Bayesian Analysis (2023) **TBA**, Number TBA, pp. 1–28

Simulation-Based Calibration Checking for Bayesian Computation: The Choice of Test Quantities Shapes Sensitivity*

Martin Modrák[†], Angie H. Moon[‡], Shinyoung Kim[§], Paul Bürkner[¶], Niko Huurre, Kateřina Faltejsková[‖], Andrew Gelman[**], and Aki Vehtari[††]

Abstract. Simulation-based calibration checking (SBC) is a practical method to validate computationally-derived posterior distributions or their approximations. In this paper, we introduce a new variant of SBC to alleviate several known problems. Our variant allows the user to in principle detect any possible issue with the posterior, while previously reported implementations could never detect large classes of problems including when the posterior is equal to the prior. This is made possible by including additional data-dependent test quantities when running SBC. We argue and demonstrate that the joint likelihood of the data is an especially useful test quantity. Some other types of test quantities and their theoretical and practical benefits are also investigated. We provide theoretical analysis of SBC, thereby providing a more complete understanding of the underlying statistical mechanisms. We also bring attention to a relatively common mistake in the literature and clarify the difference between SBC and checks based on the data-averaged posterior. We support our recommendations with numerical case studies on a multivariate normal example and a case study in implementing an ordered simplex data type for use with Hamiltonian Monte Carlo. The SBC variant introduced in this paper is implemented in the `SBC` R package.

Keywords: calibration, probabilistic programming, software testing.

MSC2020 subject classifications: 62C10.

*We thank Garud Iyengar and Henry Lam for helpful discussions on theory and proofs (Appendix A) and David Yao for bringing attention to stochastic ordering which motivated delving into families of test quantities. We thank Feras Saad for alerting us that a previous version of this paper did not correctly reflect their contributions. This work was supported by the ELIXIR CZ research infrastructure project (Ministry of Youth, Education and Sports of the Czech Republic, Grant No: LM2023055), including access to computing and storage facilities; the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — EXC-2075 - 390740016 (the Stuttgart Cluster of Excellence SimTech); the U.S. National Science Foundation, National Institutes of Health, and Office of Naval Research; and the Academy of Finland Flagship programme: Finnish Center for Artificial Intelligence.

[†]Institute of Microbiology of the Czech Academy of Sciences, martin.modrak@biomed.cas.cz
[‡]Massachusetts Institute of Technology
[§]Department of Computer Science, Kookmin University
[¶]Department of Statistics, Technical University of Dortmund
[‖]Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences
[**]Department of Statistics and Department of Political Science, Columbia University
[††]Department of Computer Science, Aalto University

 https://doi.org/10.1214/23-BA1404

1 Introduction

Simulation-based calibration checking (SBC; Talts et al. 2020) is a method to validate Bayesian computation, extending ideas from Cook et al. (2006).[1] While SBC is primarily intended for validating sampling algorithms such as MCMC, it can be used for validating any method implementing or approximating Bayesian inference. Published applications include variational inference (Yao et al., 2018) and neural posterior approximations (Radev et al., 2023).

Throughout this paper we assume an implicit and fixed Bayesian statistical model π with data space Y and parameter space Θ. For $y \in Y, \theta \in \Theta$ the model implies the following joint, marginal, and posterior distributions:

$$\begin{aligned}
\pi_{\text{joint}}(y, \theta) &= \pi_{\text{obs}}(y|\theta)\pi_{\text{prior}}(\theta), \\
\pi_{\text{marg}}(y) &= \int_{\Theta} \mathrm{d}\theta \, \pi_{\text{obs}}(y|\theta)\pi_{\text{prior}}(\theta), \\
\pi_{\text{post}}(\theta|y) &= \frac{\pi_{\text{obs}}(y|\theta)\pi_{\text{prior}}(\theta)}{\pi_{\text{marg}}(y)}.
\end{aligned}$$

Typically, the posterior distribution π_{post} is the target of inference but is impossible to evaluate directly. While many computational approaches exist for sampling from the posterior or its approximations, they may fail to provide a correct answer. Problems can arise from errors in how the algorithm or the statistical model are encoded or from inherent inability of the computational method to correctly handle a given model with a given dataset.

1.1 Self-consistency of Bayesian models

To discover problems with computation, several classes of checks can be derived from self-consistency properties of statistical models. One such property concerns the data-averaged posterior (Geweke, 2004):

$$\pi_{\text{prior}}(\theta) = \int_{Y} \mathrm{d}y \int_{\Theta} \mathrm{d}\tilde{\theta} \, \pi_{\text{post}}(\theta|y)\pi_{\text{obs}}(y|\tilde{\theta})\pi_{\text{prior}}(\tilde{\theta}). \tag{1}$$

SBC relies on a different property that involves the joint distribution of prior and posterior samples from the same model (Cook et al., 2006):

$$\pi_{\text{SBC}}(y, \theta, \tilde{\theta}) = \pi_{\text{prior}}(\tilde{\theta})\pi_{\text{obs}}(y|\tilde{\theta})\pi_{\text{post}}(\theta|y). \tag{2}$$

Since $\pi_{\text{obs}}(y|\tilde{\theta})\pi_{\text{prior}}(\tilde{\theta}) = \pi_{\text{marg}}(y)\,\pi_{\text{post}}(\tilde{\theta}|y)$, this implies,

$$\pi_{\text{SBC}}(y, \theta, \tilde{\theta}) = \pi_{\text{marg}}(y)\pi_{\text{post}}(\theta|y)\pi_{\text{post}}(\tilde{\theta}|y). \tag{3}$$

[1] The term in the literature is "simulation-based calibration"; here we have added the word "checking" to emphasize that these methods do not themselves produce calibration; rather, they measure departure from calibration.

Equation (3) immediately shows that conditional on a specific data $y \in Y$, the distributions of θ and $\tilde{\theta}$ in (2) and (3) are identical. In general, SBC-like checks are sensitive to different deviations from the correct posterior than checks based on the data-averaged posterior (see Section 3.5 for more details). The two families of checks coincide when Y has just a single element as in this case both reduce to directly comparing two distributions.

SBC and related methods employ two different implementations of the same statistical model and check if the results have the same distribution conditional on data. The first step is to define a *generator* capable of directly simulating draws from $\pi_{\text{prior}}(\tilde{\theta})$ and $\pi_{\text{obs}}(y|\tilde{\theta})$, and the second step is to define a *probabilistic program* that, in combination with a given *posterior approximation algorithm*, samples from the posterior distribution $\pi_{\text{post}}(\theta|y)$. Each simulation from the generator yields,

$$\begin{aligned} \tilde{\theta}^* &\sim \pi_{\text{prior}}(\tilde{\theta}), \\ y^* &\sim \pi_{\text{obs}}(y|\tilde{\theta}^*), \\ \theta_1, \dots \theta_M &\sim \pi_{\text{post}}(\theta|y^*), \end{aligned} \tag{4}$$

where M is the number of posterior draws sampled. Where confusion is possible we use asterisk to mark a random variable. We run many such simulations and then inspect the realized distributions of $\theta_1, \dots, \theta_M$ and $\tilde{\theta}^*$ conditional on y^*. Specific calibration checking methods differ in how exactly they test the conditional equality of the two distributions.

1.2 Proposed SBC variant

SBC has been believed to be insensitive to some classes of mismatches, and as described in Talts et al. (2020) would not work for discrete variables. To remove those limitations, we argue for the following variant of the SBC check: First, project the potentially high-dimensional parameter and data space into a scalar *test quantity* $f : \Theta \times Y \to \mathbb{R}$. Second, compute the rank of the prior draw in the posterior conditional on y. Specifically, we take the number of posterior sample draws where the test quantity is lower than in the prior draw, and, if there are any ties, choosing the rank randomly among the tied positions:

$$\begin{aligned} N_{\texttt{less}} &:= \sum_{m=1}^{M} \mathbb{I}\left[f(\theta_m, y) < f(\tilde{\theta}, y)\right], \\ N_{\texttt{equals}} &:= \sum_{m=1}^{M} \mathbb{I}\left[f(\theta_m, y) = f(\tilde{\theta}, y)\right], \\ K &\sim \text{uniform}(0, N_{\texttt{equals}}), \\ N_{\texttt{total}} &:= N_{\texttt{less}} + K, \end{aligned}$$

where $\mathbb{I}[P]$ denotes the indicator function for predicate P. The procedure simplifies if there are no ties, which will be true for most practical test quantities over models with

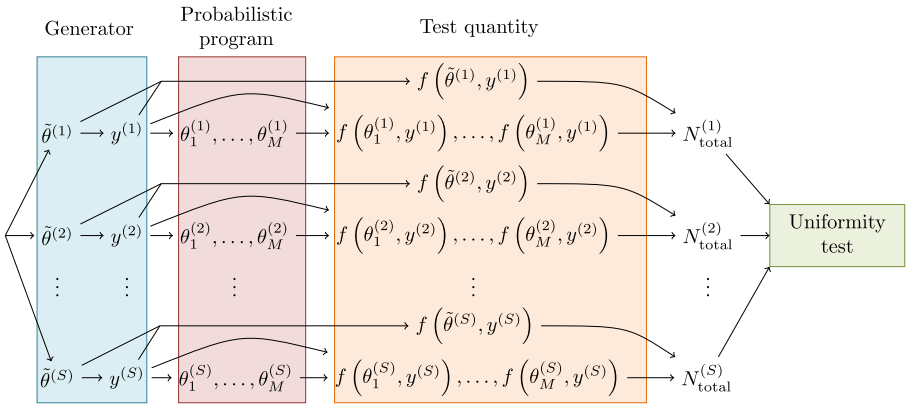

Figure 1: Schematic representation of SBC with S simulations. The generator is responsible for sampling from the prior distribution $\tilde{\theta} \sim \pi_{\text{prior}}(\tilde{\theta})$ and from the observation model $y \sim \pi_{\text{obs}}(y \mid \tilde{\theta})$. The draws from the observation model are then treated as input for the probabilistic program and the associated algorithm which takes M posterior draws $\theta_1, \dots \theta_M$. Each test quantity projects the prior draw and the posterior draws (potentially using data) onto the real line, letting us compute a single rank (N_{total}). Finally, deviations from discrete uniform distribution are assessed numerically or visually.

continuous parameter space. When no ties occur, we have $N_{\texttt{total}} = N_{\texttt{less}}$. Then, if the probabilistic program and the generator implement the same probabilistic model, we have

$$N_{\texttt{total}} \sim \text{uniform}(0, M). \tag{5}$$

See Theorems 3 and 4 for a formal statement and proof. As a result, once we obtain a set of draws from empirical distribution of $N_{\texttt{total}}$ via multiple simulations, we can perform a test for uniformity. The process is then repeated for all test quantities we want to consider. If we are using MCMC to sample from π_{post}, the posterior sample typically needs to be thinned to ensure that $\theta_1, \dots, \theta_M$ are approximately independent (Talts et al., 2020; Säilynoja et al., 2022). The overall SBC process is illustrated in Figure 1.

