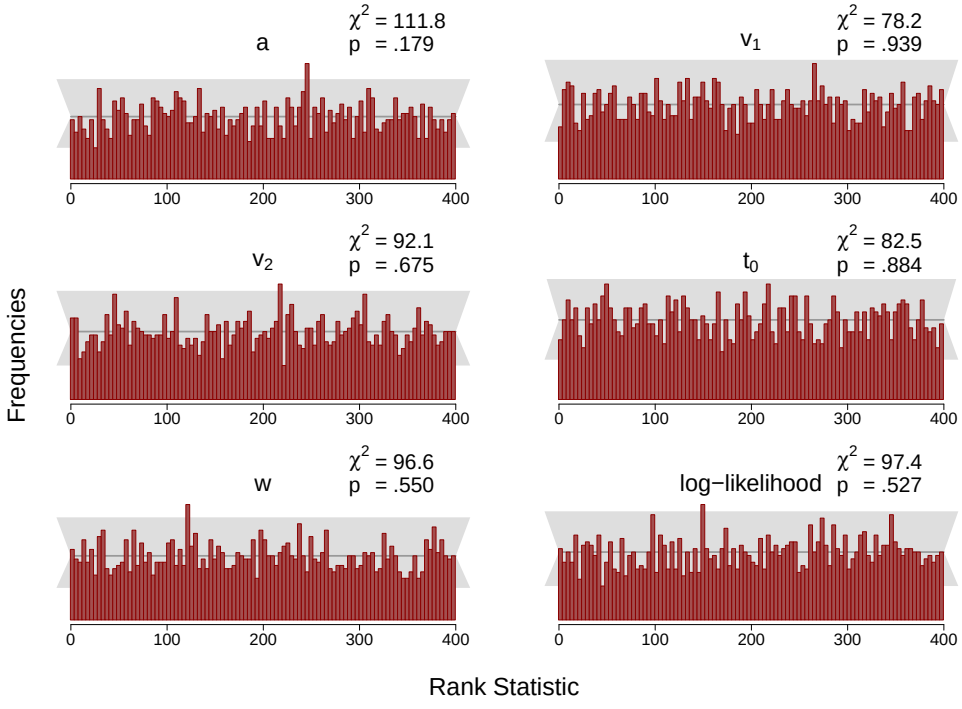


Figure 6.8: Histograms of the Rank Statistic for 500 Trials for the Truncated Model.
Note. The histograms indicate no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar).

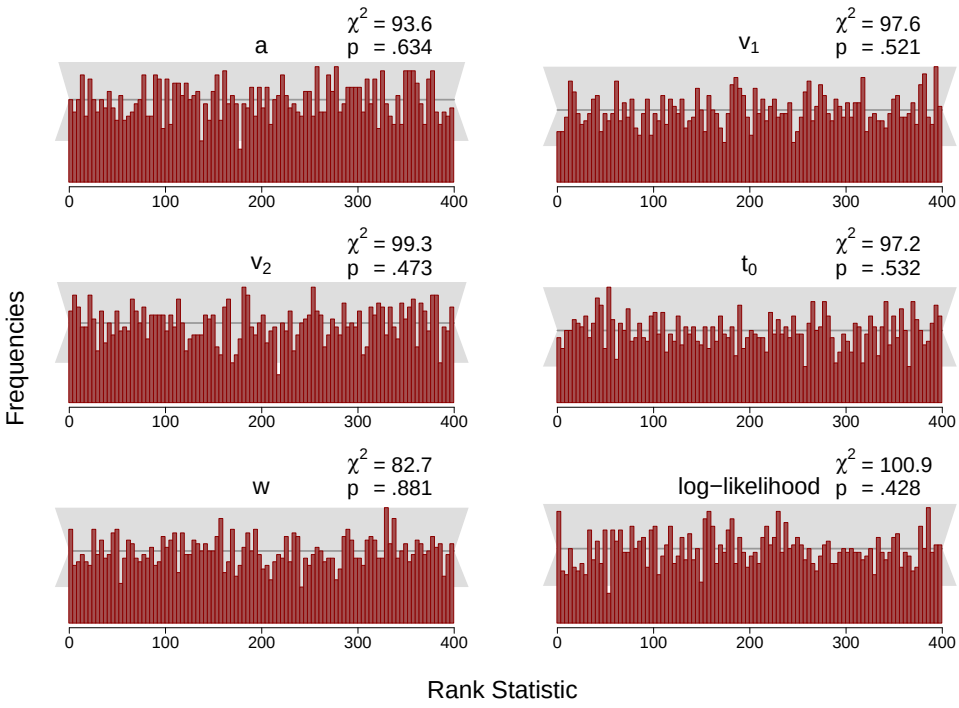


Figure 6.9: Histograms of the Rank Statistic for 100 Trials for the Censored Model.
Note. The histograms indicate no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar).

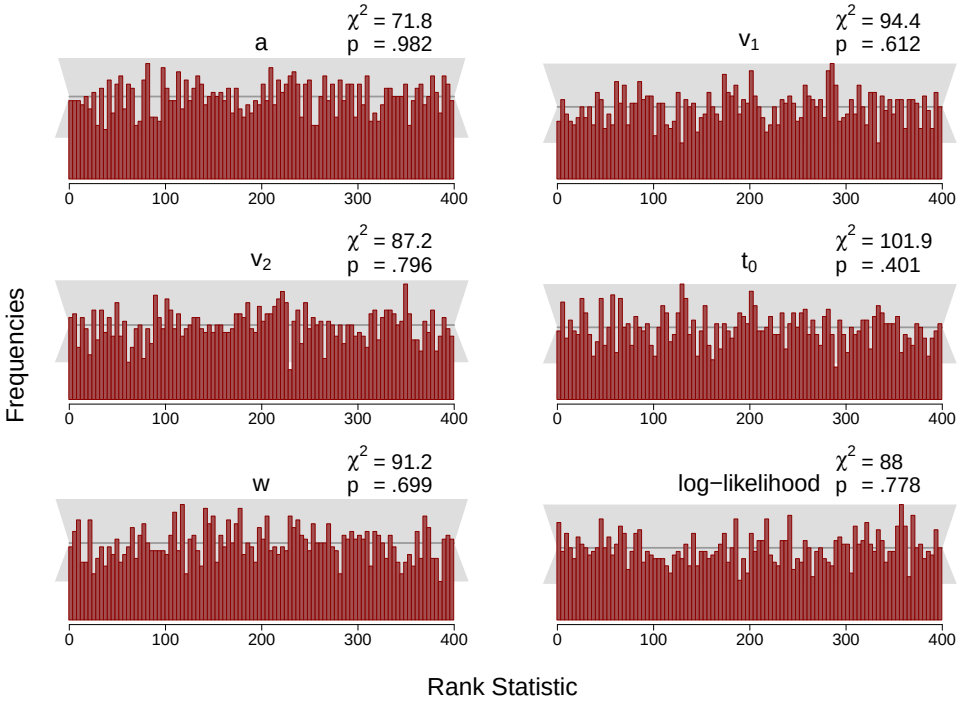


Figure 6.10: Histograms of the Rank Statistic for 500 Trials for the Censored Model.
Note. The histograms indicate no issues as the empirical rank statistics (red vertical bars) are consistent with the variation expected of a uniform histogram (gray horizontal bar).

online, (b) the design includes a response window that censors data to a right rt-bound, and (c) the authors perform a diffusion model analysis (hierarchical, basic four-parameter model).

6.7.1 The First-Person Shooter Task

As already mentioned in the Introduction, the FPST is used to study racial bias in shoot/don't-shoot decisions. Participants see pictures showing a person (Black target or White target) and an object (gun or tool). They are instructed to press the *shoot* key if the target is armed and the *not shoot* key if the target is unarmed. Typical findings are that participants are faster and more accurate to correctly decide "shoot" for Black targets than for White targets and slower and less accurate to correctly decide not to shoot in the case of unarmed Black targets than unarmed White targets (e.g., Amodio et al., 2004; Correll et al., 2002, 2007; Greenwald et al., 2003; Johnson et al., 2017; Payne et al., 2002).

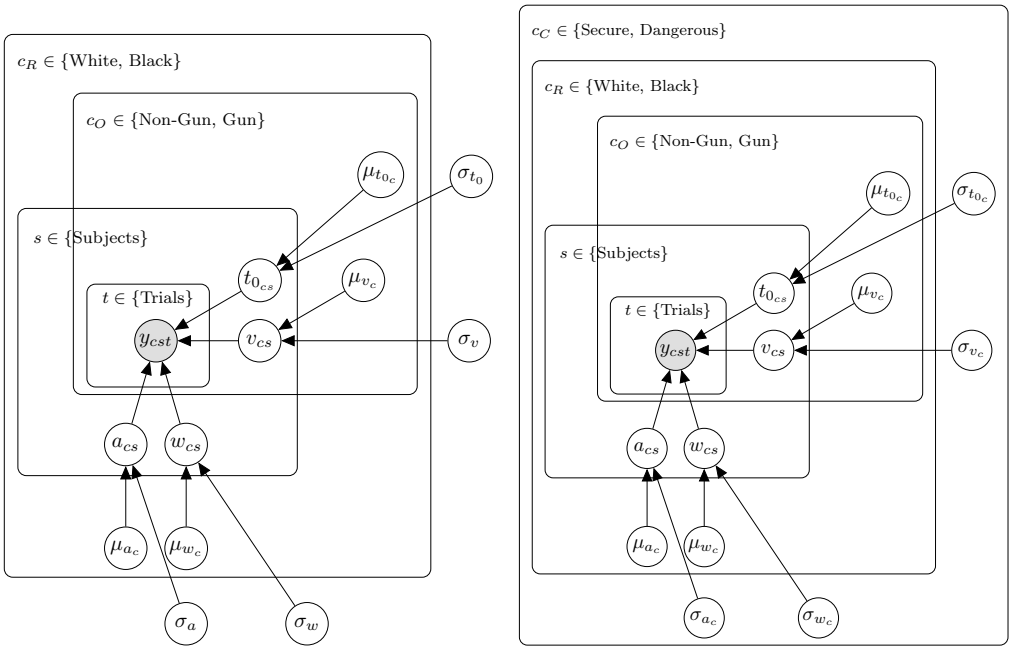
6.7.2 Study 1 by Pleskac et al. (2018)

Pleskac et al. (2018) investigate the influence of skin color on the decision to shoot using the FPST with different response deadlines and manipulations. In Study 1, targets were shown in a neutral context and a relatively liberal response deadline of 850 ms was used. In only 3% of the trials was the response deadline exceeded so that Study 1 exemplifies a situation in which we would ideally see little effect of whether the model includes censoring or truncation or neither.

The authors use a hierarchical censored basic model to analyze data. The *relative starting point*, w, and the *boundary separation*, a, are allowed to vary across race, but stay constant for the object (gun or tool), so that there are two group-level w parameters and two group-level a parameters, one per race (Black vs. White). *Drift rate* and *non-decision time* are also allowed to vary as a function of object so that there are four group-level parameters for each of drift rate and non-decision time. A graphical model representation is displayed in Figure 6.11(a).

Censored Data in This Study

Data in Study 1 were censored. That is, neither observed response nor response time was recorded for trials in which the response was made outside the response window (i.e., did not occur within 850 ms after stimulus onset). The authors built censoring into the model (using the method described by Kruschke, 2015, Chap. 25.4), see Footnote 1. This approach requires the information to which response boundary a missing rt-value belongs. To impute the missing response value, the authors used a heuristic way: They imputed missing responses so as to match the observed relative frequency of these responses for gun and non-gun objects for each subject, collapsing across the conditions (Pleskac et al., 2018, Supplementary material).



