`dynConfiR`: An R package for sequential sampling models of decision confidence

Sebastian Hellmann[1,2], Michael Zehetleitner[2], and Manuel Rausch[2,3,4]

[1]Chair of Behavioral Research Methods, TUM School of Management, Munich, Germany

[2]Philosophical-pedagogical Faculty, Catholic University of Eichstätt-Ingolstadt, Eichstätt, Germany

[3]Faculty of Society and Economics, Rhine-Waal University of Applied Sciences, Cleves, Germany

[4]Department of Psychology, University of Klagenfurt, Austria

Unreviewed Preprint

27 October 2025

Author Note

Sebastian Hellmann https://orcid.org/0000-0002-3621-6343 (Corresponding author; Adress: TUM School of Management, Arcisstraße 21, 80333 München, Germany. E-mail: sebastian.hellmann@tum.de)

Michael Zehetleitner https://orcid.org/0000-0003-3363-2680

Manuel Rausch https://orcid.org/0000-0002-5805-5544

Data and Code availability

The `dynConfiR` package is available on CRAN (https://cran.r-project.org/web/packages/dynConfiR/). This paper is based on version 1.1.1 of the package. Data and code for this paper are available at https://github.com/SeHellmann/dynConfiR_Paper. The development version of the package is available at https://github.com/SeHellmann/dynConfiR

Author Contributions

Funding acquisition: MR, MZ. Conceptualization & methodology: SH. Formal analysis, software, & visualization: SH. Writing - Original Draft: SH. Resources & project administration: MR, MZ. Writing – Review & Editing: all authors.

Funding

This work was funded by the Deutsche Forschungsgemeinschaft (grant numbers ZE887/8-1 to MZ and RA2988/3-1 to MR).

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Abstract

The modeling of response times using sequential sampling models has a long history. Because choices, confidence judgments, and reaction times are closely linked in perceptual decisions, it seems only natural to simultaneously model these three outcome variables of a decision. In the package dynConfiR, we implemented various sequential sampling models of choice, response time, and decision confidence in R. This paper gives an overview of the package, which provides probability density functions as well as high-level functions for fitting parameters to empirical data, prediction of reaction time and response distributions and simulation of artificial data sets. We describe the mathematical specification of the implemented models and give a detailed description of the implemented likelihood functions. In addition, we outline the workflow for applying the model to empirical data step-by-step: data preprocessing, model fitting, model prediction, quantitative model comparison, and visual assessment of model predictions. Finally, we present results from a parameter and model recovery analyses and assess the precision in calculating probability densities, illustrating the reliability of the implemented computations. Offering intuitive usability and high flexibility, the package is targeted at researchers in the fields of decision-making and confidence and does not require expert-level programming skills.

Keywords: R package, cognitive modeling, confidence, decision making, drift diffusion models, sequential sampling models

`dynConfiR`: An R package for sequential sampling models of decision confidence

Contents

Introduction **7**

Alternative software . . . 8

Scope and limitations of the package . . . 10

Structure of the present paper . . . 12

Sequential sampling confidence models **12**

Drift diffusion-based confidence models . . . 13

Dynamical visibility, time, end evidence model (dynaViTE) . . . 13

Dynamical weighted evidence and visibility model (dynWEV) . . . 15

Two-stage dynamic signal detection theory (2DSD) . . . 15

Drift diffusion confidence model (DDConf) . . . 15

Race Models using Wiener processes . . . 15

The multiple-threshold log-normal race model . . . 17

Common mechanism in forming confidence judgments . . . 18

Non-decision time component . . . 18

Application of models with post-decisional accumulation to data from simultaneous choice and confidence reports . . . 18

Other sequential sampling models of confidence . . . 19

Functionalities of the package **21**

Installation . . . 21

Workflow . . . 21

Fitting confidence models to experimental data . . . 22

Experimental paradigm . . . 22

Data format . . . 23

Fitted parameters . . . 23

Fitting procedure . . . 25

Predicting confidence and response time distributions . . . 26

Quantitative model comparison using information criteria . . . 26

Other functions . . . 27

Probability density functions . . . 27

Log-likelihood functions . . . 28
Simulation functions . . . 28

Example of Application in Model Comparison **29**
Experimental method . . . 29
Data . . . 30
Data preprocessing . . . 30
Analyses . . . 31
Model fitting . . . 31
Quantitative model comparison . . . 32
Prediction and visual model fit . . . 34
Exploratory analysis: Fixing model parameters . . . 36

Parameter recovery analysis **43**
Method . . . 43
Results . . . 43

Model recovery analysis **47**
Method . . . 47
Results . . . 47

Precision analysis **49**
Method . . . 49
Results . . . 50

Summary **51**

Contributions and bug reports **51**

Declarations **51**
Funding . . . 51
Competing interests . . . 52
Ethics approval . . . 52
Consent to participate . . . 52
Consent for publication . . . 52
Availability of data and materials . . . 52

Code availability 52
Authors' contributions 52
Detailed Method 69
Recovery Plots 70

Introduction

Recently, confidence has gained increasing research interest in the field of cognitive computational modeling (e.g., Adler & Ma, 2018; Aitchison et al., 2015; Desender et al., 2021; Hellmann et al., 2023, 2024; Kiani et al., 2014; Maniscalco & Lau, 2016; Maniscalco et al., 2016; Moran et al., 2015; Pleskac & Busemeyer, 2010; Ratcliff & Starns, 2009, 2013; Rausch et al., 2018; Zawadzka et al., 2017). Many experimental tasks and everyday decisions include uncertainty, so the decision-maker can not be entirely sure whether their decision was correct. The resulting degree of belief in the correctness of one's decision is referred to as confidence (Pouget et al., 2016). Because confidence is also relevant in everyday behavior and communication, for example when driving in a foggy environment or making difficult medical diagnoses, it is essential to understand how confidence arises from the decision process.

Many models of confidence are based on traditional signal detection theory (SDT, Green & Swets, 1966). We refer to these models as static models as they do not explain the single trial dynamics of a decision but assume that the decision is made by comparing a single random variable against a criterion. Traditional SDT models have been extended to account for confidence judgments, for example by introducing more criteria, additional information gain, or noise in the confidence judgment (Adler & Ma, 2018; Mamassian & de Gardelle, 2022; Rausch et al., 2018, 2023; Shekhar & Rahnev, 2021, 2024). Static confidence models have proven successful in accounting for the relationship between task difficulty and confidence and have been useful for explaining discrepancies between confidence judgments and actual accuracy. Although response times may be incorporated into these models to account for the dynamical properties of the generation of evidence on which the decisions are based, e.g. by scaling both the magnitude and noise with the response time, they do not provide an explanation for how response times are generated as a dependent variable. For this reason static models may be applied to interrogation tasks (Bogacz et al., 2006), in which the experimenter externally controls how long evidence may be accumulated, but cannot account for the empirical patterns like the negative relationship between discriminability and response times that are commonly observed in free response tasks, in which subjects themselves determine when to make a decision. In addition, confidence is also closely related to decision time in many free response tasks (Hellmann et al., 2024; Kiani et al., 2014; Rahnev et al., 2020; Vickers et al., 1985). In contrast to static models, dynamical models explain the generation of response time distributions and may thus provide insight into the causal relationship between task difficulty, decision time, and confidence. Dynamical models of decision-making assume a sequential sampling process, that is, evidence is sampled from a noisy distribution repeatedly over time, and an internal decision variable is updated until a particular stopping rule is met and the decision is triggered (Ratcliff & Smith, 2004). It should be noted that most sequential sampling models of decision making are based on signal detection

principles and thus the term dynamical signal detection theory is occasionally used for random walk models (Pleskac & Busemeyer, 2010; Smith, 2000).

A prominent example of a computational model in the field of decision-making research is the drift diffusion model (DDM) which was originally used to explain response time distributions in memory retrieval tasks (Ratcliff, 1978). Since it was initially formulated, it was extended by including additional parameters and applied in various experimental tasks (Ratcliff & Smith, 2004). However, the DDM, in its original conception, does not account for confidence judgments. In two previous studies, we compared different confidence models based on two important prototypes of dynamical models of decision-making, the DDM and the race of accumulators (Hellmann et al., 2023, 2024). We demonstrated that fitting the joint distribution of choice, response time, and confidence is useful for testing computational models of decision making and is more desirable than fitting summary statistics of the data.

The presented package includes functions to fit response times and confidence judgments in binary choice tasks based on the following models: the drift diffusion confidence model (DDConf, Hellmann et al., 2023; Ratcliff, 1978), the two-stage dynamical signal detection model (2DSD, Pleskac & Busemeyer, 2010), the dynamical weighted evidence and visibility model (dynWEV, Hellmann et al., 2023), the dynamical visibility, time, and evidence model (dynaViTE, Hellmann et al., 2024), several versions of race models (Moreno-Bote, 2010), and the multiple-threshold correlated log-normal race model (MTLNR, Reynolds et al., 2020). The models are explained in more detail in the next section.

Due to their mathematical complexity, dynamical models of decision-making and confidence are challenging to implement. By providing the `dynConfiR` package, we aim to remove the hurdle of implementing likelihood functions and fitting procedures to facilitate the application of confidence models for research questions in psychology and cognitive neuroscience.

Alternative software

Software already exists to analyze response time data for decision models, mainly in the context of the DDM. Some examples are the R packages rtdists (Singmann et al., 2020), which offers probability distribution and simulation functions for the seven-parameter DDM and the linear ballistic accumulator model (Brown & Heathcote, 2008), and RWiener (Wabersich & Vandekerckhove, 2014b), which provides an implementation of the four-parameter DDM with functions for parameter fitting. Fast-dm (Andreas Voss & Jochen Voss, 2007) is a stand-alone command line tool for fitting the seven-parameter DDM to empirical data. This tool also allows to fit the model with arbitrary parameters varying across different conditions.

The python toolbox HDDM allows for hierarchical parameter estimation of DDM parameters (Wiecki et al., 2013). Similarly, the EMC2 is an R package that allows for the Bayesian estimation of parameters in various evidence accumulation models to choice and response time data (Stevenson et al.,

2024). These packages are immensely flexible when it comes to the specification of the hypothesized effect of experimental manipulations and other predictors on model parameters. In that sense, confidence judgments could be incorporated into the analysis as an additional predictor, e.g. for guiding learning (Drugowitsch et al., 2019) or influence the decision-threshold in upcoming trials (Desender et al., 2019). However, these tools do not allow scientists to easily include confidence judgments as an additional dependent variable in the computational models.

In addition, the wiener module for JAGS (Wabersich & Vandekerckhove, 2014a) and Stan allow researchers to easily incorporate the DDM in more complex probabilistic models (see e.g. Fontanesi et al., 2019, for an application in reinforcement learning). Recently, the seven-parameter DDM was implemented in Stan (Henrich et al., 2024). Using JAGS or Stan, scientist could also build more complex models, to account for confidence data, however, this requires at least some knowledge in MCMC sampling and knowledge of either software tool. In addition, to include other models that are not based on the DDM like race models, researchers would have to write their own extension in the form of a likelihood function.

As a summary, there are a lot of alternatives when it comes to the models to the modeling of choices and response times. These alternatives are already very popular and provide a high degree of flexibility and some allow for the hierarchical estimation of model parameters, which is beneficial if only a limited number of trial are available for each subject.

For fitting confidence data, the statConfR package (Rausch et al., 2025) allows for parameter fitting in the context of static confidence models and the computation of popular measures of metacognitive performance such as meta-d′ (Maniscalco & Lau, 2012). The ReMeta toolbox is a Python library which facilitates the estimation of parameters with a high degree of flexibility in specifying the data generating process (Guggenmos, 2022). Finally, Mamassian and de Gardelle (2022) provided a Matlab toolbox to fit confidence judgments in the so-called confidence forced-choice paradigm. However, these software packages are based on signal-detection models and cannot account for response times.

Table 1 summarizes and compares available software to the `dynConfiR` package. Only very few software tools currently exist that implement sequential sampling models of confidence. The DMC software includes an implementation of the multiple-threshold linear ballistic accumulator model and the MTLNR (Heathcote et al., 2019; Reynolds et al., 2020). Unfortunately at the time of writing, the DMC software is no more actively maintained and comprehensive documentation for the application of these models is not available. Instead, the authors of DMC are planning to extend the EMC2 software to include confidence models in a future release (A. Heathcote, personal communication, September 18, 2025). `dynConfiR` provides implementations of the joint distribution of choice, response time, and confidence judgment for several sequential sampling models of decision confidence together with wrapper functions to compute the

likelihood for a whole data set and a given set of parameters. This enables advanced users to define custom likelihood functions tailored to their specific experiments. Particularly, researchers can implement their own mapping between stimulus properties and drift rates in two-alternative force-choice tasks, and incorporate attentional discounts when eye-tracking data are available (Krajbich et al., 2010).

In addition, the package includes functions for maximum-likelihood parameter estimation and for predicting or simulating distributions, which facilitates the use by less-advanced users, who want to focus on the application of the models to their data. Functions are written to provide an intuitive and straightforward way to implement the whole workflow of model fitting and comparison with only a few lines of code while still providing possibilities for customization, for example, fixing parameters that should not be fitted.

Table 1

Comparison of features of the `dynConfiR` *to other existing software.*

		Dependent variables		User-accessible functions		
Software	Programming language	Response times	Confidence judgments	Density functions	Fitting functions	Statistical paradigm
rtdists	R	✓	✗	✓	✗	frequentist
RWiener	R	✓	✗	✓	✓	frequentist
Fast-dm	stand-alone command line tool	✓	✗	✗	✓	frequentist
HDDM	Python	✓	✗	✓	✓	Bayesian
EMC2	R	✓	✗	✗	✓	Bayesian
wiener module	JAGS	✓	✗	–	–	Bayesian
lpdf_wiener	Stan	✓	✗	–	–	Bayesian
statConfR	R	✗	✓	✗	✓	frequentist
ReMeta	Python	✗	✓	✗	✓	frequentist
cfc	Matlab	✗	✓	✗	✓	frequentist
dynConfiR	R	✓	✓	✓	✓	frequentist

Note. ✓: available; ✗: not available; –: not applicable

Scope and limitations of the package

Although the dynConfiR package implements various computational models covering a broad range of concepts (e.g. models with post-decisional accumulation time, race models and drift diffusion

models) and the fitting procedure is user-friendly, there are some limitations concerning both the scope and the usability of the package we will describe below.

First, we restricted the implementation of models to confidence models for which the likelihood of the joint distribution of choices, response times, and confidence judgments is mathematically tractable. This means that the models do not capture several aspects and variations of the models discussed in the literature, e.g., collapsing boundaries and leakage. All models assume time-constant boundaries as well as stationary drift rates, which are independent of the current state of the process. In addition, the race models do not allow for inhibition. Time-collapsing boundaries are discussed in the literature to implement time-costs in the diffusion process, and leakage and inhibition are referred to as neurally plausible mechanisms of accumulation processes (Tajima et al., 2016; Usher & McClelland, 2001). In addition, several dynamic confidence models like RTCON and RTCON2 (Ratcliff & Starns, 2009, 2013) and the bounded accumulation model by Kiani et al. (2014) are not implemented in the package. As new computational models of confidence judgments keep being proposed, we plan to extend the range of models in the future with other models. In addition, we encourage contributions to the package by other researchers (see the GitHub page of the package).

Second, despite some of the models assuming post-decisional evidence accumulation, there is currently no model implemented that also accounts for confidence response times in paradigms with subsequent confidence reports. In such situations, one has to rely on the observed confidence response times to inform the post-decisional accumulation period (see Hellmann et al., 2024). There are some proposed models that do account for confidence response times, which are not yet implemented in dynConfiR (Herregods et al., 2023; Moran et al., 2015). Importantly, the models assuming post-decisional evidence accumulation accounted for response times and confidence judgments in paradigms with simultaneous choice and confidence reports and even outperformed the alternative models that did not include post-decisional accumulation (Hellmann et al., 2023).

Third, although we restrict the package to include only models with a feasible likelihood, the parameter fitting procedure requires a lot of computation time. Because the implemented maximum likelihood procedure for parameter fitting requires the computation of the joint distribution on a trial-level for each iteration, even with the effective finite sum approximations of the response time density in the diffusion-based models, the computational effort is considerable (see also the sections Model recovery analysis and Parameter recovery analysis, where we report the computation times for the likelihood evaluation and parameter fitting).

Finally, there are some conceptual limitations that come with the maximum likelihood estimation technique used for parameter fitting. Unlike in Bayesian estimation methods, the estimated parameters are

only point estimates and there is no measure for the uncertainty of the estimation. There is no implemented method to report confidence intervals for the parameters because the dependency among each other (e.g., confidence thresholds need to be ordered) makes the standard procedures for computing confidence intervals difficult. The information criteria used for assessing the goodness-of-fit of the models are also based on the maximum likelihood estimation and do not account for the functional complexity of the models (Myung, 2000). However, the good model identification presented in this article (see Model recovery analysis) indicates that using BIC for model comparison has generally a low level of mis-identifications when the models are fitted to empirical data.

Structure of the present paper

In the present paper, we will first present the confidence models that are included in the package, explaining all parameters and giving mathematical definitions. The second section provides details about the functionalities of the package. Based on the implemented functionalities, we propose workflows for different use cases and show possibilities for individual settings in the analyses. In addition, we showcase the suggested workflow for model comparison using an empirical data set. The sections Parameter recovery analysis, Precision analysis, and Model recovery analysis contain results from simulation studies examining the performance of the package. All analyses scripts and data sets used in this paper are available at https://github.com/SeHellmann/dynConfiR_Paper.

Sequential sampling confidence models

In this section, we describe the sequential sampling models included in `dynConfiR` in detail. Most dynamic models of decision-making share the idea that a decision is not based on a single sample, as in SDT. In contrast, sequential sampling models describe decisions as processes in which evidence is repeatedly sampled and accumulated over time (Ratcliff & Smith, 2004). Starting from a discrete-time perspective and normally distributed samples, reducing the time step size leads to a continuous Wiener process describing the accumulation of evidence. The use of Wiener processes is also in accordance with the idea that the internal evidence signal is represented by a sufficiently large set of neurons. The Wiener process may be interpreted as the stochastic process equivalent to the Gaussian distribution. This is because the functional invariance principle states that accordingly scaled partial sum processes, which formalize the idea of sequentially sampling and integrating evidence mathematically, converge to a Wiener process in the limit of small time steps (Klenke, 2013). However, there are dynamical models of decision-making that explicitly use other processes, like the Poisson counter model (LaBerge, 1994) or the leaky competing accumulator model (Usher & McClelland, 2001). The accumulation of evidence continues until enough information in favor of one alternative is available, formalized by a stopping criterion. When

the stopping criterion is met, a choice is triggered for the alternative favored by the accumulated evidence. All models included in **dynConfiR** assume that stopping criteria take the form of time-constant absorbing boundaries. Alternative models that suggest collapsing boundaries, which approach the starting point over time (Drugowitsch et al., 2012; Milosavljevic et al., 2010; Tajima et al., 2016), are not included in the package. Sequential sampling models of decision-making provide explanations for the correlation between discriminability and reaction time and the speed-accuracy trade-off (Lerche & Voss, 2019; Ratcliff & Rouder, 1998). The dynConfiR package features two classes of dynamical confidence models. The first class of models is based on the DDM, which assumes a single accumulation process representing evidence in favor of one choice alternative over the other. The second class of models is race models, which assumes multiple accumulation processes, each representing one choice alternative. Among the race models, the MTLNR is treated in an independent section because it does not use Wiener processes but simplifies the accumulation process to a ballistic accumulation without noise (Brown & Heathcote, 2005). In the sections that follow, we first describe the most general model of the first class, the dynamical visibility, time, and evidence model (dynaViTE), and how the other models of this class, DDConf, 2DSD, and dynWEV, are special cases of dynaViTE. We will then describe the second class of models.

Drift diffusion-based confidence models

We first present the decision mechanism that is the basis for the first class of models. In the DDM, the decision process is described as a Wiener process, which is bounded by two time-constant thresholds 0 and a. The process X starts at the starting point $X(0)$. The relative starting position between the two thresholds, i.e. $X(0)/a$, follows a uniform distribution around the parameter z with range s_z, formally $X(0)/a \sim \text{Unif}[z - s_z/2, z + s_z/2]$. The process then evolves with a drift of μ, which is normally distributed around ν with standard deviation s_ν. The diffusion constant is denoted as s. When the process first hits either the lower or the upper threshold, a decision is triggered. Decision time is thus defined as

$$T_{Dec} = min\{t|X(t) \in \{0, a\}\}.$$

The choice response R is -1 if the lower threshold was hit, i.e., if $X(T_{Dec}) = 0$, and it is +1, otherwise. Correspondingly, in discrimination tasks, the sign of the mean drift rate reflects the true stimulus identity, $S = \text{sig}(\nu)$, while its magnitude is determined by experimental manipulations in task difficulty (see section Fitting confidence models to experimental data).

Dynamical visibility, time, end evidence model (dynaViTE)

The dynaViTE model assumes that the decision process X continues after it reaches one of the two thresholds. This accumulation continues for a fixed period of time, which is represented by the parameter τ. In addition, dynaViTE postulates a second process evolving in parallel to the decision

process. The second process is denoted as visibility process V and is again a Wiener process, which always starts as 0. Its drift rate is subject to noise similar to the drift rate in the decision process. More precisely, the visibility drift is normally distributed with mean visibility drift μ_V and standard deviation σ_V. The diffusion constant of the visibility process is represented by the parameter s_V. Importantly, only one parameter in dynaViTE can be fixed to scale the other parameters without affecting model predictions. That is, if the diffusion constant of the decision process s is fixed, then s_V cannot be fixed as well without restricting the model. The psychological interpretation of the two processes is as follows: While the decision process accumulates evidence about the identity of the stimulus, i.e., whether it belongs to the class representing the upper or lower threshold, respectively, the visibility process accrues evidence about stimulus features that are indicative of task difficulty but not informative for the stimulus identity. In visual discrimination tasks – for which dynaViTE was originally proposed – visibility may be task-irrelevant stimulus features like brightness, shape, presentation time, or contour.

Confidence is then a function of the accumulated decision evidence ($X(T_{Dec}+\tau)-az$), visibility evidence ($V(T_{Dec}+\tau)$), and accumulation time ($T_{Dec}+\tau$). After the post-decisional accumulation period, accumulated evidence in the two processes is combined in a weighted sum and divided by a power of accumulation time to form an internal confidence variable

$$c_{dynaViTE} = \frac{wR(X(T_{Dec}+\tau)-az)+(1-w)V(T_{Dec}+\tau)}{(T_{Dec}+\tau)^{\lambda}}, \tag{1}$$

in which the parameter w controls the weight on decision evidence compared to visibility evidence, and λ controls the penalty of accumulation time on confidence. The factor R in the numerator of eqn (1) leads to a positive scaling of choice congruent evidence in the case when the choice is $R=-1$ (i.e., the lower threshold was hit first) because more negative values of $X(T_{Dec}-\tau)-az$ support a 'lower' decision and thus should lead to higher confidence. For perceptual decision tasks without independent manipulation of discriminability and visibility, the mean drift rate of the visibility process was previously set to the absolute mean drift of the decision process, $\mu_V=|\nu|$ (Hellmann et al., 2023, 2024). Setting the visibility drift rate to the absolute value of the decision drift rate follows the assumption that stimuli, which are easier to discriminate, are also perceived as more reliable, independent of their category. For example, when manipulating the stimulus-onset-asynchrony in a masked discrimination task, the time the stimulus was present on the screen may be perceived independently of the evidence about the stimulus category. Because stimuli that are presented longer are easier to discriminate, a longer stimulus duration increases confidence irrespective of the choice. A detailed derivation of the confidence variable in dynaViTE and justification of the form of time penalization is provided by Hellmann et al. (2024). DynaViTE includes simpler confidence models that were previously studied in the literature. The following special cases are

implemented with their own name in `dynConfiR`.

Dynamical weighted evidence and visibility model (dynWEV)

The dynWEV model is a dynamical version of a previously proposed static model of confidence, the weighted evidence and visibility model (Rausch et al., 2018). DynWEV is equivalent to the dynaViTE model without considering the accumulation time penalization in the confidence measure, i.e., $\lambda = 0$, such that

$$c_{dynWEV} = wR(X(T_{Dec} + \tau) - az) + (1 - w)V(T_{Dec} + \tau).$$

Two-stage dynamic signal detection theory (2DSD)

The 2DSD model does not assume parallel accumulation of visibility evidence and also has no penalization for accumulation time. In 2DSD, confidence only depends on whether the evidence accumulated in the post-decisional accumulation period supports or contradicts the choice (Pleskac & Busemeyer, 2010). 2DSD is a special case of dynaViTE for $\lambda = 0$ and $w = 1$. Setting $w = 1$ leads to a zero weight on the visibility process, which is thus completely ignored in the likelihood. In addition, $\lambda = 0$ implies that the denominator in eqn (1) is always 1, and accumulation time has no direct influence on confidence. The confidence variable has the form

$$c_{2DSD} = R(X(T_{Dec} + \tau) - az).$$

Drift diffusion confidence model (DDConf)

The drift diffusion confidence model is based on the the formula for optimal confidence in the DDM when drift rates are uniformly distributed (Moreno-Bote, 2010), which indicates that confidence is a monotonically decreasing function of decision time. More precisely, the confidence variable is defined as

$$c_{DDM} = \frac{1}{\sqrt{T_{Dec}}}.$$

DDConf is mathematically equivalent to the dynaViTE model with $w = 1$, $\tau = 0$, and $\lambda = 0.5$.

Race Models using Wiener processes

The drift diffusion model that serves as the basis for the previously described models assumes only one accumulation process representing relative evidence for competing decision alternatives. In contrast, race models include one accumulation process for each decision alternative. Each of the processes accrues information in favor of the corresponding decision alternative. These models are theoretically applicable to decision tasks with an arbitrary number of alternatives. In the binary setting, the two accumulators may be described as a two-dimensional Gaussian process (X_1, X_2) starting at $(0, 0)$, with constant drift (μ_1, μ_2) and covariance matrix $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{pmatrix}$. Similar to the diffusion constant, σ_1 and σ_2 may be set to 1 as

scaling factors. Each component of the process is bound from above by a time constant threshold A and B, respectively. A decision is triggered as soon as one of the accumulators hits its threshold. Decision time is thus defined by $T_{Dec} = \min\{t \,|\, X_1(t) > A \vee X_2(t) > B\}$ and the response R is 1, if $X_1(T_{Dec}) > A$ and 2, if $X_2(T_{Dec}) > B$.

In `dynConfiR`, the correlation parameter ρ is restricted to either $\rho = 0$, which results in the model denoted as the independent race model (IRM), or $\rho = -.5$, for which the respective model is denoted as the partially correlated race model (PCRM). The reason for restricting ρ to either 0 or -.5 is that for these values, there are closed-form solutions for the first passage time densities (Moreno-Bote, 2010). Closed-form solutions allow fast and precise computations of the first-passage time distribution compared to the usage of approximation methods, which are necessary if the first-passage time density can only be represented by an infinite sum. However, these two choices of ρ capture essential theoretical concepts (Teodorescu & Usher, 2013; Zylberberg et al., 2012). A value of $\rho = 0$ leads to an independent race of accumulators, which represents the assumption of evidence accumulation in the absence of interaction (Teodorescu & Usher, 2013; Zylberberg et al., 2012). A negative correlation of the noise in the accumulation processes represents the assumption of feed-forward inhibition, which means that higher input values in one accumulator partially reduce the input in the other accumulator (Teodorescu & Usher, 2013; Zylberberg et al., 2012). Note that for $\rho = -1$, the race model is equivalent to a drift diffusion model.

One possibility to compute confidence in the context of race models is the Balance of Evidence (BoE, Vickers et al., 1985), i.e., the difference in the amount of evidence in favor of the two alternatives at the time of decision. Because the winning accumulator is always at its threshold at decision time, BoE is entirely determined by the distance of the losing accumulator to its upper threshold. For instance, if $R = 1$, the confidence variable may thus be defined as

$$c_{BoE} = B - X_2(T_{Dec}).$$

The logic behind the BoE is intuitive: the less evidence there was for the non-chosen alternative, the clearer and less ambiguous the decision resulting in a higher degree of confidence associated with the decision. However, empirical studies have shown that confidence is also affected by decision time (Hellmann et al., 2023; Kiani et al., 2014). In addition, it has been shown that if confidence in a race model was computed optimally, it would be a function of both BoE and decision time (Moreno-Bote, 2010). For this reason, `dynConfiR` includes race models with a more general confidence variable in the form of a linear combination of Balance of Evidence and the inverse of decision time. Assuming again that $R = 1$, confidence is computed as

$$c_{RMt} = w_X(B - X_2(T_{Dec})) + w_{RT}\frac{1}{\sqrt{T_{Dec}}} + w_{Int}\frac{B - X_2(T_{Dec})}{\sqrt{T_{Dec}}},$$

where the weights w_X, w_{RT}, w_{Int} are greater than 0 and sum to 1 to form a trade-off between the possible predictor variables. Note that the fixed sum of weight parameters is not a restriction of the model because the confidence and confidence thresholds may be rescaled by the sum of weights to produce the same distribution of response time and confidence. `dynConfiR` implements race models with all combinations of assumptions about independent or correlated accumulators and a confidence variable that does or does not depend on decision time. The acronyms for the models used in the package are summarized in Table 2. Note that the first confidence measure c_{BoE} is a special case of the more general c_{RMt}, if the weight parameters are set accordingly ($w_X = 1, w_{RT} = w_{Int} = 0$).

