Validating Bayesian Inference Algorithms with Simulation-Based Calibration

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, Andrew Gelman

Abstract. Verifying the correctness of Bayesian computation is challenging. This is especially true for complex models that are common in practice, as these require sophisticated model implementations and algorithms. In this paper we introduce *simulation-based calibration* (SBC), a general procedure for validating inferences from Bayesian algorithms capable of generating posterior samples. This procedure not only identifies inaccurate computation and inconsistencies in model implementations but also provides graphical summaries that can indicate the nature of the problems that arise. We argue that SBC is a critical part of a robust Bayesian workflow, as well as being a useful tool for those developing computational algorithms and statistical software.

1. INTRODUCTION

Powerful algorithms and computational resources are facilitating Bayesian modeling in an increasing range of applications. Conceptually, constructing a Bayesian analysis is straightforward. We first define a joint distribution over the parameters, θ, and measurements, y, with the specification of a prior distribution and likelihood,

$$\pi(y, \theta) = \pi(y \mid \theta)\, \pi(\theta).$$

Conditioning this joint distribution on an observation, $\tilde{y}$, yields a posterior distribution,

$$\pi(\theta \mid \tilde{y}) \propto \pi(\tilde{y}, \theta),$$

ISERP, Columbia University, New York. (e-mail: sean.talts@gmail.com). Symplectomorphic LLC., New York. (e-mail: betanalpha@gmail.com). Department of Statistical Sciences, University of Toronto, Toronto. (e-mail: simpson@utstat.toronto.edu). Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Finland. (e-mail: Aki.Vehtari@aalto.fi). Department of Statistics and Department of Political Science, Columbia University, New York. (e-mail: gelman@stat.columbia.edu).

that encodes information about the system being analyzed.

Implementing this Bayesian inference in practice, however, can be computationally challenging when applied to large and structured datasets. We must make our model rich enough to capture the relevant structure of the system being studied while simultaneously being able to accurately work with the resulting posterior distribution. Unfortunately, every algorithm in computational statistics requires that the posterior distribution possesses certain favorable properties in order to be successful. Consequently the overall performance of an algorithm is sensitive to the details of the model and the observed data, and an algorithm that works well in one analysis can fail spectacularly in another.

As we move towards creating sophisticated, bespoke models with each analysis, we stress the algorithms in our statistical toolbox. Moreover, the complexity of these models provides abundant opportunity for mistakes in their specification. We must verify both that our code is implementing the model we think it is and that our inference algorithm is able to perform the necessary computations accurately. While we always get some result from a given algorithm, we have no idea how good it might be without some form of validation.

Fortunately, the structure of the Bayesian joint distribution allows for the validation of *any* Bayesian computational method capable of producing samples from the posterior distribution, or an approximation thereof. This includes not only Monte Carlo methods but also deterministic methods that yield approximate posterior distributions amenable to exact sampling, such as integrated nested Laplace approximation (INLA) (Rue, Martino and Chopin, 2009; Rue et al., 2017) and automatic differentiation variational inference (ADVI) (Kucukelbir et al., 2017). In this paper we introduce *Simulation-Based Calibration* (SBC), a generic and straightforward procedure for validating these algorithms within the scope of a given Bayesian joint distribution.

We begin with a discussion the natural self-consistency of samples from the Bayesian joint distribution and previous validation methods that have exploited this behavior. Next we introduce the simulation-based calibration framework and examine the qualitative interpretation of the SBC output, how it identifies how the algorithm being validated might be failing, and how it can be incorporated into a robust Bayesian workflow. Finally, we consider some useful extensions of SBC before demonstrating the application of the procedure over a range of analyses.

2. SELF-CONSISTENCY OF THE BAYESIAN JOINT DISTRIBUTION

The most straightforward way to validate a computed posterior distribution is to compare computed expectations with the exact values. An immediate problem with this, however, is that we know the true posterior expectation values for only the simplest models. These simple models, moreover, typically have a different structure to the models of interest in applications. This motivates us to construct a validation procedure that does not require access to the exact expectations, or any other property of the true posterior distribution.

A popular alternative to comparing the computed and true expectation values directly is to define a ground truth $\tilde{\theta}$, simulate data from that ground truth, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$, and

then quantify how well the computed posterior recovers the ground truth in some way. Unfortunately this approach is flawed, as demonstarted in a simple example.

Consider the model

$$y \mid \mu \sim \mathrm{N}(\mu, 1^2)$$
$$\mu \sim \mathrm{N}(0, 1^2)$$

and an attempt at verification that utilizes the single ground truth value $\tilde{\mu} = 0$. If we simulate from this model and draw the plausible, but extreme, data value $\tilde{y} = 2.1$, then the true posterior will be $\mu \mid \tilde{y} \sim \mathrm{N}(1.05, 0.5^2)$. As $\tilde{\mu}$ is more than two posterior standard deviations from the posterior mean, we might be tempted to say that recovery has not been successful. On the other hand, imagine that we accidentally used code that exactly fits an identical model but with the variance for both the likelihood and prior set to 10 instead of 1. In this case, the incorrectly computed posterior would be $\mathrm{N}(1.05, 5^2)$ and we might conclude that the code correctly recovered the posterior.

Consequently, the behavior of the algorithm in any *individual* simulation will not characterize the ability of the inference algorithm to fit that particular model in any meaningful way. In the example above, it might lead us to conclude that the incorrectly coded analysis worked as desired, while the correctly coded analysis failed. In order to properly characterize an analysis we need to at the very least consider multiple ground truths.

Which ground truths, however, should we consider? An algorithm might be able to recover a posterior constructed from data generated from some parts of the parameter space while faring poorly on data generated from other parts of parameter space. In Bayesian inference a proper prior distribution quantifies exactly which parameter values are relevant and hence should be considered when evaluating an analysis. This immediately suggests that we consider the performance of an algorithm over the entire Bayesian joint distribution, first sampling a ground truth from the prior, $\tilde{\theta} \sim \pi(\theta)$, and then data from the corresponding data generating process, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$. We can then build inferences for each simulated observation $\tilde{y}$ and then compare the recovered posterior distribution to the sampled parameter $\tilde{\theta}$.