(a) Study 1 (b) Study 2

Figure 6.11: Graphical Model Representation of the Models Used in Pleskac et al. (2018). *Note.* The *Context* condition, c_C, is a between-subject manipulation, the *Race*, c_R, and *Object*, c_O, conditions are within-subject manipulations. Subindex c refers to the combination of conditions belonging to the plates in which the subindex is located. y denotes the data comprising reaction time and response.

Methods

As Pleskac et al. (2018) provide all data and models online[9], we first reran the JAGS analysis in the same way the authors did. In a second step, we ran several Stan models. For all models, we choose the same priors as these authors and define four different models: (a) a basic hierarchical model without truncation or censoring, called *basic*, (b) a censored basic hierarchical model, using the responses that were heuristically inferred by the authors, called *censored*, (c) a censored basic hierarchical model, using a principled approach based on the complementary cumulative distribution function (CCDF) to deal with the missing responses as per Eq. (6.17), called *censored with CCDF*, and (d) a truncated basic hierarchical model, called *truncated*.

The basic model is the baseline model without any accommodation for censoring or truncation. Trials outside the response window are simply omitted from the analyses. The data are thereby analyzed as if there had been no such trials – an analysis that is inconsistent with the implemented response deadline.

For the censored model, we replace missing responses by the responses that were inferred by Pleskac et al. (2018). We expect the results for this model to coincide with the JAGS analysis.

---

[9] Retrieved September 23, 2025 from https://osf.io/9qku5/

Heuristically inferring the missing responses, as done by Pleskac et al. (2018) relies on strong assumptions about the missing values, namely that correct and error responses would have occurred in the same proportions above the right rt-bound as they did occur inside the response window. As this assumption need not hold in data that stem from a diffusion process, the censored model with CCDF explores an alternative, principled way to deal with missing responses that does not require additional assumptions about the distribution of the missing response values. Instead, we compute the probability of ending at the response-1 boundary or response-0 boundary (and thus, of a correct response or an error response) after the response deadline has passed (Eq. (6.17)) and multiply the data likelihood by this value for each response outside the response window. This approach correctly encodes the information implied by the event that a response does not occur prior to the deadline in the model's likelihood function and may be more appropriate when no information on the distribution of the missing response values is available (see above for an implementation of such a model). Because there was little reason to believe that the heuristic assumption required in Pleskac et al.'s (2018) approach would be grossly violated in the present case, we expect this more principled model to perform similarly as to the censored model and the JAGS analysis.

Finally, for the truncated model, unlike for the censored model, the number of trials that fall outside the response window remains unknown to the model. Like for the censored model with CCDF, the analysis based on this model is consistent with the implemented response window. It is, however, not informed by the information on the observed number of trials outside the response window and may therefore estimate parameters with somewhat greater uncertainty (expressed in, for example, larger highest density intervals), but should otherwise yield similar parameter estimates as the censored model with CCDF.

Results

Figure 6.12 shows the results of our reanalysis. All four parameters in each condition are displayed. The four Stan models are displayed as a black circle - basic model, red triangle - censored model, blue plus - censored model with CCDF, and violet diamond - truncated model. The results displayed as green cross belong to the JAGS analysis.

We make some observations:

1. The results for the censored model closely match those for the JAGS model. This means that JAGS and Stan behave similarly when applied with the same data and model.

2. The censored model with CCDF deviates little from the censored model and the JAGS model. This suggests that the heuristic assumption built into the censored model (error rates are the same for responses outside the response window as within the response window) is not grossly violated for the present data and model.

3. The basic model shows minor deviations from the censored models in the relative starting

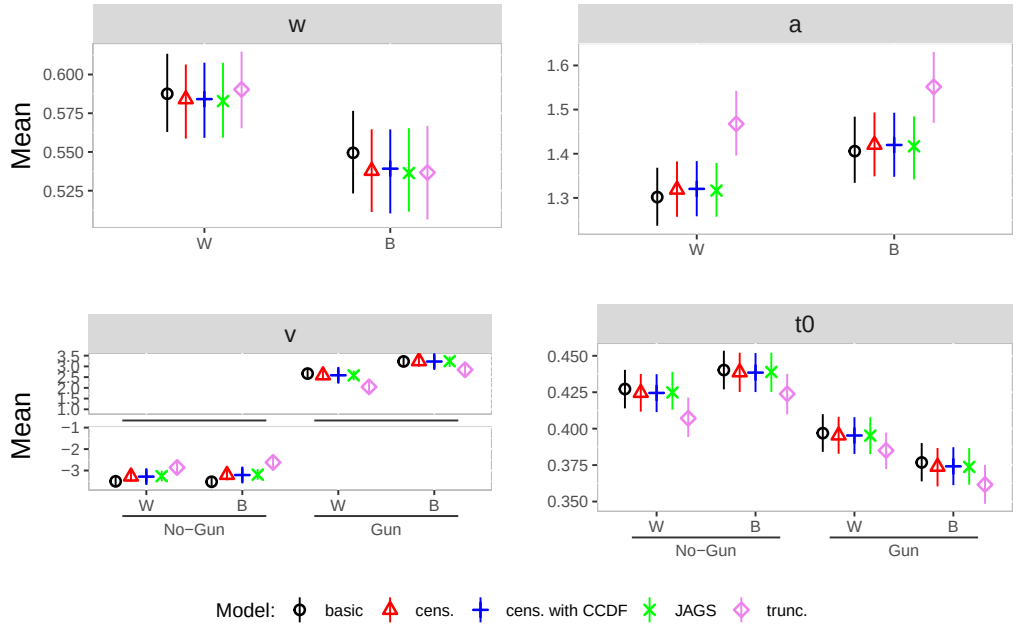


Figure 6.12: Parameter Estimates Reanalysis Study 1. *Note.* Posterior means (dots) and 95% HDI (bars) for the group-level parameter estimates of the diffusion model in each condition for the reanalysis of Study 1; W =White, B = Black, cens. = censored, trunc. = truncated.

point and the boundary separation. As only 3% of the data are censored, it is reasonable to expect similar results from the basic and the censored models.

4. Surprisingly, results for the truncated model deviate more substantially from the other models' results in nearly all parameters. With this model, the boundary separation is estimated to be larger in both conditions, whereas the drift rate and non-decision time are estimated to be smaller in all conditions. Below, we discuss additional analyses aimed at understanding the cause of this unexpected pattern of results.

6.7.3 Study 2 by Pleskac et al. (2018)

In Study 2, the target persons are shown in either a neutral or a dangerous context in a between-subjects manipulation of context. The study design thus comprises two within-subject manipulations (race: White/Black, and object: No-Gun/Gun), and one between-subject manipulation (context: neutral/dangerous). Like in Study 1, there was a response deadline, which was set to 630 ms in this study, leading to censoring for 10% of the data.

The authors again use a hierarchical censored basic model to analyze data. In this model, the *relative starting point*, w, and the *boundary separation*, a, are allowed to vary as a function of race and context, but stay constant for the object. *Drift rate* and *non-decision time* were additionally allowed to vary as a function of object. A graphical model representation is displayed in Figure 6.11(b).

Methods

As in Study 1, we first reanalyze data using JAGS with the same model and data as provided by the authors. Next, we analyze data with the four Stan models described above.

Results

Figure 6.13 shows the results of our analysis. We make some observations:

1. The results for the censored model again closely match those for the JAGS model.

2. The censored model with CCDF deviates little from the censored model and the JAGS model. This again suggests that the heuristic assumption built into the censored model (error rates are the same for responses outside the response window as within the response window) is not grossly violated for the present data and model.

3. The basic model shows some deviations from the censored models in all parameters. The estimates for boundary separation are generally smaller than estimates in the censored models, and absolute values of drift rates, one relative starting point and some non-decision times are estimated a little bit larger than in the censored models. That deviations from the basic model are somewhat more pronounced than in Study 1 was to be expected given the higher rate of censoring (10% vs. 3%).

4. Parameter estimates based on the truncated model again show unexpectedly substantial deviations from the results obtained with the other models in all parameters. With this model, the relative starting point and boundary separation are estimated to be larger in all conditions, whereas the drift rate and non-decision time are estimated to be smaller in all conditions. Next, we turn to analyses shedding some light on this unexpected pattern of results.

6.7.4 What is the Cause of the Discrepancies Between the Truncated and the Censored Models?

We did not expect substantial discrepancies between the truncated and the censored model with CCDF. Both models take the use of a response window into account, the major difference being that the censored model makes use of the information of how many trials had no response within the response window, whereas the truncated model ignores this information. We therefore expected estimates from the truncated model to be associated with some more uncertainty than those of the censored model, but little systematic deviation between the analyses based on the two models. This is in fact the pattern of results exemplified by the simulation study.

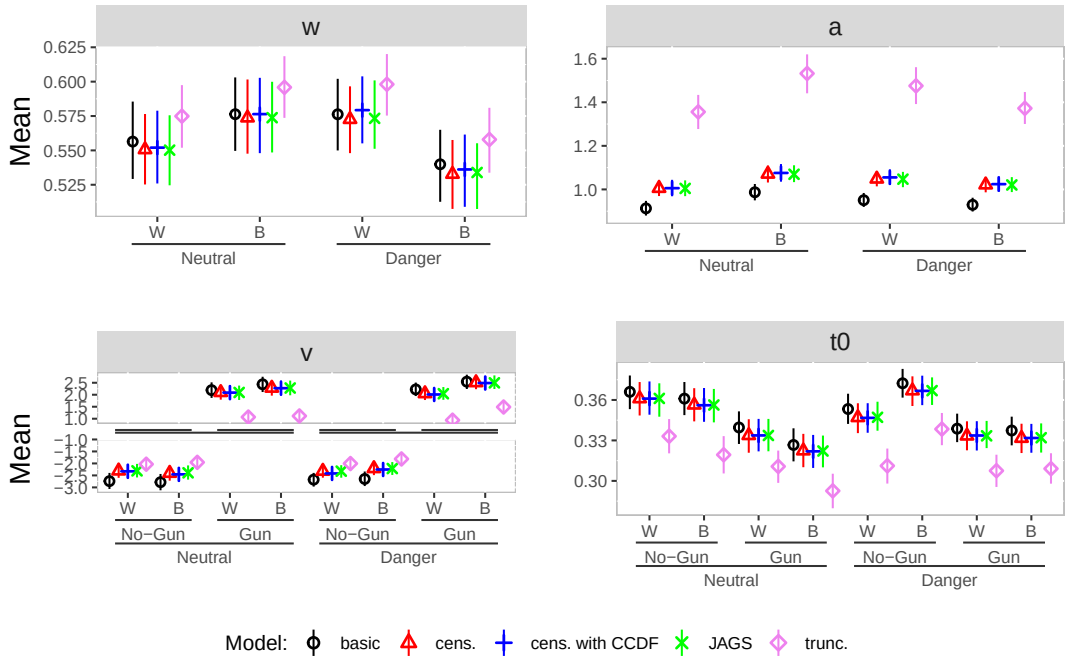


Figure 6.13: Parameter Estimates Reanalysis Study 2. *Note.* Posterior means (dots) and 95% HDI (bars) for the group-level parameter estimates of the diffusion model in each condition; W = White, B = Black, cens. = censored, trunc. = truncated.

Clearly, this expectation was not borne out, as evidenced by the substantial deviations between the analyses based on the truncated and the censored model, which in turn matched the analyses based on the basic model relatively closely in comparison. This implies that the assumptions from which our expectation was derived are wrong – our major assumption was that a diffusion process generated the analyzed data. But how exactly does this lead to the observed discrepancies?

We reasoned that the data must violate the diffusion model assumptions in such a way that the violations capitalize on the difference between the basic and the censored model, on the one hand, and the truncated model, on the other hand, to bring out the observed pattern of discrepancies in the modeling results.