While it is possible to use numerical tests for uniformity with SBC, we generally prefer to use visualisations of the rank distribution as they are more informative than numerical summaries and discourage dichotomous thinking. Most prominent are rank histograms and plots of empirical cumulative distribution functions (Säilynoja et al., 2022).

Our proposed SBC variant improves upon the way SBC has been previously reported and used in two major ways:

- We let test quantities depend on both data and parameters, while previous work only considered quantities that depend on the parameters. In practice, these test quantities were almost exclusively just the individual parameters themselves.

- Previous formulations of SBC required uniformity of $N_{\texttt{less}}$. However, even if the probabilistic program is exactly correct, $N_{\texttt{less}}$ will not be uniform if $\Pr(N_{\texttt{equals}} > 0) > 0$, that is, if ties can occur. With our improved SBC procedure, we can handle test quantities that have distributions with point masses and thus ties between $f(\tilde{\theta}, y)$ and $(f(\theta_1, y), \ldots, f(\theta_M, y))$. Resolving ties lets us use SBC for models with discrete parameters as well as in some other special cases, such as when a theoretically strictly positive test quantity suffers underflow and some prior/posterior sample draws are numerically zero. Random tie-braking has previously been used for checking that data-averaged posterior equals prior (1) over discrete parameter spaces (Saad et al., 2019).

1.3 Practical considerations

SBC will be satisfied if the generator, probabilistic program, and posterior approximation algorithm are in harmony: The generator and the probabilistic program should correspond to the same data-generating process. At the same time the posterior approximation algorithm (including the associated tuning parameters) provides samples that have at most a negligible difference from the correct posterior for the probabilistic program, given the data simulated from the prior. Failure indicates that at least one of the components is mismatched to the others. However, by itself, SBC cannot determine where exactly the problem lies. As a result, two broad uses of SBC arise:

- We have code to simulate data and a probabilistic program we trust, and the goal is to check that an algorithm correctly samples from the posterior, or
- We have an algorithm that we trust is correct and trustworthy code to simulate data, and the goal is to check that we correctly implemented our probabilistic program.

In practice, those classes overlap and mix: we are rarely completely certain of the correctness of any algorithm, generator, or probabilistic program. Additionally, SBC as a simulation method has no way to inform us about a discrepancy between the process that generated real data and the assumptions of our statistical model. For reliable inference, SBC thus needs to be combined with other elements of Bayesian workflow that can detect model misspecification, such as posterior predictive checks or analysis of residuals (Gabry et al., 2019; Gelman et al., 2020; Kay, 2021).

1.4 Importance of test quantities

It has been generally believed that methods based on (2), including SBC, are never sensitive to some classes of mismatches between the generator and the probabilistic program—most notably that it is impossible to detect if the probabilistic program samples from the prior distribution and ignores the information in the data (e.g., Equation (1.3) of Lee et al. 2019; Appendix M.2 of Lueckmann et al. 2021; Schad et al., 2022; Zhao et al., 2021; Ramesh et al., 2022; Cockayne et al., 2022).

In this paper we show that the choice of test quantities greatly influences the usefulness and sensitivity of SBC. We show that using test quantities that depend on data makes it possible to detect any conceivable mismatch between the generator and the probabilistic program. Thus, we demonstrate that the belief in inherent limitations of SBC has relied on overly restrictive and sometimes plainly incorrect assumptions. We discuss useful classes of test quantities that have not been used so far and provide characterization of possible remaining undetected failures. We provide simulation studies as well as theoretical analysis of SBC to support our findings. We hope that our theoretical framework can serve as a basis for a better understanding of the properties of SBC and related methods. All of the techniques discussed are implemented in the `SBC` R package (Kim et al., 2022).

The rest of the paper is structured as follows: Section 2 discusses related work, Section 3 summarizes the theoretical results we derived, Sections 4 and 5 show results of simulation and real-world case studies, and Section 6 discusses the results and our recommendations for practical use of SBC.

2 Related work

Prior contributions to validation of Bayesian computation can be roughly split into works that focus on the data-averaged posterior, those that focus on the SBC property, and other relevant works that do not directly invoke any self-consistency property.

2.1 Data-averaged posterior

The idea of using simulations via a generator to verify Bayesian computation can be traced back to Geweke (2004) who compared the moments of the prior and the data-averaged posterior distributions for multiple test quantities. That paper proposed to integrate a transition kernel for an MCMC sampler targeting π_{post} into a scheme that samples π_{joint} directly. This lets one obtain the data-averaged posterior from a single run of this sampler, potentially reducing the computational cost but increasing implementation burden. Geweke's formalism allows the test quantities to depend on data, although all the examples actually shown only depend on parameters. Comparing the mean vector and covariance matrix of the prior distribution and the data-averaged posterior distribution is also discussed by Yu et al. (2021), who use repeated fits to build the data-averaged posterior.

Saad et al. (2019) proposed a check for identity of two potentially high-dimensional discrete distributions by inspecting ranks generated by different total orderings over the parameter space. Their work is relevant in four ways: (i) it can be used to assess the data-averaged posterior criterion (1) for discrete domains, (ii) in close analogy to the use of test quantities in this paper, they focus on different orderings of the underlying domain and their different power to detect discrepancies, (iii) they propose breaking ties in ordering uniformly at random in the same way we do, and (iv) they prove some results that are analogous to or special cases of some of our theoretical results.

2.2 SBC-like checks

The identity of prior and posterior distributions conditional on a specific dataset as a tool to check computation was proposed by Cook et al. (2006) and further refined by Talts et al. (2020) who introduced SBC as it is currently used. Specific variants of SBC have been proposed for variational inference (Yao et al., 2018), Bayes factors (Schad et al., 2022), and Gaussian processes (Mcleod and Simpson, 2021). SBC has also been used to validate likelihood-free inference methods including neural posterior approximators with normalizing flows (Radev et al., 2020, 2021) and an SBC variant for checking joint calibration of such methods has been proposed and used in Radev et al. (2023).

Gandy and Scott (2020) proposed a procedure similar to SBC that can work with shorter sequences of Markov transitions than a full fit, reducing computational cost. This is, however, less relevant for algorithms that need a nontrivial warmup phase to adapt to the specific posterior (e.g., the adaptive Hamiltonian Monte Carlo sampler implemented in Stan; Carpenter et al., 2017). This is because warmup is a fixed cost that occurs during every model fit even if fewer post-warmup draws are needed.

Prangle et al. (2014) proposed an SBC-like procedure for approximate Bayesian computation (ABC). They note that the possibility that the probabilistic program simply samples from the prior distribution cannot be ignored in this context and resolve this issue by separately inspecting ranks for some subsets of the simulated datasets. SBC is closely related to the *coverage property* discussed by Prangle et al.: when using M posterior sample draws, SBC can be understood as checking for all posterior intervals of width $\alpha \in \left\{\frac{1}{M}, \ldots, \frac{M-1}{M}\right\}$ that the probability the interval contains the original simulated value of the test quantity is α.

A broader framework for calibration of learning procedures has been proposed by Cockayne et al. (2022). There, Bayesian inference is just one example of procedures where calibration can be empirically verified with an SBC-like check. They distinguish between *strong calibration* which corresponds to passing SBC (specifically continuous SBC as defined in Appendix A; Modrák et al. 2023) for all measurable test quantities and *weak calibration* which corresponds to having a correct data-averaged posterior (1). They however only consider test quantities that depend only on parameters.

2.3 Miscellaneous

The problem of diagnosing and understanding computational issues is transformed by Rendsburg et al. (2022). Their approach tries to find a prior distribution that would make the probabilistic program and algorithm exactly match the generator.

Both Grosse et al. (2016) and Domke (2021) proposed to use fits to multiple generated datasets to estimate the symmetrized KL-divergence between a distributional approximation to the correct posterior (e.g., Laplace or variational inference) and the true posterior. Cusumano-Towner and Mansinghka (2017) described a method to compute the symmetrized KL-divergence between a gold standard posterior and an approximate posterior.

3 Theoretical results

The formalism required for SBC is relatively heavy on definitions and syntax, so for all results in this section we also provide plain English-language summaries. Proofs, expanded definitions (as required for proofs) and some additional discussion can be found in Appendix A (Modrák et al., 2023). Some of the results for stochastic rank statistics (SRS) by Saad et al. (2019) can be understood as special cases of some of our theorems. Specifically, SRS assumes that the parameter space Θ is finite or countable and the goal is to directly compare two distributions, which is the same as assuming the data space Y has just a single element. We will refer to this special case as the *SRS assumption*.