Advantageously, this procedure also defines a natural condition for quantifying the faithfulness of the computed posterior distributions, regardless of the structure of the model itself. Integrating the exact posteriors over the Bayesian joint distribution returns the prior distribution,

$$\pi(\theta) = \int \mathrm{d}\tilde{y}\, \mathrm{d}\tilde{\theta}\, \pi(\theta \mid \tilde{y})\, \pi(\tilde{y} \mid \tilde{\theta})\, \pi(\tilde{\theta}). \tag{1}$$

In other words, for *any* model the average of any exact posterior expectation with respect to data generated from the Bayesian joint distribution reduces to the corresponding prior expectation.

Consequently, any discrepancy between the *data averaged posterior* (1) and the prior distribution indicates some error in the Bayesian analysis. This error can come either from

inaccurate computation of the posterior or a mis-implementation of the model itself. Well-defined comparisons of these two distributions then provides a generic means of validating the analysis, at least within the scope of the modeling assumptions.

3. EXISTING VALIDATION METHODS EXPLOITING THE BAYESIAN JOINT DISTRIBUTION

The self-consistency of the data-averaged posterior (1) and the prior is not a novel observation. This behavior has been exploited in at least two earlier methods for validating Bayesian computational algorithms.

Geweke (2004) proposed a Gibbs sampler targeting the Bayesian joint distribution that alternatively samples from the posterior, $\pi(\theta \mid y)$, and the likelihood, $\pi(y \mid \theta)$. If an algorithm can generate accurate posterior samples, then this Gibbs sampler will produce accurate samples from the Bayesian joint distribution, and the marginal parameter samples will be indistinguishable from any sample of the prior distribution. The author recommended quantifying the consistency of the marginal parameter samples and a prior sample with z-scores of each parameter mean, with large z-scores indicating a failure of the algorithm to produce accurate posterior samples.

The main challenge with this method is that the diagnostic z-scores will be meaningful only once the Gibbs sampler has converged. Unfortunately, the data and the parameters will be strongly correlated in a generative model and the convergence of this Gibbs sampler will be slow, making it challenging to identify when the diagnostics can be considered.

Cook, Gelman and Rubin (2006) avoided the auxiliary Gibbs sampler entirely by considering quantiles of the simulated posterior distributions. They noted that if $\tilde{\theta} \sim \pi(\theta)$ and $\tilde{y} \sim \pi(y \mid \tilde{\theta})$ then the exact posterior quantiles for each parameter,

$$q(\tilde{\theta}) = \int \mathrm{d}\theta \, \pi(\theta \mid \tilde{y}) \, \mathbb{I}[\theta < \tilde{\theta}],$$

will be uniformly distributed provided that the posteriors are absolutely continuous. Consequently any deviation from the uniformity of the computed posterior quantiles indicates a failure in the implementation of the analysis.

The authors suggest quantifying the uniformity of a quantile sample by transforming them into z-scores with an application of the inverse normal cumulative distribution function. The absolute value of the z-scores can then be visualized to identify deviations from normality of, and hence uniformity of the quantiles. At the same time these deviations can be quantified with a χ^2 test.

This procedure works well in certain examples, as demonstrated by Cook, Gelman and Rubin (2006), but it can run into problems with algorithms that utilize samples, as the empirical quantiles only asymptotically approach the true quantiles. Markov chain Monte Carlo samples present additional challenges when autocorrelations are high and effective sample sizes are low, or when a central limit theorem does not hold at all. This makes it

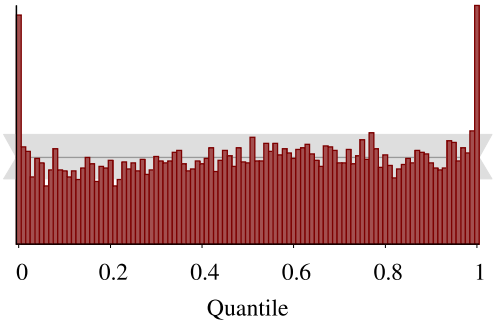

FIG 1. *The procedure of Cook, Gelman and Rubin (2006) applied to a linear regression analysis with Stan indicates significant problems despite the analysis itself being correct. In particular, the histogram of empirical quantiles (red) exhibits strong systematic deviations from the variation expected of a uniform histogram (gray).*

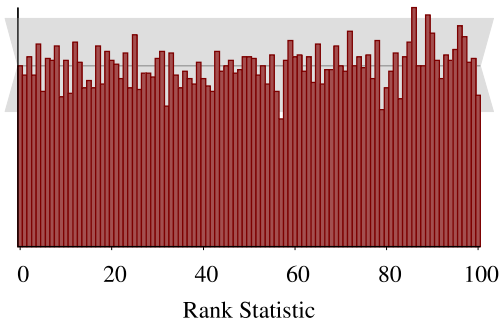

FIG 2. *SBC Algorithm 2 applied to a linear regression analysis indicates no issues as the empirical rank statistics (red) are consistent with the variation expected of a uniform histogram (gray).*

difficult to determine whether a deviation from normality is due to pre-asymptotic behavior or biases in the posterior computations.

In particular, because there are only $L + 1$ positions in a posterior sample of size L in between which the prior sample $\tilde{\theta}$ can fall, an empirical quantile is fundamentally discrete, taking one of $L + 1$ evenly spaced values on $[0, 1]$. This discretization causes artifacts when visualizing the quantiles and it requires some continuity corrections for the finite instances where the empirical quantile equals 0 or 1. At the same time, autocorrelation in the simulations creates dependence in the quantiles and modifies the distributions of test statistics that were worked out implicitly assuming independence, a point recognized in the recent correction (Gelman, 2017).