Analyses by the basic and the censored model are constrained by the proportion of censored trials (i.e., trials for which responses did not occur in the response window). For the censored model, the observed proportion of such trials is directly encoded in the data likelihood; for the basic model, such trials are nonexistent, and the data likelihood encodes the (wrong) information that such trials did not occur; their proportion is implicitly assumed to be zero.

In contrast, for the truncated model, the proportion of trials without response in the response window is an unknown. The model acknowledges the existence of a response window, but remains agnostic about the percentage of trials without responses within the response window. These trials do not inform the data likelihood of the truncated model in any way (other than by allowing for their existence).

Model	Conditions			
	W/NG	B/NG	W/G	B/G
basic	172	212	5	1
cens.	221	287	6	1
cens. CCDF	221	286	6	1
trunc.	421	555	19	3
data	52	72	29	15

Table 6.3: Observed and Predicted Numbers of Trials Without Response in Study 1.
Note. W = White; B = Black; G = Gun; NG = No-Gun; cens. = censored; trunc. = truncated.

In fitting the data, the truncated model is therefore constrained only by the reaction time distribution and responses observed within the response window. The basic and censored model must also attempt to fit the proportion of censored trials (implicitly set to zero in the basic model and equaling the observed proportion of such trials in the censored model).

Thus, discrepancies between the basic and censored model, on the one hand, and the truncated model, on the other hand, might reflect that the proportion of censored trials may be incompatible with the distribution of reaction times and responses within the window. Therefore, the truncated distribution might not be well described by diffusion model parameters, which describe the proportion of censored trials well and vice versa. Specifically, if the distributions of reaction times and responses within the window, when considered in isolation as done by the analysis via the truncated model, are best fit by diffusion model parameters which overall predict larger proportions of censored trials than the observed proportion and the zero proportion implied by the basic model, we expect the outcome of the analysis by the truncated model to differ substantially from the outcome of analyses which also try to fit these proportions.

If this analysis of our results pattern holds true, we should see (a) higher rates of predicted proportions of censored trials for the truncated model analysis than for the other model analyses along with (b) a better account of the distribution of reaction times within the window by the truncated model than by the other models.

Observed and Predicted Frequencies of Trials Without Response

Tables 6.3 and 6.4 show observed and predicted frequencies of trials without response. For Study 1, all models overestimate the number of these trials for the No-Gun conditions and underestimate their number for the Gun conditions. For Study 2, this pattern is similar for the basic and the censored models. The truncated model overestimates the number of such trials in all conditions. Furthermore, in both studies the truncated model predicts many more such trials in all conditions than the other models.

Model	Conditions							
	Neutral				Dangerous			
	W/NG	B/NG	W/G	B/G	W/NG	B/NG	W/G	B/G
basic	275	352	30	20	304	315	29	19
cens.	440	517	38	26	468	483	40	21
cens. CCDF	438	515	39	26	463	483	42	22
trunc.	931	1177	178	150	1054	1000	206	96
data	179	170	66	63	163	176	92	62

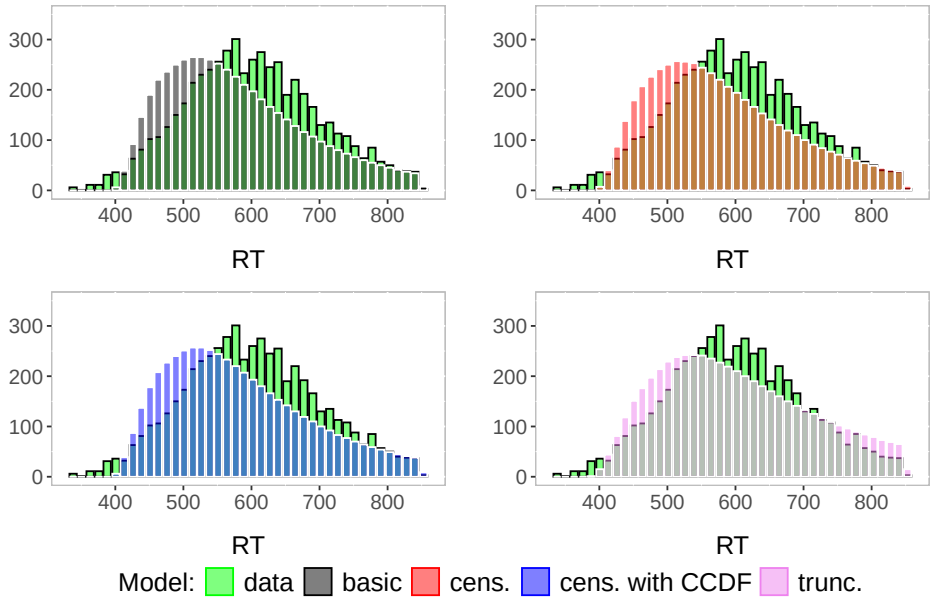
Table 6.4: Observed and Predicted Numbers of Trials Without Response in Study 2. *Note.* W = White; B = Black; G = Gun; NG = No-Gun; cens. = censored; trunc. = truncated.

Observed and Predicted RT Distributions Within the Response Window

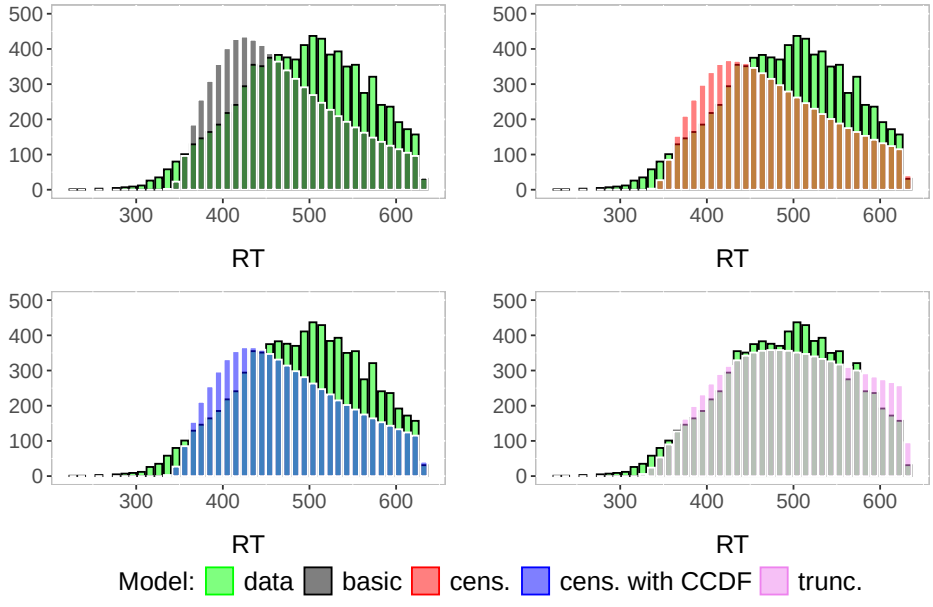
Next, we computed predicted reaction times by simulating data from the respective diffusion models using the estimated parameters and again the function `sampWiener()`. Specifically, we first simulated a large number of datasets based on the samples from the posterior distribution and counted the bin frequencies for each dataset. Then we computed the mean frequency for each bin. The resulting histograms with the mean bin frequencies of the generated datasets and the histogram of the observed data are shown in Figure 6.14 for both studies. We observe that the histograms for the basic and the censored models are slightly shifted to the left compared to the observed data. Furthermore, the histogram belonging to the truncated model matches the data histogram best in both studies.

Similar observations can be made in the quantile-quantile plots in Figure 6.15. Points align closely with the diagonal line across all quantiles for the truncated model, showing that explicitly modeling truncation improves fit. For the censored models, points also align closely with the diagonal line in the middle quantiles and deviate slightly from the diagonal line at higher quantiles. For the basic model, there is a substantial deviation from the diagonal line at higher quantiles.

To sum up these analyses, we found support for the above analysis of the causes of the discrepancies between the analyses by the truncated model, on the one hand, and the basic and censored model, on the other hand. It seems to be the case that describing the data from trials with response in the response window by the diffusion model requires parameter values that strongly overestimate the overall proportion of censored trials. The truncated model is not informed by the proportion of censored trials and thereby acquires the flexibility to account for the reaction time distribution within the response window better than the other models at the expense of predicting overall much higher proportions of censored trials than the other models. The other models are constrained by the proportion of censored trials (implicitly set to zero in the basic model and given by the observed proportion in the censored models) and predict much lower proportions of such trials at the expense of less convincing fits of reaction time distributions within the response window.

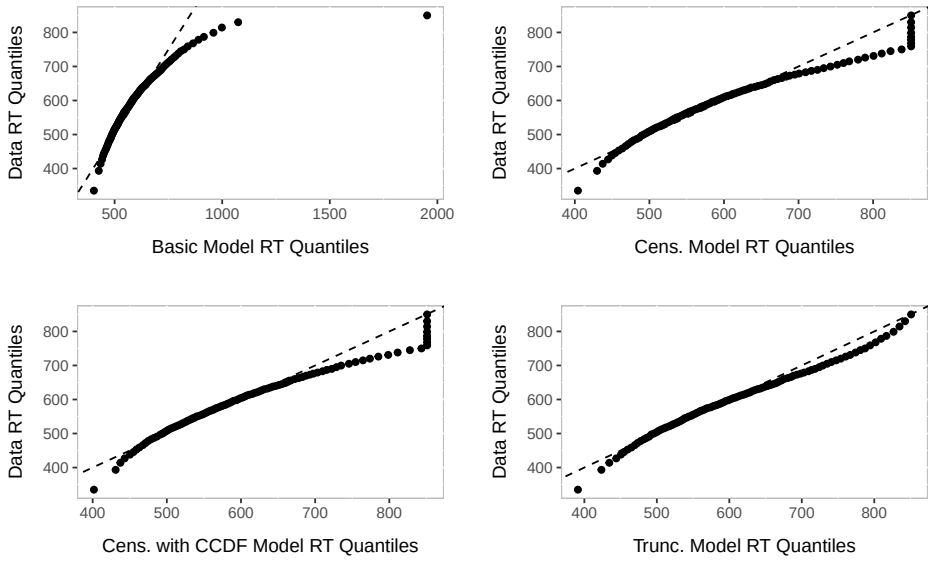


(a) Study 1

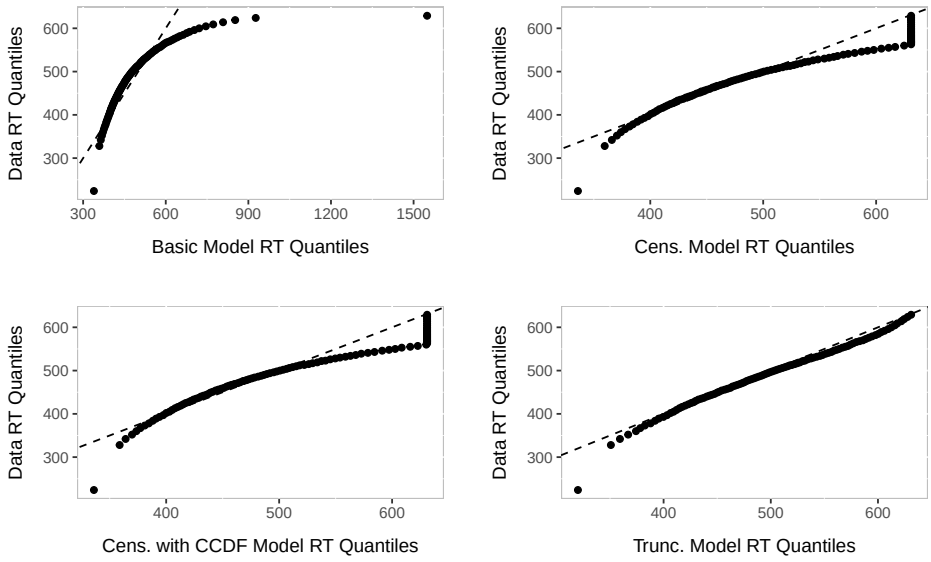


(b) Study 2

Figure 6.14: Histograms of Predicted Reaction Times vs. Observed Reaction Times.
Note. The green histograms are based on the data observed by Pleskac et al. (2018) in Study 1 (a), and Study 2 (b); cens. = censored; trunc. = truncated.