Table 2

Acronyms for the different variations of race models used in `dynConfiR`*.*

		Confidence variable	
		c_{BoE}	c_{RMt}
Correlation	0	IRM	IRMt
of noise (ρ)	-.5	PCRM	PCRMt

The multiple-threshold log-normal race model

The MTLNR model (Reynolds et al., 2020) forms a special case of racing accumulator models because it uses a simple ballistic accumulation instead of a noisy, Gaussian process (Brown & Heathcote, 2005). Variation in responses and response times are the results of variation in accumulation rates and boundary distances, which are both assumed to follow a log-normal distribution.

In detail, MTLNR assume log-normally distributed boundary distances for the two accumulators (D_1, D_2). The distribution is described by their logarithms, $(\log D_1, \log D_2)$, which follow a normal distribution with mean (μ_{d1}, μ_{d2}) and covariance matrix $\Sigma_d = \begin{pmatrix} \sigma_{d1}^2 & \sigma_{d1}\sigma_{d2}\rho_d \\ \sigma_{d1}\sigma_{d2}\rho_d & \sigma_{d2}^2 \end{pmatrix}$. Similarly, the accumulation rates (V_1, V_2) follow a log-normal distribution with mean parameter (μ_{v1}, μ_{v2}) and covariance matrix $\Sigma_v = \begin{pmatrix} \sigma_{v1}^2 & \sigma_{v1}\sigma_{v2}\rho_v \\ \sigma_{v1}\sigma_{v2}\rho_v & \sigma_{v2}^2 \end{pmatrix}$ for the underlying normal distribution on the log-rates.

Because of the linear ballistic nature of evidence accumulation, the boundary crossing times for each accumulator is determined by boundary distance and accumulation rate $T_i = D_i/V_i$. The first accumulator to hit its boundary determines the decision and decision time, similar to the race models with Wiener processes.

The confidence variable in the MTLNR is defined as the logarithm of the ratio between the boundary crossing time of the loosing accumulator over the boundary crossing time of the winning

accumulator. For instance, if the first accumulator hits its boundary first, confidence is computed by

$$c_{MTLNR} = \log \frac{T_2}{T_1}.$$

This will always lead to values greater than 0.

Common mechanism in forming confidence judgments

Although models may differ in their decision architecture and the specific computation for the confidence variable, the mechanism for the formation of a confidence report, as implemented in the package, is the same for all models. All models are built to produce discrete confidence outcomes C with an arbitrary number of levels $K \geq 2$. To generate discrete confidence reports, the models assume that some internal confidence variable c is generated according to the formula for the respective model specified above. Confidence judgments are then determined by comparing the internal confidence variable to a set of thresholds, $\theta_{R,i}, i \in \{1, ...K-1\}$, depending on the choice R. Formally, the reported confidence is

$$C = \sum_{i=1}^{K-1} \mathbb{1}_{(c>\theta_{R,i})} + 1,$$

where $\mathbb{1}$ denotes the indicator function, which is one if the condition is true and zero, otherwise. This means that observers are assumed to report a confidence level of 2 on a three-point scale if the confidence variable c falls between $\theta_{R,1}$ and $\theta_{R,2}$.

Non-decision time component

Similarly, all models share the assumption of a non-decision time component, which includes time for stimulus encoding and the formation of a motor response. The non-decision time component is not related to the decision mechanics itself but contributes to the observed response times. It is modeled as a uniformly distributed component $T_{ND} \sim \text{Unif}[t_0, t_0 + s_{t0}]$ (Ratcliff & Tuerlinckx, 2002). The formula for the response time depends on the timing of the confidence report in the experiment at hand. For experiments in which the choice and confidence judgment were reported sequentially, the models currently implemented in the package can only account for the choice response time and do not include the confidence response time. The choice response time is assumed to be

$$RT = T_{Dec} + T_{ND}. \tag{2}$$

Application of models with post-decisional accumulation to data from simultaneous choice and confidence reports

If choice and confidence are reported simultaneously, then the response time is still defined as in eqn (2) for models that do not assume post-decisional accumulation of evidence. For models that assume a post-decisional accumulation period, all processes are assumed to have finished at the time of the response.

Thus, the observed response time is the sum of decision time, post-decisional accumulation time, and non-decision time,

$$RT = T_{Dec} + \tau + T_{ND}. \tag{3}$$

However, the models with post-decisional accumulation time could also be fitted in experiments with simultaneous responses using the first definition of response time as in eqn (2) using the same fitting functions with specific arguments (see section Fitting confidence models to experimental data).

All parameters of the different models are summarized in Table 3.

Other sequential sampling models of confidence

The confidence models presented here are only a subset of previously proposed dynamical confidence models. Other models include the RTCON model (Ratcliff & Starns, 2009, 2013) or the bounded accumulation model proposed by Kiani et al. (2014). The `dynConfiR` package is restricted to models for which closed-form solutions are available for the joint distribution of response times and confidence, or the approximations of response time distributions are well-studied concerning their precision.

Table 3

List and short description of all parameters for the different models.

Parameter	Description
dynaViTE	
a	distance between upper and lower decision boundary for decision process
z	relative mean starting point of decision process
s_z	range of uniform distribution for the relative starting point in the decision process
ν	mean drift rate for decision process
s	diffusion constant of the decision process
s_ν	variation in drift rate of the decision process
τ	length of inter-rating period
μ_V	mean drift rate for the visibility process
s_V	diffusion constant of the visibility process
σ_V	variation in drift rate of visibility process
w	weight on decision evidence for confidence variable
λ	exponent of accumulation time in the denominator of the confidence variable
Race Models	
A, B	thresholds for the two accumulation processes
μ_1, μ_2	drift rates for the two accumulators
s_1, s_2	diffusion constants for the two accumulators
ρ	correlation of process noise between the two accumulators (either 0 for IRM and IRMt or -.5 for PCRM and PCRMt)
w_X, w_{RT}, and w_{Int}	weights on loosing accumulator, decision time and interaction for the confidence variable
MTLNR	
μ_{d1}, μ_{d2}	mean parameters for boundary distances of the two accumulators
μ_{v1}, μ_{v2}	mean parameters for accumulation rates of the two accumulators
σ_{d1}, σ_{d2}	variance parameters for boundary distances of the two accumulators
σ_{v1}, σ_{v2}	variance parameters for accumulation rates of the two accumulators
ρ_d	correlation between boundary distances
ρ_v	correlation between accumulation rates
Common parameters	
t_0	minimal non-decision time component
s_{t_0}	range of uniform distribution for non-decision time component
$\theta_{R,k}$	set of confidence criteria, $R = -1, 1$, $k = 1, ..., K-1$ for discretization into K steps

Functionalities of the package

In the following, we will first describe a prototypical workflow illustrating how the package may be used for model comparison studies. Afterward, the most essential functions implemented in **dynConfiR** are explained in detail.

Installation

The package is available on CRAN and may be installed with the command:

```
install.packages("dynConfiR")
```

A development version of the package is available on GitHub and may be downloaded and installed using the `devtools` package and the command:

```
devtools::install_github("SeHellmann/dynConfiR")
```

Workflow

The **dynConfiR** package provides functions for model fitting, i.e., estimation of model parameters. The function `fitRTConfModels` implements the full model fitting procedure and allows for parallelization over subjects.

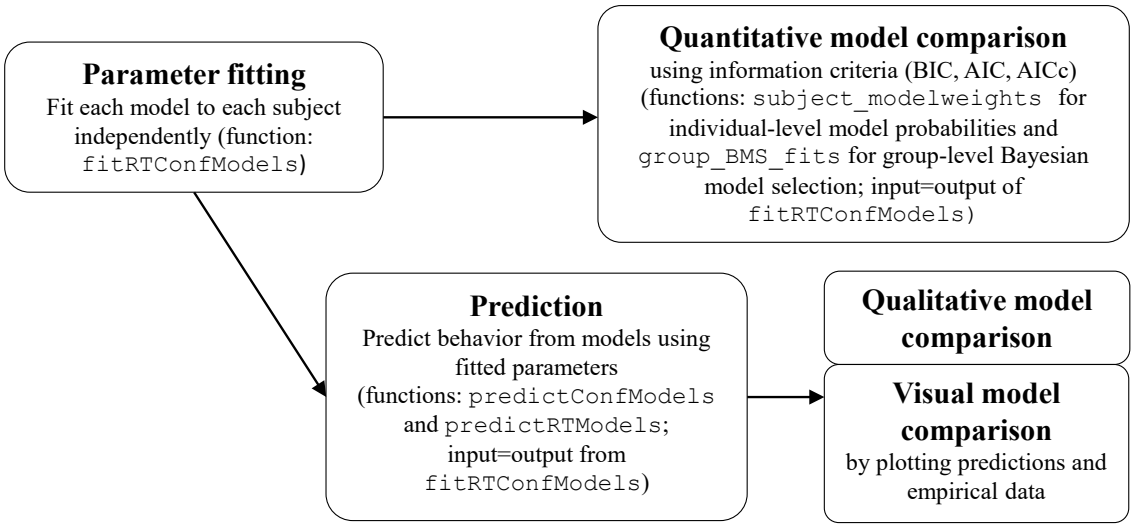
The functions of **dynConfiR** are optimized for within-subjects manipulations of discriminability. Models can be fitted independently for each subject, facilitating quantitative model comparison using information criteria like AIC and BIC (Akaike, 1974; Schwarz, 1978). Firstly, since models are fitted to individual participants, computing model weights on a subject-level, which is done by the function `subject_modelweights`, may be used to inspect models that are prominent in the group, but also to examine heterogeneity or a group-structure with respect to the best-fitting models in the data. In many cases, an conclusion on the group-level is desired. For this purpose, assuming that BIC or AIC are good approximations of model evidence, **dynConfiR** provides the function `group_BMS_fit` to conduct a group-level model comparison based on a Bayesian model selection approaches (Rigoux et al., 2014).

Cognitive modeling studies should check whether the fitted models could reproduce the main qualitative patterns of empirical data (Palminteri et al., 2017). For this purpose, the functions `predictConfModels` and `predictRTModels` compute the predicted data distributions for given parameter sets. While `predictConfModels` computes the discrete decision and confidence outcomes, `predictRTModels` provides the density for the joint distribution of decision, response time, and confidence rating. When used with previously fitted parameters, these functions can be used to visually compare the model predictions to empirical data or to check for the reproduction of qualitative data patterns. One example of a qualitative pattern that confidence models need to explain is the relationship of mean

confidence in discrimination tasks with increasing stimulus discriminability for correct and incorrect decisions, which have been referred to as a so-called folded-X or a double increase pattern (Rausch & Zehetleitner, 2019). The workflow for parameter fitting, model comparison, and prediction is summarized in Figure 1.A step-by-step tutorial on how to use the package in a modeling study is available on the GitHub page of the package and downloadable as a R markdown document at OSF. We also illustrate the workflow in section Example of Application in Model Comparison. In addition to classical model comparison studies, the fitted parameters for the individual subjects can be used for group comparisons and correlational analyses, for example, to study the relationship of specific parameters with measures of metacognitive sensitivity or neurological data.

Figure 1

Basic workflow and functions for model comparison studies.



Fitting confidence models to experimental data

In this section, we describe the fitting function in more detail, starting with which experimental data can be used, how parameters are mapped to the manipulations, and finally, the fitting procedure.

Experimental paradigm

The fitting functions in the package are tailored to perceptual, binary discrimination tasks with a single difficulty manipulation. This is a standard paradigm in the study of computational models of confidence (Kiani et al., 2014; Rahnev et al., 2020; Rausch et al., 2018). For other manipulations that are assumed to vary specific parameters only, the user can write their own likelihood and fitting functions using the density functions described later. For instance, manipulations of the speed-accuracy trade-off

(Desender et al., 2021) and post-decisional accumulation period (Desender et al., 2022; Moran et al., 2015) may also be interesting when studying the formation of confidence. Still, when all model parameters are allowed to vary across conditions, the `fitRTConfModels` function remains applicable. In such instances, the user can specify a data column as the subject identifier, which differentiates between combinations of subjects and manipulation levels such that the fitting function fits separate sets of parameters per subject and condition. Fitting independent parameter sets for each condition can be useful for critically testing the assumption that an experimental manipulation selectively influences specific parameters (Lerche & Voss, 2019; Voss et al., 2004).

Data format

The fitting function expects the data to come in a tidy data frame, with each row representing one trial. The data frame should include the following columns (expected column names in parentheses): true stimulus identity (`stimulus`), binary decision response (`response`), categorical confidence judgment (`rating`), and response time (`rt`). As an alternative to the stimulus or response column, a column for accuracy (`correct`) may be provided. In addition, a column for the experimental manipulation of discriminability of the stimulus (`condition`) may be included but is not necessary. Instead of renaming columns in the data frame, alternative column names may be added as arguments of the form `rating = "confidence"`, if, for example, the column indicating the confidence `rating` is called `confidence`. A column named `sbj`, `subject`, or `participant` may be included to fit the models independently to individual subjects.

Fitted parameters

The stimulus and response categories are denoted by $S, R \in \{-1, 1\}$, and task difficulty is assumed to be manipulated in L steps. A discrimination parameter, $d_l, l = 1, \ldots, L$, is fitted independently for each difficulty level. Thus, the condition column will be transformed into a factor, even when numeric values are supplied. In the confidence models, the drift rates of the different processes depend on the stimulus identity and the discriminability parameter of the difficulty level of the trial: For dynaViTE, the mean drift rate of the decision process is set to $\nu = Sd_l$, and the mean drift rate of the visibility process is set to $\mu_V = d_l$. For the race models, the drift rates are set to $(\mu_1, \mu_2) = (Sd_l, -Sd_l)$. This means that the first accumulator accumulates evidence for the first category, while the second one accumulates evidence for the category $S = -1$. For MTLNR, the mean parameters for the accumulation rates are set to $(\mu_{v1}, \mu_{v2}) = (d_l, 0)$, if $S = 1$ and to $(\mu_{v1}, \mu_{v2}) = (0, d_l)$, if $S = -1$.

All other parameters are assumed to be independent of the task difficulty manipulation and fitted for each subject. However, the diffusion constant of the decision process in dynaViTE s and the diffusion

constants of the two processes in the race models, σ_1 and σ_2, are fixed to 1 because other parameters may be scaled accordingly to produce the same likelihoods. This is a common approach in response time modeling (Lerche & Voss, 2016; Ratcliff & Rouder, 1998). In the MTLNR, the variance parameters for the boundary distances σ_{d1} and σ_{d2} and their correlation ρ_d are fixed to 0, such that only the variance parameters for the accumulation rates are fitted. Therefore, the variance component are simply labeled s_1 and s_2 for the variance parameters of the boundary hitting times of the first and second accumulator, and ρ for their correlation. This is in accordance with the implementation of the MTLNR in Reynolds et al. (2020). Importantly, when we implemented the model with all parameters fitted freely, we observed that the variance parameters could not be well-recovered and also model recovery was worse.

Whether choice and confidence were reported simultaneously or sequentially is determined by the `simult_conf` argument, which should be set to `TRUE` if the reports were given simultaneously and `FALSE` otherwise.

The number of confidence thresholds $\theta_{R,k}, k = 1, \ldots, K-1$ separating the internal confidence variable into discrete steps depends on the number of possible levels for the discrete confidence rating K. The confidence thresholds can vary between choice responses by default, leading to $2(K-1)$ fitted confidence threshold parameters. We recommend specifying the `nRatings` argument to provide the number of confidence levels because not every subject might have used the full range of the scale. Alternatively, the ratings column can be provided as a factor with factor levels representing the possible rating outcomes. If not all confidence levels were used, the number of fitted parameters is reduced internally because the maximum likelihood is attained by some thresholds being identical in this case. If the lowest (or highest) confidence level was not used, then the likelihood is maximized by setting the lowest confidence threshold to minus infinity (or the highest threshold to infinity). If an intermediate confidence category was not used, then the likelihood is maximized by two confidence thresholds being identical. Therefore, the concerned confidence thresholds do not need to be optimized numerically by the optimization procedure but may be set afterward to speed up the optimization. The `nRatings` argument is required to correctly format the output parameters and report the right number of fitted parameters.

To sum up, the total number of parameters depends on the number of steps in the manipulation L and the number of levels for the discrete confidence rating, K. This means that for dynaViTE, there are $11 + L + 2(K-1)$ parameters. In the race models with a time-dependent confidence variable, there are $6 + L + 2(K-1)$ parameters. Two of three weight parameters have to be fitted, while the third weight is determined by the sum of the weights being 1. In addition, the correlation parameter ρ is not estimated but fixed at -0.5 for PCRMt and 0 for IRMt. For MTLNR, the number of fitted parameters is equal to $7 + L + 2(K-1)$. For special cases like 2DSD or race models with a time-independent confidence variable,

the number of parameters is reduced by the number of fixed parameters. In addition, if the confidence thresholds are assumed to be symmetric for the two response options (by setting `fixed=list(sym_thetas=TRUE)`), the number of parameters is reduced by $K-1$.

Which models should be fitted is specified by the `models` argument. The function `fitRTConfModels` allows for all models presented in the section Sequential sampling confidence models: dynaViTE, dynWEV, 2DSD, DDConf, IRMt, PCRMt, IRM, and PCRM. In addition, the user may fix individual parameters by providing the argument `fixed` in the form of a list. For instance, researcher may want to assume an unbiased observer by setting $z = 0.5$ for the drift diffusion-based models and $A = B$ for the race models. Moreover, specifying `sym_thetas=TRUE` in the list leads to symmetric confidence thresholds for the two choice possibilities, i.e., $\theta_{1,k} = \theta_{-1,k} \forall k = 1, \ldots, K-1$.

Fitting procedure

The function `fitRTConfModels` fits the models specified in the `models` argument to each individual subject in the data set using maximum likelihood estimation, i.e., by minimizing the negative log-likelihood of model parameters. The likelihood is computed under the assumption of independent observations, which means that for trials $i = 1, ..., N$; the vectors for presented stimulus identity S and task difficulty D; and the vectors for observed outcomes response time RT, confidence rating C, and response R, the negative log-likelihood of a set of parameters ϑ is computed as

$$\mathcal{L}(RT, C, R|\vartheta, S, D) = -\sum_{i=1}^{N} \log \mathbb{P}(RT_i, C_i, R_i|\vartheta, S_i, D_i).$$

The optimization procedure starts with a grid search, in which the likelihood is computed for a broad range of possible parameter combinations. The best-performing parameter sets identified in the initial grid search are used as starting values for the optimization algorithm. The optimization algorithm is restarted several times with the previous run's output as the next run's starting point to allow the optimization to avoid local minima. The number of initial values and restarts for the optimization procedure can be set by the user using the `opts` argument. The functions offer the possibility of parallelization over both subject-model combinations and within one fitting procedure over the starting values for the optimization. The output is a data frame with one row for each combination of subject and fitted model. The columns of the output data frame are the fitted model parameters together with additional information, like the number of trials (`N`), fixed parameters (`fixed`) and the following performance measures: the final negative log-likelihood and model selection criteria AIC, AICc, and BIC.

On the one hand, the maximum-likelihood fitting procedure implemented in `fitRTConfModels` is an efficient way for estimating parameters using all the available information in the data without aggregating to quantiles (Lerche et al., 2017; Voss et al., 2013). On the other hand, the maximum likelihood

method is known to be influenced by contaminant response times, which are not generated by a DDM (Lerche et al., 2017; Ratcliff & Tuerlinckx, 2002). Therefore, it is recommended to apply a filter on trials at the level of the individual subject, e.g., by removing trials with response times that are either below a certain threshold (e.g., 300 ms) or which deviate significantly from the mean or median of the response time distribution (e.g., response times, which exceed the mean plus two standard deviations; Hellmann et al., 2023; Pleskac & Busemeyer, 2010). There are different strategies for removing contaminants in the data. Some studies use hard cut-offs (Lerche & Voss, 2017; Ratcliff et al., 2004; van den Berg et al., 2016), others use exclusion criteria based on the interquartile range (Lerche & Voss, 2019; Voss et al., 2013) or alternatively, a mixture of hard cut-off for the fast responses and a distribution-dependent cut-off for slow responses (Hellmann et al., 2023, 2024; Lerche et al., 2017; Moran et al., 2015; Pleskac & Busemeyer, 2010). However, it is hard to suggest general guidelines for exclusion criteria that suit all experiments. For new experiments, we recommend using pilot data to infer suitable exclusion criteria because different experimental paradigms produce different response time distributions. In light of ongoing replication issues in psychology (Röseler et al., 2024), we recommend to pre-register exclusion criteria and check whether results are robust concerning the specific choice of exclusion criteria (Wagenmakers et al., 2012).

Predicting confidence and response time distributions

The empirical data is often compared visually to model predictions to check for qualitative mismatches. For this purpose, `dynConfiR` includes the functions `predictConfModels` (for the discrete decision and confidence distribution) and `predictRTModels` (for the joint distribution of response time, decision and confidence). These take data frames with parameters as input. Notably, the output of the fitting procedure may be inserted directly into the prediction functions. `predictConfModels` returns a data frame with columns for stimulus identity, response, and confidence judgments and a column indicating the probability of an outcome. `predictRTModels` has an additional column for the response time, spanned equidistantly for a user-provided interval. If the input has more than one row, columns for subject ID and model are required, and the output will be accordingly structured by binding the data frame outputs for each subject and model combination one below the other. Similarly to the fitting function, `participant`, `subject`, and `sbj` column names are accepted as identifier.

Quantitative model comparison using information criteria

Beside the qualitative and visual inspection of model fits, a quantitative model comparison and model selection is often conducted to compare models when qualitative data patterns are not diagnostic (Farrell & Lewandowsky, 2018). The information criteria BIC, AIC, and AICc, are used to approximate the negative log-model evidence, such that Bayesian approaches for model comparison may be used. On a

subject level, these information criteria can be directly transformed into model weights to assess the prevalence of certain models, but also heterogeneity within the sample. This is done by the function `subject_modelweights`, which takes a data frame with columns for model names, subject identifier (one of `participant`, `subject`, or `sbj`), and the information criterion that should be used as input. The second argument `measure` gives the name of the column with the information criterion and is "BIC" by default. Therefore, the output of a `fitRTConfModels` call may be directly be used as argument for the function `subject_modelweights`.

Similar to the subject-level model comparison, the function `group_BMS_fits` takes the result of model fitting but performs a group-level Bayesian model selection based on a random effects model of model prevalence across subjects (Rigoux et al., 2014; Stephan et al., 2009). The random effects model assumes a Dirichlet distribution for model probabilities in the population, for which the α parameter is estimated based on Variational Bayesian approach and algorithm described in Stephan et al. (2009). The estimated parameter may be used to calculate models' exceedance probabilities. The exceedance probability of a model is defined as probability that the model has a higher probability compared to all other models given the Dirichlet distribution. The function also provides a scaled version of the exceedance probabilities, the protected exceedance probabilities (PEP). PEP controls for the Bayesian omnibus risk (BOR), which quantifies the risk of assuming the random effects model in contrast to a null-model, in which all models have always the same prior probability, i.e. the limit of a Dirichlet model with an α parameter with equal components that approach infinity. The estimation of BOR is based on a Variational approach and the implementation is based on the VBA toolbox for Matlab (Daunizeau et al., 2014).

In addition to the protected exceedance probability, and exceedance probability, the function `group_BMS_fits` calculates the model probabilities based on a fixed effect model, that assumes that there is a single-best model in the population. The latter is equivalent in calculating model weights based on the sum of BIC values across all participants. By default, model comparison is performed using the BIC. Using the second argument, `measure`, one could base the comparison also on the AIC or AICc in both functions. We decided to use the BIC as the default as it is the most conservative criterion for the number of data points that are usually used in studies modeling response times.

Other functions

Probability density functions

```
ddynaViTE(response, rt, th1, th2, a, v, t0, ...)
dPCRM(response, rt, th1, th2, mu1, mu2, a, b,...)
```

The implemented confidence and response time distributions form the basis for model fitting and

predictions. The distributions are implemented as probability densities in C++ and accessed in R using `Rcpp`. The usage of the different density functions is very similar. The first arguments represent the outcome variables: response time (RT), the binary choice (R), and the interval for the confidence variable (θ_1 and θ_2). The density functions return the probability $\mathbb{P}_{\text{model}}(RT, R, c_{model} \in [\theta_1, \theta_2] | \vartheta)$. The model parameters are passed as additional, individual parameters. Note that θ_1 and θ_2 are also parameters usually estimated during model fitting. The densities for drift diffusion-based models are approximated using the truncated series for the density of the drift diffusion model (see Navarro & Fuss, 2009; Voss et al., 2004). The densities for the race models are implemented according to the formulas in Moreno-Bote (2010), which are derived using the methods of images for the stochastic differential equation. The integration over the distribution of starting points and non-decision time components is conducted numerically using a rectangular approximation with equidistant steps (see Precision analysis section). The density functions may be used for theoretical calculations and to implement other model fitting algorithms instead of the maximum likelihood estimation procedure included in **`dynConfiR`**.

Log-likelihood functions

```
LogLikWEV(data, paramDf, model = "dynaViTE", simult_conf = FALSE,...)
LogLikRM(data, paramDf, model = "IRM", time_scaled = FALSE,...)
```

There are also functions for calculating the log-likelihood of a data set given some parameters for each model. The two main arguments are `data`, a data frame of the empirical data with the stimulus, response, response time and confidence, and `paramDf`, a data frame with one row and columns for the required parameters of the chosen model. The log-likelihood function is included in **`dynConfiR`** mainly to allow for the investigation of the impact of experimental manipulations on specific parameters or other relationships between stimulus discriminability and mean drift rate. For example, previous studies assumed a power function for the relationship between physical stimulus intensity and internal signal strength (Ratcliff et al., 2018; Teodorescu et al., 2016) instead of fitting discriminability parameters for each level of the experimental manipulation. The likelihood functions are wrappers of the density functions that can be used easily in custom built cost-functions for optimization.

Simulation functions

```
rdynaViTE(n, a, v, t0 = 0, z = 0.5, d = 0, sz = 0, sv = 0, st0 = 0, tau = 1, w =
  0.5, ...)
simulateRTConf(paramDf, n = 10000, model = NULL, gamma = FALSE, agg_simus = FALSE
  ,...)
```

The package includes low-level and high-level simulation functions. Because simulation is based on a discretization of the stochastic differential equation, there is an argument `delta` determining the step-size

of the discretization and an argument `maxrt`, which determines the maximal simulated decision time. The simulation of a single trial is stopped when the stopping criterion has not been met and the maximum decision time has been exceeded. When the simulation is stopped without a choice, a response of 0 is returned. First, the low-level simulation functions are similar to the simulation functions of other probability distributions in R , e.g., `rdynaViTE` and `rPCRMt`, with arguments for each parameter, most of them having default values.

A high-level function for simulating data with fitted parameters is also available. The function `simulateRTConf` takes a data frame with one row and columns for the required parameters. The high-level function simulates `n` trials per stimulus identity (which stimulus identity is used for the simulation may be changed with the `stimulus` argument) and difficulty condition. The number of difficulty levels is determined by the number of drift rates in the `paramDf` argument. To simplify the application to several parameter sets and models, the model argument can be given as a column in the `paramDf` argument. In addition, `simulateRTConf` offers the possibility to aggregate the output over response times, i.e., reporting only the discrete outcomes of choice and confidence. Finally, when `gamma=TRUE`, the function computes Kruskal's Gamma (Nelson, 1984) between confidence and several other relevant variables, e.g., between confidence and accuracy for different levels of stimulus discriminability. If `gamma=TRUE` is used, the output is a list with two components: `simus` for the data frame with the actual simulated data and `gamma` with several data frames for different Gamma correlations.

Example of Application in Model Comparison

Now, we present a complete example of an analysis including a model comparison. The data set for this demonstration was generously published by Law & Lee (Ng et al., 2021) and was downloaded from the confidence database (Rahnev et al., 2020). The data set is available at https://osf.io/vgr27.

Experimental method

The study was initially conducted to investigate serial dependence in confidence judgments using random-dot kinematograms. 16 participants reported their perceived motion direction, which was either leftwards or rightwards, simultaneously with their confidence using the keyboard. Confidence was reported on a 4-point scale.

Task difficulty was manipulated by varying motion coherence. In a 240-trial calibration phase, coherence values for target accuracy levels of .52, .65, and .78 were determined using a staircase technique. The resulting coherence values were then used in the experimental phase for coherence levels 1, 3, and 5, respectively, while the coherence values for the second and fourth levels of the manipulation were determined by averaging the values for the first and third level and the third and fifth level, respectively.

The experimental trials consisted of 20 blocks with 60 trials each. Because of the primary aim of the study, trials with medium difficulty, i.e., a coherence level of 3, were always preceded by either one or two trials with either high (level 1) or low (level 5) difficulty. This trial-by-trial dependency will be ignored in the following analysis.

Participants did not receive trial-by-trial feedback but instead received feedback at the end of each block about both their overall accuracy and their accuracy in the previous block.