Definition (Posterior family, test quantity)**.** A *posterior family* ϕ assigns a normalized posterior density to each possible $y \in Y$. That is, a posterior family is a function $\phi : \Theta \times Y \to \mathbb{R}^+$ such that $\forall y : \int \mathrm{d}\theta\, \phi(\theta|y) = 1$. For each y, we will denote the implied distribution over Θ as ϕ_y. A *test quantity* is any measurable function $f : \Theta \times Y \to \mathbb{R}$

Definition (Sample rank CDF, sample Q, sample SBC)**.** Given a test quantity f, $M \in \mathbf{N}$ and a posterior family ϕ. If $\theta_1, \ldots, \theta_M \sim \phi_y$ we can define the following random variables:

$$
\begin{aligned}
N^{\texttt{less}}_{\phi,f,\tilde{\theta},y} &:= \sum_{m=1}^{M} \mathbb{I}\left[f(\theta_m, y) < f(\tilde{\theta}, y)\right], \\
N^{\texttt{equals}}_{\phi,f,\tilde{\theta},y} &:= \sum_{m=1}^{M} \mathbb{I}\left[f(\theta_m, y) = f(\tilde{\theta}, y)\right], \\
K_{\phi,f,\tilde{\theta},y} &\sim \text{uniform}\left(0, N^{\texttt{equals}}_{\phi,f,\tilde{\theta},y}\right), \\
N^{\texttt{total}}_{\phi,f,\tilde{\theta},y} &:= N^{\texttt{less}}_{\phi,f,\tilde{\theta},y} + K_{\phi,f,\tilde{\theta},y}.
\end{aligned}
$$

The *M-sample Q* is:

$$
Q_{\phi,f}(i|y) := \int_{\Theta} \mathrm{d}\tilde{\theta}\, \pi_{\text{post}}(\tilde{\theta}|y) \Pr\left(N^{\texttt{total}}_{\phi,f,\tilde{\theta},y} \leq i\right).
$$

We then say that *ϕ passes M-sample SBC w.r.t. f* if, $\forall i \in 0, \ldots, M-1$,

$$
\int_Y \mathrm{d}y\, Q_{\phi,f}(i|y) \pi_{\text{marg}}(y) = \frac{i+1}{M+1}.
$$

This definition does not match immediately with the procedure we actually use to run SBC in practice but is more convenient for further analysis and is equivalent:

Theorem 1 (Procedural definition of sample SBC)**.** *A posterior family ϕ passes M-sample SBC w.r.t. f if and only if given $\tilde{\theta}^* \sim \pi(\tilde{\theta}), y^* \sim \pi_{\text{obs}}(y|\tilde{\theta}^*), N^{\texttt{total}} = N^{\texttt{total}}_{\phi,f,\tilde{\theta}^*,y^*}$ we have $N^{\texttt{total}} \sim \text{uniform}(0, M)$.*

Next, we define an idealized, continuous version of SBC that will be more amenable to theoretical analysis:

Definition (Continuous rank CDF, continuous q, continuous SBC)**.** We first define fitted CDF: $C^{\pi}_{\phi,f} : \bar{\mathbb{R}} \times Y \to [0,1]$, $C^{\pi}_{\phi,f}(s|y) := \int_{\Theta} \mathrm{d}\theta\, \mathbb{I}\left[f\left(\theta, y\right) \leq s\right] \phi\left(\theta|y\right)$ and fitted tie probability: $D^{\pi}_{\phi,f} : \bar{\mathbb{R}} \times Y \to [0,1]$, $D^{\pi}_{\phi,f}(s|y) := \int_{\Theta} \mathrm{d}\theta\, \phi(\theta|y) \mathbb{I}\left[f(\theta, y) = s\right]$.

We then define the *continuous* $q : [0,1] \times Y \to [0,1]$ as

$$q_{\phi,f}(x|y) := \int_{\Theta} \mathrm{d}\tilde{\theta}\, \pi_{\mathrm{post}}(\tilde{\theta}|y) \Pr\left(C_{\phi,f}\left(f\left(\tilde{\theta}, y\right)\middle|\, y\right) - UD_{\phi,f}\left(f\left(\tilde{\theta}, y\right)\middle|\, y\right) \leq x\right),$$

assuming U is a random variable distributed uniformly over the $[0,1]$ interval.

Finally, ϕ *passes continuous SBC w.r.t.* f if $\forall x \in [0,1] : \int_Y \mathrm{d}y\, q_{\phi,f}(x|y) \pi_{\mathrm{marg}}(y) = x$.

3.1 Correctness

With the definitions ready, we first establish that if a probabilistic program achieves uniform distribution of ranks in sample SBC for a given test quantity as $M \to \infty$, then it will satisfy continuous SBC as well.

Theorem 2 (Sample SBC implies continuous SBC)**.**

1. *For any fixed $y \in Y$ if as the number of sample draws $M \to \infty$ we have $\forall i \in \{0, \ldots, M\} : Q_{\phi,f}(i|y) \to \frac{i+1}{M+1}$ then $\forall x \in [0,1] : q_{\phi,f}(x|y) = x$.*
2. *If as $M \to \infty$ we have $\forall i \in \{0, \ldots, M\} : \int_Y \mathrm{d}y\, Q_{\phi,f}(i|y) \pi_{marg}(y) \to \frac{i+1}{M+1}$ then ϕ passes continuous SBC for f.*

Theorem 3 then shows that if a probabilistic program passes continuous SBC for a given test quantity, it will pass sample SBC for all M. We then show that passing continuous SBC (and thus our SBC variant) is a necessary condition for the correctness of posterior estimation (Theorem 4). That is, the correct posterior will always produce uniformly distributed ranks, including for test quantities that may have ties (see also Examples 5 and 6 in Appendix B Modrák et al. 2023). A special case of Theorem 4 under the SRS assumption was proven as Theorem 3.1 of Saad et al. (2019).

Theorem 3 (Continuous SBC implies sample SBC)**.** *For all $M \in \mathbf{N}$:*

1. *For any $y \in Y$, if $\forall x \in [0,1] : q_{\phi,f}(x|y) = x$ then $\forall i \in \{0, \ldots, M-1\} : Q_{\phi,f}(i|y) = \frac{i+1}{M+1}$.*
2. *If ϕ passes continuous SBC w.r.t. f, then ϕ passes M-sample SBC w.r.t. f.*

Theorem 4 (Correct posterior and q)**.** *For any $y \in Y$, if $\forall \theta \in \Theta : \phi(\theta|y) = \pi_{post}(\theta|y)$ then for any test quantity f we have $\forall x \in [0,1] : q_{\phi,f}(x|y) = x$.*

3.2 Characterization of SBC failures

Still, many incorrect posteriors will also pass SBC for any given test quantity, so in Theorem 5 we characterize those situations.

Theorem 5 (Characterization of SBC failures)**.** *For all* $y \in Y$ *and* $s \in \mathbb{R}$: $\int_\Theta \mathrm{d}\theta \, \mathbb{I}\left[f\left(\theta, y\right) \leq s\right] \phi\left(\theta|y\right) = \int_\Theta \mathrm{d}\theta \, \mathbb{I}\left[f\left(\theta, y\right) \leq s\right] \pi_{post}(\theta|y)$ *if and only if* $\forall x \in [0,1]$: $q_{\phi,f}(x|y) = x$.

Not only does the correct posterior yield a uniform distribution of ranks when averaging over the whole data space Y, but the ranks are uniformly distributed even when we only consider simulations that yielded data in some $\bar{Y} \subset Y$. The reverse implication also holds: when the ranks are uniformly distributed for all subsets of the data space $\bar{Y} \subset Y$, then the implied posterior distribution of the test quantity under investigation has to be exactly correct. In other words, whenever SBC "fails" and the implied posterior distribution of a given test quantity is incorrect although the rank distribution is uniform, we can find a subset of the data space, where the ranks are non-uniform. It just so happens that all the deviations in various subsets cancel each other out perfectly.

An obvious application of Theorem 5 is that we could partition our simulations based on some features of the data space and investigate uniformity separately for each part, similarly to the procedure suggested by Prangle et al. (2014). This however quickly runs into issues of multiple testing due to the lower number of simulations in each part. It is thus in our experience not practical except for the special case where interest lies only in some subset of the data space, so that the SBC checks can focus only on that data space of interest. This is a form of rejection sampling and can be practically useful if it is easy to formulate a criterion that constrains plausible real data sets but hard to construct a defensible prior distribution that would enforce this criterion implicitly. For example, prior information can be available on the plausible variance of an outcome across the whole population, which may be hard to express as a prior on coefficients associated with predictors (but see the approaches for linear models discussed in Zhang et al., 2020 and Aguilar and Bürkner, 2023).

3.3 Data-dependent test quantities

The characterization of SBC failures discussed above provides intuition why test quantities that depend on data are useful: If SBC passes for a test quantity f, but the posterior is in fact incorrect, we can always pick a test quantity g that combines f with some aspect of the data and ensures that the discrepancies in various parts of data space add up instead of canceling out. For example, we could have over-abundance of low ranks and under-abundance of high ranks in $Y_1 \subset Y$ and a matching under-abundance of low ranks and over-abundance of high ranks in $Y_2 \subset Y$. Setting

$$g(\theta, y) = \begin{cases} -f(\theta, y) & y \in Y_1, \\ f(\theta, y) & \text{otherwise}, \end{cases}$$

will ensure over-abundance of high ranks in both Y_1 and Y_2. Since such a test quantity uses all the simulations, we do not lose power from reduced number of simulations.

An even stronger reason to use data-dependent test quantities is that they make SBC in some sense complete: If there is any difference between the correct posterior and the posterior implemented by the probabilistic program, there will exist a data-dependent

test quantity that fails SBC. In fact, we can construct a specific test quantity that detects the failures, which is the ratio of the correct posterior density to the posterior density actually implemented by the probabilistic program.