(a) Study 1



(b) Study 2

Figure 6.15: Quantile-Quantile Plots with Predicted vs. Observed Data Quantiles

Model	Study 1	Study 2
basic	-8138	-12794
trunc.	-9145	-16157
cens.	-7252	-8517
cens. CCDF	-7310	-9308

Table 6.5: Goodness-of-fit Measure: WAIC. *Note.* trunc. = truncated; cens. = censored. Note that values can be compared only between the basic and the truncated model and between the two censored models (see text).

Model Selection Index: WAIC

As a model-selection index, we compute the Widely Applicable Information Criterion, also known as the Watanabe-Akaike Information Criterion (WAIC, Gelman et al., 2014; Watanabe, 2010). The WAIC is an extension of the Akaike Information Criterion (AIC) and is more appropriate for Bayesian analyses than the AIC. The WAIC estimates the effective number of parameters to adjust for overfitting. Models with smaller WAIC values are to be preferred. Table 6.5 shows the WAIC values for all models and both studies.

Note that WAIC values can only be meaningfully compared for models fitted to the same data. For the reanalyzed studies, the basic and the truncated model analyze the same datasets without the censored trials, and the two censored models analyze the same datasets that differs from the one used for the basic and truncated model in that it includes information on the censored trials. In comparing WAIC values for the basic and truncated model, the truncated model has a smaller WAIC in both studies, and in comparing WAIC values for the two censored models, the censored model based on the CCDF has the smaller WAIC in both studies.

These observations support our previous findings that the truncated model describes the analyzed data better than the basic model. Furthermore, the censored model based on the CCDF performs better on the data than the censored model based on the heuristic approach, which supports our claim from the beginning for the need for a more sophisticated modeling approach for censored data.

6.7.5 Discussion

To illustrate the behavior of the new functionality in Stan - modeling truncated and censored data with the diffusion model - we reanalyzed data of Study 1 and Study 2 by Pleskac et al. (2018). For this purpose, we analyzed data with the JAGS model specified by Pleskac et al. (2018) and with four Stan models.

In the first study, only 3% of the trials had no response within the response window. In consequence, we did not expect and did not observe pronounced differences between the analysis by the basic diffusion model without accommodation for the use of a response window and the censored models. In the second study, a stricter response deadline led to 10% of trials without response in the response window. Here, differences between the basic and the censored models were somewhat larger. For example, parameter estimates for boundary separation were smaller without censoring in the model, the non-decision time was larger, and the drift rate was larger in absolute size.

We expected similar patterns of parameter estimates for the analysis by the truncated model as for the censored model. To our surprise, the largest differences were found between the truncated model, on the one hand, and all other models, on the other hand.

We believe we have provided a plausible account of these discrepancies and have corroborated our account with additional analyses. Ultimately, the discrepancy between the analyses using truncated and censored models, along with our additional analyses, shows that the present data are quite pronouncedly incompatible with the assumption that an underlying Wiener diffusion process generates them. Note also that depending upon the pattern of violations of this assumption, we might in fact have seen other patterns of unexpected and surprising results. The takeaway recommendation here might be to routinely implement and report diagnostic model checks in addition to reporting the fitted parameter values to safeguard against surprises stemming from model misfit.

As suggested by an anonymous reviewer, violation of the diffusion model assumptions might stem from a proportion of trials in which responses and reaction times reflect a guessing process or are the result of mind wandering. Alternatively, the response-window procedure itself, which provides feedback if participants' responses fall outside the window, may have led participants to adapt the data-generating process online, which could also explain the violation of the diffusion model assumptions. Similar remarks pertain to the possibility of outliers in the data. These possibilities need further investigation.

6.8 General Discussion

The purpose of this work was to add the functionality to model censored and truncated data in diffusion model analyses in Stan. This involves implementing the cumulative distribution function of reaction time distributions arising from the diffusion model and its complement.

As mentioned in the introduction, truncated or censored data arise in paradigms that use temporal response windows outside of which responding is not possible or for which response times and/or the response falling outside the window are not recorded, as well as a consequence of post-hoc outlier analyses. For censored data, a count of these trials is kept; for truncated data,

not even a count is available.[10] As Ulrich and Miller (1994) elaborated, it is important to build the model used to analyze such data so that it accounts for censoring or truncation if data are censored or truncated. Otherwise, important characteristics of the response-time distributions such as mean, median, standard deviation, or skewness will be estimated incorrectly, biasing model-based analyses. In the case of the diffusion model, the drift rate, for example, will be overestimated when the model does not account for censoring or truncation by an upper response deadline (Pleskac et al., 2018). In order to account for truncation and censoring, we extended the diffusion model implementation in Stan, `wiener()`, tested the implementation with two consistency checks (recovery and simulation-based calibration) and reanalyzed existing datasets.

We conducted a simulation study assessing recovery from truncated and censored datasets. The results of the recovery studies are satisfactory in terms of correlations, coverage, and bias. Results for the simulation-based calibration studies do not show systematic errors, providing a more stringent test of the correctness of the current implementation than is possible via recovery studies.

We illustrated the new method by reanalyzing data from Studies 1 and 2 in Pleskac et al. (2018). Both of these studies employed response deadlines beyond which no response or response time was recorded. The reanalysis demonstrated that it can make a major difference whether data are analyzed without provision for the response deadline, using a model for censored data, or a model for truncated data (ignoring the information on the number of trials without response before the deadline). As expected, differences between a naive analysis without provision for the response deadline and the censored analysis were small in Study 1, in which most responses occurred prior to the response deadline, and somewhat larger in Study 2 in which the deadline was stricter and more trials occurred without response before the response deadline. These differences were, however, still small in comparison to the discrepancies in parameter estimates obtained from fitting the model for truncated data. Additional analyses suggest that this somewhat surprising result ultimately reflects deviations of the data from the diffusion model.

Our studies are limited by the fact that they were performed for a basic diffusion model with four parameters instead of for the full diffusion model with seven parameters. This limitation reflects the considerable increase in computing time required for fitting the seven-parameter model (Henrich et al., 2023) in a hierarchical design. This renders a large simulation study based on the seven-parameter model unrealistic in terms of required computing times. Future analyses could test the diffusion model on data that are generated from a mixture of different

---

[10] Truncation is less frequent in psychological reaction time experiments than censoring. Consider as an example a consumer-psychology study in which the stimuli are different prices presented for an item on sale on the Internet. Customers can decide whether they want to buy one item or two items at a discount. The response times to be modeled are the times to the first purchase. If prices change with a fixed interstimulus interval, we will only be able to register purchases that occur within the interval. If no purchase is registered within the interval, we do not know whether none occurred or whether a purchase occurred, but came too late. In this case, the data on purchase decisions are effectively truncated by the duration of the interstimulus interval.

models reflecting guessing or mind wandering to see how robust Bayesian diffusion modeling is against such outliers (see Ratcliff & Tuerlinckx, 2002).

In conclusion, the new features of `wiener()` produce reliable and competitive results for the basic model and enrich the landscape of diffusion modeling approaches. Using previously published datasets, we demonstrated the functionality of the new implementation and provided hands-on instructions on how to implement a censored or truncated model. We hope that these tools will prove useful for researchers wishing to analyze truncated or censored data with diffusion models.

6.9 Appendix

6.9.1 Function Call Truncated Model

To use the functionality of parallelizing the estimation process over several cores, the `target+=` notation has to be used in the **model block** of the Stan file. The two variables `left_bound` and `right_bound` are handed over in the **data block** or set directly in the .stan-file. Furthermore, the function that parallelizes the model calls, `partial_sum_fullddm()`, has to be defined in a **functions block**. Note that in the input data for a Stan analysis, no NA values are allowed. Make sure to delete all trials with missing reaction time values.

The following code corresponds to Eq. (6.13) on the log-scale:

```
// Truncated model with parallelization, both rt-bounds
// all rt of the input data are within the response window
functions { // function to parallelize each chain
 real partial_sum_fullddm(array[] real rt_slice, int start,
 int end, real a, real t0, real w, real v, real sv, real sw,
 real st, array[] int resp, real left_bound, real right_bound){
  real ans = 0;
  for (i in start:end) {
   if (resp[i] == 1) { // upper response boundary
    ans += wiener_lpdf(rt_slice[i+1-start] |
           a, t0, w, v, sv, sw, st);
   } else { // lower response boundary (mirror v and w)
    ans += wiener_lpdf(rt_slice[i+1-start] |
                       a, t0, 1 - w, -v, sv, sw, st);
   }
   ans += -log_diff_exp(
    log_sum_exp(
     wiener_lcdf(right_bound | a, t0, w, v, sv, sw, st),
     wiener_lcdf(right_bound | a, t0, 1 - w, -v, sv, sw, st)),
    log_sum_exp(
     wiener_lcdf(left_bound | a, t0, w, v, sv, sw, st),
     wiener_lcdf(left_bound | a, t0, 1 - w, -v, sv, sw, st))
    ) // parenthesis log_diff_exp
  } // end for
  return ans; } } // end for and end functions

// ... // data-, parameter-block

model { // ... // definition of priors for all model parameters
 target += reduce_sum(partial_sum_fullddm, rt, 1,
  a, t0, w, v, sv, sw, st, resp, left_bound, right_bound);
}
```

For data that are only left-truncated, change the selected lines in the above code as follows. This corresponds to Eq. (6.14) on the log-scale:

```
// lines 16 to 25
ans += -log1m_exp(log_sum_exp(
 wiener_lcdf(left_bound | a, t0, w, v, sv, sw, st),
 wiener_lcdf(left_bound | a, t0, 1 - w, -v, sv, sw, st)));
```

For data that are only right-truncated, change the selected lines in the above code as follows. This corresponds to Eq. (6.15) on the log-scale:

```
// lines 16 to 25
ans += -log_sum_exp(
 wiener_lcdf(right_bound | a, t0, w, v, sv, sw, st),
 wiener_lcdf(right_bound | a, t0, 1 - w, -v, sv, sw, st);
```

For a general introduction to Stan, see Stan Development Team (2023a).

6.9.2 Additional Reanalysis of the Simulation Study Data

Following a suggestion by a reviewer, we additionally analyzed the censored datasets from the simulation study with the seven-parameter model with and without censoring and with the four-parameter model without censoring to see whether the seven-parameter model would capture censored data and underlying parameters better.

Due to the high computational demand imposed by fitting the seven-parameter models, we only reanalyzed 1000 datasets in the 100 trials condition for these models each. Therefore, the plots are thinned compared to the four-parameter model plots which are based on 2000 datasets.

The results for correlations and coverage in 50% and 95% HDI are shown in Table 6.6. For the seven-parameter model, the inter-trial variabilities are omitted as the comparison focuses on the parameters shared by the four-parameter and the seven-parameter model. The results for assessing the bias in parameter recovery are shown in Figures 6.16 to 6.19.

Results show that the seven-parameter model does not capture the data and underlying parameters better than the four-parameter model in terms of coverage, correlations and bias. If anything, there appears to be somewhat more bias in the recovery of drift rates under the 7-parameter model whether censored or not. From these findings, we can conclude that the seven-parameter model with or without censoring does not necessarily fit censored data better than the four-parameter model with censoring included.