Data

After downloading the data from the confidence database, we selected the relevant columns, converted their names to lowercase, and removed the calibration trials from the data set. We then renamed the columns for the subject ID to `participant` (note that the functions accept the column names `subject` and `sbj` as well) and response times to `rt`. As confidence was measured on a 4-point scale, we did not have to bin the rating response. The resulting data set has the following form.

```
> head(Data)
  participant stimulus response confidence        rt coh_level coherence
1           1        1        2          3 2.7568960         1 0.1146677
2           1        2        2          3 0.5194411         1 0.1146677
3           1        2        2          4 0.4860291         3 0.1998495
4           1        1        2          4 3.0795050         4 0.5150529
5           1        1        2          4 0.5935152         5 0.8302563
6           1        2        1          4 0.3307259         5 0.8302563
> dim(Data)
[1] 15360     7
```

Data preprocessing

A typical step before fitting sequential sampling models is to remove possible outliers from the empirical response time distribution. Filtering individual data by response times is recommended because the maximum likelihood method, which is used by **`dynConfiR`**, is known to be specifically influenced by outliers, and there is no implementation of lapses in the current version of the package.

We removed responses that were faster than the median minus one standard deviation and slower than the mean plus 4 standard deviations for each participant, resulting in the removal of 1.7% of all trials.

In the second step, we removed one participant because they did not perform above chance (0.53 compared against 0.5), which delivered no evidence of being above chance in a Bayesian proportion test against chance level accuracy with a prior scale conducted via the `proportionsBF` function from the

BayesFactor package (Morey et al., 2024) with a prior scale parameter of 0.5.

Analyses

Model fitting

We wanted to conduct a model comparison on the data, comparing a broad range of possible models. We considered models previously compared on similar datasets from visual discrimination experiments: dynaViTE, dynWEV, 2DSD, PCRMt, and IRMt (Hellmann et al., 2023, 2024). Note that although we did not explicitly fit all models, simpler models are special cases of the models fitted in this section. For instance, race models with time-independent confidence variables, IRM and PCRM, are special cases of IRMt and PCRMt, respectively. In addition, we fitted the MTLNR model, which was not previously compared to the above models on empirical data. The first step was to fit the model parameters for each participant, which was achieved with the following command:

```
parfits <- fitRTConfModels(Data, models=c("dynaViTE", "dynWEV",
                                          "2DSD", "PCRMt", "IRMt", "MTLNR"),
                nRatings=4, restr_tau = "simult_conf",
                opts=list(nAttempts=4, nRestarts=4),
                parallel="both", n.cores = c(5, 4),
                condition = "coh_level", rating="confidence")
```

The function `fitRTConfModels` splits the data for the participant column and fits each of the models given in the respective `models` argument. Providing `nRatings=4` ensured that three confidence thresholds are fitted for participants that did not used the full range of the confidence scale. The argument `restr_tau="simult_conf"` indicates that choice and confidence responses were reported simultaneously.

The argument `opts` is optional and allows for adaptations of the fitting procedure. We used only the four best parameter sets (default: 5) from the initial grid search as starting values for the optimization routine and started the optimization algorithm with a broad trust region four times (default: 5).

With the combination of `parallel=TRUE` and `n.cores=c(5,4)`, we distributed the fitting procedure across different CPUs, fitting five participants in parallel with four cores per participant. Therefore, for each participant, the four optimization routines with the different initial parameter settings were run in parallel.

Finally, it was necessary to include column names that deviate from the default ones, which was achieved with the last two arguments.

Quantitative model comparison

With few lines of code, the first central part of the analysis was achieved. With the outputs, a quantitative model comparison may be conducted using the information criteria available in `parfits`. Information criteria for models that were fit to participants independently could be used in different ways for model comparison. Model comparison could be performed on the participant-level by calculating BIC weights independently for each participant. This approach may already indicate dominating individual models but is also able to represent heterogeneity across participants or groups of participants. Still, sometimes it is desirable to investigate the performance of models on a group-level. The results for both analysis levels are easily obtained by using the functions `subject_modelweights` respectively `group_BMS_fits` with using `parfits` as input.

```
BICweights <- subject_modelweights(parfits)
groupBMS <- group_BMS_fits(parfits)
```

In the following, we visualize only the results based on the BIC. The **`dynConfiR`** package also includes AIC and AICc in the output, which are not visualized here because the results are almost identical. On a participant-level, the dynaViTE model provided the best account for 9 out of 15 participants, while the MTLNR model provided the best account for 3 participants (Figure2, upper panel). Visualizing the average BIC values over the sample, we see that the MTLNR and therace models achieved the lowest average BIC (Figure 2, middle panel). Among the DDM-based models, dynaViTE performed best. Despite the close match between the two race models, the results of the group-level model selection according to a fixed effect model clearly favors the anti-correlated race model (dotted lines in the lower panel of Figure 2). In contrast, the random effects model decisively favors the dynaViTE model as indicated by protected exceedance probability ($pep_{dynaViTE} = .99$, Figure 2).

Appendix Figure A1 demonstrates that the average differences and thus the fixed effect model probabilities are dominated by two participants, who showed an extreme difference in BICs between race models and DDM-based models. Inspecting the Bayesian omnibus risk for the fixed effect model compared to the null model, which is indistinguishable from 1, tells us that the fixed effect model is generally a bad fit to the data because of the heterogeneity in the data. In contrast, the random effects model shows a BOR of .0016, clearly indicating a more adequate account for the data.

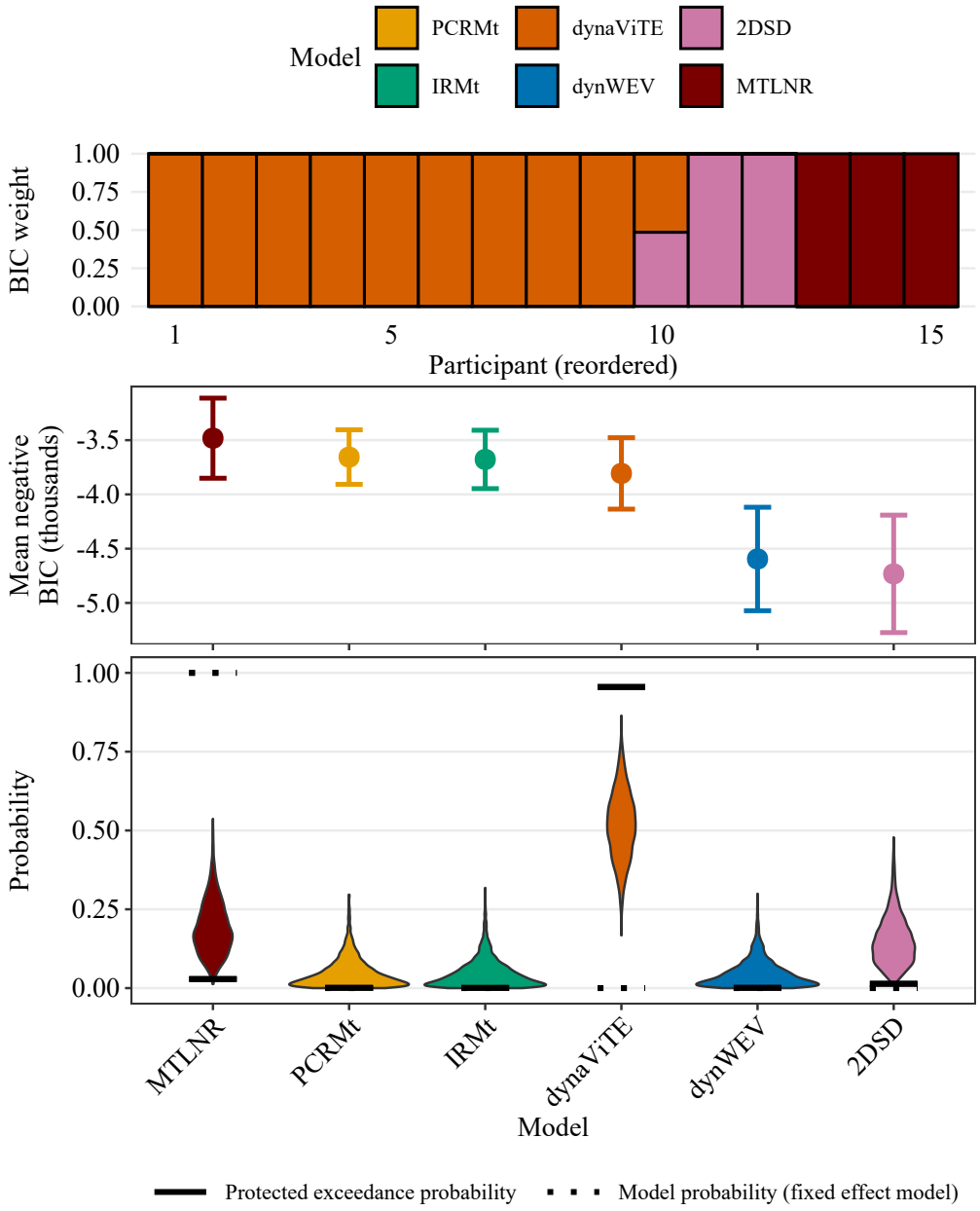


Figure 2

***Upper panel**: BIC weights across participants (reordered). **Middle panel**: Negative mean BIC values. Error bars represent within-subject standard errors. **Lower panel**: Results from a Bayesian model selection analysis. Violin plots show simulated model probabilities from a Dirichlet distribution fitted to BIC values according to a random effects model. Solid bars indicate the corresponding protected exceedance probability, dotted bars indicate the model probabilities resulting from a fixed effect model.*

Prediction and visual model fit

The output of the fitting function may be passed directly to the prediction functions, which also use parallelization over participants. The prediction functions automatically split the first data frame argument by the participant and model columns and select the respective prediction function for the model of each row.

```
prediction_ConfDist <- predictConfModels(parfits, simult_conf=TRUE,
                parallel = TRUE)
prediction_RTConfDist <- predictRTModels(parfits, simult_conf=TRUE, maxrt=6,
                scaled=TRUE, DistConf = prediction_ConfDist,
                parallel=TRUE)
```

Furthermore, visualizations may be generated with the predictions to compare model fits with empirical distributions. The output of the function `predictConfModels` has the following form (note that only the first digits are printed for readability):

```
> print(head(prediction_ConfDist), row.names=FALSE)
condition stimulus response correct rating     p info     err    model participant
1         1        1        1       1      1 0.033   OK 7.9e-05 dynaViTE           1
2         1        1        1       1      1 0.033   OK 7.9e-05 dynaViTE           1
3         1        1        1       1      1 0.029   OK 7.7e-05 dynaViTE           1
4         1        1        1       1      1 0.020   OK 7.4e-05 dynaViTE           1
5         1        1        1       1      1 0.014   OK 7.1e-05 dynaViTE           1
1        -1        1        1       0      1 0.031   OK 7.8e-05 dynaViTE           1
```

In the above data frame, `p` represents the probability of a confidence rating and response given the stimulus identity and discriminability condition in the respective row. The columns `info` and `err` reproduce the output of the call to `integrate` used to compute the probabilities. We can compare the predicted distribution to the observed data distribution (see Figure 3), which can be used to assess the overall precision of the model fit. It seems that all models fitted the overall data pattern well. The strongest deviation is the overestimation of low confidence response for easy conditions in the MTLNR. In addition, there is generally a slight overestimation of low confidence for correct leftward responses in easy conditions, particularly in the race models (IRMt and PCRMt). In addition, for difficult stimuli, the probability of very low confidence was underestimated, while it was overestimated for high confidence. Interestingly, for all other conditions, there was a tendency to underestimate confidence in the leftward motion choices (second and third column) but an overestimation of confidence for rightward motion choices (first and fourth column). Participants seem to have shown a motion-dependent confidence bias leading to

higher confidence ratings for correct leftwards motion responses compared to rightwards motion responses, which all the models, but particularly the three race-based models, were not able to capture very accurately (Appendix Figure A2).

Using the full response distribution, it is possible to aggregate on different levels to examine specific data patterns. One possibility is to visualize the increase in accuracy with easier decisions (Figure 4, top row), for which the MTLNR shows a strong underestimation of accuracy in medium to high discriminability trials. Researchers might also be interested in the relationship between confidence and stimulus discriminability for correct and incorrect decisions (Figure 4, bottom row). In the present example, we see increasing mean confidence with higher stimulus discriminability for both correct and incorrect decisions, which is referred to as a double-increase pattern (Rausch & Zehetleitner, 2019). Many computational models of confidence are not able to produce such a pattern but can only account for a negative relationship between confidence and discriminability in incorrect decisions, resulting in the folded-X pattern (Hellmann et al., 2023; Rausch et al., 2018, 2020). One example of a model that only accounts for a folded-X pattern is the 2DSD model, which also showed this pattern in the present example. Although the PCRMt and IRMt are in principle able to account for a double-increase pattern, they showed a flat curve for incorrect responses, indicating constant confidence across difficulty levels for incorrect choices. Finally, also the MTLNR produces a folded-X pattern. The dynWEV model underestimated the steepness of the increase in confidence for incorrect decisions. The dynaViTE model showed the most pronounced double-increase pattern but overall overestimated confidence in incorrect decisions.

To compute the predicted response time distributions for the fitted parameters, **`dynConfiR`** offers the `predictRTModels` function, which, similarly to the `predictConfModels`, computes the joint distribution of choice, confidence rating, and response time for each level of stimulus identity and discriminability. The `dens` column represents the defective distribution of response times, i.e., the integral of the density for each confidence rating and response is not 1 but equals the probability of the respective choice and confidence report. With the argument `scaled=TRUE`, the correctly scaled densities (`densscaled`) are computed by dividing the defective density values by the probability of the respective discrete response. For this purpose, an additional argument (`DistConf`) may be provided by passing the output of `predictConfModels` to prevent the repeated computations of the discrete distributions. Note that the `DistConf` argument must have the same participants, models, and response and stimulus coding used for the models as the first argument. The resulting data frame has the following form:

```
> print(head(prediction_RTConfDist), row.names=FALSE)
condition stimulus response correct rating    rt  dens densscaled    model participant
1         1        1        1       1      1 0.000 0.000        0.0 dynaViTE           1
1         1        1        1       1      1 0.061 0.035        1.0 dynaViTE           1
1         1        1        1       1      1 0.121 0.046        1.4 dynaViTE           1
1         1        1        1       1      1 0.182 0.053        1.6 dynaViTE           1
1         1        1        1       1      1 0.242 0.059        1.8 dynaViTE           1
1         1        1        1       1      1 0.303 0.064        1.9 dynaViTE           1
```

In the decision-making literature, response time distributions are commonly visualized using quantiles (Figures 6 and 5 Ratcliff & Smith, 2004). For convenience, the package includes the function `PDFtoQuantiles`, which computes quantiles from a vector of probability density values. It also allows for data frame inputs with several columns that may be used as groups (like computing quantiles for different experimental conditions or participants). The relationship between task difficulty and response times was rather weak in the present example, illustrated by the flat quantile curves in Figure 5. Response times slightly decreased for easier stimuli, which was captured by all models. Here we see that the most notable deviation from the data is produced by the MTLNR in the correct choices for the easiest condition, for which the MTLNR underestimates the lower tail of the distribution. Concerning the relationship of confidence with response times, there was a weak negative relationship, which was more pronounced in incorrect choices (Figure 6). This pattern was again well captured by all models. The most pronounced deviations are visible in incorrect choices, for which the race models slightly overestimated the decrease in response times in the upper quantiles, and the 2DSD model did not reproduce the speed up for the lowest quantile in high confidence.

Exploratory analysis: Fixing model parameters

The drift diffusion-based models, i.e., DDConf, 2DSD, dynWEV, and dynaViTE, include all between-trial variability parameters. It is easy to fit these models additionally without allowing for between-trial variability in the starting point and non-decisional time by setting the respective parameters to 0 in the fitting routine. The following code demonstrates the `fixed` argument. We additionally fixed the diffusion constant in the visibility process for dynWEV and dynaViTE to 1, equal to the diffusion constant in the decision process. Note fixing any parameter only affects the models, which include those parameters. After the model fitting procedure, we renamed the models for the more restricted versions and combined all parameter fits in one data frame.

```
parfits_fixed <- fitRTConfModels(Data,
        models=c("dynaViTE", "dynWEV", "2DSD"),
        nRatings=4, restr_tau = "simult_conf",
        fixed = list(sz=0, st0=0, svis=1),
        opts=list(nAttempts=4, nRestarts=4), parallel="both", n.cores = c(5, 4),
        condition = "coh_level", rating="confidence")
parfits_fixed[, setdiff(names(parfits), names(parfits_fixed))] <- NA
allfits <- rbind(parfits,
        mutate(parfits_fixed, model = paste0(model, " (fixed)")))
```

The quantitative comparisons show that the restricted dynaViTE model performed best on an average level as well as for six individual participants, while the full dynaViTE model still fitted best for seven participants. Although the restricted dynWEV and the PCRMt model still had the second and third best average BICs, no participant was best fit by either model. This illustrates again that quantitative model comparisons can lead to different results when performed on a group level compared to the individual participant level (Figure 7). Over the extended set of models, the fully flexible dynaViTE still achieved a PEP of .75, the restricted version of the dynaViTE a PEP of .17, and the full 2DSD of .03. Concerning the visual model fit, the restricted models did not deviate strongly from the more complex models (Appendix Figures A3-A5).

```
selected_models <- c("dynaViTE (fixed)", "IRMt", "PCRMt", "dynWEV (fixed)")
selected_fits <- subset(allfits, model %in% selected_models)
prediction_ConfDist <- predictConfModels(selected_fits, simult_conf=TRUE,
                parallel = TRUE)
prediction_RTConfDist <- predictRTModels(selected_fits, simult_conf=TRUE, maxrt=4,
                scaled=TRUE, DistConf = prediction_ConfDist,
                parallel=TRUE)
```

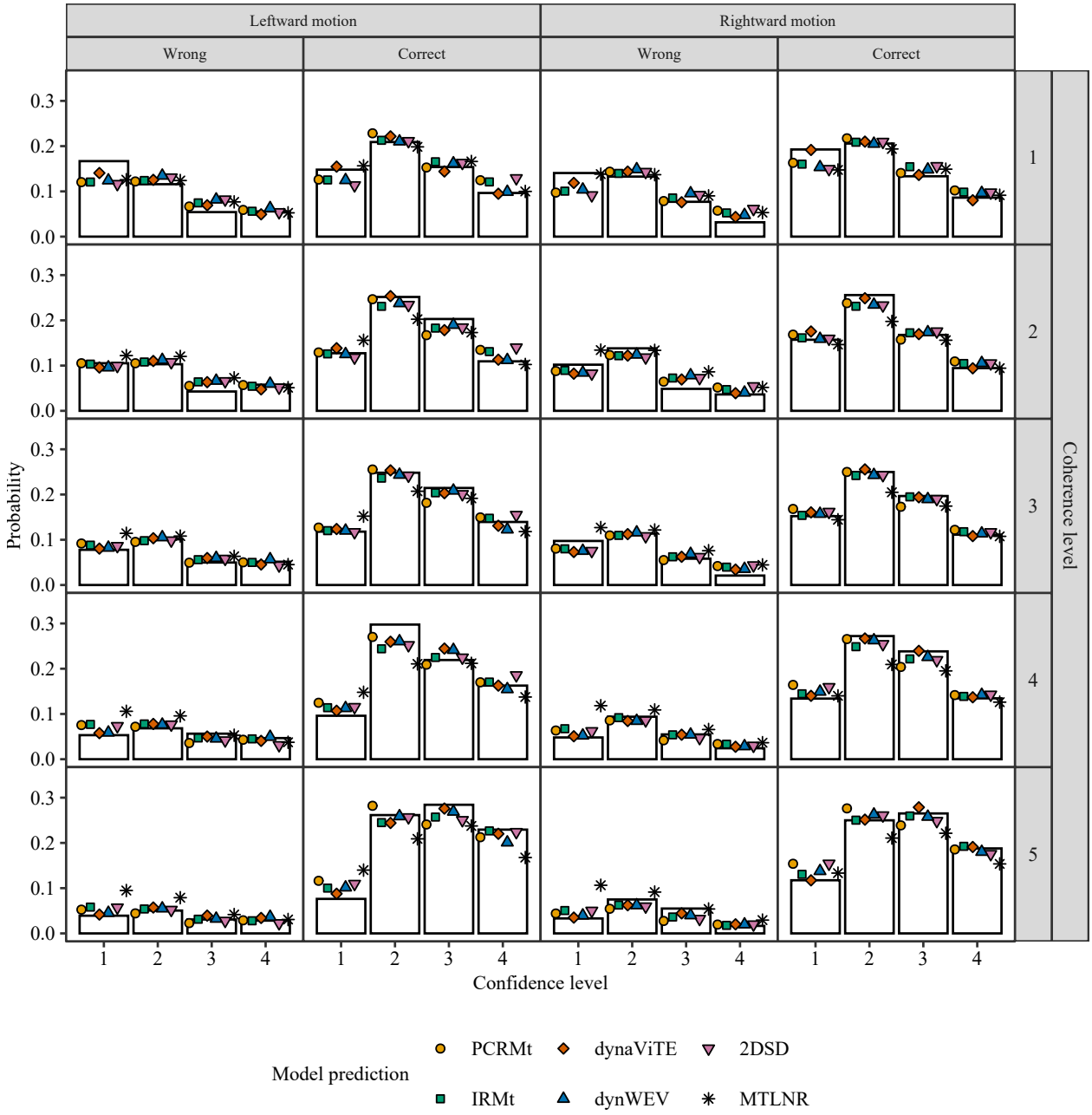


Figure 3

Observed (bars) and predicted (points) response distribution for the different models (shape and color of points) across stimulus identity (columns, high level) and levels of stimulus discriminability (rows). Probabilities within each row and stimulus identity column add to 1 for each group of data shown, i.e., height represents the conditional probability of a given accuracy and confidence rating given the stimulus discriminability and identity.

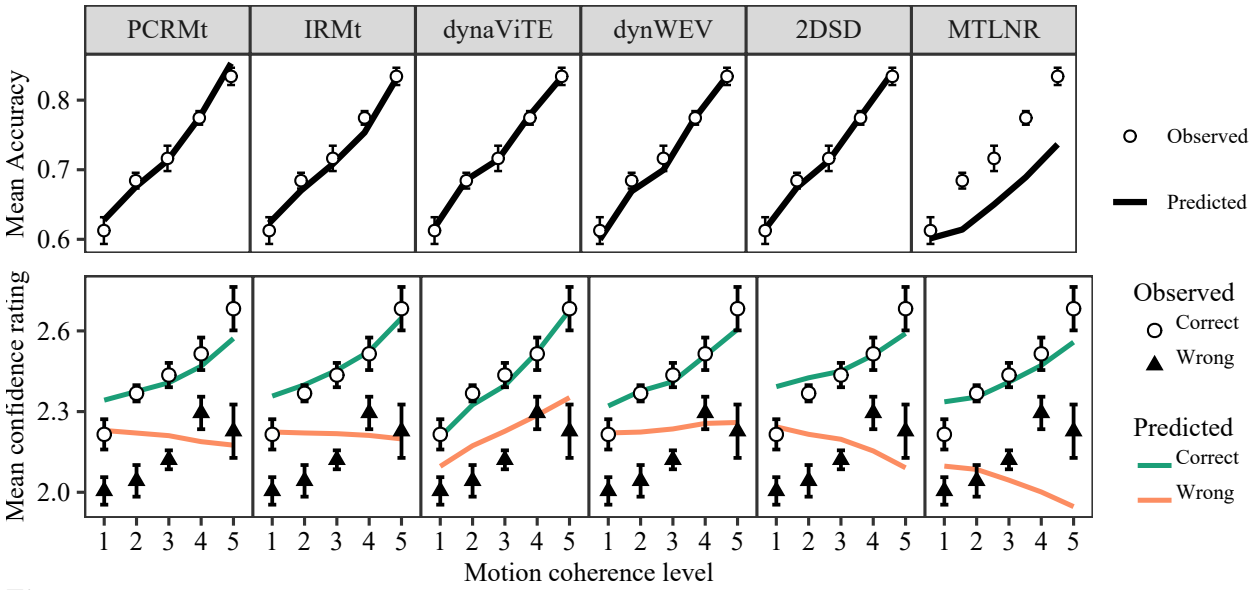


Figure 4

Accuracy (top row) and mean confidence rating (bottom row) for empirical data (points and triangles) and model predictions (lines). Error bars represent within-subject standard errors.

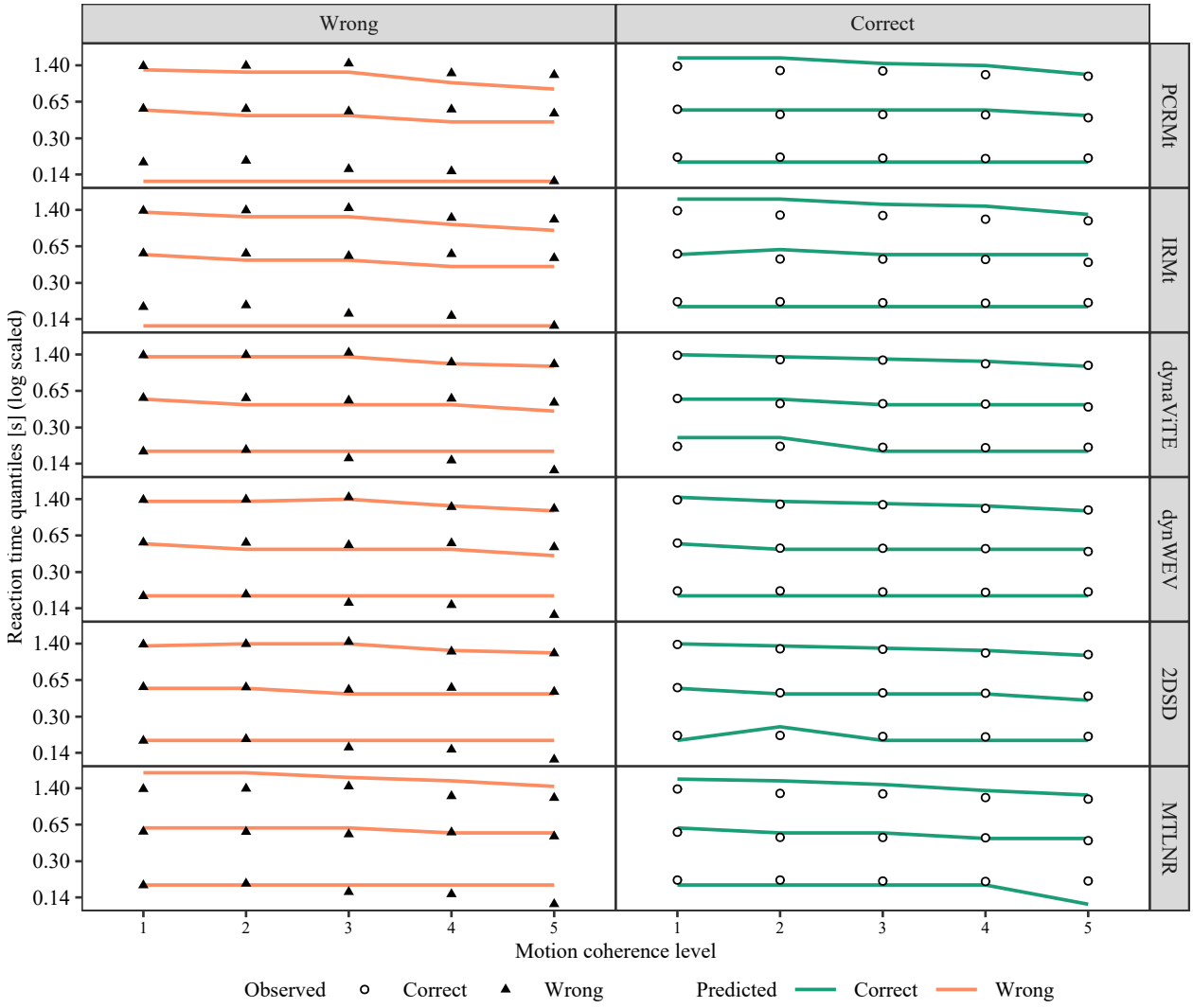


Figure 5

Response time quantiles for observed (points) and predicted (lines) response time distributions across correct and incorrect decisions (columns) and levels of stimulus discriminability (x-axis). Probabilities for quantiles: .1, .5, .9.