Theorem 6 (Density ratio)**.** *For any posterior family ϕ, take $g(\theta, y) = \frac{\pi_{\text{post}}(\theta|y)}{\phi(\theta|y)}$. Then ϕ passes continuous SBC w.r.t. g if and only if π_{post} and ϕ are equal except for a set of measure 0:*

$$\int_Y \mathrm{d}y \int_\Theta \mathrm{d}\theta \, \pi_{joint}(y, \theta) \mathbb{I}\left[\pi_{post}(\theta|y) \neq \phi(\theta|y)\right] = 0.$$

Here g is not a practical test quantity, as it (a) depends on the specific probabilistic program we implemented and (b) requires that we already have the correct posterior density. However our empirical results in this paper, and our experience with using SBC in model development more generally, shows that the model likelihood $\pi_{\text{obs}}(y|\theta)$ is frequently useful as a general-purpose test quantity. This makes sense intuitively, as the likelihood is an important contributor to the density ratio. In their Theorem 3.1, Saad et al. (2019) proved an analogous result under the SRS assumption, although relying on a different test quantity. Under the SRS assumption, they also show that the *difference* of the two densities will fail M-sample SBC for all $M > 1$ (their Theorem 3.6) and has maximum power against discrepancies (their Theorem 3.7).

3.4 Ignoring data

We generalize the result that probabilistic programs sampling from the prior distribution will pass SBC against all test quantities that do not depend on data.

Theorem 7 (Incomplete use of data)**.** *Assume a model π with observation space Y and parameter space Θ, a space Y', and a measurable function $t : Y \to Y'$. Denote the set $t^{-1}(y') = \{y \in Y : t(y) = y'\}$. Consider the model π' with parameter space Θ and observation space Y' such that for all $\theta \in \Theta, y' \in Y'$:*

$$\pi'_{\text{prior}}(\theta) = \pi_{\text{prior}}(\theta),$$
$$\pi'_{\text{obs}}(y'|\theta) = \int_{t^{-1}(y')} \mathrm{d}y \, \pi_{\text{obs}}(y|\theta).$$

Assume a test quantity $f' : Y' \times \Theta \to \mathbb{R}$. If we have a posterior family ϕ' on Y', Θ such that ϕ' passes continuous SBC w.r.t. f' and set test quantity $f : Y \times \Theta \to \mathbb{R}, f(\theta, y) = f'(\theta, t(y))$ and posterior family ϕ on Θ, Y such that $\phi(\theta|y) = \phi'(\theta|t(y))$ then ϕ passes continuous SBC w.r.t. f.

Here, the choice of t lets us choose which aspects of the data are ignored, if $\forall y \in Y : t(y) = 1$, we recover the case where all data are ignored: $\pi'_{\text{post}}(\theta|y) = \pi_{\text{prior}}(\theta)$ and thus $\phi(\theta|y) = \pi_{\text{prior}}(\theta)$ will pass SBC w.r.t. f. If t is a bijection, no information is lost. Other choices of t then let us interpolate between those two extremes, for example ignoring just a subset of the data points, treating some data points as censored, rounding all data to integers.

3.5 Detailed analysis of simple models and test quantities

Appendix B (Modrák et al., 2023) provides full theoretical analysis of SBC for simple models and test quantities where we can actually characterize all possible posterior distributions that will satisfy SBC. This is aimed at providing intuition on what SBC actually does and also serves as counterexamples to some claims. In some publications (e.g., Lee et al., 2019, Lueckmann et al., 2021, Schad et al., 2022, Grinsztajn et al., 2021, Ramesh et al., 2022, Saad et al., 2019), it is assumed that SBC is based on the data-averaged posterior (1). We show that this is incorrect: Example 2 not only explicitly constructs posterior distributions that will satisfy (1) for some test quantity while not passing SBC, but also posterior distributions that pass SBC while not satisfying (1). One possibly more general lesson is that SBC is most naturally understood as enforcing constraints on the quantile function of the test quantity while having a correct data-averaged posterior is most naturally seen as constraint on the density of the test quantity.

This implies there might be some gains from using both the data-averaged posterior and SBC when verifying the correctness of Bayesian computation. We however suspect that the additional practical benefit of using the data-averaged posterior is small in the sense that the incorrect posteriors that pass SBC but are ruled out by (1) are mostly contrived and unlikely to be a result of a computational problem or an inadvertent mistake. Lemma 2.19 of Cockayne et al. (2022) proves that if a posterior passes SBC for *all possible* test quantities that do not depend on data, it will have the correct data-averaged posterior for all test quantities that do not depend on data, so SBC is stronger at least in the limit of using infinitely many test quantities. We leave a more thorough examination of the relationship between data-averaged posterior and SBC as future work.

Additionally, we show the behavior of SBC when ties are present, whether induced by a test quantity (Example 5) or by discrete parameter space (Example 6). Discrete parameter spaces may induce additional structure on the space of posterior families passing SBC.

3.6 Monotonic transformations of test quantities

Finally, transforming a test quantity by a strictly monotonic function produces equivalent SBC results:

Theorem 8 (Monotonic transformations)**.** *Assume test quantities f, g and a set of measurable functions $h_y : \mathbb{R} \to \mathbb{R}$ such that $\forall y \in Y, \theta \in \Theta : f(\theta, y) = h_y(g(\theta_1, y))$ and a posterior family ϕ. If either for all $y \in Y : h_y$ is strictly increasing or for all $y \in Y : h_y$ is strictly decreasing then 1) ϕ passes continuous SBC w.r.t. f if and only if ϕ passes continuous SBC w.r.t. g and 2) ϕ passes M-sample SBC w.r.t. f if and only if ϕ passes M-sample SBC w.r.t. g.*

The result cannot be easily strengthened as many non-monotonic transformations lead to different, non-equivalent SBC checks. Example 3 shows that flipping the ordering of values only for some subset of the data space yields a different SBC check. Example 4

shows that we can also obtain a different check if we combine a test quantity with a non-monotonic bijection, and Example 5 shows the same for the case when a whole range of values is projected onto a single point. In all those examples, the transformed test quantities rule out some sets of posteriors that pass SBC for the original quantity, but there are also sets of posteriors not passing SBC for the original quantity but passing SBC for the transformed quantity.

4 Numerical case studies

The theoretical analysis in previous section primarily deals with the behavior of SBC in the limit of both infinitely many posterior draws per fit and infinitely many simulations. Here, we further support the results by numerical experiments which let us understand not only whether a certain problem is detectable at all but also how much computational effort is required for SBC to detect the problem.

4.1 Setup

To illustrate some of the properties of various types of test quantities, we use a simple bivariate normal model,

$$\begin{aligned} \boldsymbol{\mu} &\sim \mathrm{MVN}(0, \boldsymbol{\Sigma}), \\ \mathbf{y}_1, \ldots, \mathbf{y}_n &\sim \mathrm{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \\ \boldsymbol{\Sigma} &= \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}, \end{aligned} \tag{6}$$

where the two-element vector μ is the target of inference and $\mathbf{y}_1, \ldots, \mathbf{y}_n$ are observed. Introducing $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i$, the correct analytic posterior is $\mathrm{MVN}\left(\frac{N\bar{\mathbf{y}}}{n+1}, \frac{1}{n+1}\boldsymbol{\Sigma}\right)$. Unless mentioned otherwise we will use $n = 3$. In most previous use cases of SBC, the only test quantities used would have been the parameters themselves, that is, the elements of $\boldsymbol{\mu}$ in the above example. Below, we also check a host of derived quantities: the sum, difference, and product of the $\boldsymbol{\mu}$ elements, the joint likelihood of all the data, and pointwise likelihoods for the first two data points.

To quantify the discrepancy between an observed distribution of posterior ranks and the uniform distribution, we take the likelihood of observing the most extreme point on the empirical CDF if the rank distribution was indeed uniform:

$$\gamma = 2 \min_{i \in \{1, \ldots, M+1\}} \big(\min\{\mathrm{Bin}(R_i | S, z_i), 1 - \mathrm{Bin}(R_i - 1 | S, z_i)\} \big). \tag{7}$$

Here, M is the number of draws in the sample obtained from the posterior, S is the number of simulations (and thus the number of observed ranks), $z_i = \frac{i}{M+1}$ is the expected proportion of observed ranks smaller than i, R_i is the observed count of ranks smaller than i, and $\mathrm{Bin}(R|S, p)$ is the CDF of the binomial distribution with S trials and probability of success p evaluated at R. This metric was introduced in a paper by Säilynoja et al. (2022), where we can also find computational methods to evaluate

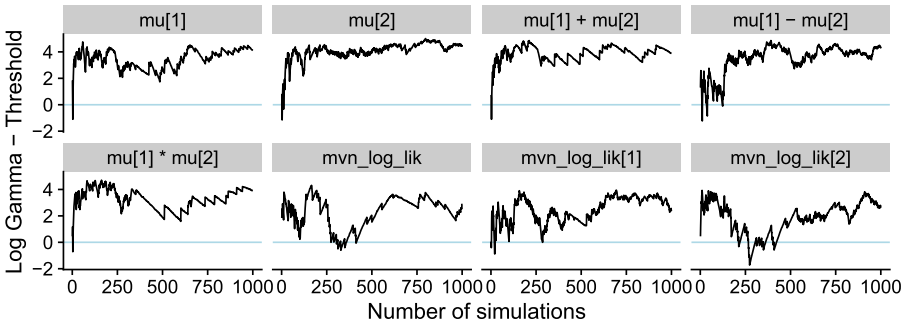

Figure 2: *Case study 1: Evolution of the difference between the gamma statistic and threshold ($\log \bar{\gamma}$) for rejecting uniformity at 5% for the correct posterior.* `mvn_log_lik[1]` *and* `mvn_log_lik[2]` *are the pointwise likelihoods $\pi(\mathbf{y}_1|\mu)$ and $\pi(\mathbf{y}_2|\mu)$ respectively, while* `mvn_log_lik` *is the joint likelihood. As expected when using a 5% level for rejection, false positives (values below the threshold) do happen, but they tend to correspond to only small discrepancies.*

the distribution of γ under uniform distribution of ranks for given M and S. Our primary metric of interest would then be $\log \frac{\gamma}{\bar{\gamma}}$, where $\bar{\gamma}$ is the 5th percentile of the null distribution. That is, if you adopt a hypothesis-testing framework, then $\log \frac{\gamma}{\bar{\gamma}} < 0$ implies a rejection of the hypothesis of uniform distribution at the 5% level. Having $\log \frac{\gamma}{\bar{\gamma}} < 0$ also corresponds to situations where visual checks of the ECDF plots would show problems (for a single test quantity). This diagnostic is typically more sensitive than the Kolmogorov-Smirnoff or χ^2 test.