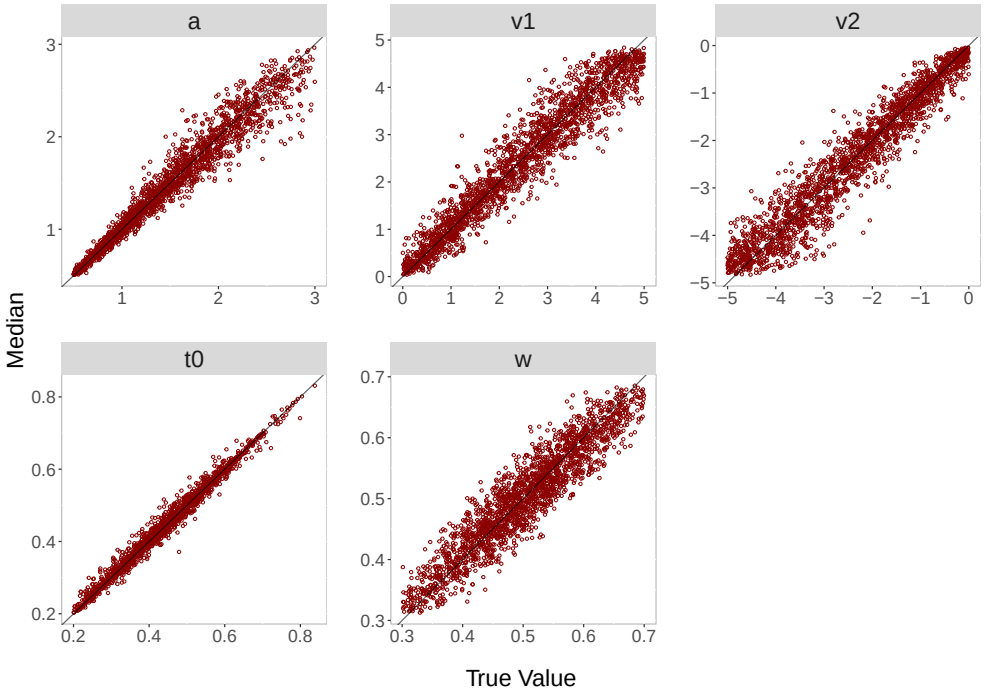


Figure 6.16: Diagonal Plot of Posterior Median Against True Value for 100 Trials for the Four-Parameter Non-Censored Analysis.

Par.	r	50% [a]	95% [a]	Par.	r	50% [a]	95% [a]
— 100 Trials, 4-p. nc —				— 100 Trials, 7-p. nc —			
a	.98	49	95	a	.95	53	95
v_1	.96	51	95	v_1	.95	46	91
v_2	.96	48	94	v_2	.94	43	90
t_0	.99	48	94	t_0	.99	41	95
w	.93	49	95	w	.92	48	96
— 500 Trials, 4-p. nc —				— 100 Trials, 7-p. c —			
a	.99	49	95	a	.95	52	96
v_1	.99	49	95	v_1	.94	44	91
v_2	.99	51	95	v_2	.94	39	89
t_0	1.00	49	94	t_0	.99	41	95
w	.99	50	95	w	.91	49	95

Table 6.6: Parameter Recovery Study: Evaluation Criteria (Correlations, Coverage) for Parameters Estimated from 100 and 500 Simulated Trials, Respectively, for the Four-Parameter Model without Censoring, and the Seven-Parameter Model with and without Censoring. *Note.* Par.=Parameters; r=Correlations (between true parameter values and posterior medians), 4-p. nc = four-parameter model non-censored, 7-p. nc = seven-parameter model non-censored, 7-p. c = seven-parameter model censored

[a] Percent of simulated datasets with true value in the HDI of this percentage.

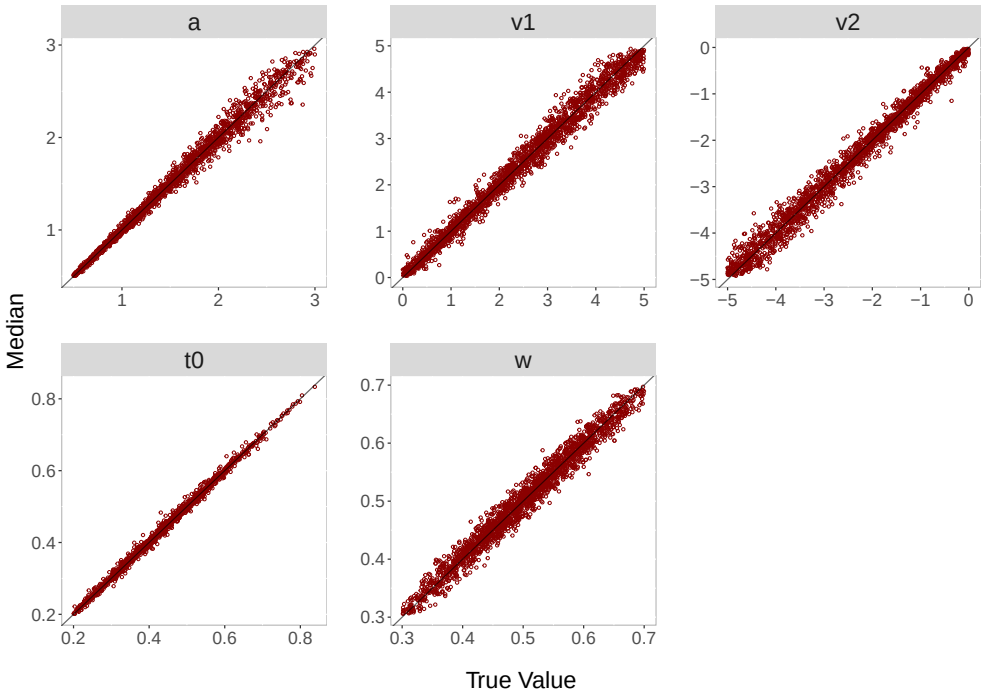


Figure 6.17: Diagonal Plot of Posterior Median Against True Value for 500 Trials for the Four-Parameter Non-Censored Analysis.

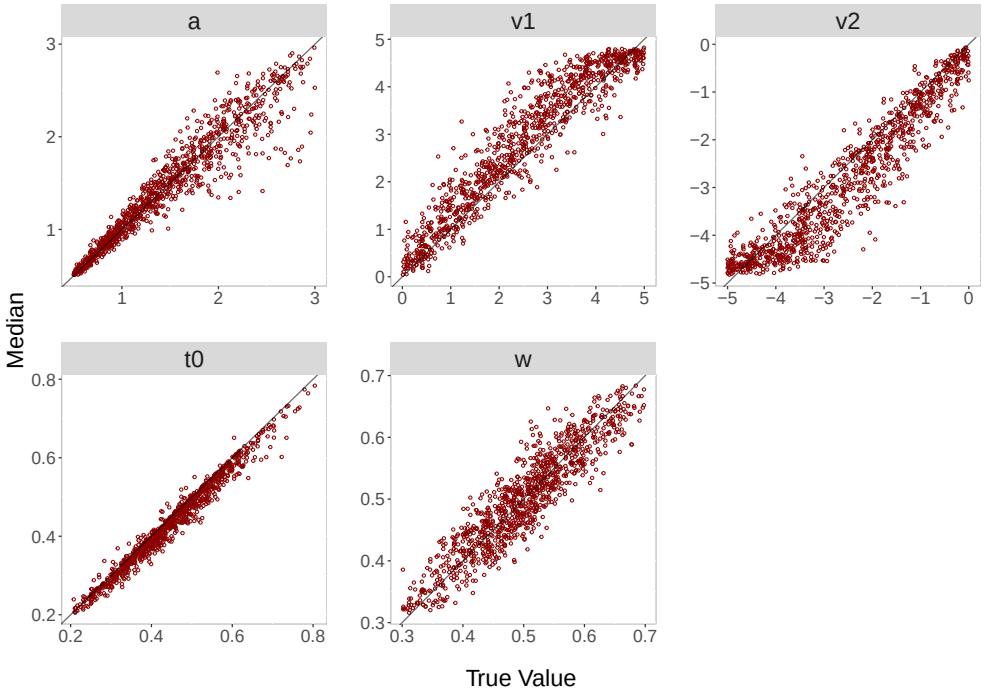


Figure 6.18: Diagonal Plot of Posterior Median Against True Value for 100 Trials for the Seven-Parameter Non-Censored Analysis.

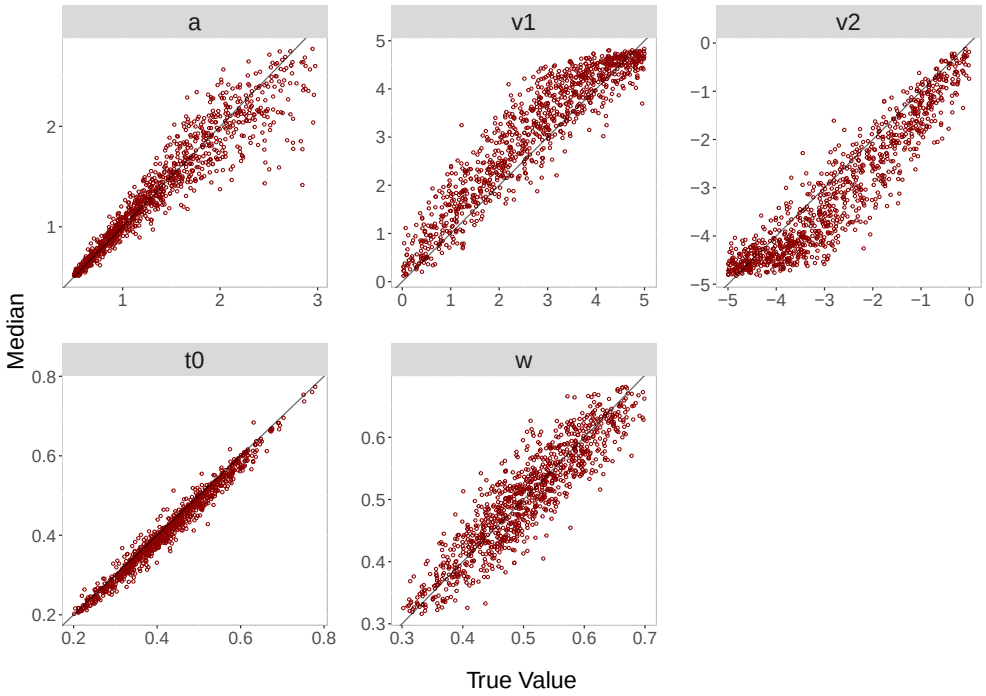


Figure 6.19: Diagonal Plot of Posterior Median Against True Value for 100 Trials for the Seven-Parameter Censored Analysis.