To demonstrate these issues, we run the Cook, Gelman and Rubin (2006) procedure for a straightforward linear regression model (Listings 1 and 2 in the Appendix) in Stan 2.17.1 (Carpenter et al., 2017). Although Stan is known to be extremely accurate for this analysis, a histogram of the empirical quantiles demonstrates strong deviations from uniformity (Figure 1) that immediately suggests algorithmic problems that aren't there. Moreover, a numerical quantification with the proposed inverse Normal CDF followed by a χ^2 test quantifying the resulting normality would immediately fail due to the large number of empirical quantiles exactly equaling 0 or 1. We also see evidence of autocorrelation in the posterior samples manifesting in the histogram, an issue we consider more thoroughly in Section 5.1.

4. SIMULATION-BASED CALIBRATION

We can work around the discretization artifacts of Cook, Gelman and Rubin (2006), however, by considering a similar consistency criterion that is immediately compatible with sampling-based algorithms. In this section we introduce *simulation-based calibration* (SBC), a procedure that builds histograms of *rank statistics* that will follow a discrete uniform distribution if the analysis has been correctly implemented.

SBC requires just one assumption: that we have a generative model for our data. Given such a model, we can run any given algorithm over many simulated observations and the self consistency condition (1) provides a target to verify that the algorithm is accurate over that ensemble, and hence sufficiently *calibrated* for the assumed model. This calibration ensures that certain one dimensional test statistics are correctly distributed under the assumed model and is similar to checking the coverage of a credible interval under the assumed model.

Importantly, this calibration is limited exclusively to the computational aspect of our analysis. It offers no guarantee that the posterior will cover the ground truth for any single observation or that the model will be rich enough to capture the truth at all. Understanding the range of posterior behaviors for a given observation requires a more careful *sensitivity analysis* while validating the model assumptions themselves requires a study of *predictive performance*, such as posterior predictive checks (PPCs, e.g., Gelman et al. (2014), chapter 6). Where SBC uses draws from the joint prior distribution $\pi(\theta, y)$, PPCs use the posterior predictive distribution for predicting new data $\tilde{y}$, $\pi(\tilde{y}|y)$. We view both of these checks as a vital part of a robust Bayesian workflow.

In this section we first demonstrate the expected behavior of rank statistics under a proper analysis and construct the SBC procedure to exploit this behavior. We then demonstrate how deviations from the expected behavior are interpretable and help identify the exact nature of implementation error.

4.1 Validating Consistency With Rank Statistics

Consider the sequence of samples from the Bayesian joint distribution and resulting posteriors,

$$\begin{aligned} \tilde{\theta} &\sim \pi(\theta) \\ \tilde{y} &\sim \pi(y \mid \tilde{\theta}) \\ \{\theta_1, \ldots, \theta_L\} &\sim \pi(\theta \mid \tilde{y}). \end{aligned} \tag{2}$$

The relationship (1) implies that the prior sample, $\tilde{\theta}$, and an exact posterior sample, $\{\theta_1, \ldots, \theta_L\}$, will be distributed according to the the same distribution. Consequently, for any one-dimensional random variable, $f : \Theta \to \mathbb{R}$ the *rank statistic* of the prior sample

Algorithm 1 SBC generates a histogram from an ensemble of rank statistics of prior samples relative to corresponding posterior samples. Any deviation from uniformity of this histogram indicates that the posterior samples are inconsistent with the prior samples. For a multidimensional problem the procedure is repeated for each parameter or quantity of interest to give multiple histograms.

Initialize a histogram with bins centered around $0, \ldots, L$.

for n in N **do**
  Draw a prior sample, $\tilde{\theta} \sim \pi(\theta)$
  Draw a simulated data set, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$
  Draw posterior samples $\{\theta_1, \ldots, \theta_L\} \sim \pi(\theta \mid \tilde{y})$
  for each one-dimensional random variable, f **do**
    Compute the rank statistic $r(\{f(\theta_1), \ldots, f(\theta_L)\}, f(\tilde{\theta}))$ as defined in (4.1)
    Increment the histogram with $r(\{f(\theta_1), \ldots, f(\theta_L)\}, \mid f(\tilde{\theta}))$

Analyze the histogram for uniformity.

relative to the posterior sample,

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

will be uniformly distributed across the integers $[0, L]$.

THEOREM 1. *Let $\tilde{\theta} \sim \pi(\theta)$, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$, and $\{\theta_1, \ldots, \theta_L\} \sim \pi(\theta \mid \tilde{y})$ for any joint distribution $\pi(y, \theta)$. The rank statistic of any one-dimensional random variable over θ is uniformly distributed over the integers $[0, L]$.*

The proof is given in Appendix B.

There are many ways of testing the uniformity of the rank statistics, but the SBC procedure, outlined in Algorithm 1, exploits a histogram of rank statistics for a given random variable to enable visual inspection of uniformity (Figure 3). We first sample N draws from the Bayesian joint distribution. For each replicated generated dataset we then sample L exact draws from the posterior distribution and compute the corresponding rank statistic. We then bin the L rank statistics in a histogram spanning the $L + 1$ possible values, $\{0, \ldots, L\}$. If only correlated posteriors samples can be drawn then the procedure can be modified as discussed in Section 5.1.

In order to help identify deviations, each histogram is complemented with a gray band indicating 99% of the variation expected from a uniform histogram. Formally, the vertical extent of the band extends from the 0.005 quantile to the 0.995 quantile of the Binomial$(N, (L + 1)^{-1})$ distribution so that under uniformity we expect that, on average, the counts in only one bin a hundred will deviate outside this band.