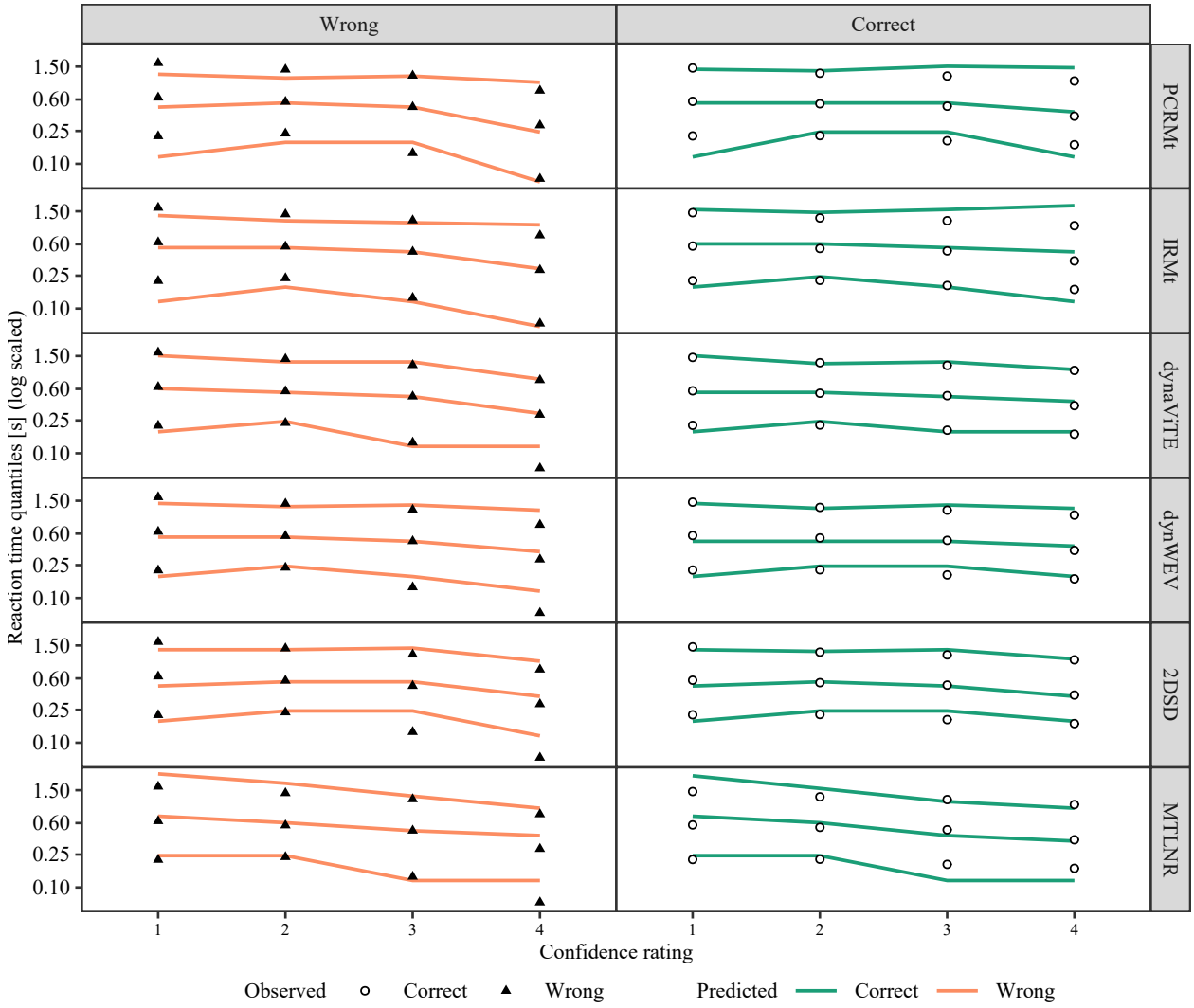


Figure 6

Response time quantiles for observed (points) and predicted (lines) response time distributions across correct and incorrect decisions (columns) and confidence ratings (x-axis). Probabilities for quantiles: .1, .5, .9.

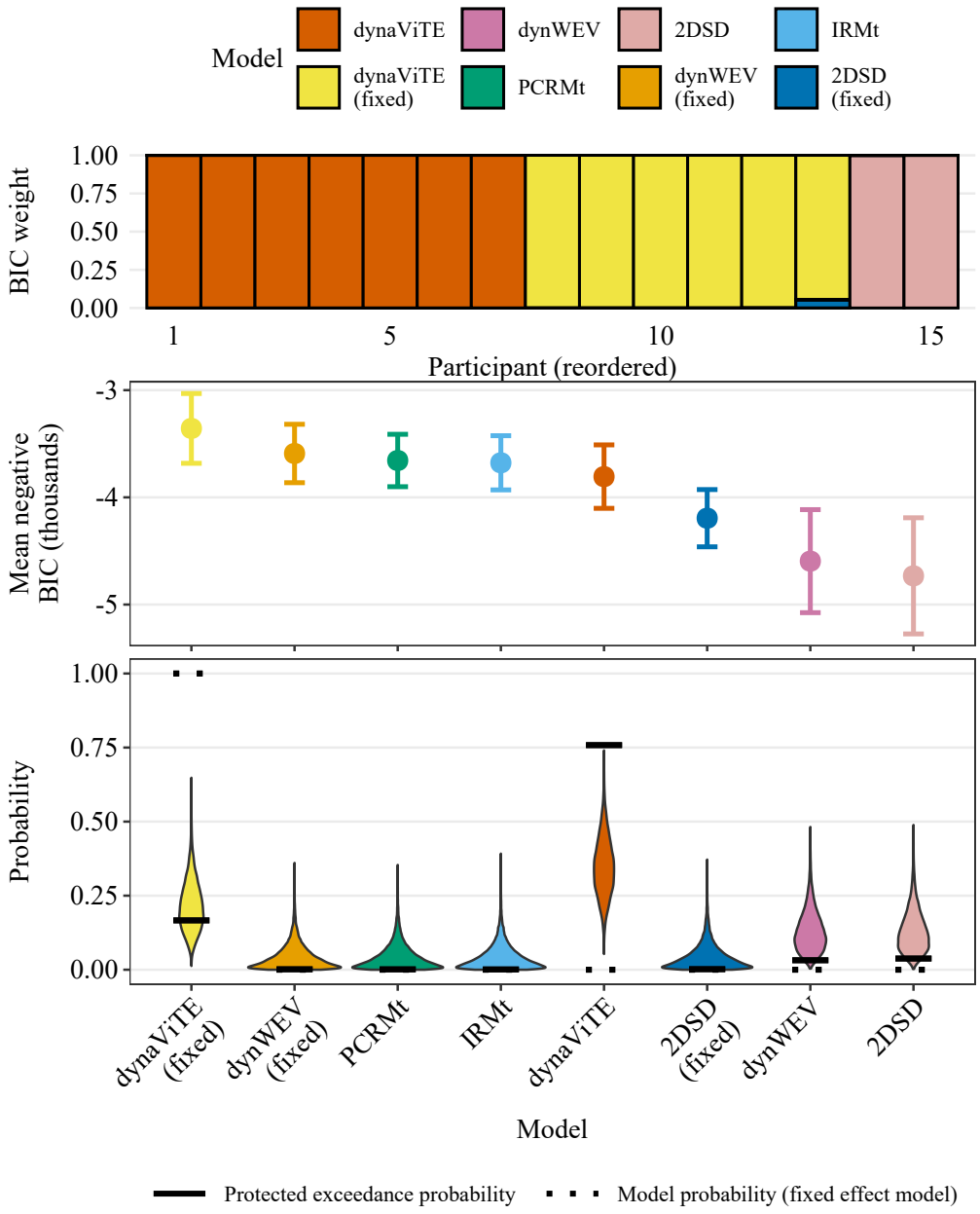


Figure 7

***Upper panel**: BIC weights across participants (reordered). **Middle panel**: Negative mean BIC values. Error bars represent within-subject standard errors. **Lower panel**: Results from a Bayesian model selection analysis. Violin plots show simulated model probabilities from a Dirichlet distribution fitted to BIC values according to a random effects model. Solid bars indicate the corresponding protected exceedance probability, dotted bars indicate the model probabilities resulting from a fixed effect model.*

Parameter recovery analysis

In order to compare fitted model parameters between groups or within subjects across different experimental conditions, it is necessary that the estimation of model parameters is robust. To assess the robustness of the model fitting procedure, we conducted a parameter recovery analysis using artificial data sets for the four most general models: dynaViTE, PCRMt, IRMt, and MTLNR, which include other models implemented in `dynConfiR` as special cases.

Method

To assess parameter recovery, we generated artificial data sets using known parameter values and then refitted each model to the simulated data. This procedure allowed us to evaluate how well the recovered parameters matched the generating ones. We assumed five confidence levels ($K = 5$) and five stimulus discriminability levels ($L = 5$), resulting in 26 fitted parameters for dynaViTE, 21 for IRMt and PCRMt, and 22 for MTLRN. Generating parameter sets were drawn from previous empirical model fits (Hellmann et al., 2023, 2024; Ng et al., 2021; Orchard et al., 2022; Shekhar & Rahnev, 2021). Confidence thresholds were not taken directly from these fits but were instead computed from quantiles of the simulated confidence variable, ensuring all confidence levels were represented.

Parameter recovery was evaluated across different data set sizes (50–500 trials per condition) using the concordance correlation coefficient (CCC; Lin, 1989) as a measure of agreement between generating and recovered parameters. Full methodological details and preprocessing steps are provided in the Appendix.

Results

Figure 8 illustrates the parameter recovery performance as measured by the concordance correlation coefficients across model parameters (also see Appendix Figures B1-12 for more detailed results). The results indicate that the decision-related parameters are generally more robust to recover for all models but the MTLNR.

In dynaViTE, the only choice-specific parameter that is hard to recover is the between trial variability in starting point (sz). Concerning the confidence-related parameters in dynaViTE, only the diffusion constant in the visibility accumulation (s_{Vis}) has a relatively low CCC, which requires at least 200 trials per condition and stimulus identity (2,000 trials in total) to achieve a CCC of .57. A possible solution to improve the robustness of parameter fitting might be to fix s_{Vis} to 1, similar to the diffusion constant in the decision process.

For race models, the confidence thresholds show a slightly lower recovery rate, although the general recovery is relatively good, even with fewer trials. The recovery of the confidence thresholds may also be

increased by assuming a parametric relationship between them instead of fitting each threshold independently and only restricting them to be monotonic.

In MTLNR, the decision parameters show in general slightly worse but still satisfying recovery. Importantly, the recovery does not strongly improve with the number of trials.

Figure 9 shows the time it took to fit the parameters to each data set. This illustrates that, without parallelization, fitting times may exceed several hours for large data sets, which limits the application of the package in such situations to machines with sufficient computing power. In addition, the MTLNR and dynaViTE models take considerably longer compared to PCRMt and IRMt with fitting times exceeding 48 hours for big data sets.

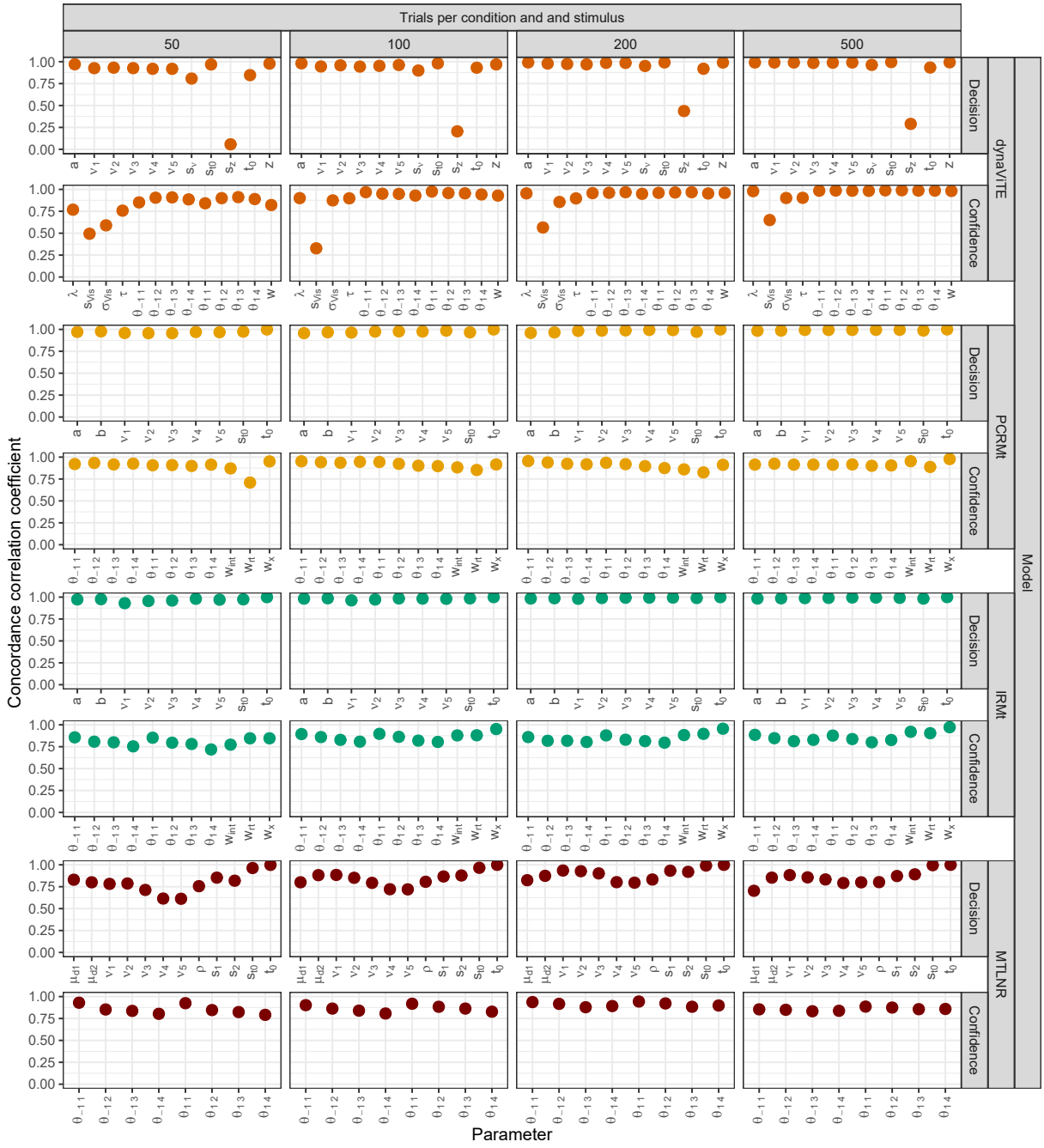


Figure 8

Concordance correlation coefficients (Lin, 1989) between the true and recovered parameters from the parameter recovery across the number of trials per condition and stimulus identity (columns) and generative model (rows).

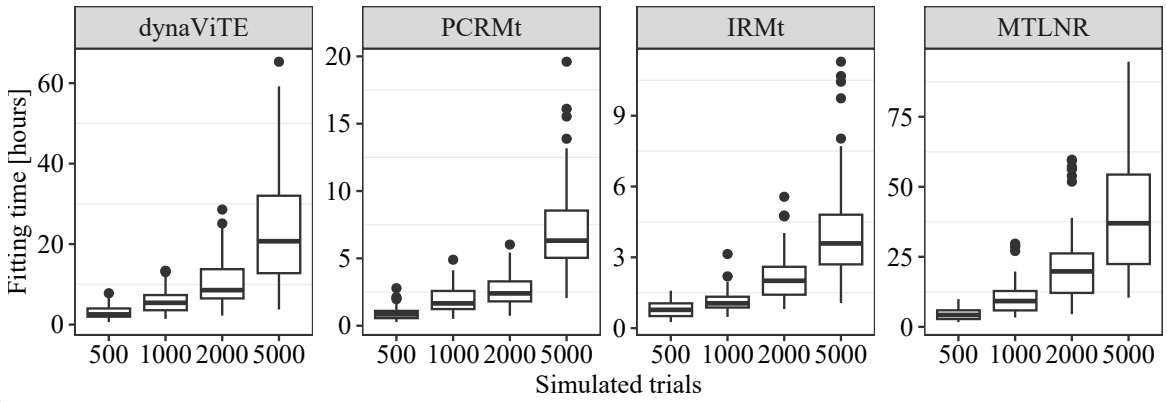


Figure 9

Fitting times for the simulated data sets in the parameter recovery.

Model recovery analysis

Besides the importance of being able to robustly estimate model parameters for each individual model, it is important to be able to identify the true data generating model among a set of candidate models, given some model is reasonably close to the data generating process.

We assessed model recovery in a similar way as parameter recovery, simulating data for four of the implemented models: 2DSD, dynaViTE, IRMt, and PCRMt. We then fitted each of these four models to the artificial datasets. Based on the fits, we identified the best fitting model in terms of the PEP for the whole simulated sample using the implemented `group_BMS_fit` function. In addition, we classified models in terms of the minimal BIC, AIC, and AICc for individual simulations and computed the classification precision.

Method

For the simulation of artificial data, we used the same method as in the parameter recovery analysis, sampling from parameter sets obtained by previous model fits to empirical data.

We sampled 50 parameter sets for each of the following models: 2DSD, dynaViTE, IRMt, PCRMt, and MTLNR. For each parameter set, we simulated artificial data with 100 trials per combination of stimulus identity and discriminability, resulting in 1000 trials per individual.

We then fitted each of the five models used for data generation to each of the simulated data sets. We fitted the models using the initial grid search, the four most promising parameter sets for two iterative calls to the optimization algorithm each. We reduced the number of optimization routines to speed up the fitting. The results show, however, that model recovery is still very good.

Results

Figure 10 shows the time needed to fit each data set. The race models are generally fitted faster with an expected time of 1 hour for the PCRMt and slightly less for IRMt. For 2DSD, most of the cases also take 1 hour, however values up to 2 hours occasionally occur. The longest fitting times are observed for dynaViTE—the most complex model—with up to 4 hours, while the most cases fall below 2 hours. MTLNR shows the highest median fitting time with around 1.5 hours, despite the relatively small number of parameters. This is probably due to the high numbers of steps in the numerical integration necessary to achieve a sufficient precision in the likelihood (see Precision analysis). Note that this is the time to fit each participant if no parallelization is used within the participants. The results suggest a very good model recovery on the individual level (Fig. 11, upper panel)for all models but MTLNR. Race models and DDM-based models can be clearly separated, and also among the model architectures, there is only slight confusion for individual subjects. The fitting procedure misidentifies the MTLNR with the 2DSD in 13 out

of the 50 simulated data sets (26%), which indicates that there is a high risk of model mimicry between these two models. On a group-level, the PEP for each generative model was close to 1 for the whole sample of 50 data sets, with MTLNR achieving the lowest PEP with .9991 and values indistinguishable from 1 for the other models. Because experimental studies with a big number of trials often rely on smaller sample sizes, we conducted a bootstrap analysis, in which we draw 1000 sub-samples of 10 data sets for each generative model and computed PEP on each subset. This analyses is rather conservatives because studies with only 10 participants and 500 trials per participant seem rather unreliable in general when using diffusion models and non-hierarchical methods. The results indicated that the model recovery of 2DSD, dynaViTE, IRMt, and PCRMt was robust (Fig. 11, lower panel). The only exception was MTLNR, which got occasionally identified incorrectly as 2DSD. Note that all selection analyses were conducted based on the BIC, which is the most conservative criterion. When using the less conservative AIC, results look similar on an individual level, but slightly better on a group-level (Appendix Figure C1) for the comparison between 2DSD and dynaViTE. The AIC is therefore preferable over the BIC when comparing these two models.

These results indicate while a comparison between diffusion-based models with race models is very robust in general both on the individual and group level, some models may show model mimicry on an individual level but also on the group level if sample size is low. Group-level comparisons are very robust for a sufficiently large sample size. In the case of small samples and small number of trials, we therefore recommend to conduct a model mimicry analysis for the fitted models using the parameter estimates for the data at hand.

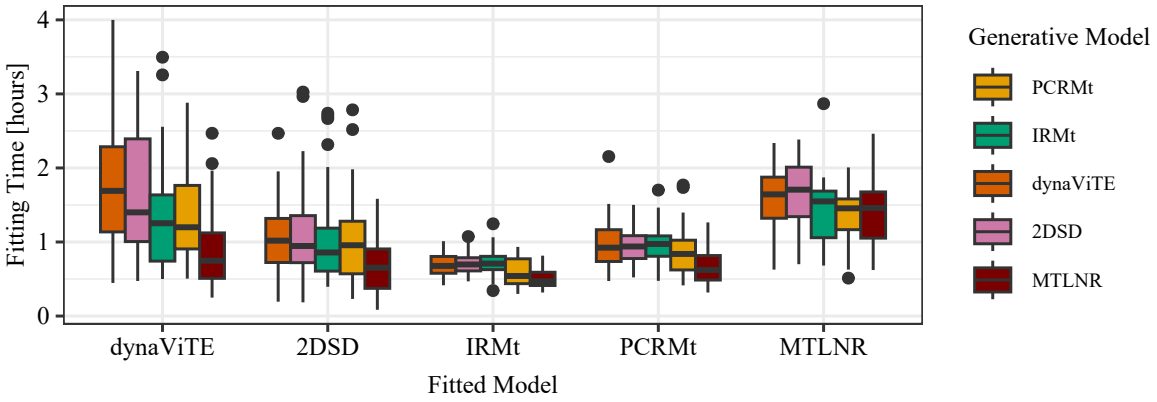


Figure 10

Fitting times for the simulated data sets in the model recovery.

Precision analysis

Two numerical approximations are involved in the computation of the probability densities. First, in the drift diffusion-based models (dynaViTE, dynWEV, 2DSD, and DDConf), the infinite series in the formula for the first-passage time density is approximated using a truncated summation for which an upper bound for the error is available (Navarro & Fuss, 2009). However, the integration over the variation in starting point and the variation in the non-decision time component are not analytically solvable and thus is computed numerically. This second approximation also leads to uncertainty in the precision of the density computations. The numerical computation of the integral uses a rectangular approximation of the density with an equidistant grid of support points. The step size for this approximation is controlled by the `precision` argument. The value of the precision argument is transformed into step sizes using similar computations as in the rtdists package (Singmann et al., 2020). Because there are no mathematical guarantees in the form of upper error bounds, we conducted a simulation study to assess the expected error for different values of the precision arguments.

Method

We estimated the expected error by computing the densities to simulated observations from different parameter sets several times with different values for the `precision` argument. Afterward, we computed the mean difference between the calculated density values as a measure of the error in the computation. Parameter sets were simulated in the following way.

Confidence judgments were assumed to be measured live on a three-point scale (i.e., $K = 3$), and there were three levels of stimulus discriminability (i.e., $L = 3$). This means that for dynaViTE, there were $11 + 3 + 2 \cdot 2 = 18$ parameters, for IRMt and PCRMt, there were $6 + 3 + 4 = 13$ parameters, and for MTLNR, there were $10 + 3 + 2 \cdot 2 = 17$ parameters.

Parameters were sampled independently and uniformly, with some exceptions. First, the discriminability parameters were uniformly distributed for the first level, and the differences between discriminability levels were uniformly distributed. This ensured increasing levels of discriminability. Second, starting point variability in dynaViTE was uniformly distributed across the admissible range dependent on the mean starting point parameter z. Similarly, the three weight parameters w_X, w_{RT}, and w_{Int} are sampled sequentially. In addition, for the upper decision boundaries of the two processes in race models, their sum and relative height were independently uniformly distributed to more closely resemble how boundary separation and relative starting point in dynaViTE were sampled. Finally, the confidence thresholds were computed based on a simulated proportion of ratings, which ensured a minimum number of observations in each category.

We derived parameter ranges from previously conducted model fits to empirical data from Hellmann et al. (2023, 2024), and the example in this paper.

We sampled 50 random parameter sets per model for dynaViTE, IRMt, PCRMt, and MTLNR. A data set with 50 trials for each combination of discriminability condition and stimulus category was generated for each parameter set (600 trials per data set in total). Finally, we computed the trial-wise likelihood for the simulated data with different precision arguments. For dynaViTE and MTLNR, we used values from 2 to 8 in steps of 0.5 plus a value of 9. For the race models, we used values from 2 to 7 in steps of 0.5 plus a precision of 9. The probabilities attained with the highest precision, i.e., 9, were used as references for the other precision values. To estimate the absolute error, we computed the mean absolute distance between the probabilities for each precision to the reference. In addition, we computed the mean absolute difference between two consecutive precision values to estimate the expected improvement of the density calculation. For full details, we refer the reader to the analysis code.

Results

The mean absolute differences for the computed probability densities between different values of precision arguments are depicted for dynaViTE in Figure 12, for the race models in Figure 13, and for MTLNR in Figure 14.

For all models, the estimated error and the difference between subsequent values of the precision argument both start far below the required value for low precision values and then decrease exponentially as functions of the precision argument for higher values. Assuming that the exponential decrease of subsequent differences continues, we can infer that the magnitude of the error in the computed densities is proportional to the estimated error in our simulation. This means that with the default value of 6 for the precision argument in the densities, the expected absolute error is about 10^{-6} of magnitude, while for a value of 7.5, it is $10^{-7.5}$. In general, the transformation between the precision argument and the step size used for the numerical integration is chosen such that the provided precision represents the number of digits correctly calculated on average.

Notably, the computation time for the probability densities also increases exponentially with the precision argument (Figures 12-14, lower rows), which clarifies the trade-off between precision and computation time. Because only one numerical integration is necessary for race models – IRMt, PCRMt, and MTLNR do not include between-trial variability in starting points – the computation time should be expected to be less influenced by the precision. While we see a lower intercept and slope for IRMt and PCRMt, i.e., faster computations, the computation time for MTLNR is similar to the dynaViTE, suggesting that the step-size required for obtaining a required precision is much smaller compared to the other models.

Fitting the parameters to experimental data using the default arguments for the fitting function may require up to 140,000 evaluations of the negative log-likelihood of the data (around 12,000 parameter sets in the initial grid search) plus 125,000 evaluations from the optimization (5 starting parameter sets each with five calls of the optimization routine each with a maximum of 5,000 function evaluations per optimization call). Keeping the computation time low is essential for the applicability of these models (see Figure 9). Therefore, we use a default precision of 3 in the fitting functions. Evaluating the likelihood with 600 trials takes much less than a second. Still, the results from the recovery analyses show that this is sufficient to produce a high model and parameter recovery.

In addition, using the default values, the precision of parameter estimates and likelihoods is often limited by other factors like the timing of stimulus presentation and the measurement precision of reaction times. For experimental response time data, the precision of reaction time measurements is often limited to milliseconds, and the precision depends on the hardware and the software used to conduct the experiment (Bridges et al., 2020; Plant & Turner, 2009). For online studies, the precision is often lower (Anwyl-Irvine et al., 2021; Semmelmann & Weigelt, 2017).

Summary

The `dynConfiR` package implements state-of-the-art computational models of choice, response time, and decision confidence based on the drift diffusion model and race models of choice. The R package may prove to be an attractive tool for psychology and cognitive neuroscience researchers because it offers a user-friendly implementation of the probability distributions of observed data and functions for parameter fitting, prediction, and simulation. The package is freely available via the CRAN repository, which provides sustainable access and facilitates its installation for the user.

Contributions and bug reports

Any issue and observed bugs in the package may be reported here: https://github.com/SeHellmann/dynConfiR/issues. Finally, we encourage any contributions in the form of additional implemented models or extensions of the package functionalities in the form of pull requests. A brief instruction on how to contribute new models is available at https://github.com/SeHellmann/dynConfiR/.

Declarations

Funding

This work was partly funded by the Deutsche Forschungsgemeinschaft (grants ZE887/8-1, ZE887/9-1 to MZ and RA2988/3-1, RA2988/4-1 to MR).

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

Data sets are available for download at https://github.com/SeHellmann/dynConfiR_Paper.

Code availability

Code for all analyses and production of figures is available for download at https://github.com/SeHellmann/dynConfiR_Paper.

Authors' contributions

Funding acquisition: MR, MZ. Conceptualization & methodology: SH. Formal analysis, software, & visualization: SH. Writing – Original Draft: SH. Resources & project administration: MR, MZ. Writing – Review & Editing: all authors.

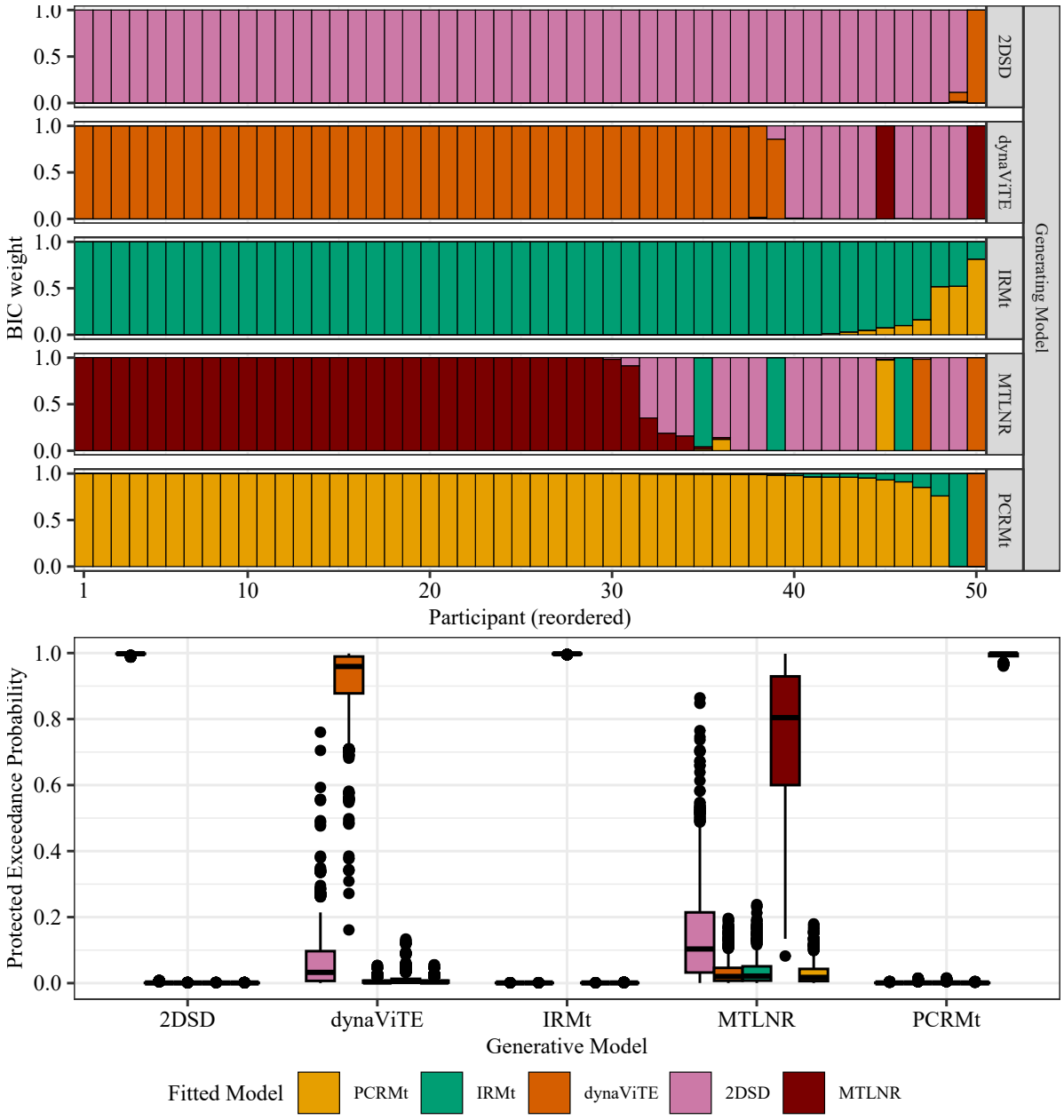


Figure 11

Upper panel: Individual model weights for each generative model (rows) across simulated participants.