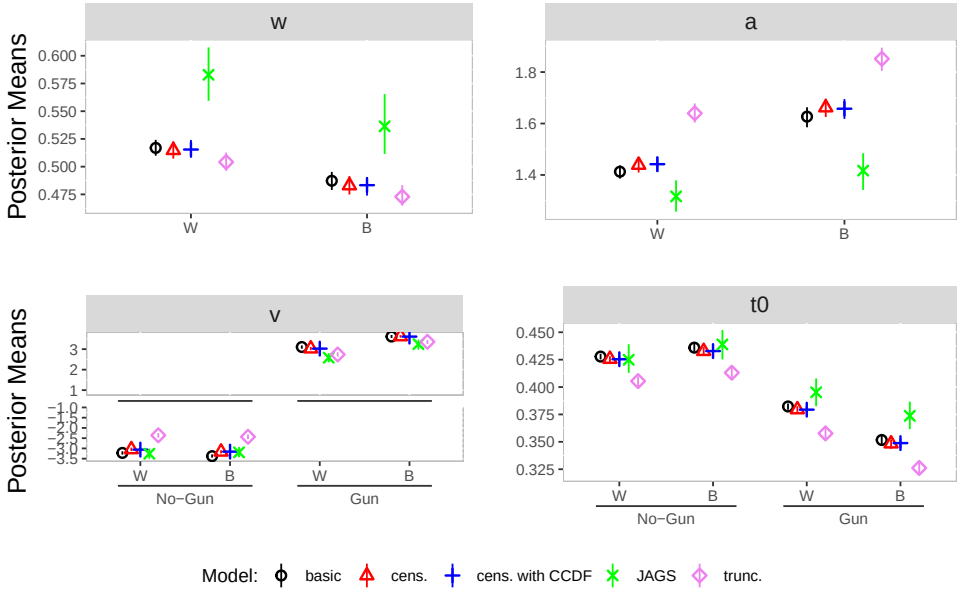
6.9.3 Reanalysis of the Pleskac Data with Non-Hierarchical Models

Following another suggestion by a reviewer, we additionally analyzed the datasets of Pleskac et al. (2018) with non-hierarchical models. We examined two different variants of non-hierarchical modeling. This means, in contrast to the analyses shown in the main body, we did not fit the data per condition in a hierarchical manner, but, for the first variant, we fitted each participant separately. We generated the same number of samples from the posterior distribution for each participant. For each parameter, we then averaged the individual parameter estimates from the $i-$th sample across participants for each $i = 1\ldots$ to obtain a sample of the posterior distribution of parameter means under the non-hierarchical model applied jointly and independently to each participant. For the second variant, we fitted all data with a non-hierarchical model as if all data were from one person.

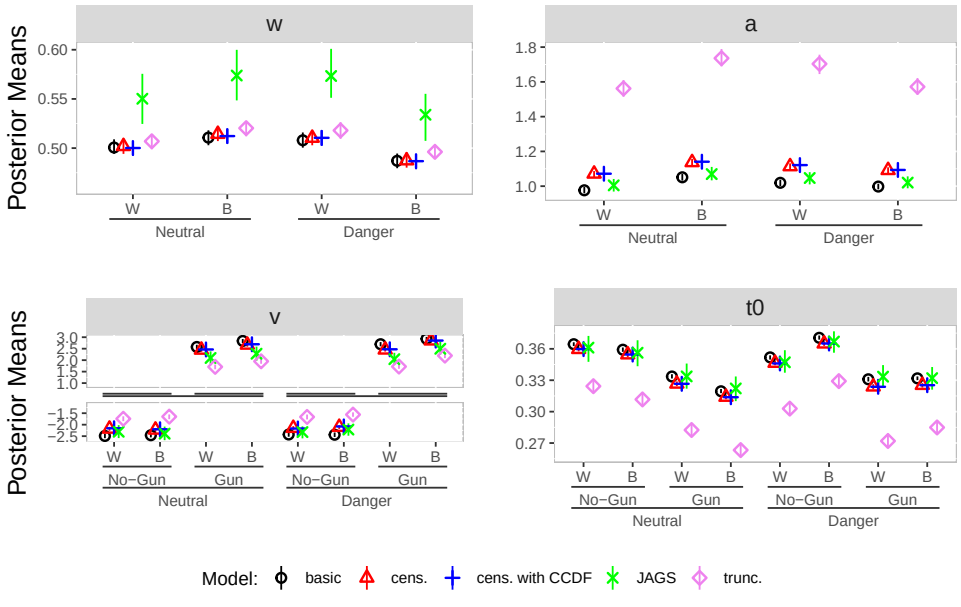
The plots in Figure 6.20 show the posterior means and their HDIs for the first variant to which we added the JAGS results from the above hierarchical analyses for the sake of comparison. It can be seen that the non-hierarchical results capture major trends that the JAGS results show. However, for both studies, results show sizeable deviations from the JAGS results in the relative starting point, the boundary separation and the non-decision time. Furthermore, it is also noticeable that the HDIs of the non-hierarchical posterior means are smaller than the HDIs of the population-level parameters of the hierarchical analyses (compare with Figures 6.12 and 6.13), reflecting the fact that hierarchical modeling achieves *partial* pooling, leading to more appropriate assessments of uncertainty in estimation as quantified by HDIs than does either no pooling or complete pooling. Again, the results of the truncated model deviate from the results of the other models as seen before in the hierarchical analyses.

The plots in Figure 6.21 show posterior means and HDIs for the second variant, where all data were fit with one model. We also added the JAGS results from the above hierarchical analyses to the plots. In this variant, not all results capture the main trends in the data as the hierarchical analyses. For example, in Study 2, results for the relative starting point deviate for both the neutral and the danger condition, between White and Black targets. Again, in both studies for two of four parameters (boundary separation and non-decision time), results show sizeable deviations from the JAGS results.

To sum up, the results of the non-hierarchical analyses capture main trends in the data for the non-hierarchical models fitted to each participant separately. Therefore, in this variant, one would probably not come to different conclusions regarding the effects of the factors that Pleskac et al. (2018) manipulated in this case. However, when all data are fit with one model as if they were from one person, the non-hierarchical analyses do not capture all trends that the hierarchical analyses capture, and even show opposite trends for some conditions and parameters. This probably reflects the fact that analyzing data with nonlinear models without taking the heterogeneity of the involved data sources into account is known to introduce systematic biases in parameter estimates (e.g., Rouder & Lu, 2005).

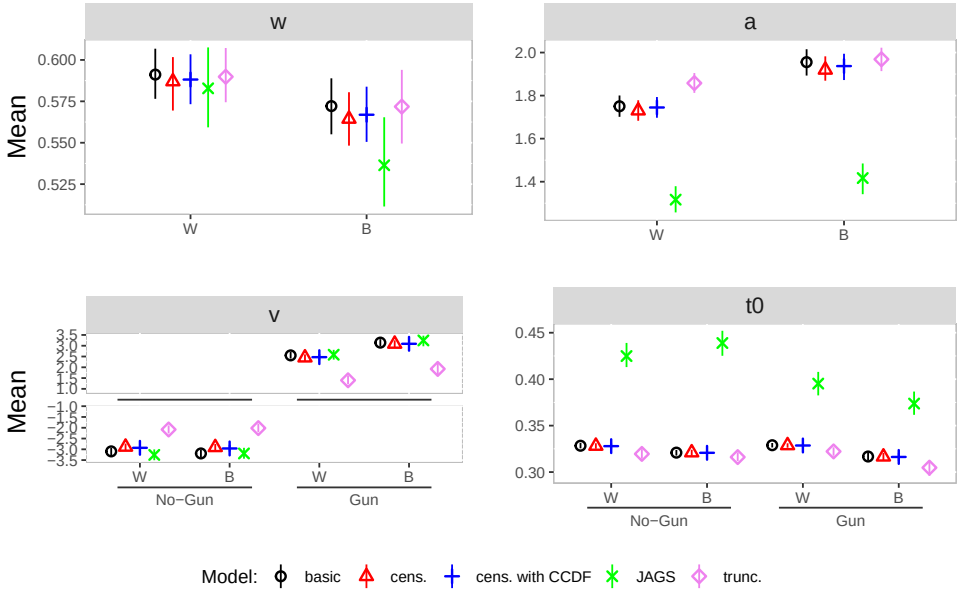


(a) Study 1

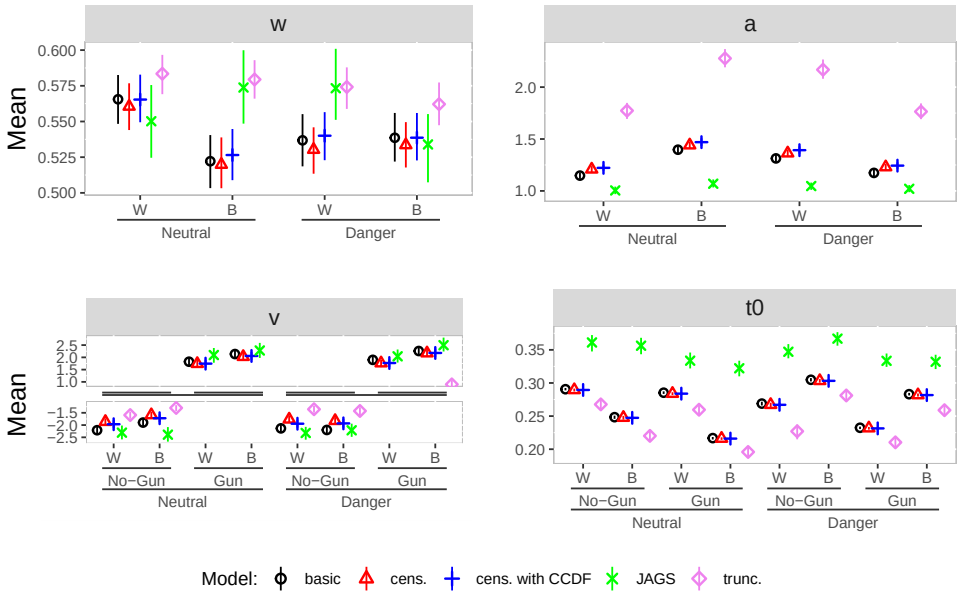


(b) Study 2

Figure 6.20: Parameter Estimates Non-Hierarchical Reanalysis, one Model per Person. *Note.* Means (dots) and 95% HDI (bars) for the group-level parameter estimates of the diffusion model in each condition; W = White, B = Black, cens. = censored, trunc. = truncated.



(a) Study 1



(b) Study 2

Figure 6.21: Parameter Estimates Non-Hierarchical Reanalysis, one Model for all Data.
Note. Means (dots) and 95% HDI (bars) for the group-level parameter estimates of the diffusion model in each condition; W = White, B = Black, cens. = censored, trunc. = truncated.

References

Amodio, D. M., Harmon-Jones, E., Devine, P. G., Curtin, J. J., Hartley, S. L., & Covert, A. E. (2004). Neural signals for the detection of unintentional race bias. *Psychological Science*, *15*(2), 88–93. https://doi.org/10.1111/j.0963-7214.2004.01502003.x

Arnold, N. R., Bröder, A., & Bayen, U. J. (2015). Empirical validation of the diffusion model for recognition memory and a comparison of parameter-estimation methods. *Psychological Research*, *79*(5), 882–898. https://doi.org/10.1007/s00426-014-0608-y

Barchard, K. A., & Russell, J. A. (2024). Distorted correlations among censored data: causes, effects, and correction. *Behavior Research Methods*, *56*(3), 1207–1228. https://doi.org/10.3758/s13428-023-02086-5

Carlston, D. E., Johnson, K., & Hugenberg, K. (Eds.). (2024). *The Oxford Handbook of Social Cognition (2nd edition)*. Oxford University Press.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, *76*(1). https://doi.org/10.18637/jss.v076.i01

Correll, J., Park, B., Judd, C. M., & Wittenbrink, B. (2002). The police officer's dilemma: Using ethnicity to disambiguate potentially threatening individuals. *Journal of Personality and Social Psychology*, *83*(6), 1314–1329. https://doi.org/10.1037//0022-3514.83.6.1314

Correll, J., Park, B., Judd, C. M., & Wittenbrink, B. (2007). The influence of stereotypes on decisions to shoot. *European Journal of Social Psychology*, *37*(6), 1102–1117. https://doi.org/10.1002/ejsp.450

Correll, J., Wittenbrink, B., Crawford, M. T., & Sadler, M. S. (2015). Stereotypic vision: How stereotypes disambiguate visual stimuli. *Journal of Personality and Social Psychology*, *108*(2), 219–233. https://doi.org/10.1037/pspa0000015

Draine, S. C., & Greenwald, A. G. (1998). Replicable unconscious semantic priming. *Journal of Experimental Psychology: General*, *127*(3), 286–303.