In complex problems computational resources often limit the number of replications, N, and hence the sensitivity of the resulting SBC histogram. In order to reduce the noise from

small replications it can be beneficial to uniformly bin the histogram, for example by pairing neighboring ranks together into a single bin to give $B = L/2$ total bins. Our experiments have shown that keeping $N/B \approx 20$ lead to a good trade-off between the expressiveness of the binned histogram and the necessary variance reduction. Choosing $L + 1$ to be divisible by a large power of 2 makes this re-binning easier; for example, instead of generating 1000 samples in a problem with known computational limitations we recommend generating 999 samples or, even better, $1024 - 1 = 1023$ samples from the posterior distributions.

Regardless of the binning, however, it will be difficult to identify sufficiently small deviations in the SBC histogram and it can be useful to consider alternative visualizations of the rank statistics. We consider this Section 5.2.

4.2 Interpreting SBC

What makes the SBC procedure particularly useful is that the deviations from uniformity in the SBC histogram can indicate *how* the computed posteriors are incorrect. We follow an observation from the forecast calibration literature (Anderson, 1996; Hamill, 2001), which suggests that the way the rank histogram deviates from uniformity can indicate bias or mis-calibration of the computed posterior distributions.

A histogram without any appreciable deviations is shown in Figure 3. The histogram of rank statistics is consistent with the expected uniform behavior, here shown with the 99% interval in light gray and the median in dark gray.

Figure 4 demonstrates the deviation from uniformity exhibited by correlated posterior samples. The correlation between the posterior samples causes them to cluster relative to the proceeding prior sample, biasing the ranks to extremely small or large values. The similarity to Figure 1 is no coincidence. We describe how to process correlated posterior samples generated from Markov chain Monte Carlo algorithms in Section 5.1.

Next, consider a computational algorithm that produces, on average, posteriors that are *overdispersed* relative to the true posterior. When averaged over the Bayesian joint distribution this results in a data-averaged posterior distribution (1) that is overdispersed relative to the prior distribution (Figure 5a), and hence rank statistics that are biased towards the extremes that manifests as a characteristic $\cap$-shaped histogram (Figure 5b).

Conversely, an algorithm that computes posteriors that are, on average, *under-dispersed* relative to the true posterior produces a histogram of rank statistics with a characteristic $\cup$ shape (Figure 6).

Finally, we might have an algorithm that produces posteriors that are biased above or below the true posterior. This bias results in a data-averaged posterior distribution biased in the same direction relative to the prior distribution (Figure 7a) and rank statistics that are biased in the opposite direction (Figure 7b). For example, posterior samples biased to smaller values results in higher rank statistics, where as posterior samples biased to larger values results in lower rank statistics.

A misbehaving analysis can in general manifest many of these deviations at once. Because each deviation is relatively distinct from the others, however, in practice the systematic

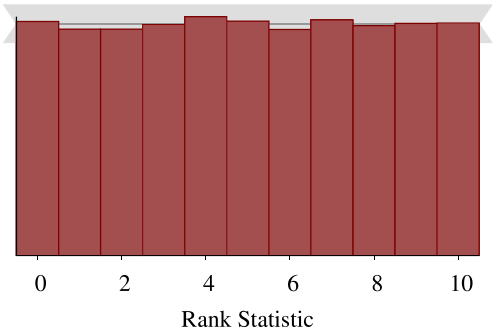

FIG 3. *Uniformly distributed rank statistics are consistent with the ranks being computed from independent samples from the exact posterior of a correctly specified model.*

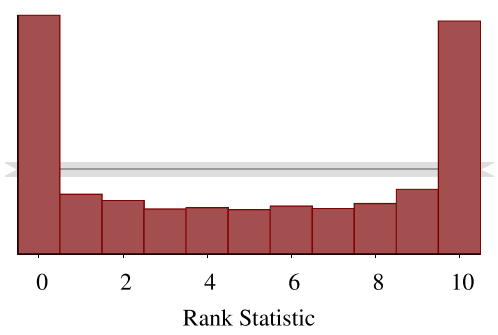

FIG 4. *The spikes at the boundaries of the SBC histogram indicate that posterior samples possess non-negligible autocorrelation.*

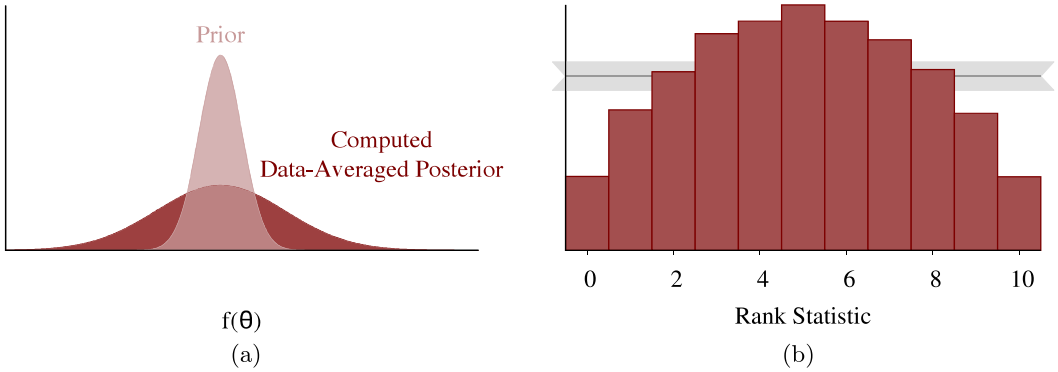

FIG 5. *A symmetric, ∩-shaped distribution indicates that the computed data-averaged posterior distribution (dark red) is overdispersed relative to the prior distribution (light red). This implies that on average the computed posterior will be wider than the true posterior.*