4.2 Correct posterior – Case study 1

Figure 2 shows how the γ statistic evolves in a fairly typical SBC run as we add more simulations using a probabilistic program that samples from the correct posterior. There is some variability, but most of the time all quantities would indicate uniformity and if they indicate some non-uniformity, the discrepancies tend to be small so we are unlikely to reject this model as incorrect.

4.3 Ignoring data – Case studies 2–4

For comparison, case study 2 (Figure 3) shows the evolution of the same quantities for a typical run with an incorrect posterior that is completely equal to the prior. All quantities that do not depend on data pass SBC, barring small short-term deviations as seen for the correct posterior. But all the likelihood-based quantities start showing big discrepancies after just a handful of simulations. While the overall distribution of ranks for the parameters themselves is uniform, when we look separately at data with large average y and low average y, the ranks are strongly non-uniform in both regions (Figure 4).

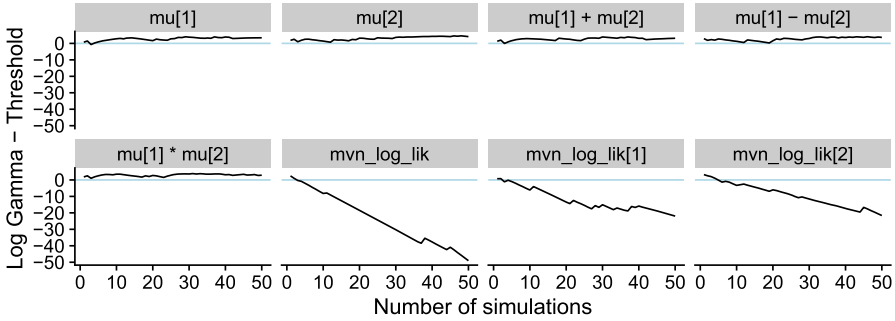

Figure 3: *Case study 2: Evolution of the difference between the gamma statistic and threshold for rejecting uniformity at 5% for an incorrect posterior that equals the prior. Note how quickly large discrepancies accumulate for the likelihood-based quantities, despite the horizontal axis being zoomed to show only first 50 simulations.*

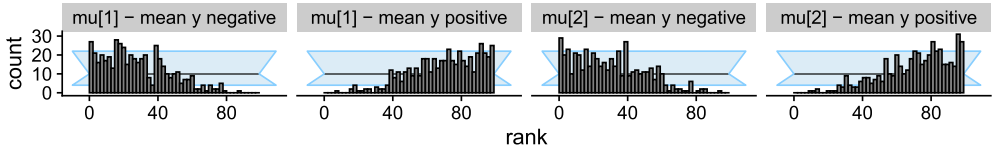

Figure 4: *Case study 2: Rank distribution for the elements of μ split by the average value of the corresponding y elements for the incorrect posterior that is completely equal to the prior. The distributions for the two cases exactly compensate to make the overall distribution uniform. The gray horizontal line represents exact uniform distribution and the blue areas represent an approximate 95% prediction interval for the observed ranks, assuming uniform rank distribution.*

In case study 3, we observe similar behaviour for the posterior that ignores only the first data point; see Figure 5. The biggest difference is that now the pointwise likelihood for the second data point—which was not ignored—passes SBC, while the joint likelihood as well as the pointwise likelihood for the first ignored data point show problems. Additionally, the pointwise likelihood for the ignored data point now shows bigger discrepancy than the joint likelihood. For both quantities, the discrepancy is smaller and requires about $S = 20$ simulations to reliably uncover, because ignoring a single data point produces a posterior that is closer to the correct one than when ignoring all the data. For case study 4 we increase the number of data points to $n = 20$ (Figure 6), ignoring just a single data point produces a posterior that is close to correct and even after 1000 simulations, the discrepancy for the joint likelihood is small. The pointwise likelihood for the first (ignored) data point still detects the problem relatively quickly.

More generally, if the model (partially) ignores data, then adding a test quantity that involves both data and parameters can detect this failure. Specifically adding the joint log-likelihood of the data as a derived quantity seems to be a useful default. If

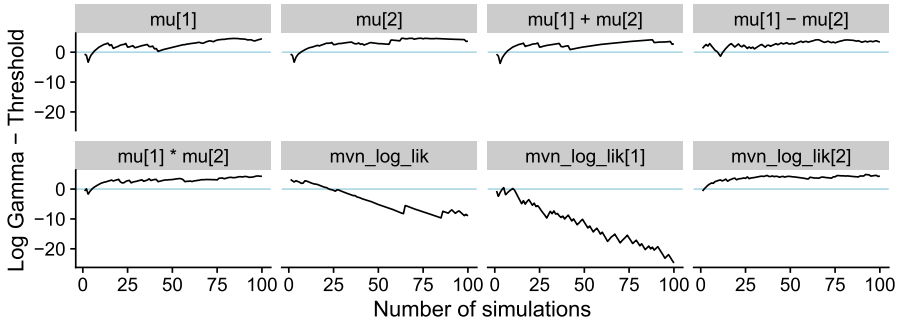

Figure 5: *Case study 3: Evolution of the difference between the gamma statistic and threshold for rejecting uniformity at 5% for an incorrect posterior that ignores the first datapoint among a small data set ($n = 3$).*

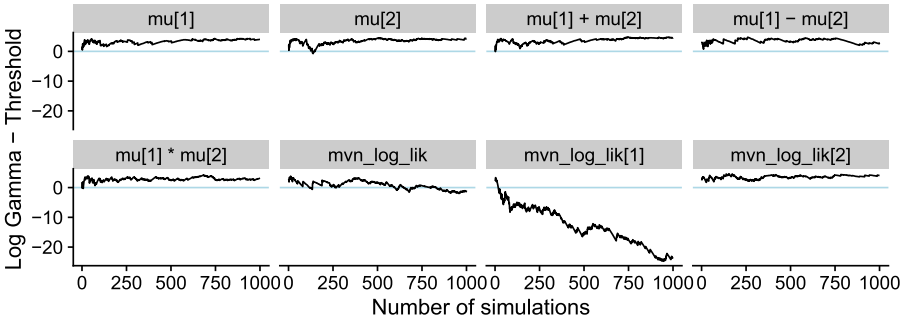

Figure 6: *Case study 4: Evolution of the difference between the gamma statistic and threshold for rejecting uniformity at 5% for an incorrect posterior that ignores the first datapoint among a larger dataset ($n = 20$).*

only a small part of the data is missing, using the joint likelihood in SBC will turn it into a problem of precision. Missing just a single datapoint in a large dataset (e.g., an off-by-one error in the probabilistic program) may change the posterior only slightly and be undetectable with realistic computational effort.

4.4 Incorrect correlations – Case study 5

Suppose we have an incorrect posterior that has the correct marginal distributions for both parameters, i.e., sampling is done from independent univariate normal distributions, $\mu_i \mid \mathbf{y}_1, \ldots, \mathbf{y}_n \sim N\left(\frac{n\bar{\mathbf{y}}_i}{n+1}, \frac{1}{n+1}\mathbf{\Sigma}_{i,i}\right)$. The evolution of the discrepancy as simulations are added is shown in Figure 7. If the test quantities are the univariate parameters, SBC passes without any indication of problems, while the likelihood-based quantities as well as the difference, product, and sum of the variables show problems relatively quickly. The joint likelihood is the first to show serious issues.

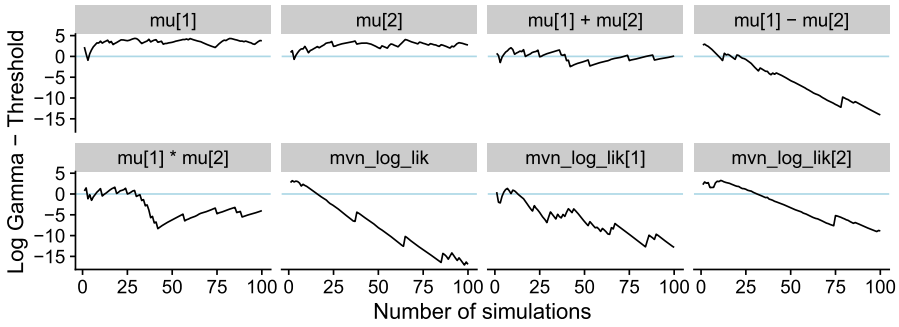

Figure 7: *Case study 5: Evolution of the difference between the gamma statistic and threshold for rejecting uniformity at 5% for incorrect posterior that has wrong correlation structure.*

If the inference does not represent correlations in the posterior correctly, this should as well manifest in an SBC failure for some function of the parameters. This can be directly targeted by using products ("interactions") of model parameters, but the log-likelihood once again seems to be generally useful as a highly nonlinear function of all model parameters.

4.5 Less plausible problems – Case study 6

In this subsection our results get less practical and more theoretical. The (partially) unused data case may easily arise in practice due to a bug in the probabilistic program such as an indexing bug or a deficient overall approach. For example, an approximate Bayesian computation algorithm may not learn from the data at all and just stick to the prior (Prangle et al., 2014). Incorrect correlations or more general higher-order structure of the posterior may also easily arise due to a problem with an approximate inference algorithm. For example, mean-field variational inference will never recover any correlations by design. Beyond those examples, we have found it hard to find incorrect probabilistic programs that would satisfy the SBC identity and could plausibly arise from unintentional mistakes in program code or problems with an algorithm. We see this as anecdotal evidence that SBC augmented with a few well-chosen test quantities that probe usage of data and higher order posterior structure such as the likelihood can robustly detect these kinds of mistakes. That said, for specific models, wide sets of artificial counterexamples that incorrectly pass SBC can be constructed.