Lower panel: Bootstrapped protected exceedance probabilities (PEP).

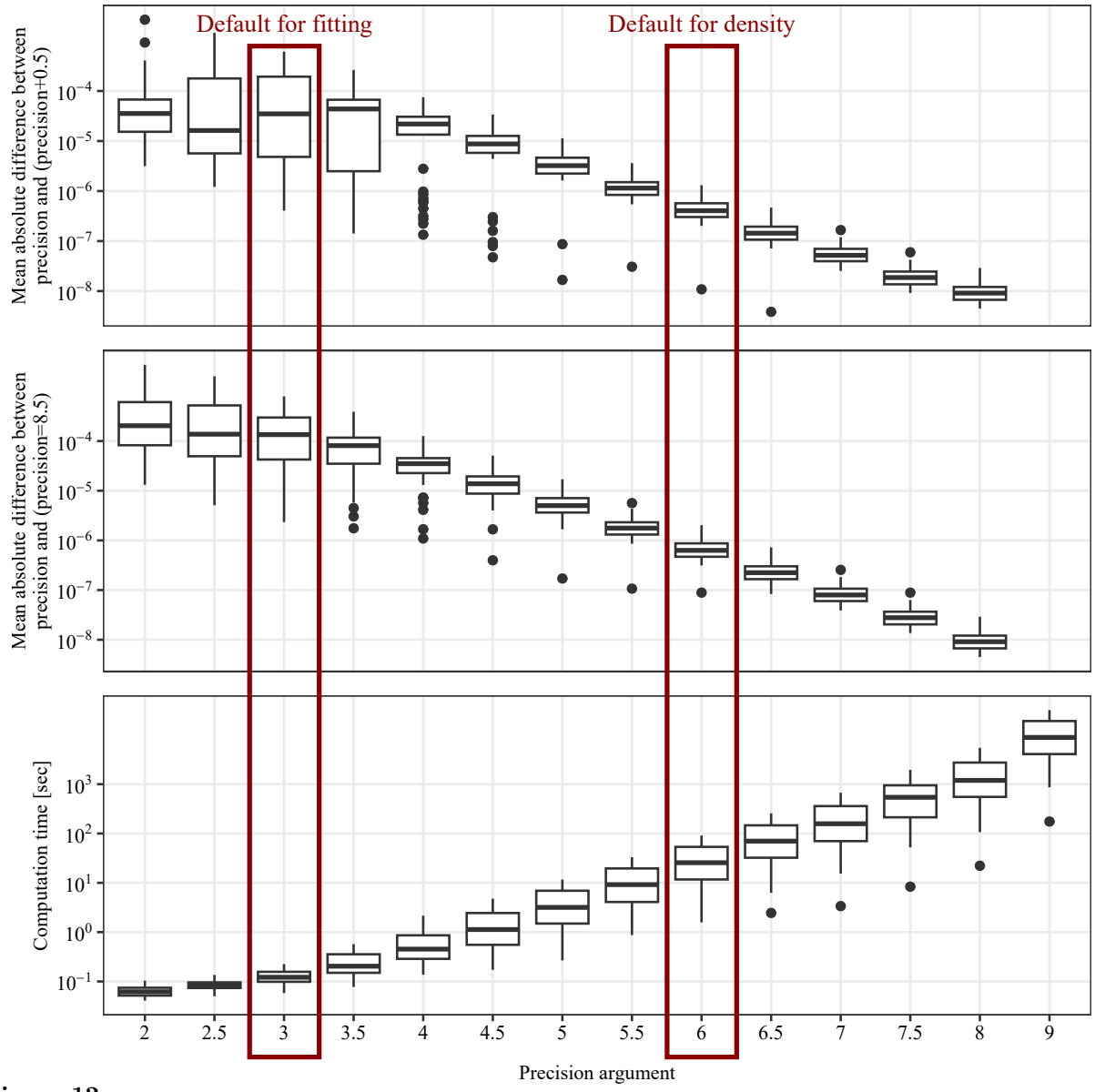


Figure 12

First Row: Distribution of mean absolute differences in computed densities between different choices of the precision argument and the argument +0.5. Second Row: Distribution of mean absolute differences in computed densities between different choices of the precision argument and the computed densities for precision = 8.5. Third Row: Distribution of computation time of the densities for a vector of 600 observations (log scaled). Each observation is based on 600 simulated trials for a random parameter set (50 trials for each combination of discriminability condition and stimulus category).

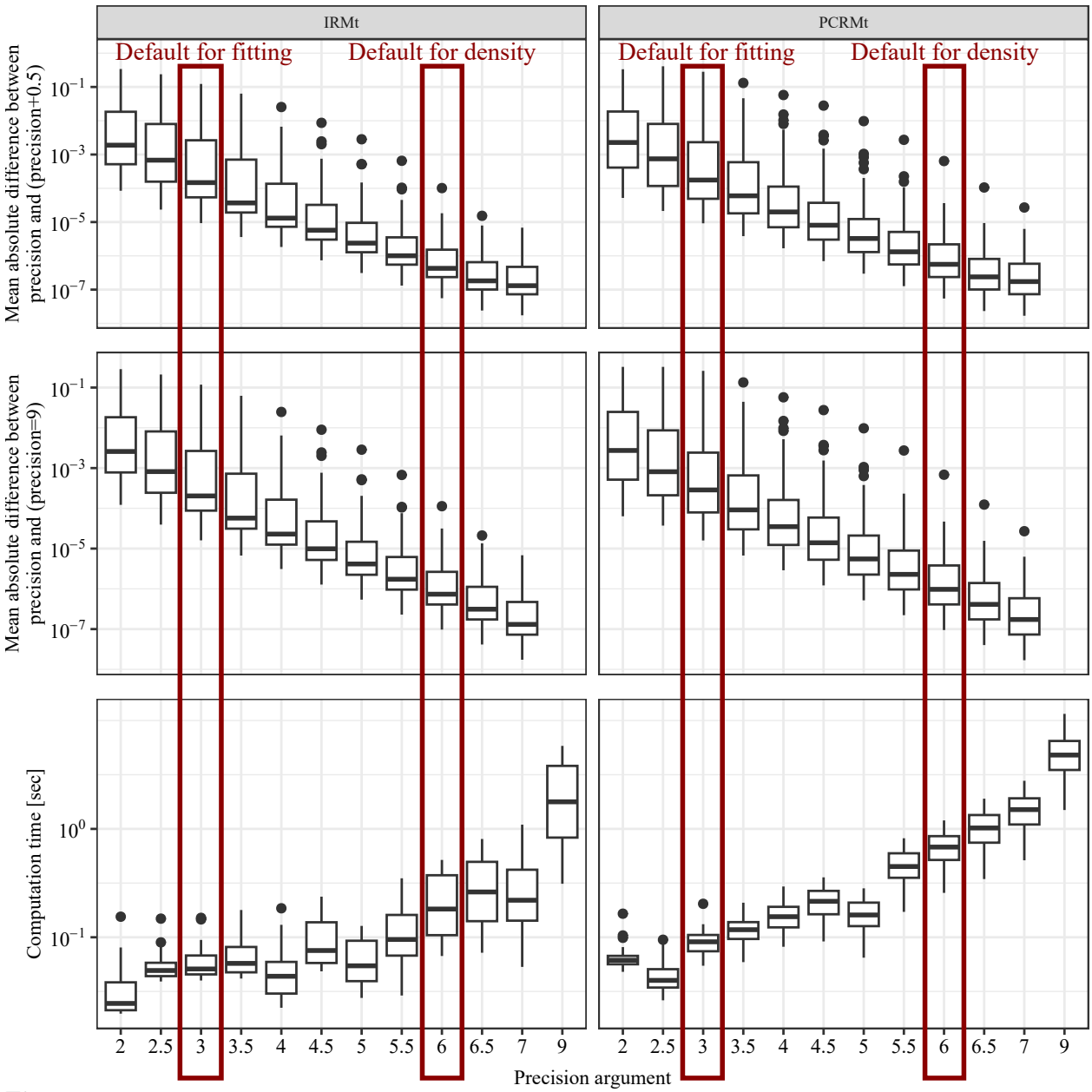


Figure 13

First Row: Distribution of mean absolute differences in computed densities between consecutive increases in the exponent of the precision argument. Second Row: Distribution of mean absolute differences in computed densities between different choices of the precision argument and the computed densities for precision = 9. Third Row: Distribution of computation time of the densities for a vector of 600 observations (log scaled). Each observation is based on 600 simulated trials for a random parameter set (50 trials for each combination of discriminability condition and stimulus category).

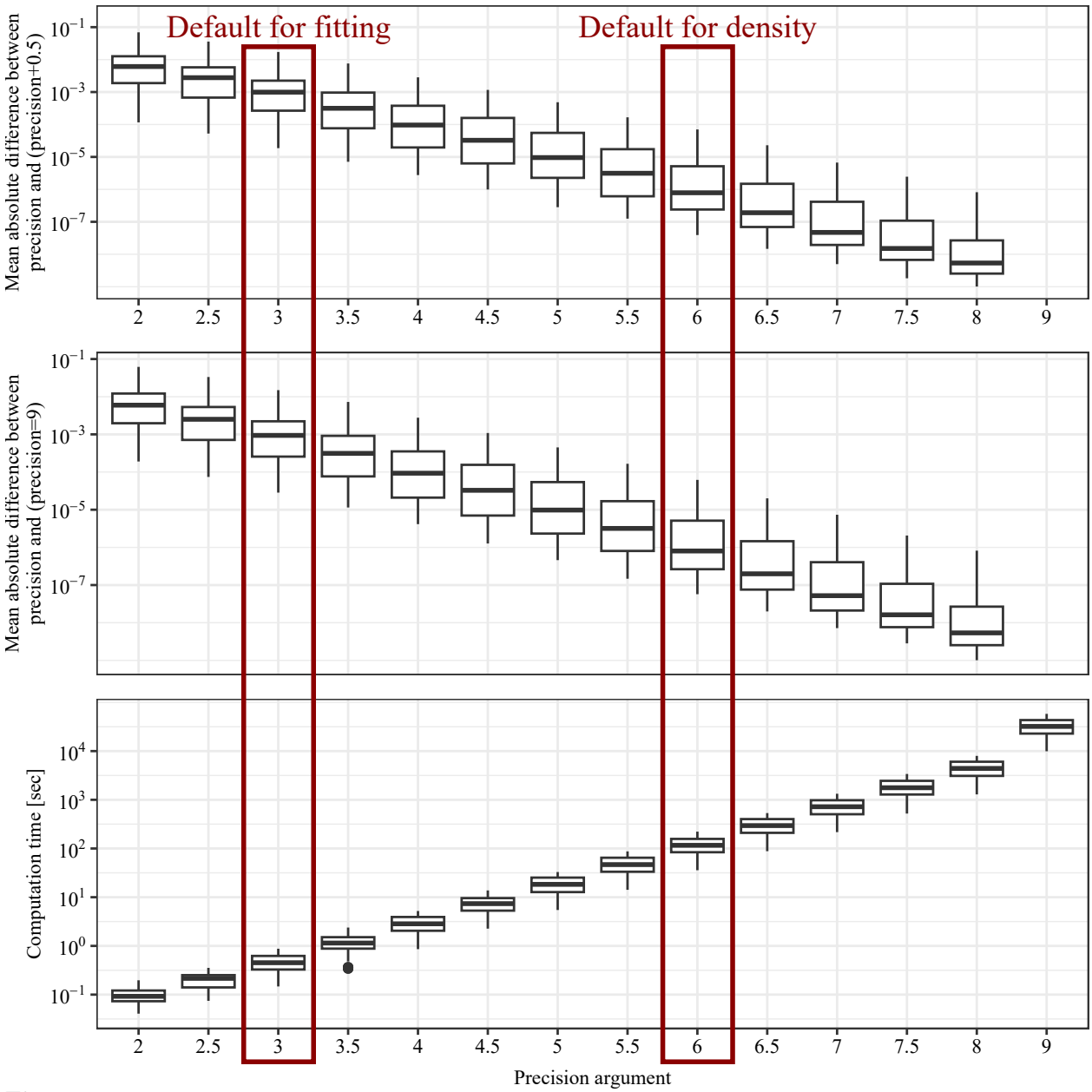


Figure 14

First Row: Distribution of mean absolute differences in computed densities between consecutive increases in the exponent of the precision argument. Second Row: Distribution of mean absolute differences in computed densities between different choices of the precision argument and the computed densities for precision = 9. Third Row: Distribution of computation time of the densities for a vector of 600 observations (log scaled). Each observation is based on 600 simulated trials for a random parameter set (50 trials for each combination of discriminability condition and stimulus category).

References

Adler, W. T., & Ma, W. J. (2018). Comparing bayesian and non-bayesian accounts of human confidence reports. *PLoS computational biology*, *14*(11), Article e1006572. https://doi.org/10.1371/journal.pcbi.1006572

Aitchison, L., Bang, D., Bahrami, B., & Latham, P. E. (2015). Doubly bayesian analysis of confidence in perceptual decision-making. *PLOS Computational Biology*, *11*(10), Article e1004519. https://doi.org/10.1371/journal.pcbi.1004519

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. https://doi.org/10.1109/TAC.1974.1100705

Andreas Voss & Jochen Voss. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, *39*(4), 767–775. https://doi.org/10.3758/BF03192967

Anwyl-Irvine, A., Dalmaijer, E. S., Hodges, N., & Evershed, J. K. (2021). Realistic precision and accuracy of online experiment platforms, web browsers, and devices. *Behavior Research Methods*, *53*(4), 1407–1425. https://doi.org/10.3758/s13428-020-01501-5

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological review*, *113*(4), 700–765. https://doi.org/10.1037/0033-295X.113.4.700

Bridges, D., Pitiot, A., MacAskill, M. R., & Peirce, J. W. (2020). The timing mega-study: Comparing a range of experiment generators, both lab-based and online. *PeerJ*, *8*, e9414. https://doi.org/10.7717/peerj.9414

Brown, S., & Heathcote, A. (2005). A ballistic model of choice response time. *Psychological review*, *112*(1), 117–128. https://doi.org/10.1037/0033-295X.112.1.117

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

Daunizeau, J., Adam, V., & Rigoux, L. (2014). Vba: A probabilistic treatment of nonlinear models for neurobiological and behavioural data. *PLOS Computational Biology*, *10*(1), e1003441. https://doi.org/10.1371/journal.pcbi.1003441

Desender, K., Boldt, A., Verguts, T., & Donner, T. H. (2019). Confidence predicts speed-accuracy tradeoff for subsequent decisions. *eLife*, *8*, Article e43499. Retrieved December 14, 2022, from https://elifesciences.org/articles/43499

Desender, K., Donner, T. H., & Verguts, T. (2021). Dynamic expressions of confidence within an evidence accumulation framework. *Cognition*, *207*, Article 104522. https://doi.org/10.1016/j.cognition.2020.104522

Desender, K., Vermeylen, L., & Verguts, T. (2022). Dynamic influences on static measures of metacognition. *Nature Communications*, *13*(1), 4208. https://doi.org/10.1038/s41467-022-31727-0

Drugowitsch, J., Mendonça, A. G., Mainen, Z. F., & Pouget, A. (2019). Learning optimal decisions with confidence. *Proceedings of the National Academy of Sciences of the United States of America*, *116*(49), 24872–24880. https://doi.org/10.1073/pnas.1906787116

Drugowitsch, J., Moreno-Bote, R., Churchland, A. K., Shadlen, M. N., & Pouget, A. (2012). The cost of accumulating evidence in perceptual decision making. *Journal of Neuroscience*, *32*(11), 3612–3628. https://doi.org/10.1523/JNEUROSCI.4010-11.2012

Farrell, S., & Lewandowsky, S. (2018). *Computational modeling of cognition and behavior*. Cambridge University Press. https://doi.org/10.1017/9781316272503

Fontanesi, L., Gluth, S., Spektor, M. S., & Rieskamp, J. (2019). A reinforcement learning diffusion decision model for value-based decisions. *Psychonomic Bulletin & Review*, *26*(4), 1099–1121. https://doi.org/10.3758/s13423-018-1554-2

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics*. Wiley. https://books.google.de/books?id=E-04zQEACAAJ

Guggenmos, M. (2022). Reverse engineering of metacognition. *eLife*, *11*, Article e75420. https://doi.org/10.7554/eLife.75420

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

Hellmann, S., Zehetleitner, M., & Rausch, M. (2023). Simultaneous modeling of choice, confidence, and response time in visual perception. *Psychological review*. https://doi.org/10.1037/rev0000411

Hellmann, S., Zehetleitner, M., & Rausch, M. (2024). Confidence is influenced by evidence accumulation time in dynamical decision models. *Computational Brain & Behavior*, *7*(3), 287–313. https://doi.org/10.1007/s42113-024-00205-9

Henrich, F., Hartmann, R., Pratz, V., Voss, A., & Klauer, K. C. (2024). The seven-parameter diffusion model: An implementation in stan for bayesian analyses. *Behavior Research Methods*, *56*(4), 3102–3116. https://doi.org/10.3758/s13428-023-02179-1

Herregods, S., Le Denmat, P., Vermeylen, L., & Desender, K. (2023). *Modelling speed-accuracy tradeoffs in the stopping rule for confidence judgments*. https://doi.org/10.1101/2023.02.27.530208

Kiani, R., Corthell, L., & Shadlen, M. N. (2014). Choice certainty is informed by both evidence and decision time. *Neuron*, *84*(6), 1329–1342. https://doi.org/10.1016/j.neuron.2014.12.015

Klenke, A. (2013). *Wahrscheinlichkeitstheorie*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36018-3

Krajbich, I., Armel, C., & Rangel, A. (2010). Visual fixations and the computation and comparison of value in simple choice. *Nature Neuroscience*, *13*(10), 1292–1298. https://doi.org/10.1038/nn.2635

LaBerge, D. (1994). Quantitative models of attention and response processes in shape identification tasks. *Journal of Mathematical Psychology*, *38*(2), 198–243. https://doi.org/10.1006/jmps.1994.1015

Lerche, V., & Voss, A. (2016). Model complexity in diffusion modeling: Benefits of making the model more parsimonious. *Frontiers in Psychology*, *7*, 1324. https://doi.org/10.3389/fpsyg.2016.01324

Lerche, V., & Voss, A. (2017). Retest reliability of the parameters of the ratcliff diffusion model. *Psychological Research*, *81*(3), 629–652. https://doi.org/10.1007/s00426-016-0770-5

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, L. I. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, *45*(1), 255–268.

Mamassian, P., & de Gardelle, V. (2022). Modeling perceptual confidence and the confidence forced-choice paradigm. *Psychological Review*, *129*(5), 976–998. https://doi.org/10.1037/rev0000312

Maniscalco, B., & Lau, H. (2012). A signal detection theoretic approach for estimating metacognitive sensitivity from confidence ratings. *Consciousness and cognition*, *21*(1), 422–430. https://doi.org/10.1016/j.concog.2011.09.021

Maniscalco, B., & Lau, H. (2016). The signal processing architecture underlying subjective reports of sensory awareness. *Neuroscience of consciousness*, *2016*(1). https://doi.org/10.1093/nc/niw002

Maniscalco, B., Peters, M. A. K., & Lau, H. (2016). Heuristic use of perceptual evidence leads to dissociation between performance and metacognitive sensitivity. *Attention, Perception, & Psychophysics*, *78*(3), 923–937. https://doi.org/10.3758/s13414-016-1059-x

Milosavljevic, M., Malmaud, J., Huth, A., Koch, C., & Rangel, A. (2010). The drift diffusion model can account for the accuracy and reaction time of value-based choices under high and low time pressure. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1901533

Moran, R., Teodorescu, A. R., & Usher, M. (2015). Post choice information integration as a causal determinant of confidence: Novel data and a computational account. *Cognitive psychology*, *78*, 99–147. https://doi.org/10.1016/j.cogpsych.2015.01.002

Moreno-Bote, R. (2010). Decision confidence and uncertainty in diffusion models with partially correlated neuronal integrators. *Neural computation*, *22*(7). https://doi.org/10.1162/neco.2010.12-08-930

Morey, R. D., Rouder, J. N., Jamil, T., Urbanek, S., Forner, K., & Ly, A. (2024). Bayesfactor: Computation of bayes factors for common designs (version 0.9.12-4.7). https://cran.r-project.org/web/packages/BayesFactor

Myung, I. J. (2000). The importance of complexity in model selection. *Journal of Mathematical Psychology*, *44*(1), 190–204. https://doi.org/10.1006/jmps.1999.1283

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

Nelson, T. O. (1984). A comparison of current measures of the accuracy of feeling-of-knowing predictions. *Psychological bulletin*, *95*(1), 109–133.

Ng, L. C. H., Law, F. H. F., Lam, A. M. W., Or, C. C.-F., & Lee, A. L. F. (2021). Metacognitive adaptation revealed in serial dependence of visual confidence judgments. *Journal of Vision*, *21*(9), 2487. https://doi.org/10.1167/jov.21.9.2487

Orchard, E. R., Dakin, S. C., & van Boxtel, J. J. A. (2022). Internal noise measures in coarse and fine motion direction discrimination tasks and the correlation with autism traits. *Journal of Vision*, *22*(10), 19. https://doi.org/10.1167/jov.22.10.19

Palminteri, S., Wyart, V., & Koechlin, E. (2017). The importance of falsification in computational cognitive modeling. *Trends in cognitive sciences*, *21*(6), 425–433. https://doi.org/10.1016/j.tics.2017.03.011

Plant, R. R., & Turner, G. (2009). Millisecond precision psychological research in a world of commodity computers: New hardware, new problems? *Behavior Research Methods*, *41*(3), 598–614. https://doi.org/10.3758/BRM.41.3.598

Pleskac, T. J., & Busemeyer, J. (2010). Two-stage dynamic signal detection: A theory of choice, decision time, and confidence. *Psychological review*, *117*(3). https://doi.org/10.1037/a0019737

Pouget, A., Drugowitsch, J., & Kepecs, A. (2016). Confidence and certainty: Distinct probabilistic quantities for different goals. *Nature neuroscience*, *19*(3), 366–374. https://doi.org/10.1038/nn.4240

Rahnev, D., Desender, K., Lee, A. L. F., Adler, W. T., Aguilar-Lleyda, D., Akdoğan, B., Arbuzova, P., Atlas, L. Y., Balcı, F., Bang, J. W., Bègue, I., Birney, D. P., Brady, T. F., Calder-Travis, J., Chetverikov, A., Clark, T. K., Davranche, K., Denison, R. N., Dildine, T. C., … Zylberberg, A. (2020). The confidence database. *Nature human behaviour*, *4*(3), 317–325. https://doi.org/10.1038/s41562-019-0813-1

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological review*, *85*(2), 59–108. https://doi.org/10.1037/0033-295X.85.2.59

Ratcliff, R., Gomez, P., & McKoon, G. (2004). A diffusion model account of the lexical decision task. *Psychological review*, *111*(1), 159–182. https://doi.org/10.1037/0033-295X.111.1.159

Ratcliff, R., & Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, *9*(5), 347–356. https://doi.org/10.1111/1467-9280.00067

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., & Starns, J. J. (2009). Modeling confidence and response time in recognition memory. *Psychological review*, *116*(1), 59–83. https://doi.org/10.1037/a0014086

Ratcliff, R., & Starns, J. J. (2013). Modeling confidence judgments, response times, and multiple choices in decision making: Recognition memory and motion discrimination. *Psychological review*, *120*(3), 697–719. https://doi.org/10.1037/a0033152

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

Ratcliff, R., Voskuilen, C., & Teodorescu, A. (2018). Modeling 2-alternative forced-choice tasks: Accounting for both magnitude and difference effects. *Cognitive psychology*, *103*, 1–22. https://doi.org/10.1016/j.cogpsych.2018.02.002

Rausch, M., Hellmann, S., & Zehetleitner, M. (2018). Confidence in masked orientation judgments is informed by both evidence and visibility. *Attention, Perception, & Psychophysics*, *80*(1), 134–154. https://doi.org/10.3758/s13414-017-1431-5

Rausch, M., Hellmann, S., & Zehetleitner, M. (2023). Measures of metacognitive efficiency across cognitive models of decision confidence. *Psychological Methods*. https://doi.org/10.1037/met0000634

Rausch, M., Meyen, S., & Hellmann, S. (2025). Statconfr: An r package for static models of decision confidence and metacognition. *Journal of Open Source Software*, *10*(106), 6966. https://doi.org/10.21105/joss.06966

Rausch, M., & Zehetleitner, M. (2019). The folded x-pattern is not necessarily a statistical signature of decision confidence. *PLoS computational biology*, *15*(10), Article e1007456. https://doi.org/10.1371/journal.pcbi.1007456

Rausch, M., Zehetleitner, M., Steinhauser, M., & Maier, M. E. (2020). Cognitive modelling reveals distinct electrophysiological markers of decision confidence and error monitoring. *NeuroImage*, *218*, Article 116963. https://doi.org/10.1016/j.neuroimage.2020.116963

Reynolds, A., Kvam, P. D., Osth, A. F., & Heathcote, A. (2020). Correlated racing evidence accumulator models. *Journal of Mathematical Psychology*, *96*, 102331. https://doi.org/10.1016/j.jmp.2020.102331

Rigoux, L., Stephan, K. E., Friston, K. J., & Daunizeau, J. (2014). Bayesian model selection for group studies - revisited. *NeuroImage*, *84*, 971–985. https://doi.org/10.1016/j.neuroimage.2013.08.065

Röseler, L., Kaiser, L., Doetsch, C. A., Klett, N., Seida, C., Schütz, A., Aczel, B., Adelina, N., Agostini, V., Alarie, S., Albayarak-Aydemir, N., Aldoh, A., Al-Hoorie, A. H., Azevedo, F., Baker, B. J., Barth, C. L., Beitner, J., Brick, C., Brohmer, H., … Zhang, Y. (2024). *The replication database: Documenting the replicability of psychological science*. https://doi.org/10.31222/osf.io/me2ub

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464. https://doi.org/10.1214/aos/1176344136

Semmelmann, K., & Weigelt, S. (2017). Online psychophysics: Reaction time effects in cognitive experiments. *Behavior Research Methods*, *49*(4), 1241–1260. https://doi.org/10.3758/s13428-016-0783-4

Shekhar, M., & Rahnev, D. (2021). The nature of metacognitive inefficiency in perceptual decision making. *Psychological review*, *128*(1), 45–70. https://doi.org/10.1037/rev0000249

Shekhar, M., & Rahnev, D. (2024). How do humans give confidence? a comprehensive comparison of process models of perceptual metacognition. *Journal of experimental psychology. General*, *153*(3), 656–688. https://doi.org/10.1037/xge0001524

Singmann, H., Brown, S. D., Gretton, M., Heathcote, A., Voss, A., Voss, J., & Terry, A. (2020). Rtdists: Response time distributions.