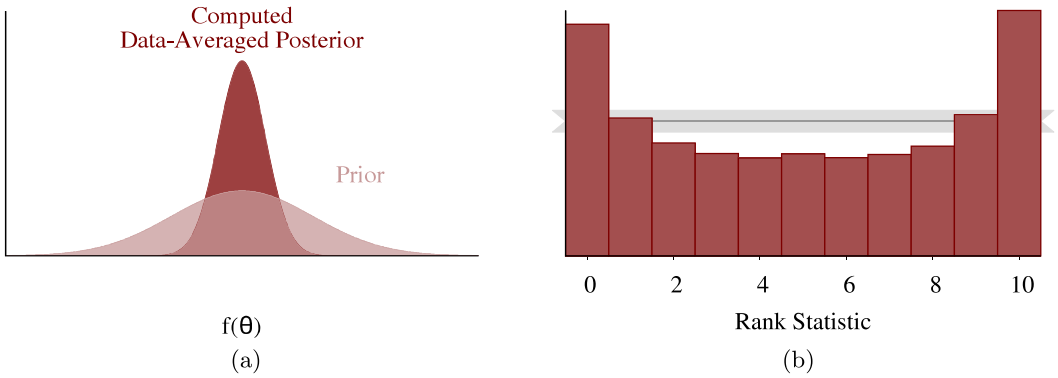

FIG 6. *A symmetric ∪ shape indicates that the computed data-averaged posterior distribution (dark red) is under-dispersed relative to the prior distribution (light red). This implies that on average the computed posterior will be narrower than the true posterior.*

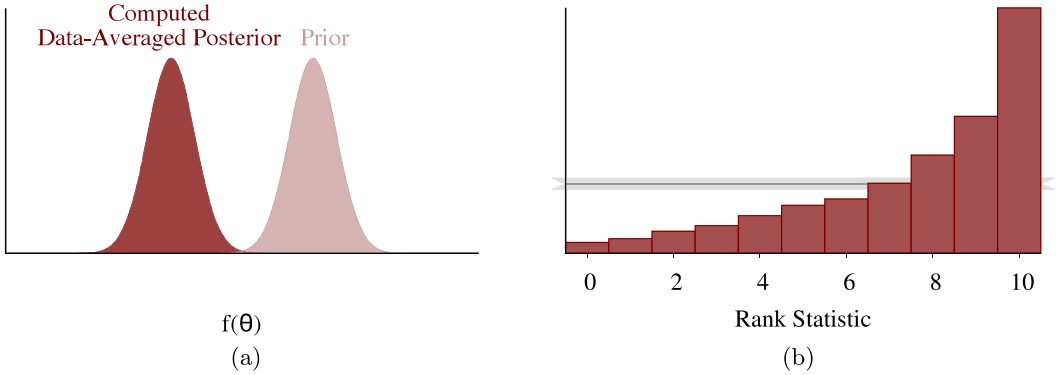

FIG 7. *Asymmetry in the rank histogram indicates that the computed data-averaged posterior distribution (dark red) will be biased in the opposite direction relative to the prior distribution (light red). This implies that on average the computed posterior will be biased in the same opposite direction.*

deviations are readily separated into the different behaviors if they are large enough.

4.3 Simulation-Based Calibration Plays a Vital Role in a Robust Bayesian Workflow

SBC is one of the few tools for evaluating the critical but frequently unexamined choice of computational method made in any Bayesian analysis. We have already argued that performance on a single simulated observation is, at best, a blunt instrument. Moreover, while most theoretical results only provide asymptotic comfort, SBC adapts to the specific model design under consideration.

Furthermore, because SBC validates accuracy through one-dimensional random variables we can use carefully chosen random variables to make targeted assessments of an analysis based on our inferential needs and priorities. As these needs and priorities change we can run SBC again to verify the analysis anew.

The downside of using SBC in practice is that it is expensive; instead of fitting a single observation we have to fit N simulated observations before even considering the measured data. These fits, however, are embarrassingly parallel, which makes it possible to leverage access to computational resources through multicore personal computers, computing clusters, and cloud computing. For example, all of the examples in Section 6 were run on clusters and took, at most, a few hours.

The procedure can be sped up further by reducing the number of independent draws needed from the posterior at the cost of losing some sensitivity. Even a few simulations are useful to catch gross problems in an analysis.

5. EXTENDING SIMULATION-BASED CALIBRATION

SBC provides a straightforward procedure for validating simulation-based algorithms applied to Bayesian analyses, but the procedure can be limited in a few circumstances. In this section we discuss some small modifications that allow SBC to remain useful in some common practical circumstances.

5.1 Mitigating the Effect of Autocorrelation

As we saw in Section 4.2, SBC histograms will deviate from uniformity if the posterior samples are correlated, making it difficult to identify any bias in the samples. Unfortunately this limits the utility of the ideal SBC procedure when applied to Markov chain Monte Carlo (MCMC) algorithms. Given the popularity of these algorithms in practice, and the consequent need for validation schemes, however, it would be highly beneficial to mitigate the effects of autocorrelation to best utilize the SBC procedure. Fortunately, we can readily ameliorate the effects of autocorrelation with an appropriate thinning scheme.

Under certain ergodicity conditions, Markov chain Monte Carlo estimators achieve a central limit theorem,

$$\frac{1}{N_{\text{eff}}} \sum_{n=1}^{N_{\text{eff}}} f(\theta_n) \sim \text{N}\left(\mathbb{E}[f], \frac{\mathbb{V}[f]}{N_{\text{eff}}[f]}\right),$$

where $\mathbb{E}[f]$ is the posterior expectation of a function f, $\mathbb{V}[f]$ is the variance of f, and $N_{\text{eff}}[f]$ is the effective sample size for f,

$$N_{\text{eff}}[f] = \frac{N_{\text{samp}}}{1 + 2\sum_{m=0}^{\infty} \rho_m[f]},$$

with $\rho_m[f]$ the lag-m autocorrelation of f, which we estimate from the realized Markov chain (Gelman et al., 2014, Ch. 11). In words, N_{samp} correlated samples contains roughly the same information as N_{eff} exact samples when estimating the expectation of f.