In case study 6, we show a specific case of a more general class of setups where we can create an incorrect posterior approximation that produces overabundance of low ranks for datasets with average of $\mathbf{y}$ positive and compensates by producing overabundance of high ranks for other datasets. If this is done right, the test quantity will pass SBC. The distribution of the ranks conditional on the average of $\mathbf{y}$ for one such setup is shown in Figure 8—here we transform draws from the correct posterior distribution by first applying the correct CDF, manipulating the results to achieve the

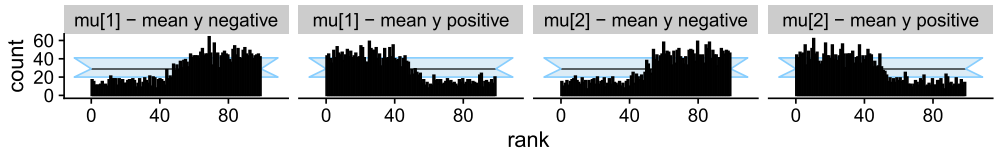

Figure 8: *Case study 6: Rank distribution for the elements of μ split by the average value of the corresponding y elements for the incorrect posterior that satisfies SBC for individual parameters. The distributions for the two cases exactly compensate to make the overall distribution uniform.*

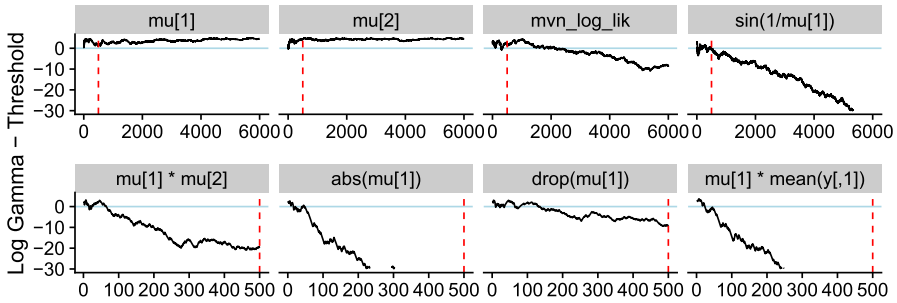

Figure 9: *Case study 6: Evolution of the difference between the gamma statistic and threshold for rejecting uniformity at 5% for incorrect posterior that satisfies SBC for individual parameters. Note the different horizontal axis between top row (quantities that detect the problem slowly or not at all) and bottom row (quantities that detect the problem quickly). The vertical red dashed line marks 500 simulations. We only show quantities derived from the first element of μ; the situation is analogous for the second element. The `drop(mu[1])` quantity is defined as μ_1 if $\mu_1 < 1$ and as $\mu_1 - 5$ otherwise.*

desired shape of ranks and then transform back via the quantile function. See the associated code for more details. As seen in Figure 9, when averaging over all datasets, SBC indeed passes for the univariate parameter test quantities, but if we instead look at, say, the absolute value of μ (as well as some other non-monotonic transformations of μ), we immediately see problems as now some of the previously low ranks flip to high ranks and the discrepancies accumulate instead of canceling each other. In this particular case, the problem is also eventually picked up by the product of the μ values and with enough simulations even by the joint likelihood, but there is no guarantee this will always happen. In general, non-monotonic transformations can discover incorrect posteriors that would be otherwise hidden when looking at the original variables. Still, the practical relevance of non-monotonic transforms in SBC is, in our view, likely limited, as it required careful work to construct posteriors that manifested this behaviour. We were unable to find even remotely plausible scenarios where an issue with Bayesian computation was best discovered by using a non-monotonic transformation of another test quantity.

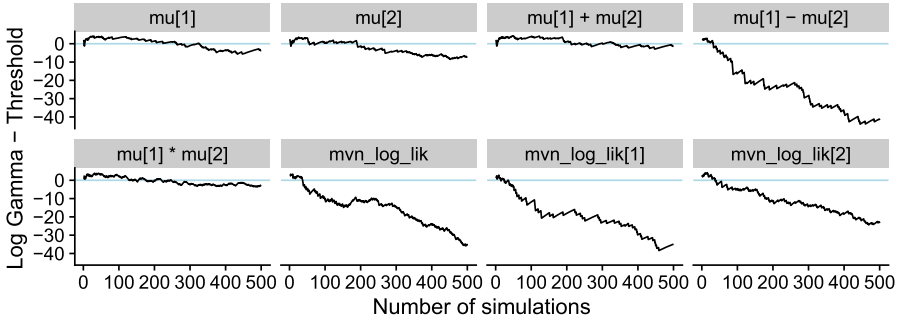

Figure 10: *Case study 7: Evolution of the difference between the gamma statistic and threshold for rejecting uniformity at 5% for incorrect posterior that introduces a small bias for each parameter.*

4.6 Small discrepancies – Case study 7

A final case study considers small discrepancies in the posterior. To be specific, we introduce a small bias in the posterior drawn from normal$(0, 0.3)$ independently for each simulation and element of μ. The resulting SBC history is shown in Figure 10. While all of the monitored quantities will eventually show the problem, the likelihood-based quantities and the difference of μ do that noticeably sooner than others. This demonstrates that derived quantities can somewhat improve precision of SBC: small changes in the univariate marginals can result in big (and thus easy to detect) changes for some test quantities combining the univariate marginals with data and other parameters.

5 Real-world case study

We present a case study adapted from an actual user discussion on forums of the Stan probabilistic programming language. Our goal is to use Stan and its Hamiltonian Monte Carlo implementation to sample from a distribution over an ordered K-dimensional simplex which is then to be used as a component in a larger model:[2]

$$\text{OrdSimplex}_K = \left\{ \mathbf{x} \in \mathbb{R}^K | 0 < x_1 < \cdots < x_K < 1, \sum_{i=1}^{K} x_i = 1 \right\}.$$

To do that, we need to construct an ordered simplex from primitive data types available in Stan and compute the logarithm of the Jacobian determinant of the transformation (up to a constant).

[2]The discussion can be found at https://discourse.mc-stan.org/t/ordered-simplex-constraint-transform/24102. We thank Sean Pinkney, Bob Carpenter and Ben Goodrich for contributing to the discussion and suggesting solutions.

5.1 Proposed implementations

We consider three variants. Mimicking the fallibility of methods proposed by real statisticians, not all of the following derivations will be correct. A reader interested in little mathematical puzzles may try to pin down any errors. In the next subsection, we will then show how to use SBC to discover the error(s) without the painstaking attention to detail required for checking the math. We then also remedy the error(s).

The first variant will be called `min`. Here, we start with an unordered bounded vector $\mathbf{u} \in [0,1]^{K-1}$ (which is a primitive in Stan). The minimal element of the simplex needs to satisfy $x_1 < \frac{1}{K}$, so we set $x_1 = \frac{u_1}{K}$. Given x_1, if we set $\mathbf{x}' \in \mathbb{R}^{K-1}, x'_i = \frac{x_{i+1}-x_1}{1-Kx_1} = \frac{x_{i+1}-x_1}{1-u_1}$, then $\mathbf{x}' \in \text{OrdSimplex}_{K-1}$, giving us a recursive formula for the transformation, which we can unroll as:

$$\begin{aligned} b_1 = 0,\ r_1 = 1,& \\ \text{for } 1 \le i < K:\ x_i = b_i + r_i \frac{u_i}{K+1-i},\ b_{i+1} = x_i,\ r_{i+1} = r_i(1-u_i),& \\ x_k = b_k + r_k = 1 - \sum_{i=1}^{K-1} x_i.& \end{aligned}$$

Here r_i can be understood as tracking the remaining amount to be distributed to ensure x_i sum to 1 if all the following elements will be at least b_i. For $1 \le i < K$, we have $\frac{\partial x_i}{\partial u_i} = \frac{r_i}{K+1-i}$, and when also $i \le j < K$ then $\frac{\partial x_i}{\partial u_j} = 0$, so the Jacobian matrix is triangular and the Jacobian determinant is thus

$$\det \mathbf{J} = \prod_{i=1}^{K-1} \frac{r_i}{K+1-i}. \tag{8}$$

The second variant, called `softmax` starts with a positive ordered vector $\mathbf{v} \in (0, +\infty)^{K-1}$, $v_1 < \cdots < v_{K-1}$ (also a primitive in Stan). We then prepend 0 to the vector and normalize it with the softmax function:[3]

$$s = 1 + \sum_{i=1}^{K-1} \exp(v_i),\ x_1 = \frac{1}{s},\ x_k = \frac{\exp(v_{k-1})}{s}.$$

For $k > 1, 1 \le j \le K-1, j \ne k-1$ the partial derivatives are:

$$\begin{aligned} \frac{\partial x_k}{\partial v_{k-1}} &= \frac{\exp(v_{k-1})}{s} - \frac{\exp(2v_{k-1})}{s^2} = \frac{\exp(v_{k-1})(s - \exp v_{k-1})}{s^2}, \\ \frac{\partial x_k}{\partial v_j} &= -\frac{\exp(v_{k-1} + v_j)}{s^2}. \end{aligned}$$

[3] One could also base the normalization on the arithmetic sum of the elements, but this results in problematic geometry of the posterior and the sampler has trouble converging.