Smith, P. L. (2000). Stochastic dynamic models of response time and accuracy: A foundational primer. *Journal of Mathematical Psychology*, *44*(3), 408–463. https://doi.org/10.1006/jmps.1999.1260

Stephan, K. E., Penny, W. D., Daunizeau, J., Moran, R. J., & Friston, K. J. (2009). Bayesian model selection for group studies. *NeuroImage*, *46*(4), 1004–1017. https://doi.org/10.1016/j.neuroimage.2009.03.025

Stevenson, N., Donzallaz, M. C., Innes, R. J., Forstmann, B., Matzke, D., & Heathcote, A. (2024). *Emc2: An r package for cognitive models of choice*. https://doi.org/10.31234/osf.io/2e4dq

Tajima, S., Drugowitsch, J., & Pouget, A. (2016). Optimal policy for value-based decision-making. *Nature Communications*, *7*(1), Article 12400. https://doi.org/10.1038/ncomms12400

Teodorescu, A. R., Moran, R., & Usher, M. (2016). Absolutely relative or relatively absolute: Violations of value invariance in human decision making. *Psychonomic Bulletin & Review*, *23*(1), 22–38. https://doi.org/10.3758/s13423-015-0858-8

Teodorescu, A. R., & Usher, M. (2013). Disentangling decision models: From independence to competition. *Psychological Review*, *120*(1), 1–38. https://doi.org/10.1037/a0030776

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: The leaky, competing accumulator model. *Psychological review*, *108*(3), 550–592. https://doi.org/10.1037/0033-295x.108.3.550

van den Berg, R., Anandalingam, K., Zylberberg, A., Kiani, R., Shadlen, M. N., & Wolpert, D. M. (2016). A common mechanism underlies changes of mind about decisions and confidence. *eLife*, *5*, Article e12192. https://doi.org/10.7554/eLife.12192

Vickers, D., Smith, P., Burt, J., & Brown, M. (1985). Experimental paradigms emphasising state or process limitations: Ii effects on confidence. *Acta Psychologica*, *59*(2), 163–193. https://doi.org/10.1016/0001-6918(85)90018-6

Voss, A., Nagler, M., & Lerche, V. (2013). Diffusion models in experimental psychology: A practical introduction. *Experimental psychology*, *60*(6). https://doi.org/10.1027/1618-3169/a000218

Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, *32*(7), 1206–1220. https://doi.org/10.3758/BF03196893

Wabersich, D., & Vandekerckhove, J. (2014a). 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

Wabersich, D., & Vandekerckhove, J. (2014b). The rwiener package: An r package providing distribution functions for the wiener diffusion model. *The R Journal*, *6*(1), 49. https://doi.org/10.32614/RJ-2014-005

Wagenmakers, E.-J., Wetzels, R., Borsboom, D., van der Maas, H. L. J., & Kievit, R. A. (2012). An agenda for purely confirmatory research. *Perspectives on psychological science : a journal of the Association for Psychological Science*, *7*(6), 632–638. https://doi.org/10.1177/1745691612463078

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

Zawadzka, K., Higham, P. A., & Hanczakowski, M. (2017). Confidence in forced-choice recognition: What underlies the ratings? *Journal of experimental psychology. Learning, memory, and cognition*, *43*(4), 552–564. https://doi.org/10.1037/xlm0000321

Zylberberg, A., Barttfeld, P., & Sigman, M. (2012). The construction of confidence in a perceptual decision. *Frontiers in integrative neuroscience*, *6*, Article 79. https://doi.org/10.3389/fnint.2012.00079

Appendix A

Example of Application in Model Comparison

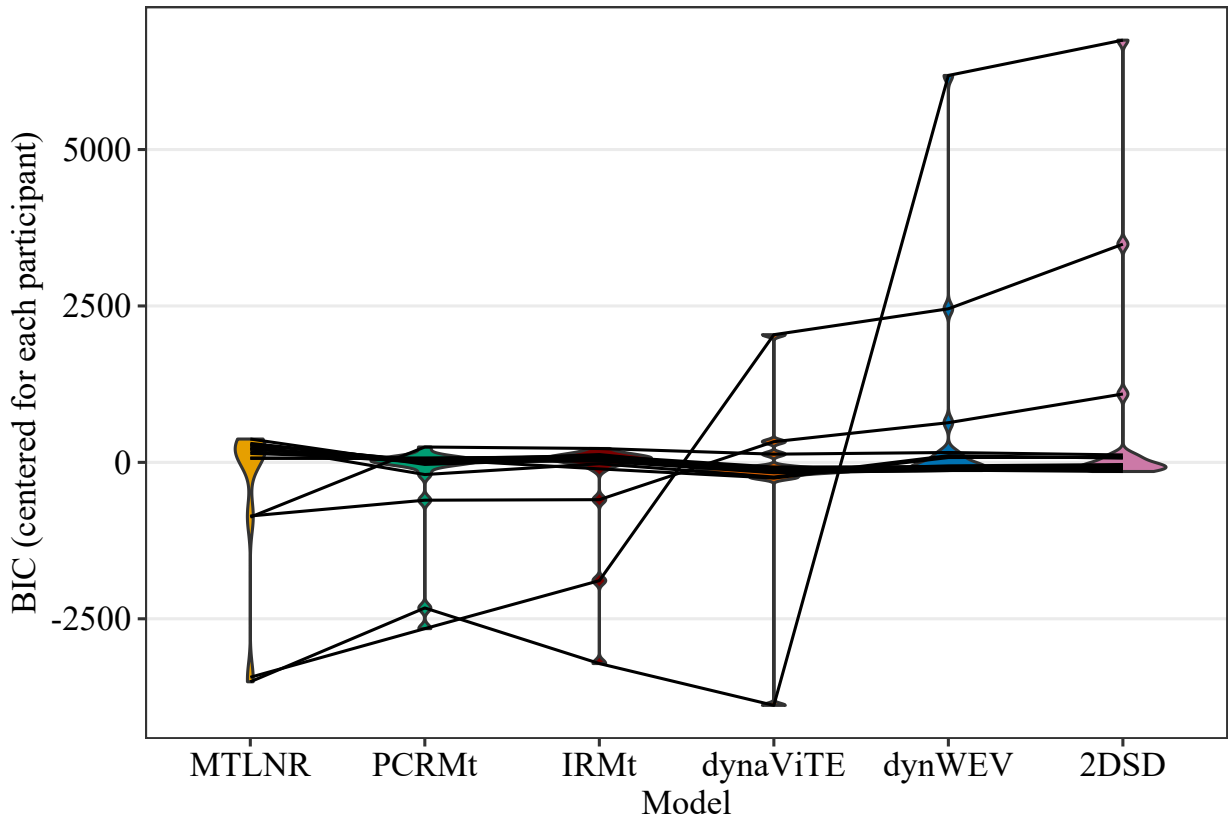


Figure A1

Distribution of BIC values, centered for each participant, across different models. Each line represents a participant.

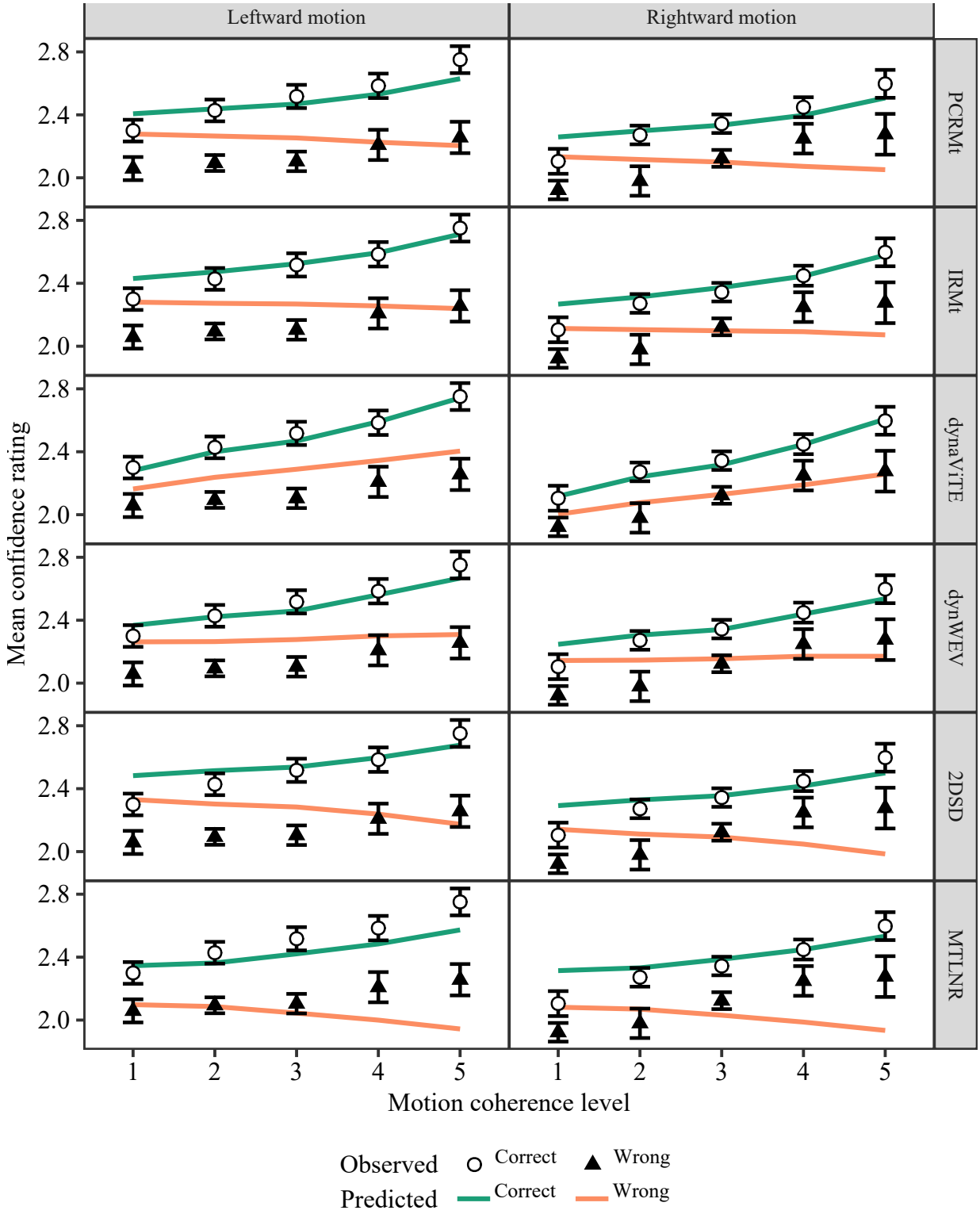


Figure A2

Observed (points) and fitted (lines) mean confidence judgments by accuracy (line color, shape) and different responses (columns). Error bars represent within-subject standard errors.

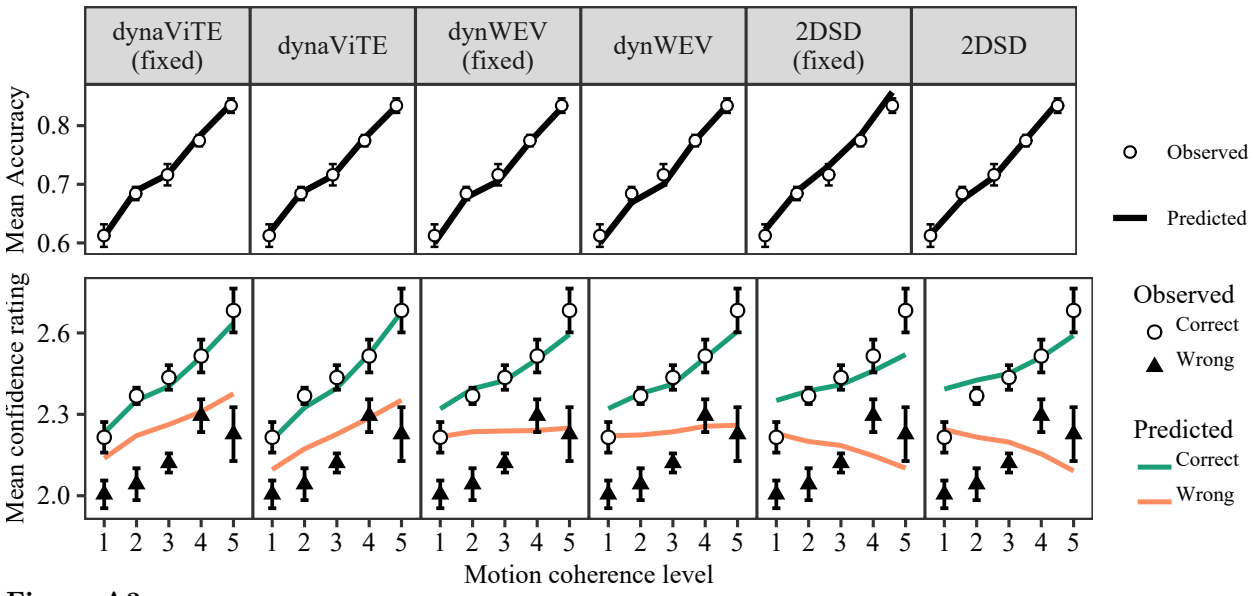


Figure A3

Accuracy (top row) and mean confidence rating (bottom row) for empirical data (points and triangles) and model predictions from the full and restricted DDM-based models (lines). Error bars represent within-subject standard errors.

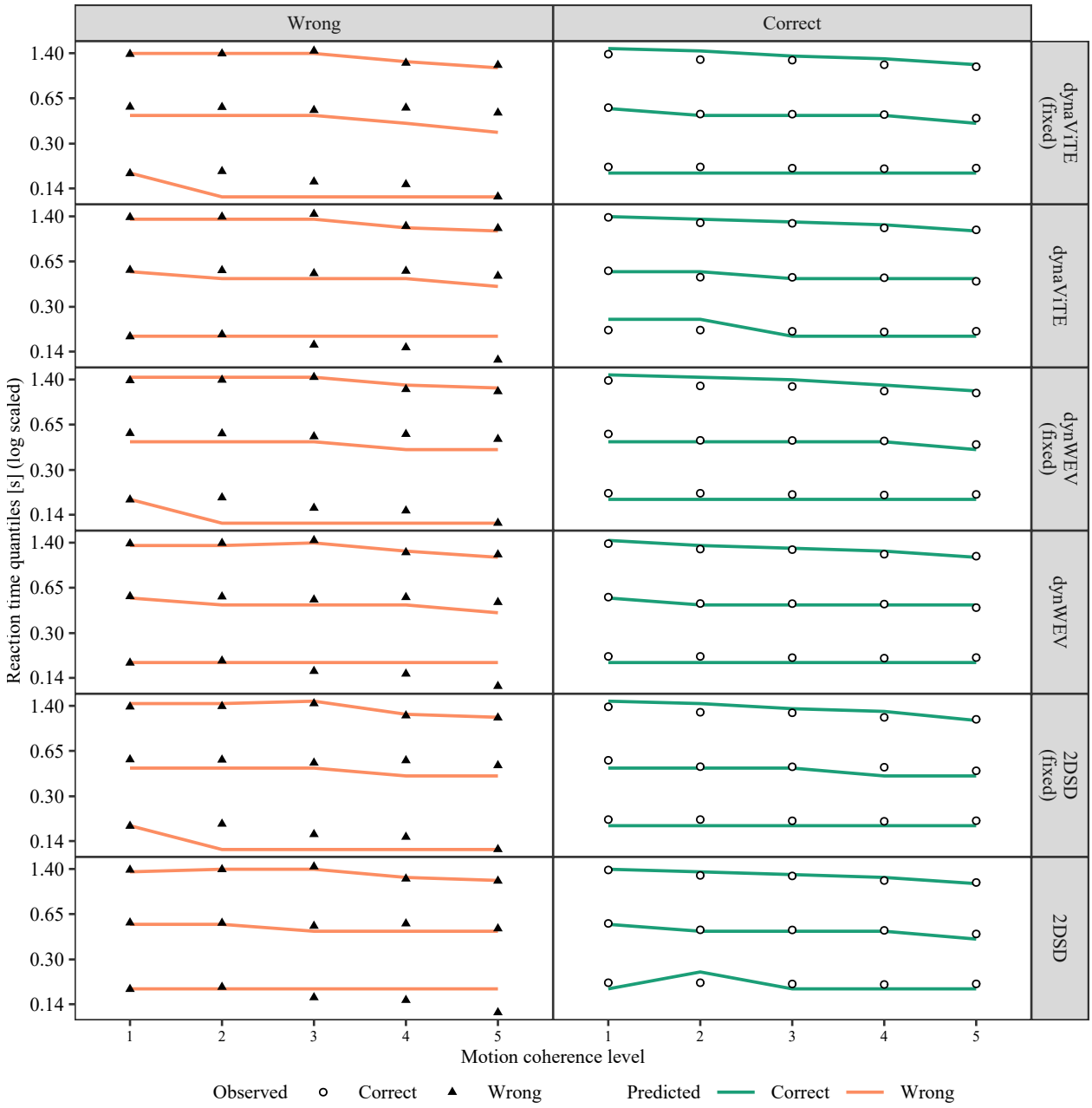


Figure A4

Response time quantiles for observed (points) and predicted (lines) response time distributions across correct and incorrect decisions (columns) and levels of stimulus discriminability (x-axis). Probabilities for quantiles: .1, .5, .9.

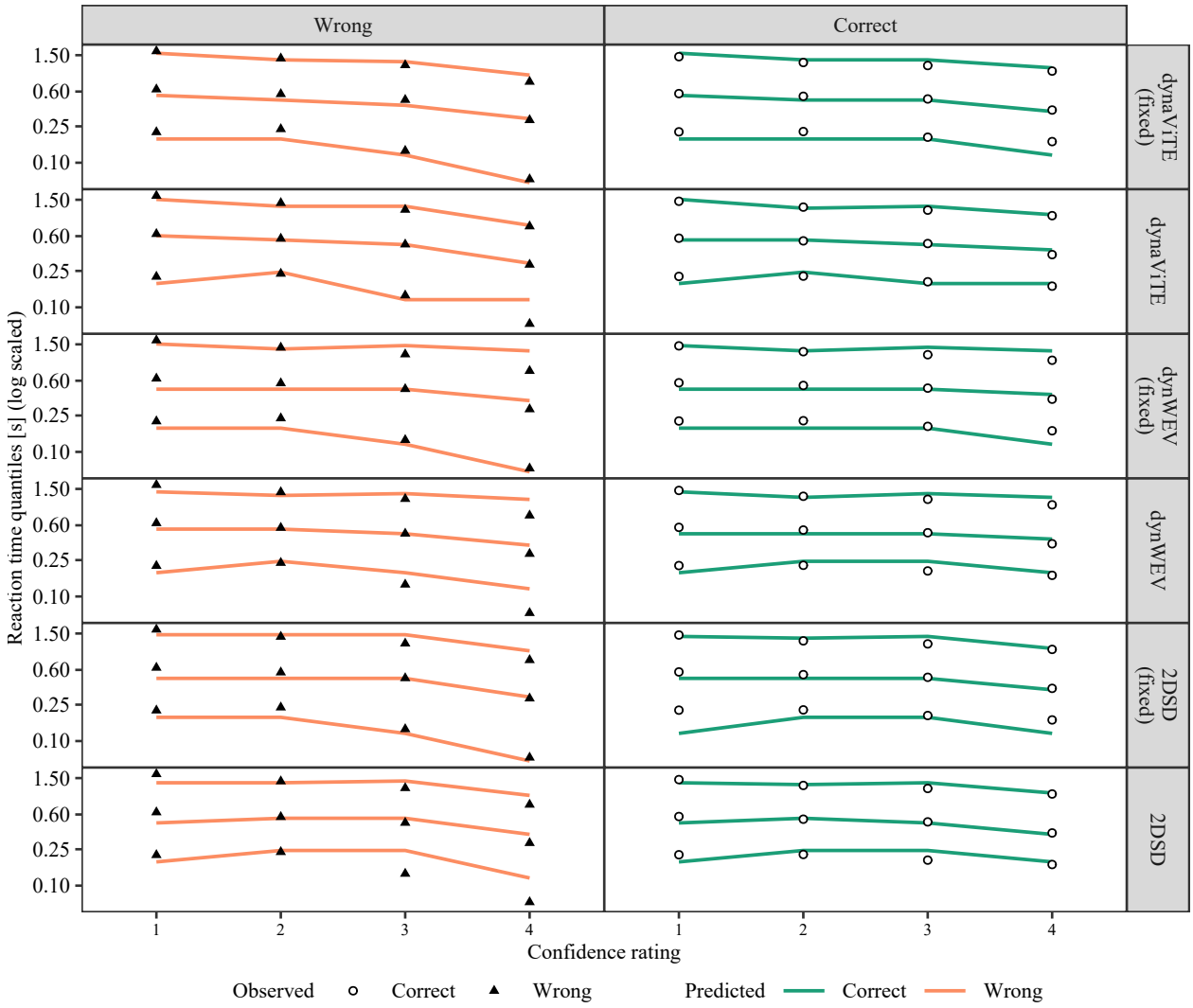


Figure A5

Response time quantiles for observed (points) and predicted (lines) response time distributions across correct and incorrect decisions (columns) and confidence ratings (x-axis). Probabilities for quantiles: .1, .5, .9.

Appendix B

Parameter Recovery

Detailed Method

For measuring parameter recovery, we generated artificial data sets from a known set of model parameters, fitted the model to the synthetic data, and compared the recovered parameters to the generating parameters.

We assumed that there were five levels of confidence (i.e., $K = 5$), and there were five levels of stimulus discriminability (i.e., $L = 5$). This means that for dynaViTE, there were $11 + 5 + 2 \cdot 5 = 26$ fitted parameters, for IRMt and PCRMt, there were $6 + 5 + 10 = 21$ parameters, and for MTLRN, there were $7 + 5 + 10 = 22$ parameters.

The generating parameter sets (except for the confidence thresholds) were sampled from parameters derived from previously conducted model fits to empirical data.

We gathered parameter sets from Hellmann et al. (2023, 2024), and the example in this paper. For all four models we used the estimates to the data from Hellmann et al. (2023) and the estimates to the data from Law and Lee (Ng et al., 2021). In addition, dynaViTE, PCRMt, and MTLNR were additionally fitted to the data from Shekhar and Rahnev (2021, see the example in this article). Finally, for MTLNR, we also used estimates to the data from Experiment 2 in Orchard et al. (2022). In total, this resulted in 93 parameter sets for dynaViTE and PCRMt, 73 parameter sets for IRMt, and 110 parameter sets for MTLNR. For the 20 parameter sets fitted to the data of Shekhar and Rahnev (2021), which had only three experimental conditions, the means of the estimated drift rates from two consecutive conditions were used as additional experimental conditions (i.e. the fitted drift rates (ν_1, ν_2, ν_3) were mapped to $(\nu_1, (\nu_1 + \nu_2)/2, \nu_2, (\nu_2 + \nu_3)/2\nu_3))$. For the estimates to the data from Orchard et al. (2022), which had eight difficulty levels, we used the sensitivity parameters of the third and sixth level as sensitivity parameters for the second and fourth level in the simulation. In addition, we computed the average of the first and second, the fourth and fifth, and the seventh and eighth level for the first, third, and fifth level in the simulation, respectively. For more details, see the accompanying code.

Concerning the confidence thresholds, we opted not to utilize the thresholds from the previous model fits. The reason for not using the fitted confidence thresholds was that some participants did not use all confidence categories, resulting in some thresholds being either fitted to plus or minus infinity or coinciding with one another. In contrast, we used the fact that when simulating artificial data, we can simulate the continuous confidence variable in the model and compute the confidence thresholds as quantiles of the confidence variable, given the proportions of confidence ratings. We fitted a Dirichlet distribution to the observed proportions of confidence ratings from the empirical data of all participants.

We then drew random probability vectors from the Dirichlet distribution as proportions of confidence reports, resampling if any proportion was less than 2%. The confidence thresholds were then computed as quantiles of the confidence variable in the simulated data set. This procedure ensured that each confidence level contained a non-zero proportion of responses.

To assess the number of trials necessary to recover the parameters, we sampled either 50, 100, 200, or 500 trials per condition and stimulus identity. For two stimulus identities and five levels of discriminability, this leads to 500, 1,000, 2,000, and 5,000 trials per simulated data set for five discriminability conditions.

For each number of artificial trials, we sampled 100 parameter sets and generated one data set per parameter set.

To measure parameter recovery performance, we computed the concordance correlation coefficient (CCC; Lin, 1989), which, in contrast to Pearson correlation, is reduced by non-zero intercepts and non-unit slopes. Therefore, in contrast to Pearson's correlation coefficient, it is sensitive to deviations from the identity line.