This suggests that thinning a Markov chain by keeping only every $N_{\text{samp}}/N_{\text{eff}}[f]$ states should yield a sample with negligible autocorrelation that is well-suited for the SBC procedure with f, giving us (Algorithm 2). By carefully thinning the autocorrelated samples we should be able to significantly reduce the $\cup$ shape demonstrated in Figure 4 and maximize the sensitivity to any remaining issues with the model or algorithm. When running the SBC procedure over multiple quantities of interest we suggest thinning the chain using the minimum $N_{\text{eff}}[f]$.

Algorithm 2 Simulation-based calibration can be applied to the correlated posterior samples generated by a Markov chain provided that the Markov chain can be thinned to L effective samples at each iteration.

Initialize a histogram with bins centered around $0, \ldots, L$.

for n in N **do**
  draw a prior sample $\tilde{\theta} \sim \pi(\theta)$
  draw a simulated data set $\tilde{y} \sim \pi(y \mid \tilde{\theta})$
  run a Markov chain for L' iterations to generate the correlated posterior samples,
    $\{\theta_1, \ldots, \theta_{L'}\} \sim \pi(\theta \mid \tilde{y})$
  compute the effective sample size, $N_{\text{eff}}[f]$ of $\{\theta_1, \ldots, \theta_{L'}\}$ for the function f
  if $N_{\text{eff}}[f] < L$ **then**
    rerun the Markov for $L' \cdot L/N_{\text{eff}}[f]$ iterations
  uniformly thin the correlated sample to L states and truncate any leftover draws at L
  compute the rank statistic $r(\{f(\theta_1), \ldots, f(\theta_L)\}, f(\tilde{\theta}))$ as defined in (4.1)
  increment the histogram with $r(\{f(\theta_1), \ldots, f(\theta_L)\}, f(\tilde{\theta}))$

Analyze the histogram for uniformity.

Although some autocorrelation will remain in a sample that has been thinned by effective sample size, our experience has been that this strategy is sufficient to remove the autocorrelation artifacts from the SBC histogram. If desired, more conservative thinning strategies, such as the truncation rules of Geyer (1992) can remove autocorrelation completely from the sample. A sample thinned with these rules is typically much smaller than the sample achieved by thinning based on the effective sample size, and we have not seen any significant benefit for SBC from the increased computation time needed for these more elaborate thinning methods to date.

Deviations that cannot be mitigated by thinning provide strong evidence that the Markov chain Monte Carlo estimators do not follow a central limit theorem and the Markov chains

are not adequately exploring the target parameter space. This is particularly useful given that establishing central limit theorems for particular Markov chains and particular target distributions is a notoriously challenging problem even in relatively simple circumstances.

5.2 Simulation-Based Calibration for Small Deviations

The SBC histogram provides a general and interpretable means of identifying deviations from uniformity of the rank statistics and hence inaccuracies in our posterior computation, at least when the inaccuracies are large enough. For small deviations, however, the SBC histogram may not be sensitive enough for the deviations to be evident and other visualization strategies may be advantageous.

One option is to bin the SBC histogram multiple times to see if any deviation persists regardless of the binning. This approach, however, is ungainly to implement when there are many parameters and can be difficult to interpret. In particular, considering multiple histograms introduces a vulnerability to multiple testing biases.

Another approach is to pair the SBC histogram with the empirical cumulative distribution function (ECDF) which reduces variation at small and large ranks, making it easier to identify deviations around those values (Figure 8b). The deviation of the empirical CDF away from the expected uniform behavior is especially useful for identifying these small deviations (Figure 8c).

More subtle deviations can be isolated by considering more particular summary statistics, such as ranks quantiles or averages. While these have the potential to identify small biases they can also be harder to interpret and not as sensitive to the systematic deviations that manifest in the SBC histogram. Identifying a robust suite of diagnostic statistics is an open area of research and at present we recommend using the SBC histogram whenever possible.

6. EXPERIMENTS

In this section we consider the application of SBC on a series of examples that demonstrates the utility of the procedure for identifying and correcting incorrectly implemented analyses. For each example we implement the SBC procedure using posterior samples $L = 100$ so that, if the algorithm is properly calibrated, then the rank statistics will follow a $U[0, 100]$ discrete uniform distribution. The experiments in Section 6.1 through Section 6.3 used $N = 10,000$ replicated observations while the experiment in Section 6.4 used $N = 1000$ replicated observations.

6.1 Misspecified Prior

Let's first consider the case where we build our posterior using a different prior than that which we use to generate prior samples. This is not an uncommon mistake, even when models are specified in probabilistic programming languages.

Consider the linear regression model that we used before (Listing 2 in the Appendix) but with the prior on β modified to $\mathrm{N}(0, 1^2)$. With the prior samples still drawn according to $\mathrm{N}(0, 10^2)$, we expect that the posterior for β will be under-dispersed relative to the prior