We notice the repeated elements and define a $K-1$ dimensional diagonal matrix $\mathbf{D}$, where $\mathbf{D}_{i,i} = \frac{\exp(y_i)}{s^2}$. We can now express the Jacobian matrix as

$$\mathbf{J} = \left(\mathbf{D} \begin{pmatrix} -\exp(v_1) & \cdots & -\exp(v_{K-1}) \\ \vdots & \ddots & \vdots \\ -\exp(v_1) & \cdots & -\exp(v_{K-1}) \end{pmatrix} + s\mathbf{I}_{K-1} \right).$$

We now define a $K-1$ dimensional column vector $\mathbf{c}, c_k = -\exp(v_k)$ and a row vector $\mathbf{r}, r_k = 1$ and obtain $\mathbf{J} = \mathbf{D}\left(\mathbf{cr} + s\mathbf{I}_{K-1}\right)$. By the matrix determinant lemma, $\det(\mathbf{cr} + \mathbf{X}) = \det(\mathbf{X})(1 + \mathbf{r}\mathbf{X}^{-1}\mathbf{c})$, for any invertible matrix $\mathbf{X}$. Since $\mathbf{rc} = \sum_{i=1}^{K-1}(-\exp v_i) = 1 - s$, we have:

$$\det(\mathbf{cr} + s\mathbf{I}_{K-1}) = \left(1 + \frac{1}{s}\mathbf{rc}\right) s^{K-1} = \left(1 + \frac{1-s}{s}\right) s^{K-1} = s^{K-2}.$$

Since $\det(\mathbf{D}) = \frac{\exp(\sum_{i=1}^{K-1} y_i)}{s^{2(K-1)}}$, we finally have

$$\det(\mathbf{J}) = \det(\mathbf{D}) \det(\mathbf{cr} + s\mathbf{I}_{K-1}) = \frac{\exp(\sum_{i=1}^{K-1} v_i)}{s^{K-1}}. \tag{9}$$

As a different approach, if we are willing to restrict our priors over the ordered simplex to Dirichlet distributions, we may employ the fact that if $\mathbf{w} \in (0, +\infty)^K, w_i \sim \Gamma(\alpha_i, 1)$ then $\frac{\mathbf{w}}{\sum_{i=1}^K w_i} \sim \text{Dirichlet}(\boldsymbol{\alpha})$. So if we start with $\mathbf{w}$ positive ordered (a primitive in Stan), then $\mathbf{x} = \frac{\mathbf{w}}{\sum_{i=1}^K w_i}$ will be Dirichlet distributed over OrdSimplex_K and no Jacobian adjustment is required. A downside of this approach is that the mapping is many-to-one and in models where $\mathbf{x}$ is tightly constrained by data, the implied geometry on $\mathbf{w}$ will likely pose difficulty for most samplers. This variant will be referred to as `gamma`.

At this point the interested reader is welcome to try to find issues with any of the above approaches.

5.2 Testing with SBC

Whether the reader managed to find errors or not, we can use SBC to test all approaches. To run SBC we embed the ordered simplex into a simple model:

$$\begin{aligned} &\mathbf{x} \in \text{OrdSimplex}_4, \pi(\mathbf{x}) \propto \text{Dirichlet}(2,\ 2,\ 2,\ 2), \\ &\mathbf{y} \sim \text{Multinomial}(10,\ \text{x}). \end{aligned} \tag{10}$$

Implementing the simulator code is straightforward: due to symmetry, we can sample $\mathbf{x}$ simply by ordering a sample from the unordered Dirichlet distribution. Both `min` and `gamma` variant show no problems in SBC and are indeed correct, but `softmax` exhibits issues. Figure 11 shows the evolution of the discrepancies. The problems are most quickly picked up by the first element of $\mathbf{x}$ and the log Dirichlet prior density. Although the

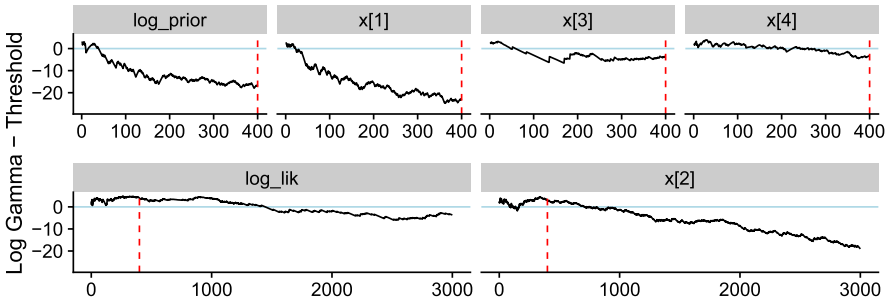

Figure 11: *Evolution of the difference between the gamma statistic and threshold for rejecting uniformity at 5% for the incorrectly implemented* `softmax` *variant of an ordered simplex model.* `log_lik` *is the multinomial log likelihood of the data and* `log_prior` *is the log density of the prior Dirichlet distribution. Note the different horizontal axis between top row (quantities that detect the problem quickly) and bottom row (quantities that detect the problem slowly). The vertical red dashed line marks 400 simulations.*

problem is found relatively quickly with SBC, the bias in the inferences would likely not be noticed in an informal assessment of the model: the results are not completely wrong, just somewhat biased. The source of the issue is an off-by-one error in the exponent for s in equation (9); the correct Jacobian determinant is

$$\det(\mathbf{J}) = \det(\mathbf{D}) \det(\mathbf{c}\mathbf{r} + s\mathbf{I}_{K-1}) = \frac{\exp \sum_{i=1}^{K-1} v_i}{s^K}.$$

Indeed, if we correct the Jacobian, SBC passes.

5.3 Remarks

The previous section showed the type of modeling problem where SBC is in our view the most useful: deriving and implementing the probabilistic program is relatively involved and offers plenty of opportunities for error, but building a simulator is straightforward. Jacobian adjustments for changes of variables are also in our experience one of the most confusing concepts to Stan users and SBC offers a good way to check if one's reasoning is correct.

The examples in this section introduce several non-obvious conceptual questions: For `min` and `softmax` we compute the Jacobian only considering $K - 1$ elements of the ordered simplex, even when the Dirichlet prior then acts on all elements. Is that correct? For `gamma`, will ordering $\mathbf{w}$ imply the correct ordered simplex distribution? Running SBC is then a useful (although not completely definitive) check that our reasoning is correct.

In this example, the log likelihood did not reveal the error quickly, showing that it is not a panacea, especially in cases where the problem lies with the prior. The log prior density seems potentially useful in this case as it shows the problem as quickly as the most problematic individual parameter. Another lesson is that SBC is useful not only

for testing a full model but also for testing components of a model in isolation, akin to unit tests in software engineering. Additionally, by running SBC, we get a simulation study for free: in the specific setup described by (10), `min` is the most efficient in terms of effective sample size per second, followed by `gamma`. The correct version of `softmax` performs worst. The `softmax` variant also fails to converge in 6 of the 1000 simulations, while the other two are slightly more stable (convergence problems in 3 and 1 of the simulations respectively). Finally, our posterior uncertainty is large and the data do not really provide a lot of information about the parameter values. See the rendered output of the supplementary code for details.

6 Conclusions

6.1 Choosing test quantities for SBC

We have found that enriching the repertoire of test quantities used in SBC provides both qualitative and quantitative improvements to the ability of SBC to detect problems in Bayesian computation. For practical use of SBC in everyday model and algorithm development, we recommend to use by default the individual model parameters as test quantities as well as the joint likelihood of the data and potentially a small number of other quantities.

Individual parameters are recommended as they are always immediately available and are able to diagnose a large number of problems with a posterior approximation. Also, the parameters are themselves often of primary interest for inference, so it is desirable to check that their uncertainty is correctly calibrated.

The joint likelihood is a highly useful quantity to detect the types of problems discussed in Section 4 (especially ignoring data and incorrect correlations). In all of the cases presented in our simulations, the joint likelihood was able to detect the discrepancies and in many cases it was even able to detect them with the fewest simulations among all considered quantities. While, for some specific problems, we could find quantities that are more sensitive than the joint likelihood, none other was useful in all cases. Section 3.3 provides theoretical justification for why we could expect this to hold frequently and not only in the examples we discussed. We think this generality makes the joint likelihood a good default quantity to monitor in SBC. If not using all the data correctly is a potential issue (e.g., because the code handling the data is particularly complex), then adding selected likelihoods for subsets of the data might also be sensible.

As shown in Section 5, knowing where a potential problem lies can let us design more sensitive problem-specific checks (e.g., when we are not sure our prior density is correct, the log prior can be highly useful). It also makes sense to add test quantities tailored to the specific inferential goals we have built the model for (e.g., some specific model predictions). These quantities often let us implicitly check the correctness of parameter correlations or other dependency structures and safeguard the user against problems that they care about the most. If correlations or other dependencies in the posterior are directly of interest, then pairwise products or differences of the model parameters can also be sensible test quantities.

6.2 Limitations

Although we have shown that SBC can in principle diagnose any problem, limitations for practical use remain. For nontrivial models, adding a finite number of test quantities cannot guard against all possible ways the SBC identity may be satisfied by an incorrect posterior. However, as we check more quantities, the potential counterexamples become contrived, hard to construct, and unlikely to be the result of an inadvertent bug in model or algorithm code. At the same time, adding more test quantities increases the risk of false SBC failures simply due to the number of tests performed (if no corrections for multiple comparisons are made for the SBC checks) or it may reduce the overall power of the check (if corrections for multiple comparisons are made), so choosing test quantities carefully remains important.

This problem could potentially be alleviated by improving our understanding of the expected dependency structure of different test quantities' uniformity checks, letting us correct for multiple comparisons without loosing that much power. However, even similar test quantities can lead to in principle different SBC checks (see Section 3.6). So any practical measure of dependency or orthogonality between test quantities would need to reflect not only existence of a difference, but also its magnitude. We leave that as future work. In practice, we have seen similarity in the degree of uniformity violation between different test quantities using the same inputs, making the need for multiple comparison correction less urgent.

Moreover, there are practical limitations imposed by the fact that we always have only limited computational resources for SBC: We can produce only a limited number of simulated datasets to fit the model on and only a limited number of posterior draws per fitted model. Both contribute to the stringency and precision of the uniformity test we can perform. The difference between continuous SBC and any practical implementation of sample SBC arises due to (a) approximating $q_{\phi,f}(x|y)$ by $Q_{\phi,f}(\lfloor xM \rfloor \,|\, y)$, and (b) using finite number of simulations to assess uniformity of $N_{\texttt{total}}$. In both cases, the underlying difference can be understood as estimating a CDF by an empirical CDF and should therefore have similar rate of decrease with more draws. This suggests that for a given computational budget a user is likely to obtain the highest sensitivity using the same order of magnitude of simulated datasets as posterior draws per dataset. However, in practice most algorithms incur a substantial cost in a warmup phase, before any samples can be extracted. We also want to assess that our fitting algorithm has converged for each dataset, which typically requires the equivalent of at least 100 independent posterior samples (as measured by effective sample size) to do that (e.g., to get a low $\hat{R}$ statistic, as discussed by Vehtari et al. 2021). It is thus hard to get a speedup by reducing the number of posterior draws. Unless we can afford to run many thousands of simulations, we are also unlikely to benefit substantially from getting more than this minimal number of draws.