Recovery Plots

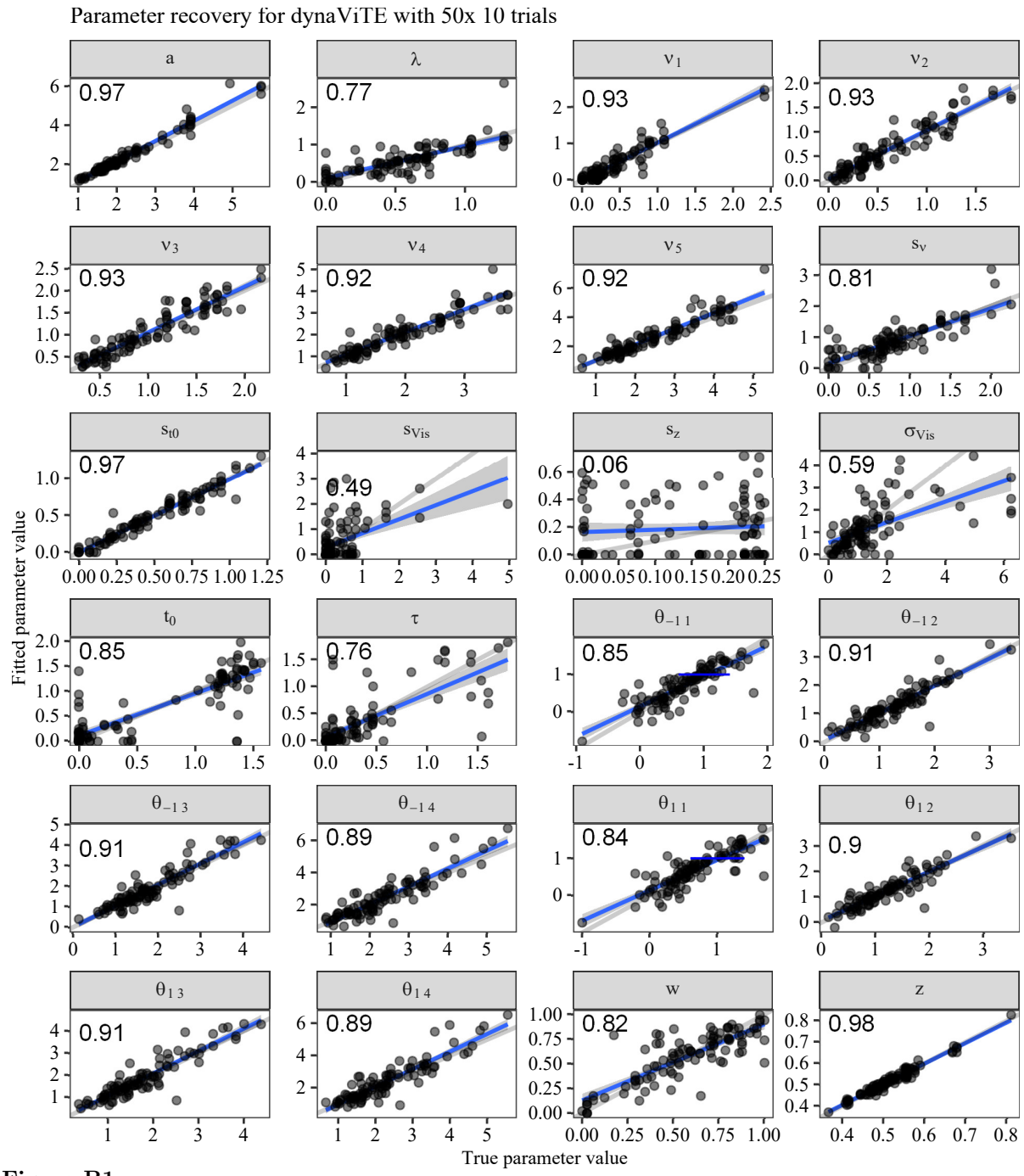


Figure B1

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

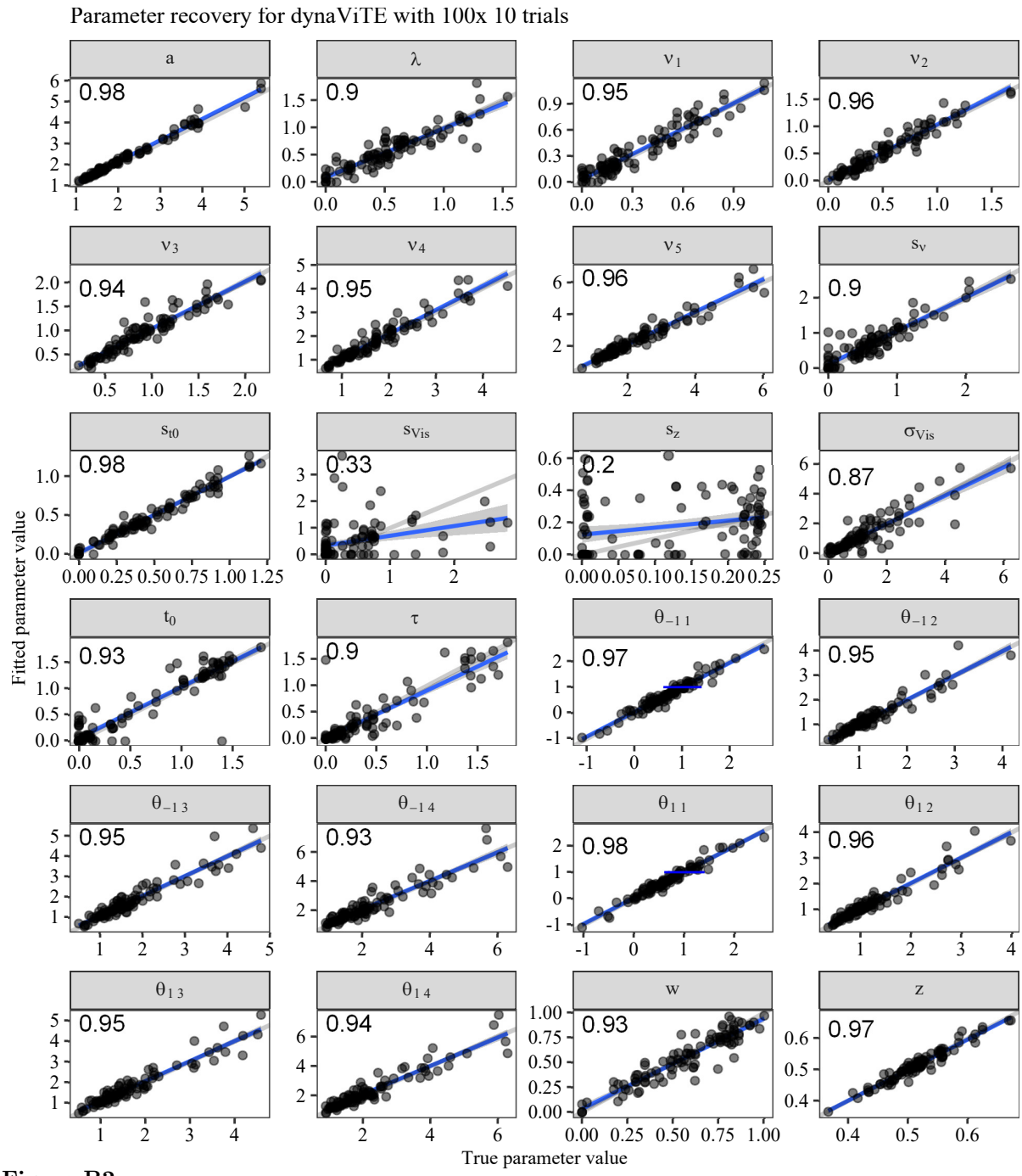


Figure B2

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

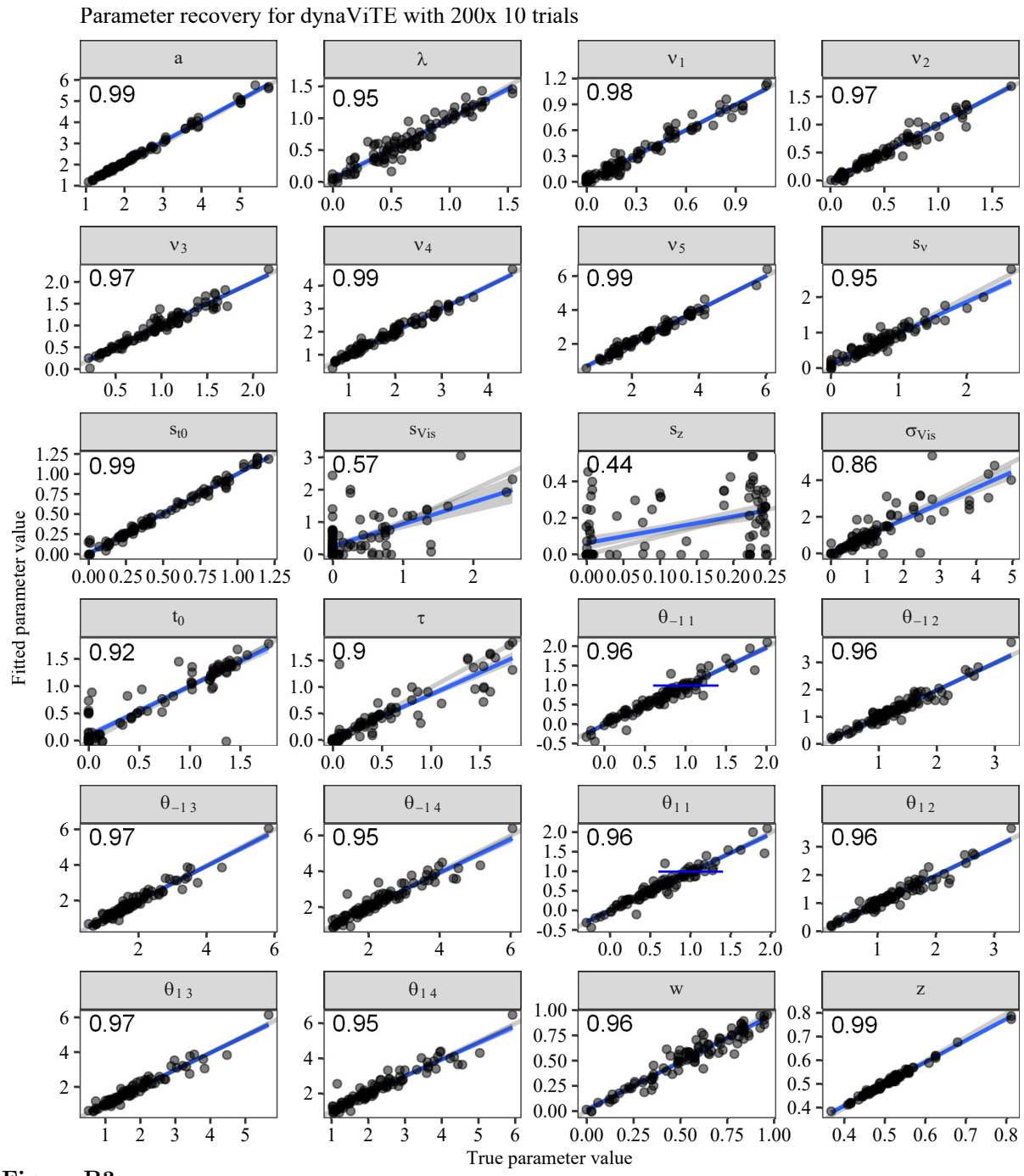


Figure B3

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

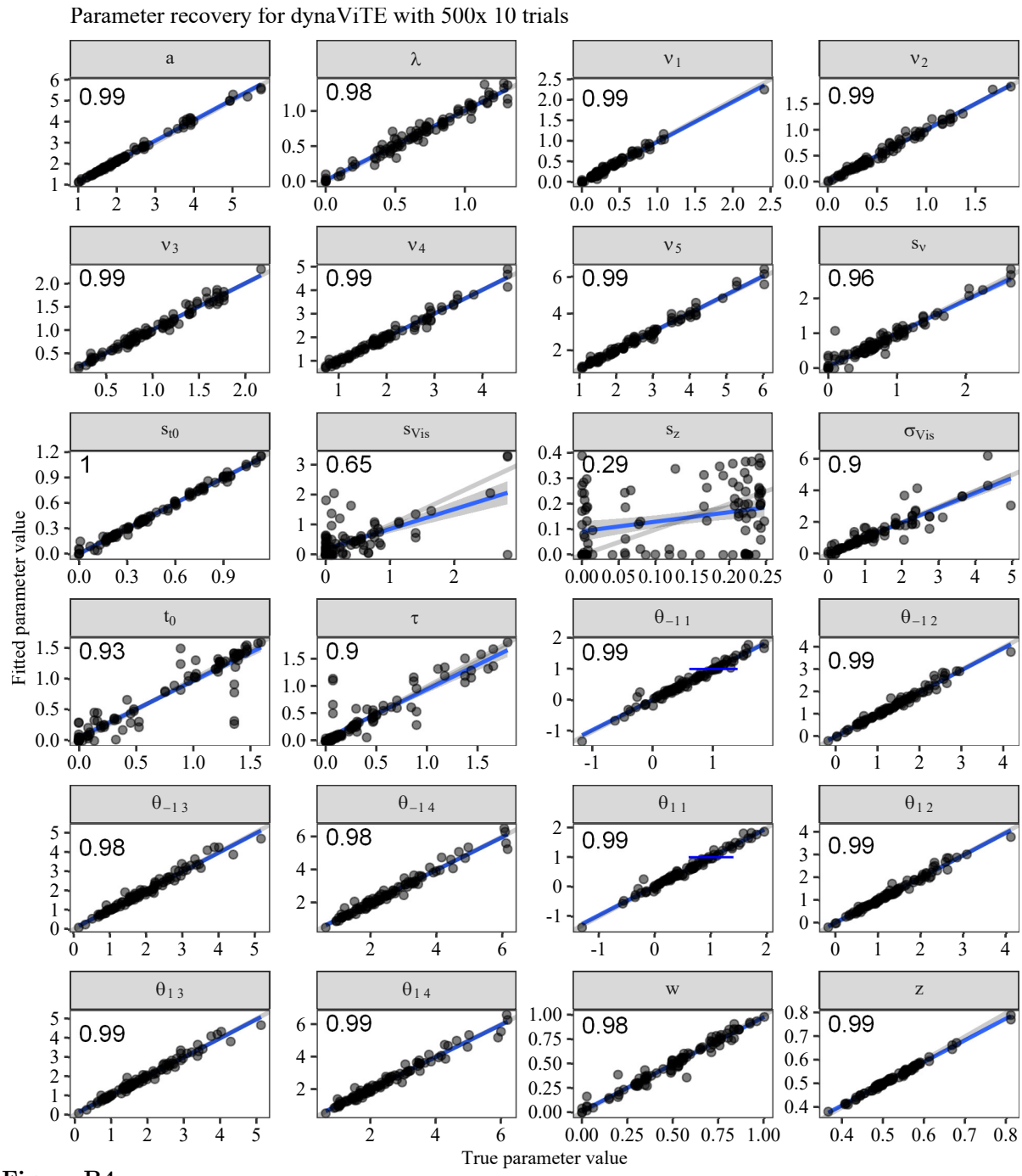


Figure B4

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

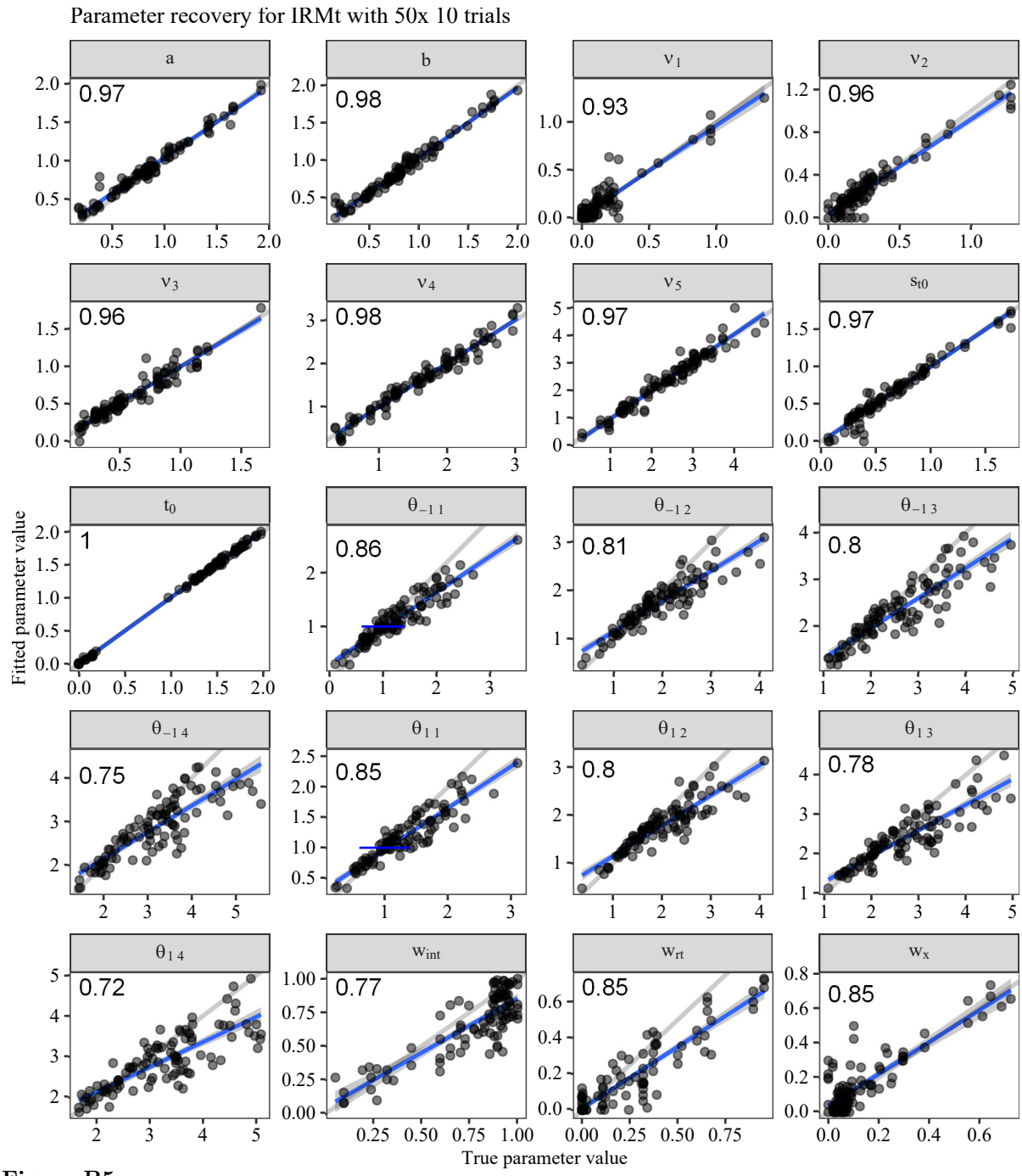


Figure B5

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

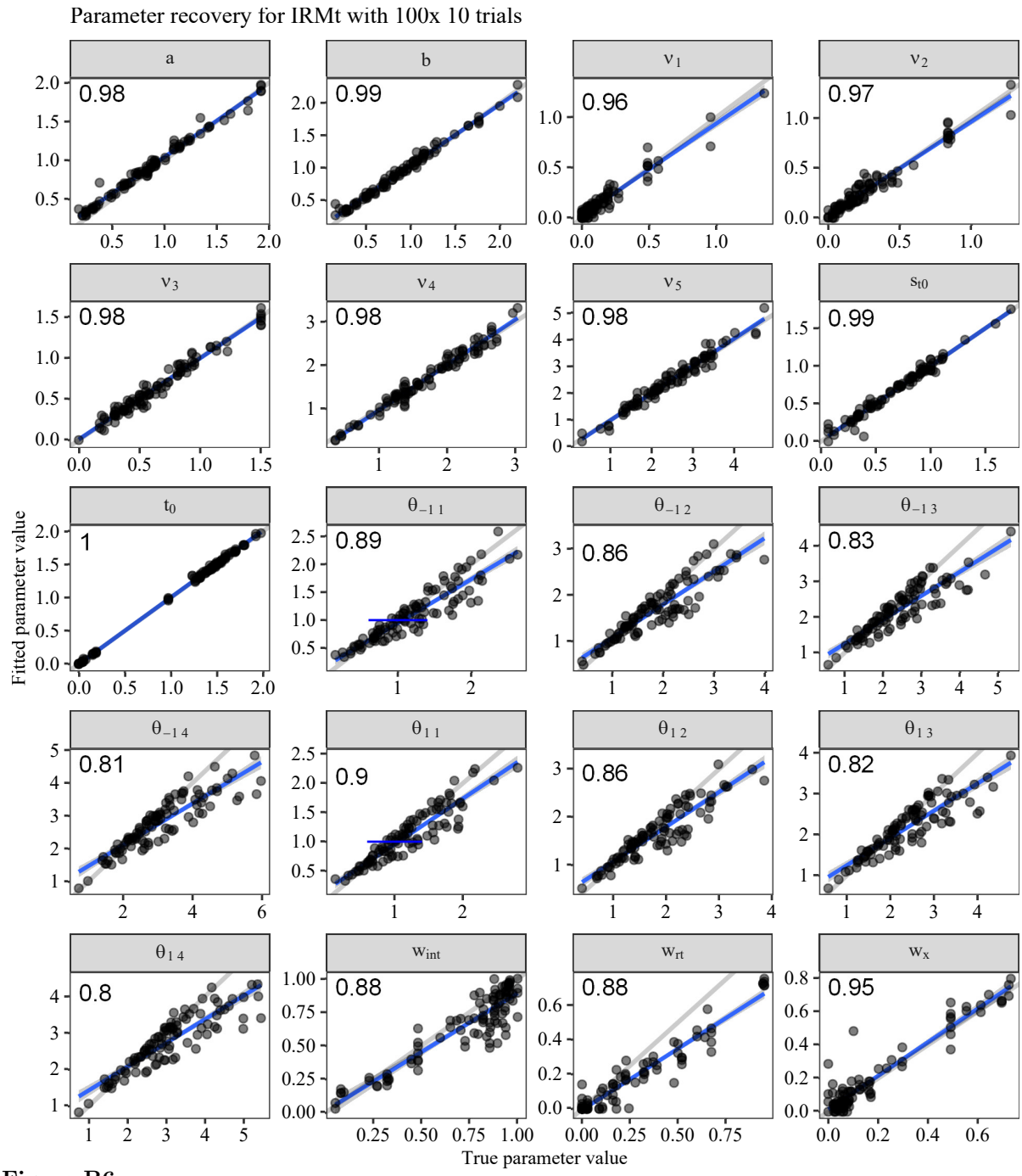


Figure B6

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

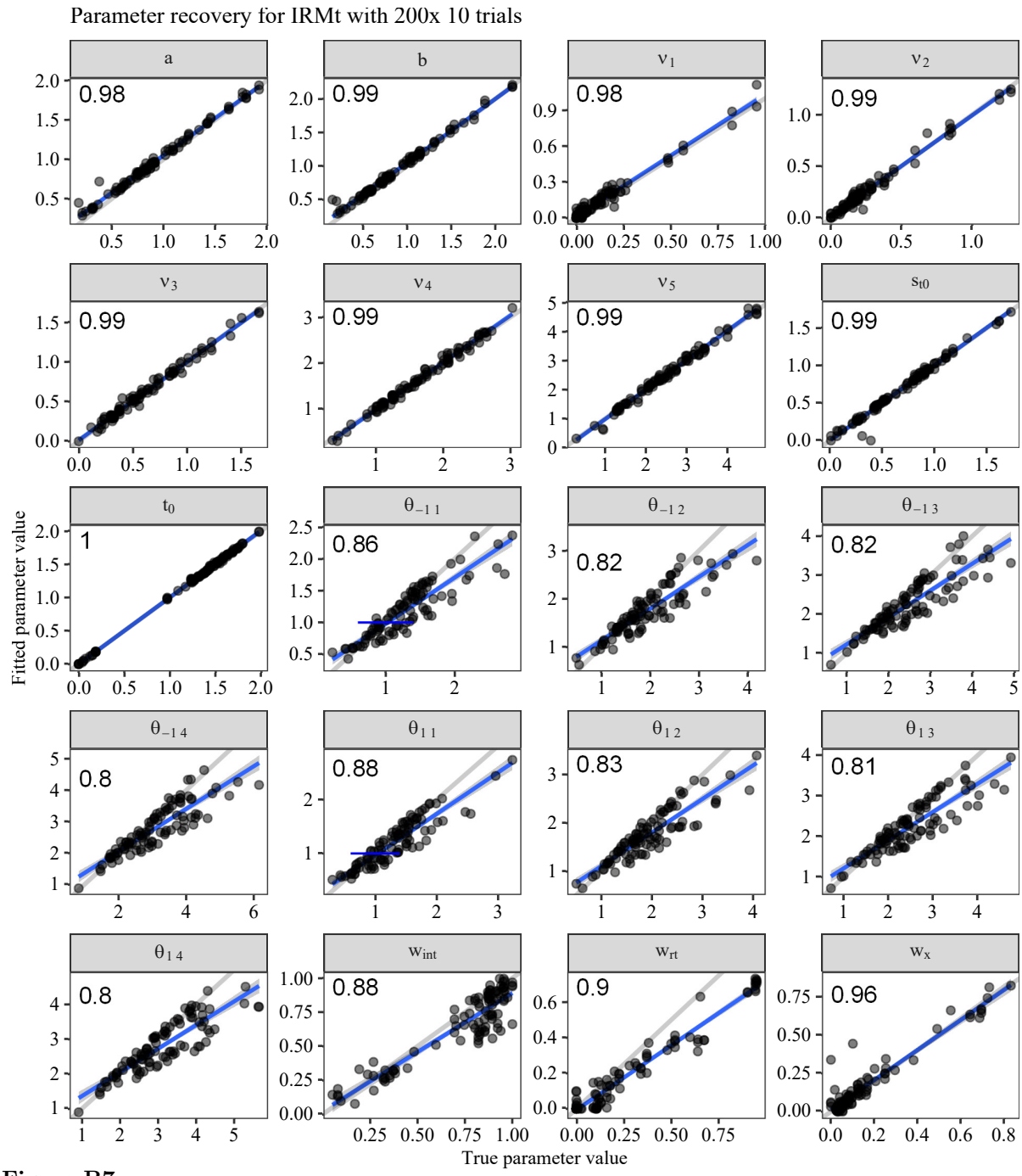


Figure B7

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

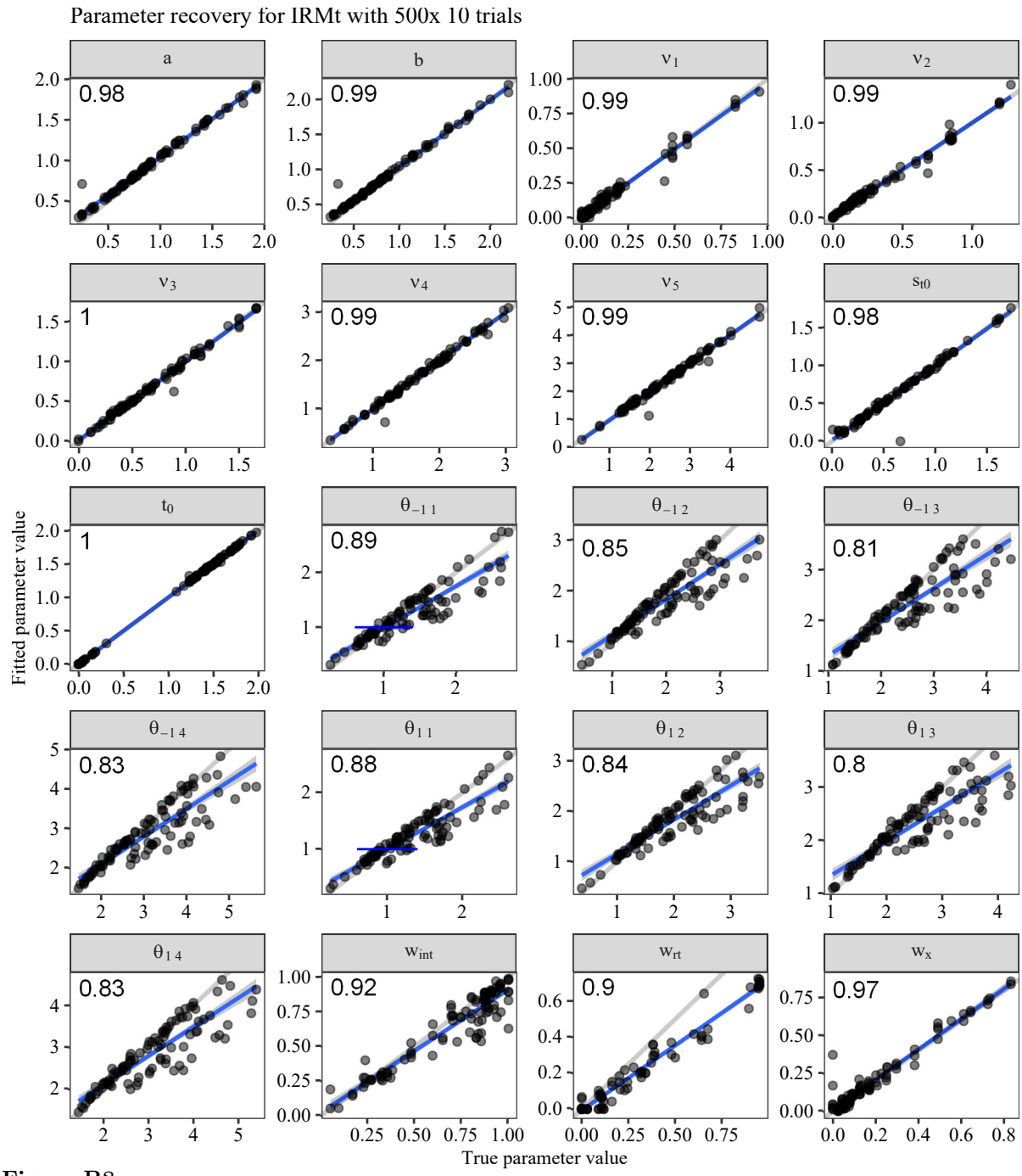


Figure B8

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

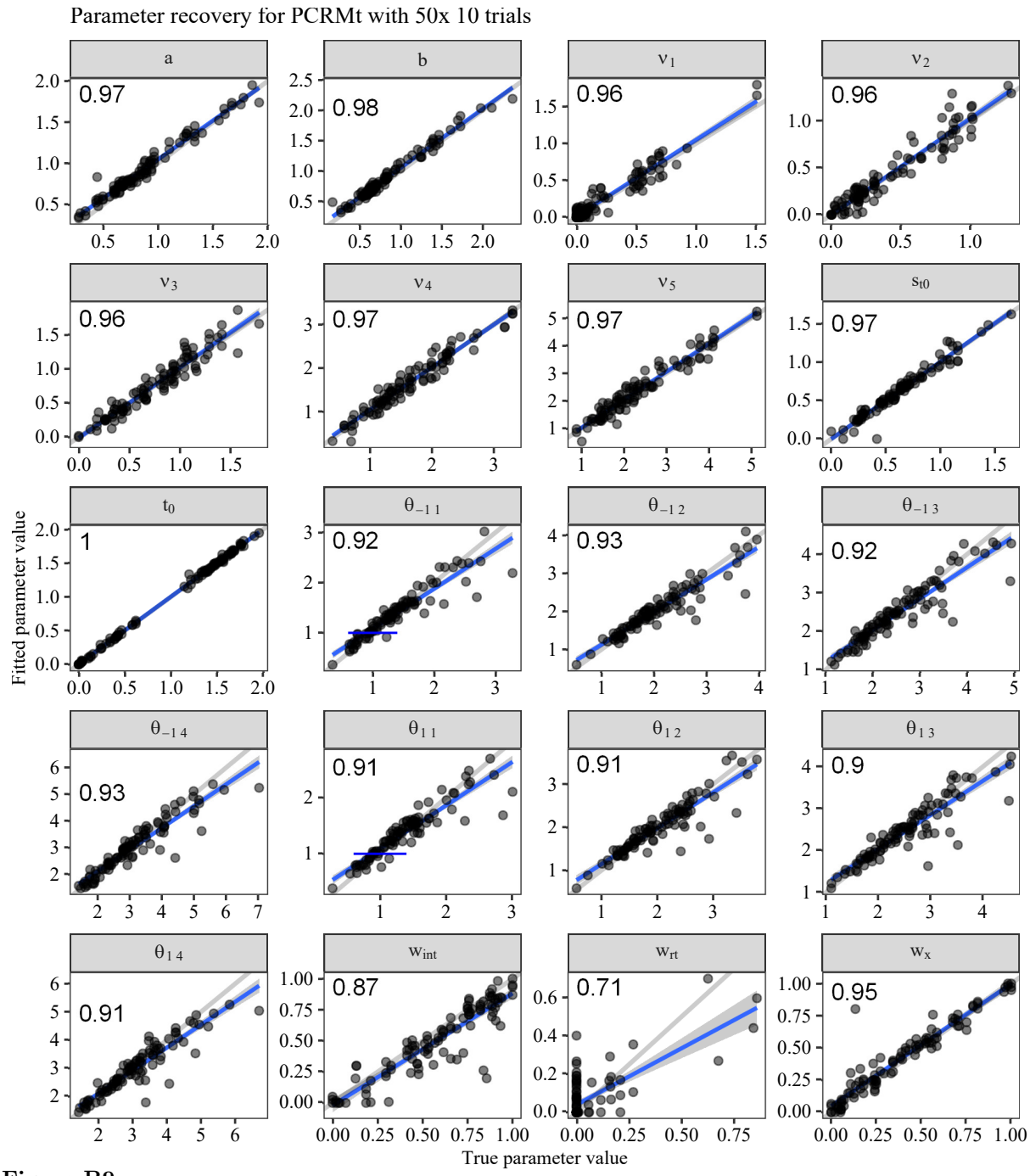


Figure B9

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

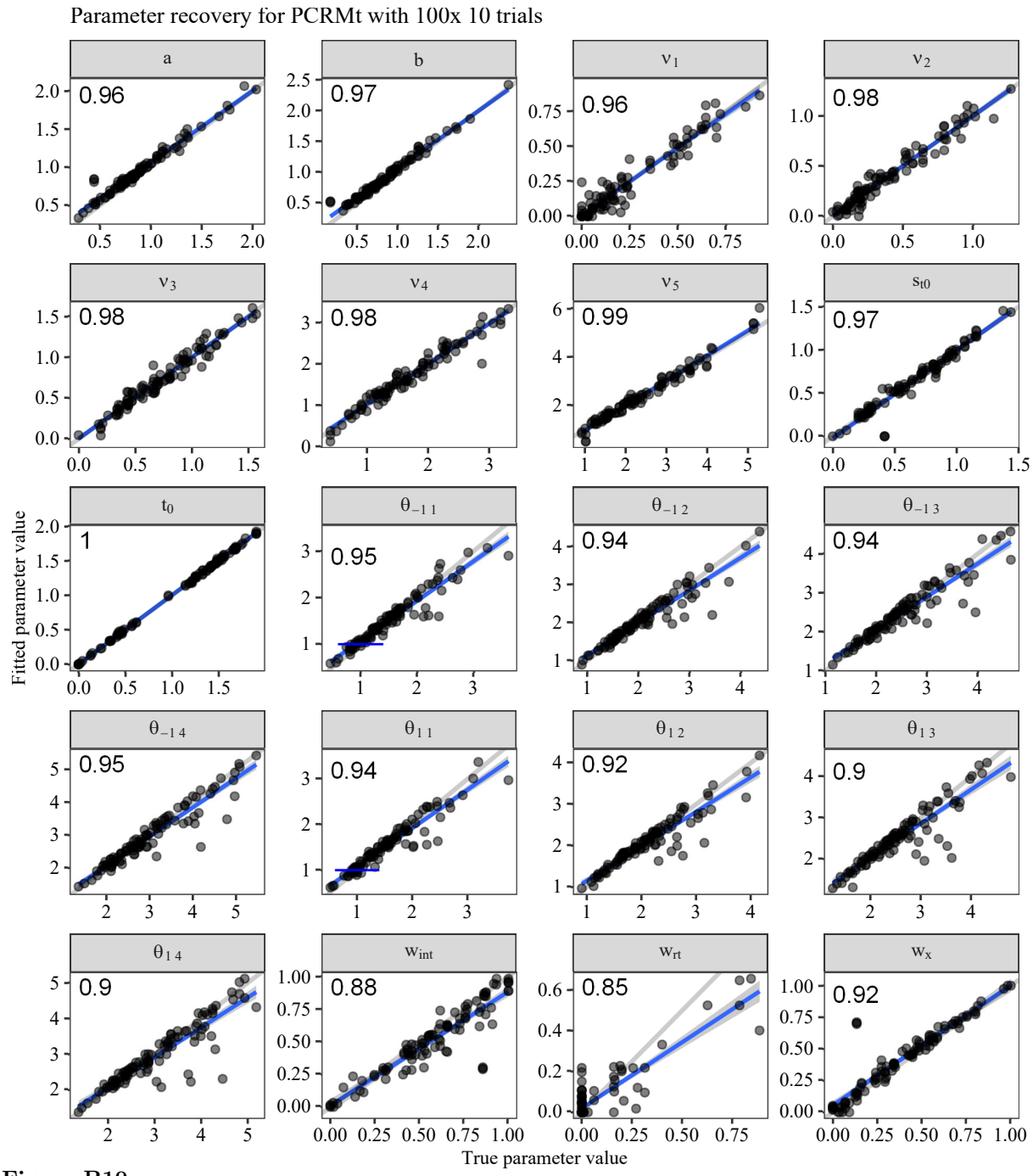


Figure B10

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