Fengler, A., Omar, A., Xu, P., Bera, K., & Frank, M. J. (2023). HSSM Documentation. Retrieved August 17, 2023, from https://lnccbrown.github.io/HSSM/

Gelman, A., Hwang, J., & Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, *24*(6), 997–1016. https://doi.org/10.1007/s11222-013-9416-2

Greenwald, A. G., Oakes, M. A., & Hoffman, H. G. (2003). Targets of discrimination: Effects of race on responses to weapons holders. *Journal of Experimental Social Psychology*, *39*(4), 399–405. https://doi.org/10.1016/S0022-1031(03)00020-9

Hartmann, R., & Klauer, K. C. (2021). Partial derivatives for the first-passage time distribution in Wiener diffusion models. *Journal of Mathematical Psychology*, *103*, 102550. https://doi.org/10.1016/j.jmp.2021.102550

Heathcote, A., Lin, Y.-S., Reynolds, A., Strickland, L., Gretton, M., & Matzke, D. (2019). Dynamic models of choice. *Behavior Research Methods*, *51*(2), 961–985. https://doi.org/10.3758/s13428-018-1067-y

Henrich, F., Hartmann, R., Pratz, V., Voss, A., & Klauer, K. C. (2023). The Seven-parameter Diffusion Model: An Implementation in Stan for Bayesian Analyses. *Behavior Research Methods*. https://doi.org/10.3758/s13428-023-02179-1

Johnson, D. J., Hopwood, C. J., Cesario, J., & Pleskac, T. J. (2017). Advancing Research on Cognitive Processes in Social and Personality Psychology: A Hierarchical Drift Diffusion Model Primer. *Social Psychological and Personality Science*, *8*(4), 413–423. https://doi.org/10.1177/1948550617703174

Johnson, D. J., Stepan, M. E., Cesario, J., & Fenn, K. M. (2020). Sleep Deprivation and Racial Bias in the Decision to Shoot: A Diffusion Model Analysis. *Social Psychological and Personality Science*, *17*(3), 194855062093272. https://doi.org/10.1177/1948550620932723

Krause, S., Back, M. D., Egloff, B., & Schmukle, S. C. (2012). A New Reliable and Valid Tool for Measuring Implicit Self-Esteem. *European Journal of Psychological Assessment*, *28*(2), 87–94. https://doi.org/10.1027/1015-5759/a000095

Kruschke, J. K. (2015). *Doing bayesian data analysis: A tutorial with R, JAGS, and Stan* (2nd ed.). Elsevier.

Lambert, A. J., Payne, B. K., Jacoby, L. L., Shaffer, L. M., Chasteen, A. L., & Khan, S. R. (2003). Stereotypes as dominant responses: On the "social facilitation" of prejudice in anticipated public contexts. *Journal of Personality and Social Psychology*, *84*(2), 277–295. https://doi.org/10.1037/0022-3514.84.2.277

Lin, Y.-S., & Strickland, L. (2020). Evidence accumulation models with R: A practical guide to hierarchical Bayesian methods. *The Quantitative Methods for Psychology*, *16*(2), 133–153. https://doi.org/10.20982/tqmp.16.2.p133

Matzke, D., & Wagenmakers, E.-J. (2009). Psychological interpretation of the ex-Gaussian and shifted Wald parameters: A diffusion model analysis. *Psychonomic Bulletin & Review*, *16*(5), 798–817. https://doi.org/10.3758/PBR.16.5.798

Mitrinović, D. S. (1970). *Analytic Inequalities*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-99970-3

Modrak, M., Moon, A. H., Kim, S., Bürkner, P., Huurre, N., Faltejskova, K., Gelman, A., & Vehtari, A. (2022). Simulation-Based Calibration Checking for Bayesian Computation: The Choice of Test Quantities Shapes Sensitivity. *arXiv*, *arXiv:2211.02383v1*.

Payne, B. K. (2001). Prejudice and perception: The role of automatic and controlled processes in misperceiving a weapon. *Journal of Personality and Social Psychology*, *81*(2), 181–192. https://doi.org/10.1037/0022-3514.81.2.181

Payne, B. K. (2006). Weapon Bias: Split-second decisions and unintended stereotyping. *Current Directions in Psychological Science*, *15*(6), 287–291.

Payne, B. K., Lambert, A. J., & Jacoby, L. L. (2002). Best laid plans: Effects of goals on accessibility bias and cognitive control in race-based misperceptions of weapons. *Journal of Experimental Social Psychology*, *38*, 384–396.

Pleskac, T. J., Cesario, J., & Johnson, D. J. (2018). How race affects evidence accumulation during the decision to shoot. *Psychonomic Bulletin & Review*, *25*(4), 1301–1330. https://doi.org/10.3758/s13423-017-1369-6

R Core Team. (2021). *R: A Language and Environment for Statistical Computing*. https://www.R-project.org/

Ratcliff, R. (1978). A Theory of Memory Retrieval. *Psychological Review*, *85*(2), 59–108.

Ratcliff, R., & Rouder, J. N. (1998). Modelling response times for two-choice decisions. *Psychological Science*, *9*(5), 347–356.

Ratcliff, R., & Smith, P. L. (2004). A comparison of sequential sampling models for two-choice reaction time. *Psychological Review*, *111*(2), 333–367. https://doi.org/10.1037/0033-295X.111.2.333

Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion Decision Model: Current Issues and History. *Trends in Cognitive Sciences*, *20*(4), 260–281. https://doi.org/10.1016/j.tics.2016.01.007

Ratcliff, R., & Tuerlinckx, F. (2002). Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review*, *9*(3), 438–481. https://doi.org/10.3758/BF03196302

Rivers, A. M. (2017). The Weapons Identification Task: Recommendations for adequately powered research. *PloS one*, *12*(6), e0177857. https://doi.org/10.1371/journal.pone.0177857

Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review*, *12*(4), 573–604. https://doi.org/10.3758/BF03196750

Stan Development Team. (2023a). Sampling Statements (version 2.32). Retrieved November 2, 2023, from https://mc-stan.org/docs/reference-manual/sampling-statements.html

Stan Development Team. (2023b). Truncated Data (version 2.32). Retrieved November 2, 2023, from https://mc-stan.org/docs/stan-users-guide/truncated-data.html

Stevenson, N., Donzallaz, M. C., Innes, R. J., Forstmann, B., Matzke, D., & Heathcote, A. (2024). EMC2: An R Package for cognitive models of choice. *Preprint*. https://doi.org/10.31234/osf.io/2e4dq

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating Bayesian Inference Algorithms with Simulation Based Calibration. *arXiv*, *arXiv:1804.06788v2*.

Thiem, K. C., Neel, R., Simpson, A. J., & Todd, A. R. (2019). Are Black Women and Girls Associated With Danger? Implicit Racial Bias at the Intersection of Target Age and Gender. *Personality & Social Psychology Bulletin*, *45*(10), 1427–1439. https://doi.org/10.1177/0146167219829182

Todd, A. R., Johnson, D. J., Lassetter, B., Neel, R., Simpson, A. J., & Cesario, J. (2020). Category salience and racial bias in weapon identification: A diffusion modeling approach. *Journal of Personality and Social Psychology*. https://doi.org/10.1037/pspi0000279

Todd, A. R., Simpson, A. J., Thiem, K. C., & Neel, R. (2016). The generalization of implicit racial bias to young black boys: Automatic stereotyping or automatic prejudice? *Social Cognition*, *34*(4), 306–323.

Ulrich, R., & Miller, J. (1994). Effects of truncation on reaction time analysis. *Journal of Experimental Psychology: General*, *123*(1), 34–80.

Vandekerckhove, J., & Tuerlinckx, F. (2007). Fitting the Ratcliff diffuion model to experimental data. *Psychonomic Bulletin & Review*, *14*(6), 1011–1026.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *arXiv*, *arXiv:1903.08008v5*.

Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, *32*(7), 1206–1220.

Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, *39*(4), 767–775.

Wabersich, D., & Vandekerckhove, J. (2013). Extending JAGS: A tutorial on adding custom distributions to JAGS (with a diffusion model example). *Behavior Research Methods*, *46*(1), 15–28. https://doi.org/10.3758/s13428-013-0369-3

Watanabe, S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. *Journal of Machine Learning Research*, *11*, 3571–3594.

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in Neuroinformatics*, *7*, 14. https://doi.org/10.3389/fninf.2013.00014

Chapter 7

General Discussion

The aim of this dissertation was to enrich the field of cognitive modeling with a flexible and robust implementation of the diffusion model in the probabilistic programming environment Stan to enable (a) modeling data with the seven-parameter diffusion model, and (b) modeling truncated or censored data with the diffusion model. By extending the diffusion model framework in Stan the work addressed major methodological gaps in experimental decision-making research, particularly in the family of two-alternative forced-choice tasks. Such tasks are commonly used in prejudice research in social psychology, among many other fields of application. One prominent paradigm in this area is the first-person shooter task (FPST).

At the time when the work on this dissertation started, there did not exist an implementation in the broad landscape of diffusion model implementations that was able to

- model all reaction time patterns in the data,
- handle sparse data,
- model truncated and censored data properly,

and additionally carries other desirable properties as being flexible in the sense of a free choice of the prior distributions, or not being limited to only one special programming language.

One of two central contributions of this work lies in the development and rigorous validation of a Stan-based implementation of the seven-parameter diffusion model. This implementation comprises the probability density function (PDF) and its partial derivatives. To show validity and correctness of the implemented algorithms, three studies were performed: a large-scaled recovery study, a simulation-based calibration study (both in Paper 1), and a comparison study to the already existing software package HDDM (Paper 2).

For the recovery study, we took typical parameter distributions reported in the literature as priors for the parameters. From these priors, we drew 2000 parameters each and obtained 2000 *true* parameter sets. From each parameter set we simulated one dataset with 100 trials and one dataset with 500 trials. Then, we analyzed all datasets with the new implementation and tested how precise the resulting parameters recovered the true parameters. In summary, the recovery study showed satisfactory to good parameter recovery of all model parameters, with better results for the four basic model parameters than for the three inter-trial variability parameters. This is in line with previous findings in the literature.

In a Bayesian framework, the interpretive power of recovery studies is inherently limited, as even strong parameter recovery does not guarantee the correctness of the algorithm's implementation. Therefore, we additionally performed a simulation-based calibration study (SBC), which aims to show the correct implementation of the underlying algorithm. An SBC evaluates whether the prior parameter distributions and the posterior parameter distributions are in agreement. When this is the case, it indicates that the implementation is free from systematic errors. In the present analysis, the SBC revealed no indications of such errors, thereby providing strong evidence for the validity of the implementation.

After validating the recovery and the correctness of the algorithm, we added a third validation test to embed the new implementation in the landscape of existing implementations. At that time, HDDM was one of the most popular software solutions in use when it came to seven-parameter diffusion models in a Bayesian hierarchical framework. HDDM (*hierarchical drift diffusion model*, Wiecki et al., 2013) is a software package written in Python that comprises several Bayesian hierarchical model formulations for the diffusion model and the linear ballistic accumulator model (Brown & Heathcote, 2008). In our analyses, we were interested in its diffusion model implementations. The main differences between Stan and HDDM are (a) that Stan has many interfaces to different data analysis languages, whereas HDDM is Python-based, and (b) that in Stan it is quite easy for the user to set the prior distributions for the model parameters, whereas HDDM limits the user to a small choice of predefined prior distributions.

We adopted the simulation study that was once performed when HDDM was published (see Wiecki et al., 2013) and in a second step we extended the design from the four-parameter model to the seven-parameter model. We set the prior distributions in the Stan models to equal the predefined informative prior set given in HDDM, analyzed the same data with both implementations, and compared the results.