Additional test quantities do not help much with precision problems—if the posterior is close to correct, the test quantities will also be close to correct. Although in some cases, some test quantities can slightly increase the sensitivity of the check by combining multiple parameters, so small imprecisions in each of the parameters can

get compounded (once again the nonlinearity of the likelihood seems to be at least sometimes useful in this regard).

6.3 Implications for non-SBC checks

As a contribution to the broader discussion about validation of Bayesian computation, we show that SBC and the data-averaged posterior provide different checks, despite being repeatedly conflated in the literature (see Section 3.5). We leave a more detailed comparison of SBC and data-averaged posterior as future work, although there are some tentative arguments to believe that SBC provides stricter checks.

SBC is not the only approach to validating Bayesian computation that relies on choosing specific test quantities—test quantities are fundamental to the methods of Geweke (2004), Prangle et al. (2014), Gandy and Scott (2020), and Cockayne et al. (2022). We suspect that many of the considerations regarding their choice for SBC are applicable also in these other approaches.

Supplementary Material

Appendixes for Simulation-Based Calibration Checking for Bayesian Computation: The Choice of Test Quantities Shapes Sensitivity (DOI: 10.1214/23-BA1404SUPP; .pdf). Appendix A contains mathematical theory and proofs, and Appendix B contains examples of simple models where we can fully characterize the space of posteriors that satisfy simulation-based calibration checking with respect to several test quantities. Code for simulations and all the figures in Sections 4 and 5 can be found at https://github.com/martinmodrak/sbc_test_quantities_paper. All code output and associated commentary can also be viewed at https://martinmodrak.github.io/sbc_test_quantities_paper/

References

Aguilar, J. E. and Bürkner, P.-C. (2023). "Intuitive joint priors for Bayesian linear multilevel models: The R2D2M2 prior." *Electronic Journal of Statistics*, 17(1): 1711–1767. MR4609453. doi: https://doi.org/10.1214/23-ejs2136. 10

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). "Stan: A probabilistic programming language." *Journal of Statistical Software*, 76(1). URL https://www.jstatsoft.org/index.php/jss/article/view/v076i01 7

Cockayne, J., Graham, M. M., Oates, C. J., Sullivan, T. J., and Teymur, O. (2022). "Testing whether a learning procedure is calibrated." *Journal of Machine Learning Research*, 23(203): 1–36. URL http://jmlr.org/papers/v23/21-1065.html MR4577156. 5, 7, 12, 25

Cook, S. R., Gelman, A., and Rubin, D. B. (2006). "Validation of software for Bayesian models using posterior quantiles." *Journal of Computational and Graphical Statistics*,

15(3): 675–692. MR2291268. doi: https://doi.org/10.1198/106186006X136976. 2, 7

Cusumano-Towner, M. F. and Mansinghka, V. K. (2017). "AIDE: An algorithm for measuring the accuracy of probabilistic inference algorithms." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 3004–3014. Red Hook, NY, USA: Curran Associates Inc. 7

Domke, J. (2021). "An easy to interpret diagnostic for approximate inference: Symmetric divergence over simulations." arXiv:2103.01030 7

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019). "Visualization in Bayesian workflow." *Journal of the Royal Statistical Society: Series A*, 182: 389–402. MR3902665. doi: https://doi.org/10.1111/rssa.12378. 5

Gandy, A. and Scott, J. (2020). "Unit testing for MCMC and other Monte Carlo methods." arXiv:2001.06465 7, 25

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., and Modrák, M. (2020). "Bayesian workflow." arXiv:2011.01808. MR4298989. doi: https://doi.org/10.1214/20-ba1221. 5

Geweke, J. (2004). "Getting it right." *Journal of the American Statistical Association*, 99: 799–804. MR2090912. doi: https://doi.org/10.1198/016214504000001132. 2, 6, 25

Grinsztajn, L., Semenova, E., Margossian, C. C., and Riou, J. (2021). "Bayesian workflow for disease transmission modeling in Stan." *Statistics in Medicine*, 40: 6209–6234. MR4339396. doi: https://doi.org/10.1002/sim.9164. 12

Grosse, R. B., Ancha, S., and Roy, D. M. (2016). "Measuring the reliability of MCMC inference with bidirectional Monte Carlo." In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/0e9fa1f3e9e66792401a6972d477dcc3-Paper.pdf 7

Kay, M. (2021). "Extracting and visualizing tidy residuals from Bayesian models." URL http://mjskay.github.io/tidybayes/articles/tidybayes-residuals.html 5

Kim, S., Moon, A. H., Modrák, M., and Säilynoja, T. (2022). "SBC: Simulation based calibration for rstan/cmdstanr models." URL https://github.com/hyunjimoon/SBC/ 6

Lee, J. E., Nicholls, G. K., and Ryder, R. J. (2019). "Calibration procedures for approximate Bayesian credible sets." *Bayesian Analysis*, 14: 1245–1269. MR4044852. doi: https://doi.org/10.1214/19-BA1175. 5, 12

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. (2021). "Benchmarking simulation-based inference." *Proceedings of Machine Learning Research*, 130: 343–351. URL https://proceedings.mlr.press/v130/lueckmann21a.html 5, 12

Mcleod, J. and Simpson, F. (2021). "Validating Gaussian process models with simulation-based calibration." In *2021 IEEE International Conference on Artificial Intelligence Testing (AITest)*, 101–102. 7

Modrák, M., Moon, A. H., Kim, S., Bürkner, P., Huurre, N., Faltejsková, K., Gelman, A., and Vehtari, A. (2023). "Supplementary Material for "Simulation-based calibration checking for Bayesian computation: The choice of test quantities shapes sensitivity"." 7, 8, 9, 12

Prangle, D., Blum, M. G. B., Popovic, G., and Sisson, S. A. (2014). "Diagnostic tools for approximate Bayesian computation using the coverage property." *Australian & New Zealand Journal of Statistics*, 56: 309–329. MR3300163. doi: https://doi.org/10.1111/anzs.12087. 7, 10, 17, 25

Radev, S. T., D'Alessandro, M., Mertens, U. K., Voss, A., Köthe, U., and Bürkner, P.-C. (2021). "Amortized Bayesian model comparison with evidential deep learning." *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. MR3796894. doi: https://doi.org/10.1109/tnnls.2017.2665555. 7

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. (2020). "BayesFlow: Learning complex stochastic models with invertible neural networks." *IEEE Transactions on Neural Networks and Learning Systems*, 33(4): 1452–1466. MR4516681. 7

Radev, S. T., Schmitt, M., Pratz, V., Picchini, U., Köthe, U., and Bürkner, P.-C. (2023). "JANA: Jointly amortized neural approximation of complex Bayesian models." In *Uncertainty in Artificial Intelligence (UAI) Conference Proceedings*. 2, 7

Ramesh, P., Lueckmann, J.-M., Boelts, J., Tejero-Cantero, Á., Greenberg, D. S., Goncalves, P. J., and Macke, J. H. (2022). "GATSBI: Generative Adversarial Training for Simulation-Based Inference." In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=kR1hC6j48Tp 5, 12

Rendsburg, L., Kristiadi, A., Hennig, P., and Von Luxburg, U. (2022). "Discovering inductive bias with Gibbs priors: A diagnostic tool for approximate Bayesian inference." *Proceedings of Machine Learning Research*, 151: 1503–1526. URL https://proceedings.mlr.press/v151/rendsburg22a.html 7

Saad, F. A., Freer, C. E., Ackerman, N. L., and Mansinghka, V. K. (2019). "A family of exact goodness-of-fit tests for high-dimensional discrete distributions." *Proceedings of Machine Learning Research*, 89: 1640–1649. URL https://proceedings.mlr.press/v89/saad19a.html 5, 6, 8, 9, 11, 12

Säilynoja, T., Bürkner, P.-C., and Vehtari, A. (2022). "Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison." *Statistics and Computing*, 32(2). MR4402179. doi: https://doi.org/10.1007/s11222-022-10090-6. 4, 13

Schad, D. J., Nicenboim, B., Bürkner, P.-C., Betancourt, M., and Vasishth, S. (2022). "Workflow techniques for the robust use of Bayes factors." *Psychological Methods*. URL https://doi.org/10.1037/met0000472 5, 7, 12

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2020). "Validating Bayesian inference algorithms with simulation-based calibration." URL http://www.stat.columbia.edu/~gelman/research/unpublished/sbc.pdf 2, 3, 4, 7

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021). "Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion)." *Bayesian Analysis*, 16(2): 667–718. MR4298989. doi: https://doi.org/10.1214/20-ba1221. 24

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018). "Yes, but did it work?: Evaluating variational inference." *Proceedings of Machine Learning Research*, 80: 5581–5590. URL https://proceedings.mlr.press/v80/yao18a.html 2, 7

Yu, X., Nott, D. J., Tran, M.-N., and Klein, N. (2021). "Assessment and adjustment of approximate inference algorithms using the law of total variance." *Journal of Computational and Graphical Statistics*, 30: 977–990. MR4356599. doi: https://doi.org/10.1080/10618600.2021.1880921. 6

Zhang, Y. D., Naughton, B. P., Bondell, H. D., and Reich, B. J. (2020). "Bayesian regression using a prior on the model fit: The R2-D2 shrinkage prior." *Journal of the American Statistical Association*, 117: 862–874. MR4436318. doi: https://doi.org/10.1080/01621459.2020.1825449. 10

Zhao, D., Dalmasso, N., Izbicki, R., and Lee, A. B. (2021). "Diagnostics for conditional density models and Bayesian inference algorithms." *Proceedings of Machine Learning Research*, 161: 1830–1840. URL https://proceedings.mlr.press/v161/zhao21b.html 5