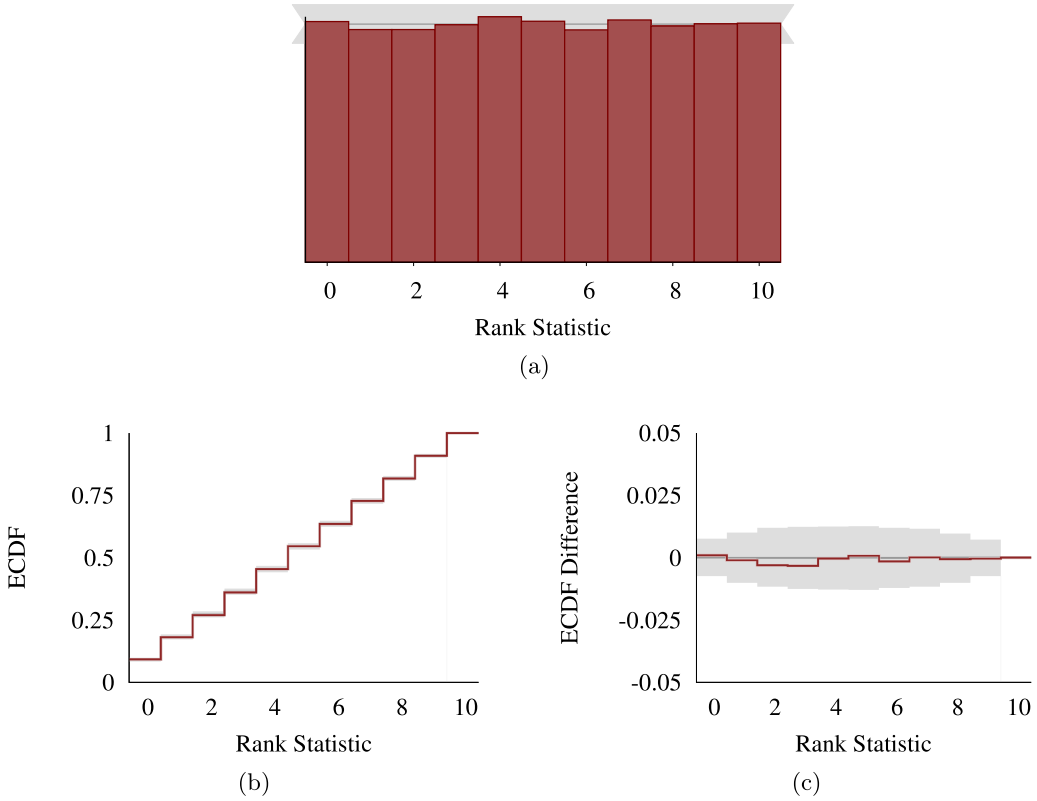

FIG 8. *In order to emphasize small deviations at low and large ranks we can pair the (a) SBC histogram with the corresponding (b) empirical cumulative distribution function (dark red) along with the variation expected of the empirical cumulative distribution function under uniformity. (c) Deviations are often easier to identify by subtracting the expected uniform behavior from the empirical cumulative distribution function.*

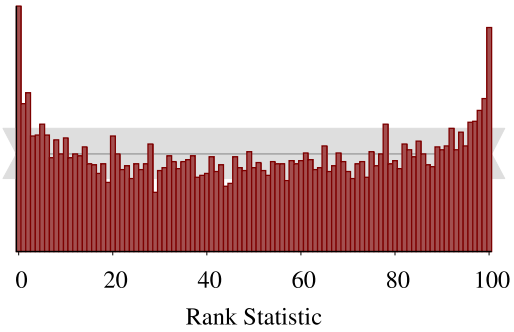

FIG 9. *When the data are simulated using a much wider prior than was used to fit the model, the SBC histogram for a regression parameter β exhibits a characteristic $\cup$-shape.*

even when the computation is exact. This should then lead to the deviation demonstrated in Figure 6 and, indeed, we see the characteristic $\cup$ shape in the SBC histogram for β (Figure 9).

6.2 Biased Markov chain Monte Carlo

Hierarchical models implemented with a centered parameterization (Papaspiliopoulos, Roberts and Sköld, 2007) are known to exhibit a challenging geometry that can cause MCMC algorithms to return biased posterior samples. While some algorithms, such as Hamiltonian Monte Carlo (Neal et al., 2011; Betancourt and Girolami, 2013) provide diagnostics capable of identifying this problem, these diagnostics are not available for general MCMC algorithms. Consequently the SBC procedure will be particularly useful in hierarchical models if it can identify this problem.

Here we consider a hierarchical model of the eight schools data set Rubin (1981) using a centered parameterization (Listing 3 in the Appendix). In this example the centered parameterization exhibits a classic funnel shape that contracts into a region of strong curvature around small values of τ, making it difficult for most Markov chain methods to adequately explore.

The SBC rank histogram for τ produced from Algorithm 1 clearly demonstrates that the posterior samples from Stan's dynamic Hamiltonian Monte Carlo extension of the NUTS algorithm (Hoffman and Gelman, 2011; Betancourt, 2017) are biased below the prior samples, consistent with the known pathology (Figure 12b). Here we used Algorithm 1 instead of 2 because the algorithm's unfaithfulness is evident over the deviation caused by the autocorrelation. Moreover, the extra computation required to return $L = 100$ effective samples post-thinning is impractical here as the centered parameterization, among other failing HMC diagnostics, has a low effective sample size per sample rate.

The corresponding non-centered parameterization should behave much better. Indeed, the SBC histogram thinned using Algorithm 2 (Figure 11) shows no deviation from uni-

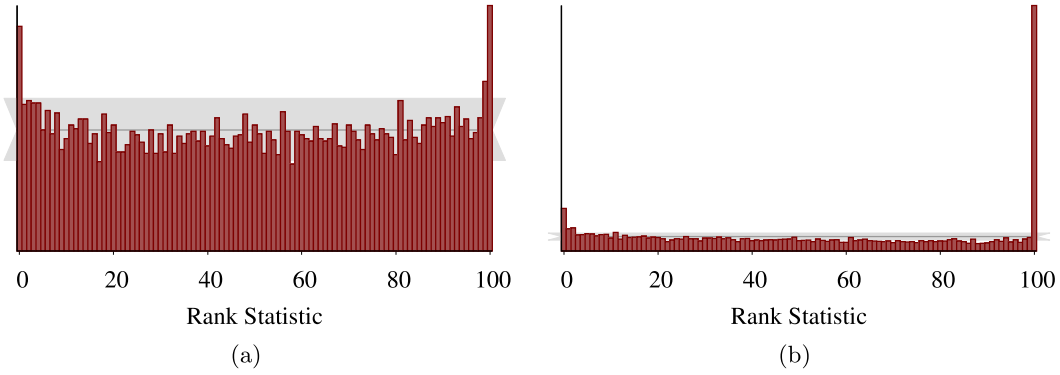

FIG 10. *Even without thinning, the underlying Markov chains, the SBC histograms for* $\theta[1]$ *and* τ *in the 8 schools centered parameterization of Section 6.2 demonstrate that Hamiltonian Monte Carlo yields samples that are biased towards larger values of* τ *than were used to generate the data.*