Interestingly, we observed minor discrepancies in both the mean error and correlation analyses that appeared to exhibit a potentially systematic nature. To investigate this further, we conducted a simulation-based calibration study for HDDM to assess the correctness of the implemented algorithm. Indeed, the SBC revealed that there are issues in the underlying code. Upon examining the documented versus actually implemented prior distributions, we discovered discrepancies in three priors: the priors for the relative starting point, the group-level mean of the relative starting point, and the group-level standard deviation of the boundary separation are implemented more informatively than documented. After rerunning the Stan analyses with the corrected priors the oberseved anomalies largely disappeared. Nonetheless, a second run of the SBC for HDDM still detected minor irregularities, leaving open the possibility of residual misspecification of the priors given the difficulty of discerning the priors from the complex source code on Github.

As another potential source of discrepancy, we also explored differences in the precision of the computations. Notably, some likelihood values differed beyond expectation between Stan and HDDM. This lack of precision could indeed be another reason that the SBC still revealed indications for implementation errors.

However, given the overall agreement in parameter recovery and correlations between both implementations, these deviations might not be relevant in practical applications. Our findings could reflect a trade-off between precision of the calculations and runtime, as can also be seen in the runtime-analysis: For the four-parameter model, Stan outperformed HDDM and was much faster. In contrast, for the seven-parameter implementation, HDDM was faster than Stan. This might also be due to the different MCMC-samplers and integration-routines implemented in Stan and HDDM.

The second central contribution of this work lies in the extension of the Stan-based implementation of the seven-parameter diffusion model to enable modeling truncated and censored data. After the implementation of the seven-parameter PDF functions was finished, the Stan-community asked whether we could also implement the cumulative distribution function (CDF) and its complement (CCDF) for the diffusion model. This directly played into our cards as these functions are needed to enable modeling truncated and censored data, which occur in experiments that include a response window. Therefore, after we successfully validated the PDF functions, we continued with the implementation of the CDF and CCDF functions and their partial derivatives. We performed analog validation tests for both functions as for the PDF - a recovery study and an SBC. Results of the recovery study are satisfactory in terms of correlation, coverage, and bias. Results for the SBC do not show systematic errors, indicating that the algorithm is implemented correctly.

Subsequently, we illustrated the new method by reanalyzing an already existing dataset from an FPST-study, which brought interesting insights in the different models' behavior. We took the data from Studies 1 and 2 in Pleskac et al. (2018), where the authors investigated the influence of skin color on the decision to shoot with different response windows. In Study 1, a relatively liberal response window with upper response deadline of 850 ms was used, which caused only 3% of the trials to fall outside the response window. In Study 2, the response window was decreased to 630 ms, which caused 10% of the trials to be outside the response window. The authors treated data as censored and used a heuristical approach to impute the missing responses. This approach assumes that the relative frequency of the missing responses matched the relative frequency of the observed responses.

We analyzed data with four different hierarchical models: (a) the basic four-parameter model without accounting for the response window, (b) a censored model using the heuristically imputed responses by the authors, (c) another censored model where we compute the probability of ending at the response-1 boundary or response-0 boundary using the CCDF, and (d) a truncated model. For all models we chose the same prior distributions as the authors.

As extensively investigated and discussed in the third paper above, we found that the results of the truncated model deviated substantially from the results obtained by the basic or the censored models. In order to examine the cause of the discrepancies, we performed several analyses. First, we compared observed and predicted frequencies of trials without response, second, we inspected histograms of predicted reaction times, third, we plotted quantile-quantile plots with predicted vs. observed data quantiles, and fourth, we computed the model selection

index WAIC. At the first glance, it seemed that the truncated model outperformed the other models as it better fits the actual data within the response window. But the results of the analysis of the predicted frequencies of trials outside the response window show that the truncated model largely overestimates the number of these trials. Taken together, the analyses suggest that the underlying data-generating process in this experiment might not be a pure Wiener diffusion process. The response window might distort the diffusion process, and therefore, we would see it as important to additionally perform diagnostic model checks to make sure that the applied models fit to the data.

Additionally, there might also be other processes that mix with the diffusion process. These could be mind wandering or guessing processes. Such possibilities need further investigation in future research.

The new implementation brings many advantages. It is a great step to be able to model data with the seven-parameter model hierarchically in a Bayesian framework. Our implementation allows one to compose the model freely with all combinations of parameters and complexity between the basic model and the full model.

Furthermore, it is now possible to model truncated and censored data in a sophisticated way with a diffusion model in Stan. This will facilitate the modeling effort for many researchers that deal with response windows or other experiments that produce censored or truncated data.

It is also worth noting that with the third model option described above to model the censored data using the CCDF function and the probability function to hit one or the other boundary, we offer researchers a more sophisticated way of dealing with censored data. With this opportunity, the researcher is not anymore forced to make a heuristical assumption on the distribution of the censored responses, but can let the model derive the frequencies.

Nevertheless, there are also limitations. One main disadvantage is that the computational time for the full model in the hierarchical setting is very long and large computational resources are needed. This makes it difficult to fit the model. This might be connected to the complexity of the functions themselves, as for the inter-trial variabilities integrals have to be computed which need many resources. Another reason could be that there is not enough information in the data to estimate the inter-trial variabilities quickly. However, it is unusual to define a model with all three inter-trial variabilities hierarchically. Usually, only the four basic parameters are set hierarchically and the inter-trial variabilities are not hierarchical.

Another option to circumvent the slow fitting is not to include all three inter-trial variabilities at once. Instead, only one or two inter-trial variabilities can be included in the model definition. In our tests, it seemed that the variability in drift rate is very fast, as for this variability no integral has to be computed, and the variability in relative starting point is very slow and problematic to fit.

It would be interesting for future research to collect data in a study with a simple design with only the most necessary conditions, and specifically investigate the behavior of the different models.

Now, with a Bayesian implementation in Stan where the user is free in the choice of the prior

distributions, another limitation occurs. The more complex the model is, the more options exist to define a model and to set the priors. The question how to set the priors is a complex and important one, but this holds for all models in a Bayesian context.

The question of how to properly deal with outliers is still open. As the model is very sensitive to outliers it is important to detect outliers reliably to obtain interpretable results. This, together with analyses regarding mind wandering and guessing processes, is another topic for future research.

Besides methodological questions regarding the behavior of the model in different situations, with a varying number of parameters, with different priors, or in combination with secondary processes that distort the underlying data-generating process, also questions from the application side are interesting for future research.

One prominent example for the application of a diffusion model analysis is, as already seen, the first-person shooter task, FPST. Research in this area still continues and develops. Also on the side of the experimental design new methods are developed constantly. For the FPST, recently, the immersive shooting simulator was taken to a new level. In such a simulator study, the participant interacts with a suspect in life-sized video scenarios filmed from a first-person perspective. If the participant decides to shoot, he fires a modified handgun that produces a realistic sound and recoil and records the response time (Pleskac et al., 2025). In this setting, only the *shoot* decision is recorded. The *don't shoot* decision is measured indirectly. This is the case when no shoot-reaction was shown and an implicit boundary crossing is present.

Such modified FPST experiments no longer belong to the family of two-alternative forced-choice tasks as participants are not compelled to respond. Instead, they fall under the category of Go/No-Go tasks (Chen et al., 2025; Donders, 1969). Nevertheless, the resulting data can be treated as censored, given that the number of non-response trials is known, and therefore, can be modeled with a censored diffusion model. This is the point where our novel implementation and modeling approach prove essential. Through our work, the analytical toolkit is broadened, to enable researchers with enhanced means to probe such questions to their depth.

We hope that the work in this dissertation will serve many researchers as helpful in the analysis of their data and that many new and interesting insights can be generated using our methods.

References

Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: linear ballistic accumulation. *Cognitive Psychology*, *57*(3), 153–178. https://doi.org/10.1016/j.cogpsych.2007.12.002

Chen, Y., Fang, J., Cesario, J., Liu, T., & Pleskac, T. J. (2025). Comparing one-boundary and two-boundary evidence accumulation models for go/no-go processes: An application to the decision to shoot. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *47*(0).

Donders, F. C. (1969). On the speed of mental processes. In W. G. Koster (Ed.), *Attention and performance II*. North Holland. (Original work published in 1868).

Pleskac, T. J., Cesario, J., & Johnson, D. J. (2018). How race affects evidence accumulation during the decision to shoot. *Psychonomic Bulletin & Review*, *25*(4), 1301–1330. https://doi.org/10.3758/s13423-017-1369-6

Pleskac, T. J., Cesario, J., Johnson, D. J., & Gagnon, G. (2025). Modeling police officers' deadly force decisions in an immersive shooting simulator. *Journal of Experimental Psychology: Applied*. https://doi.org/10.1037/xap0000542

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in Neuroinformatics*, *7*, 14. https://doi.org/10.3389/fninf.2013.00014

Acknowledgments

I thank all people that supported me on my way to this wonderful achievement.

I thank my family that always believed in me and always supported me in the things I do.

I thank my advisor, Christoph Klauer, for all his great efforts in helping me, especially when I did not know how to proceed. I am thankful for your kindness, your generosity, and patience with me. You always had an open door and open ear and always took the time to explain complicated methods and to think about a solution to my problems. I learned a lot from you and for that I am very thankful.

I thank my co-workers, Raphael Hartmann, and Valentin Pratz. I was so glad to have you around me. We had many conversations on the mathematical and programming specific issues.

I thank SMiP for giving this great opportunity. I value the workshops, courses, and retreats you provided us with and the unforgettable time we spent together in Mannheim, Tübingen, and Freiburg. I thank Anke Söllner and Annette Förster for their help, whether with hotel booking or answering questions about administrative stuff. I also thank the many PhD candidates and PIs with whom I had conversation. Special thanks go to Thorsten Meiser, Andreas Voss, and Andrea Kiesel for their great help.

I thank the Stan-development team that accompanied me nearly the whole phase of my dissertation. You introduced me to the deep programming skills needed for the Stan language. I thank Steve Bronder, Andrew Johnson, and Bob Carpenter who did not get tired of reviewing thousands and thousands of lines of code and eventually accepted the code for the inclusion in Stan.

I thank Tim Pleskac for all the effort he put into my problems in recovering his data for my third paper. I really enjoyed our exchange on the censored data and the tricky R/Matlab-problem, and I am glad that we eventually solved it.

I thank the Freiburger Orchestergesellschaft, that gave me a place where I could make music and where I learned much about a classical orchestra. I thank Antje Hain, Kirsten Brügmann and Sabine Mierbach with whom I spent wonderful hours outside university and outside the orchestra.

Last but not least I thank one special person for accompanying me on my way and encouraging me to go to Freiburg and start the PhD.

Further Acknowledgments

This work was performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC.

Funding information

All three papers of this dissertation were funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - GRK 2277 'Statistical Modeling in Psychology' (SMiP) - project number 310365261, as well as by a Koselleck grant, DFG Kl 614/39-1 awarded to Karl Christoph Klauer.

Availability of Data and Materials

All R scripts for the simulations and the empirical analyses, as well as experimental datasets, are available at the Open Science Framework:

For the first manuscript: https://osf.io/486up/ (Retrieved September 23, 2025).

For the second manuscript: https://osf.io/vb4ex/ (Retrieved September 23, 2025).

For the third manuscript: https://osf.io/vg7zf/ (Retrieved September 23, 2025).

Availability of Code

The code for the PDF functions is available in Stan since version 2.35. On the Stan github page[1], it comprises the files `wiener5_lpdf` and `wiener_full_lpdf`.

The code for the CDF functions is available in Stan since version 2.38. It comprises the files `wiener4_lcdf_unnorm`, `wiener4_lccdf_unnorm`, `wiener_full_lcdf_unnorm`, and `wiener_full_lccdf_unnorm`. Note that with this release the function call has changed. The functions have to be called with the appendix `_unnorm`, e.g. `wiener_full_lcdf_unnorm`.

[1] Retrieved January 11, 2026 from https://github.com/stan-dev/math/tree/develop/stan/math/prim/prob