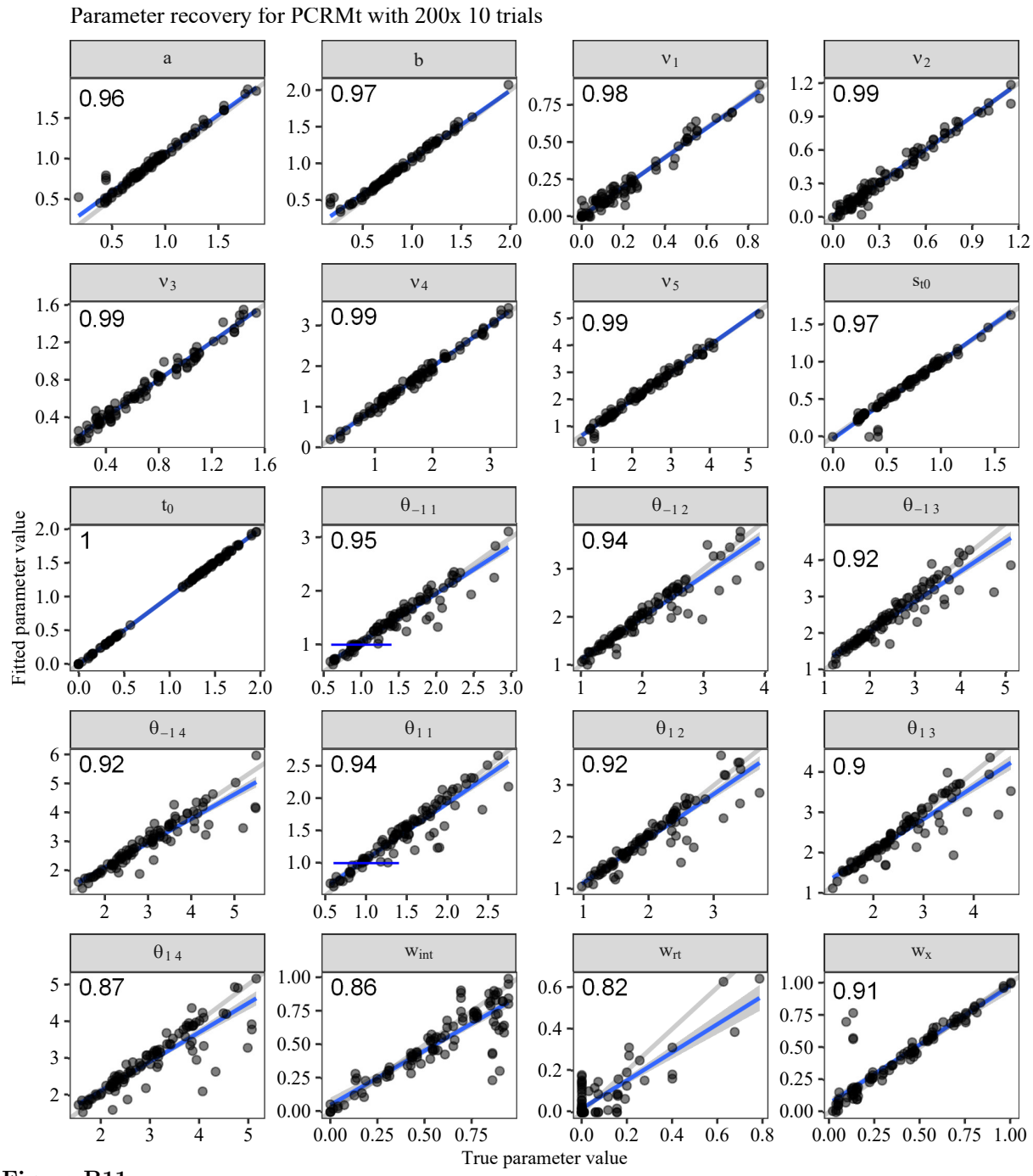


Figure B11

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

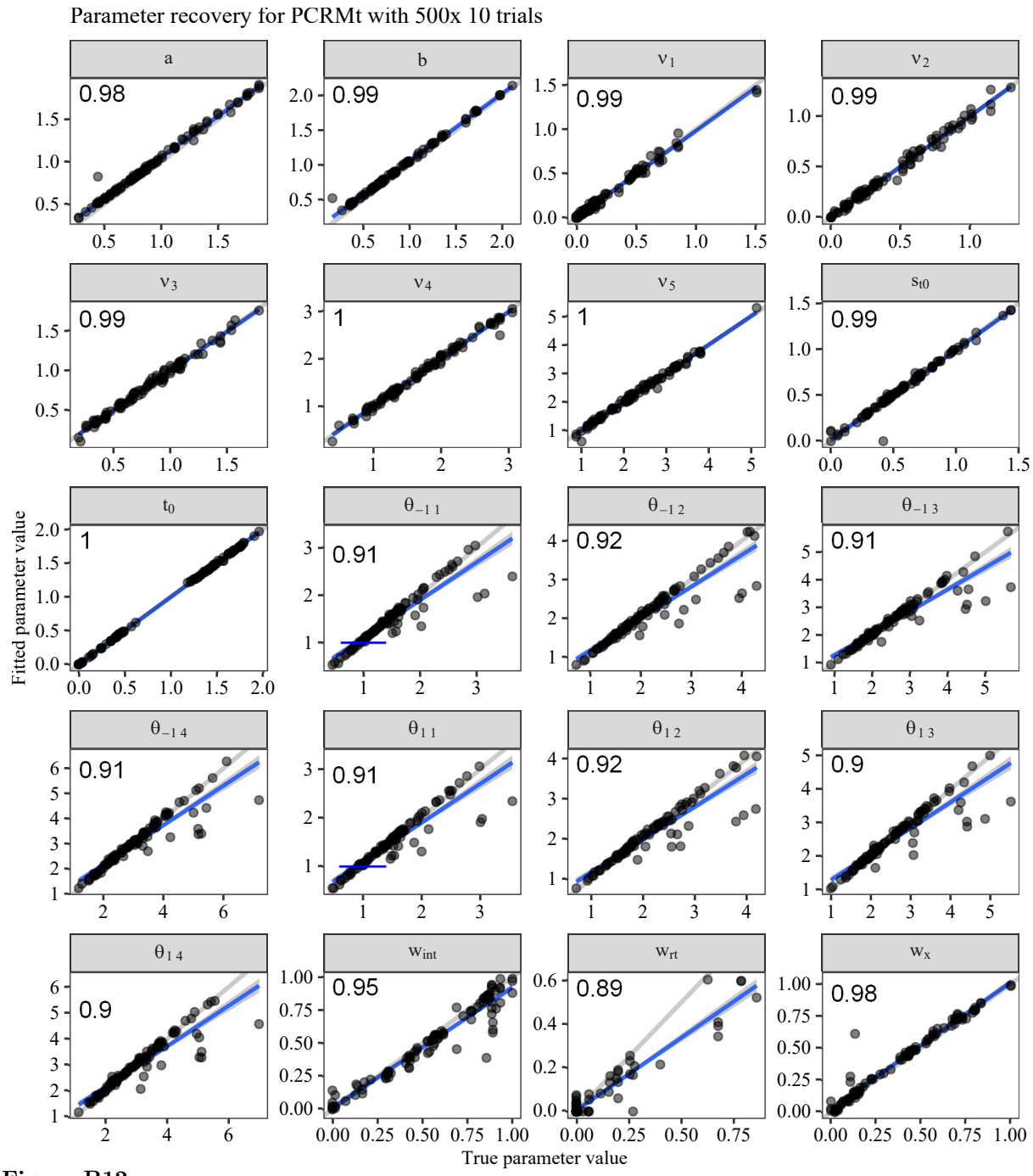


Figure B12

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

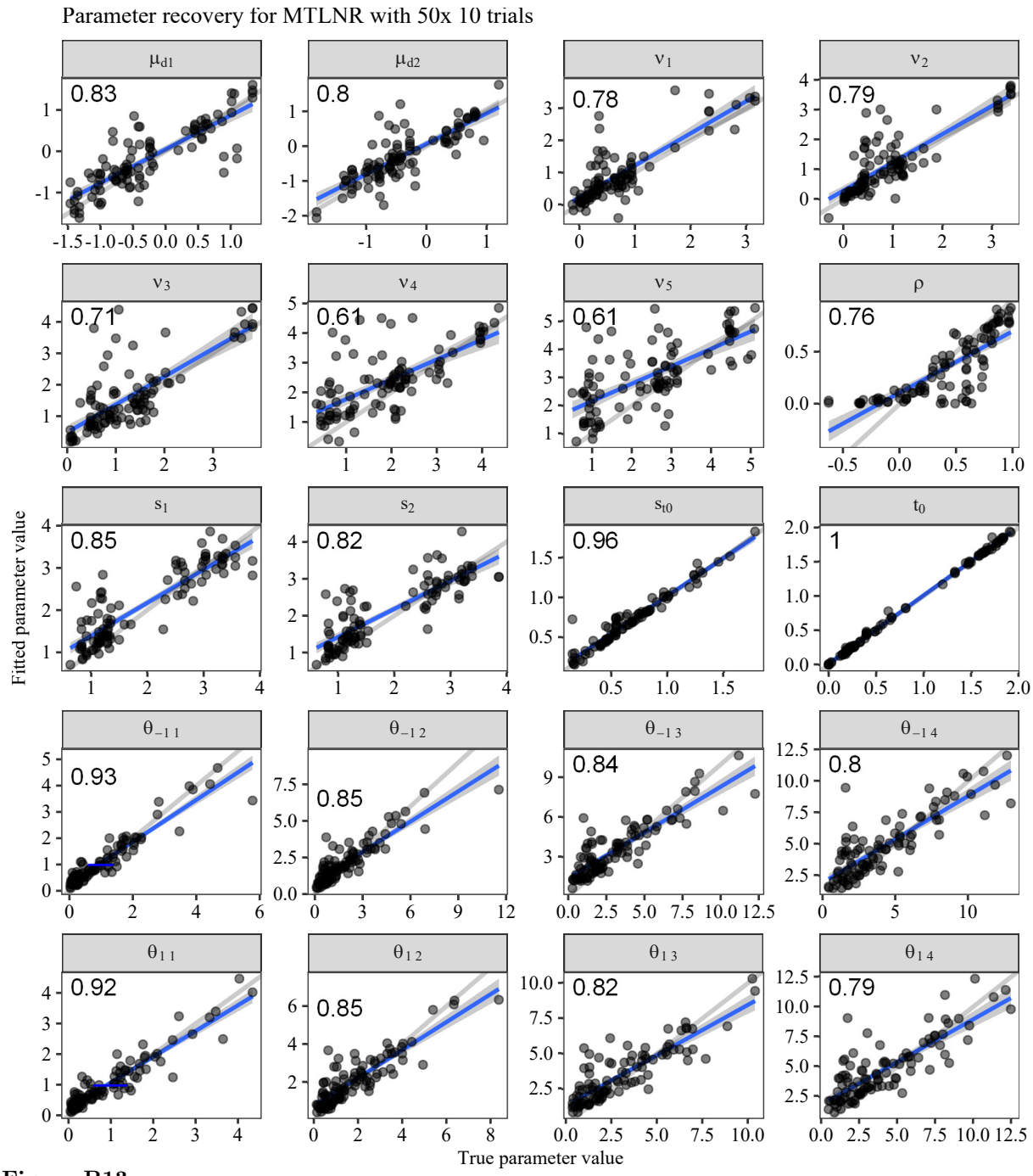


Figure B13

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

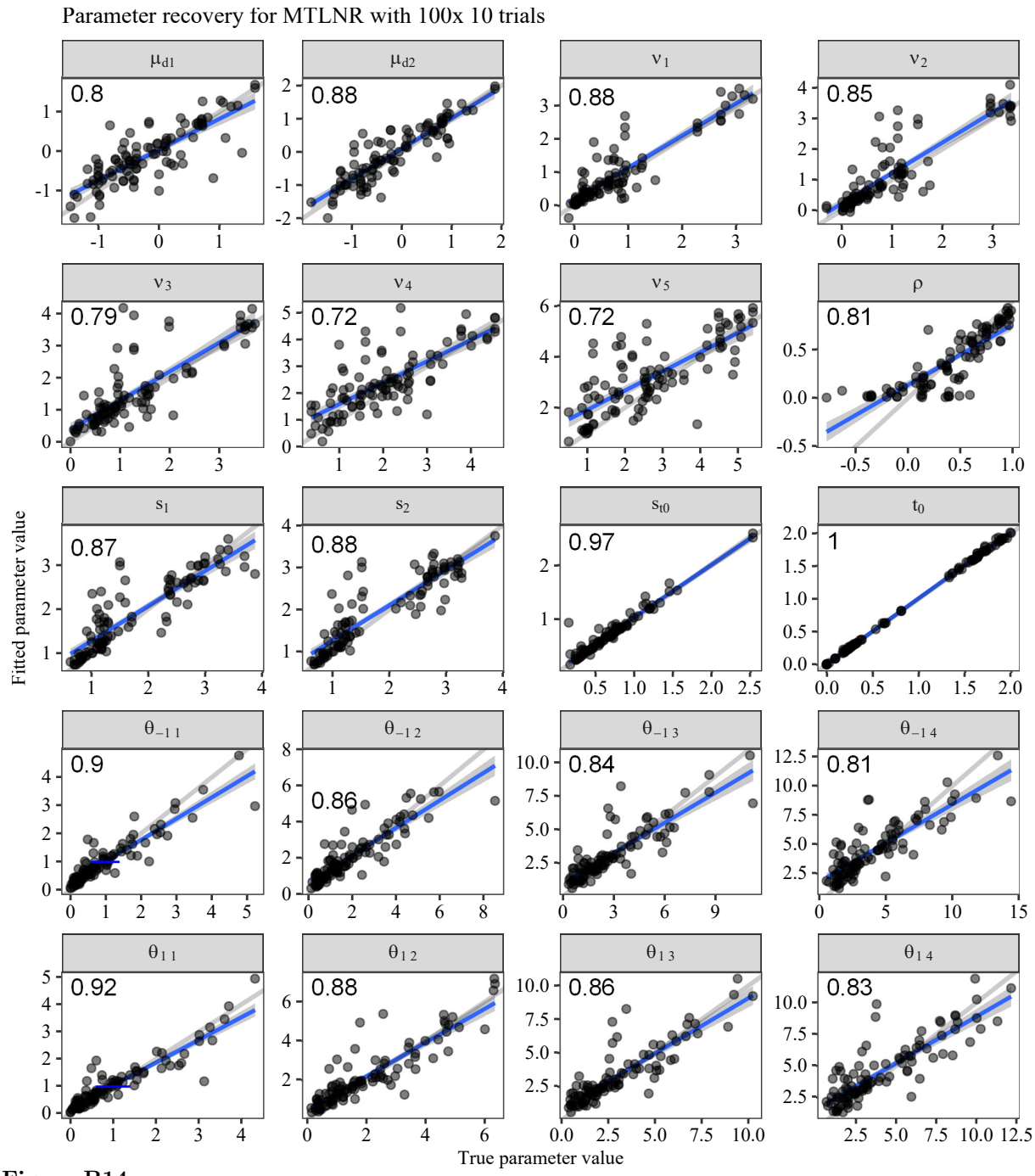


Figure B14

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

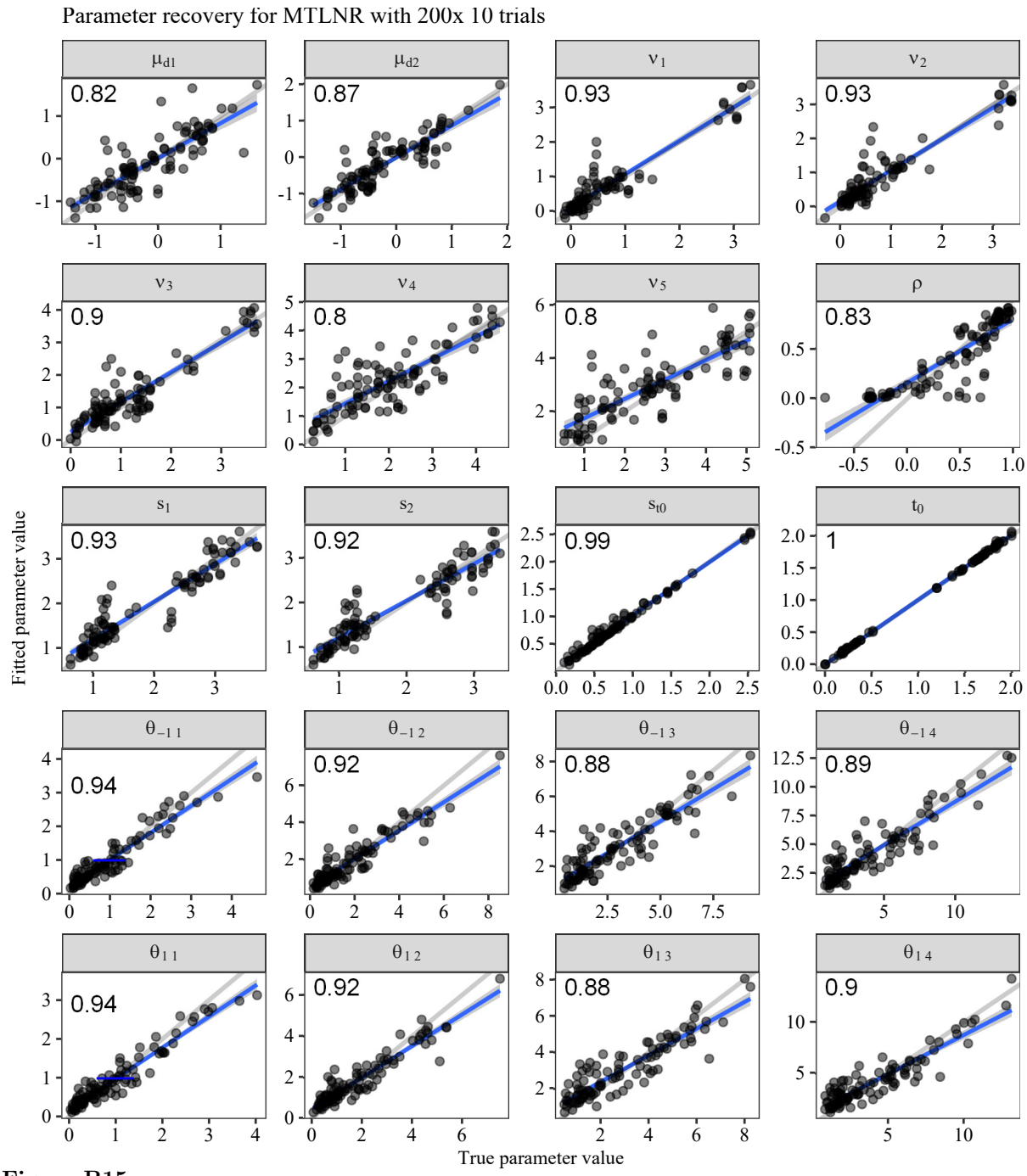


Figure B15

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

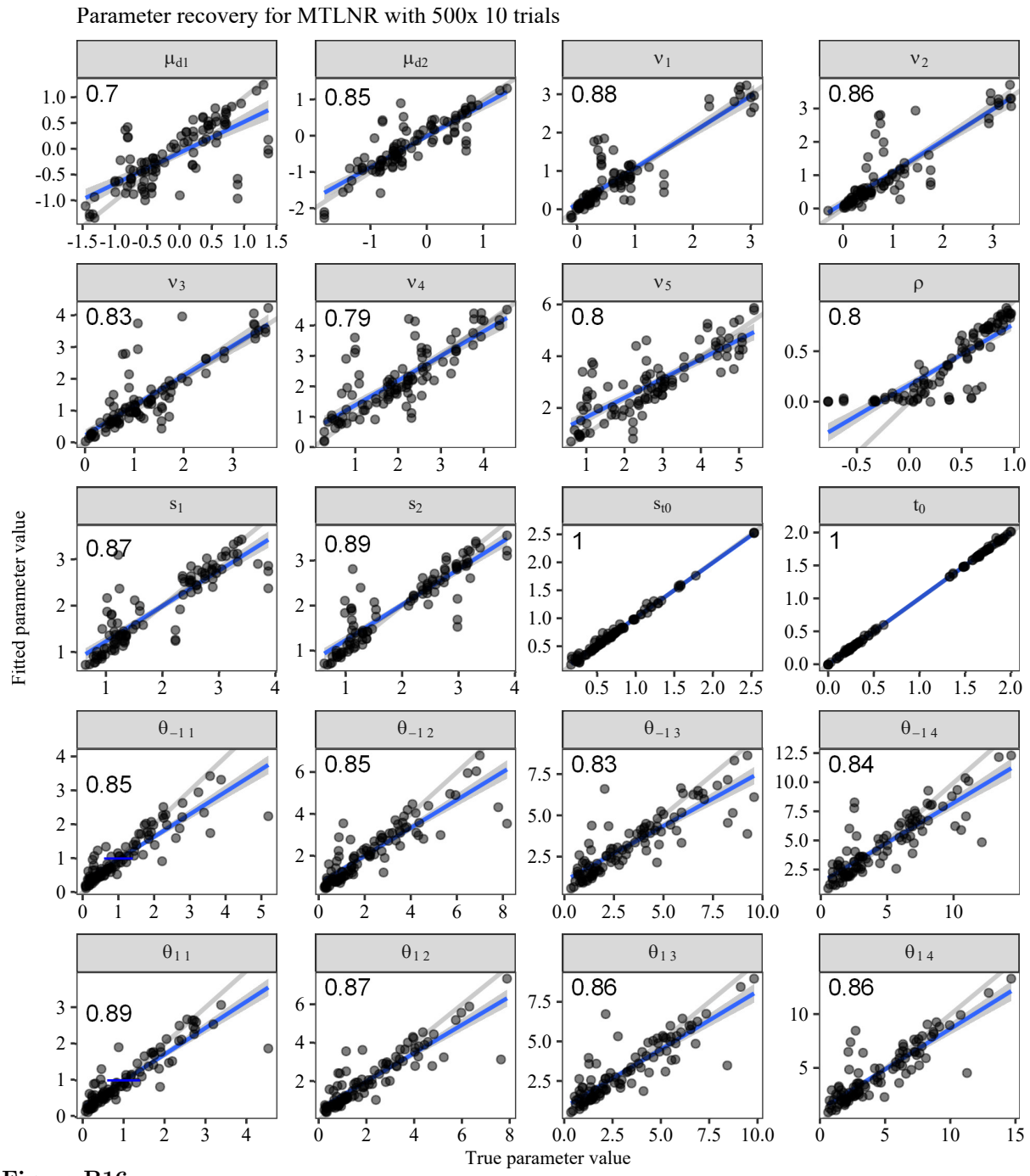


Figure B16

Recovered vs. true generative parameters across parameters. Each point represents one simulated parameter and data set. The blue line and shaded area show a linear regression line with 95% confidence band. The grey line shows the identity line. Numbers in the panels show the concordance correlation coefficient for the parameter.

Appendix C

Model Recovery

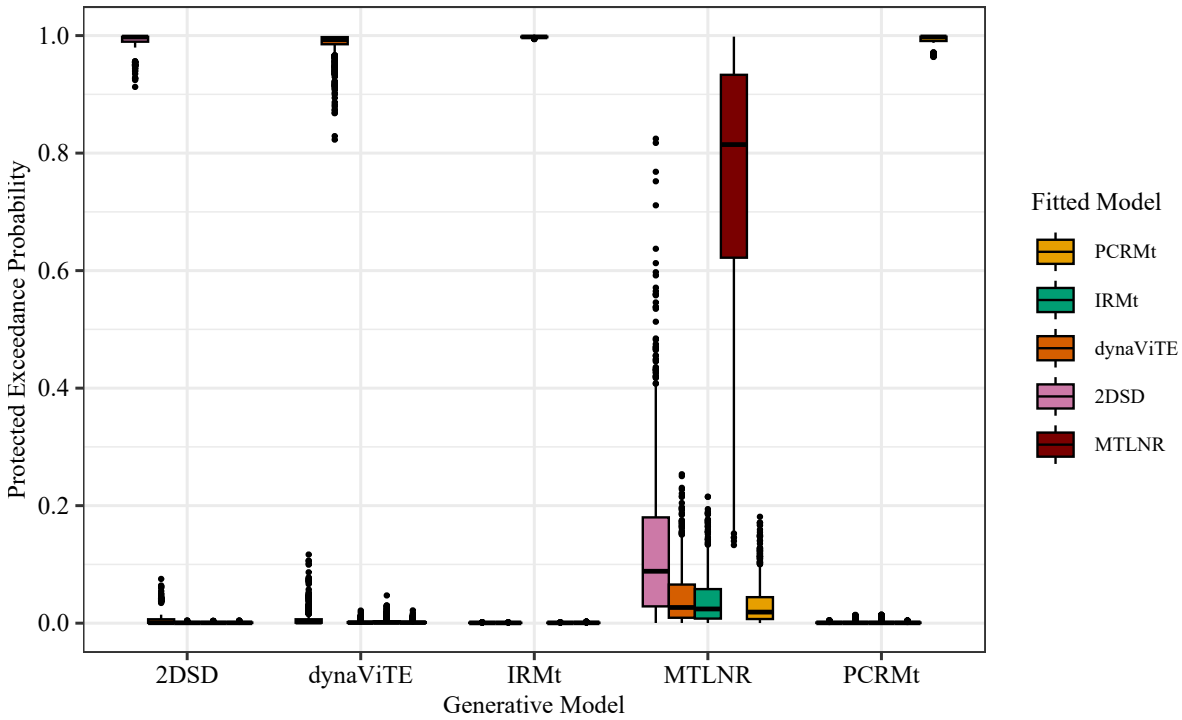


Figure C1

Bootstrapped protected exceedance probabilities (PEP).