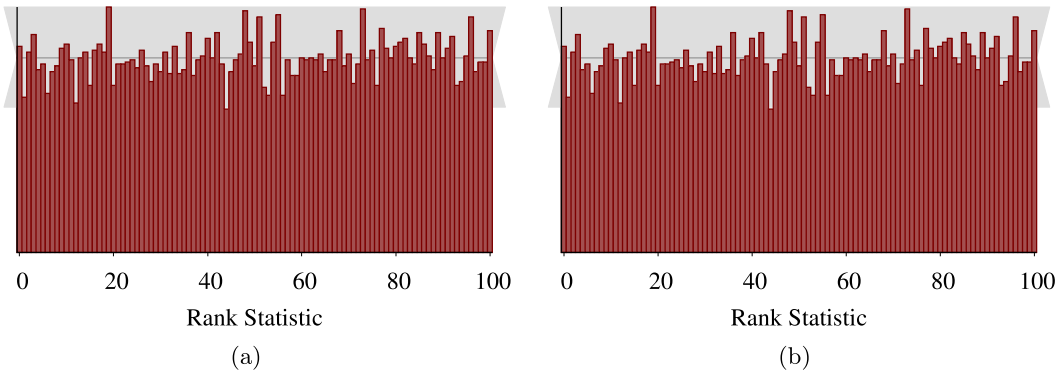

FIG 11. *Once thinned, the SBC histogram for* $\theta[1]$ *and* τ *from the 8 schools non-centered parameterization in Section 6.2 show no evidence of bias.*

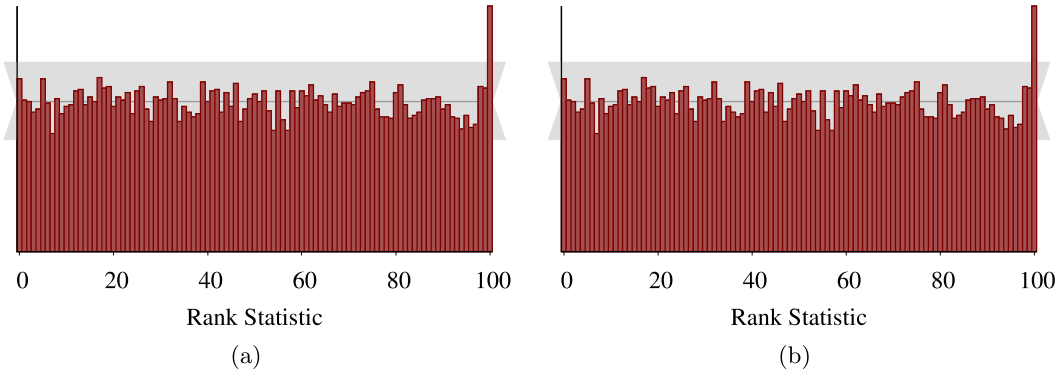

FIG 12. *Without thinning, the SBC histogram for $\theta[1]$ and τ from the 8 schools non-centered parameterization in Section 6.2 exhibits characteristic signs of autocorrelation in the posterior samples.*

formity as we expected given that Hamiltonian Monte Carlo is known to yield accurate computation for this analysis. If the SBC histogram is computed without thinning (Figure 12), the autocorrelation manifests as a large spikes at $L = 100$, consistent with the discussion in Section 5.1.

6.3 ADVI can fail for simple models

We next consider automatic differentiation variational inference (ADVI) applied to our linear regression model (Listing 2 in the Appendix). In particular, we run the implementation of ADVI in Stan 2.17.1 that returns exact samples from a variational approximation to the posterior. Here we use Algorithm 1 again because we know that ADVI does not produce autocorrelated posterior samples.

Algorithm 1 immediately identifies that the variational approximation found by ADVI drastically underestimates the posterior for the slope, β (Figure 13). Compare this with the results from Hamiltonian Monte Carlo (Figure 2), which yields a rank histogram consistent with uniformity.

6.4 INLA is slightly biased for spatial disease prevalence mapping

Finally let's consider a sophisticated spatial model for HIV prevalence fit to data from the 2003 Demographic Health Survey in Kenya (Corsi et al., 2012). We follow the experimental setup of (Wakefield, Simpson and Godwin, 2016) and fit the model using INLA.

The data were collected by dividing Kenya into 400 enumeration areas (EAs) and in the ith EA randomly sampling m_i households, with the jth household containing N_{ij} people. Both m_i and N_{ij} are chosen to be consistent with the Kenya DHS 2003 AIDS recode. The

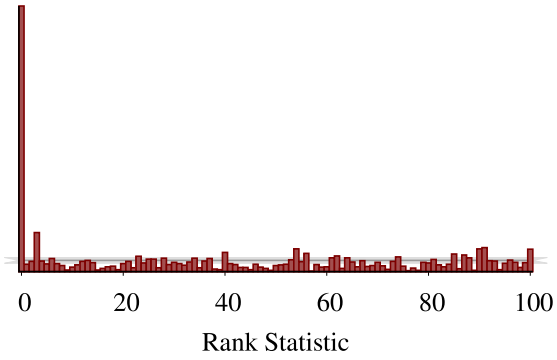

FIG 13. *The SBC histogram resulting from applying ADVI on the simple linear regression model indicates that the algorithm is strongly biased towards larger values of β in the true posterior.*

number of positive responses y_{ij} is modeled as

$$\begin{aligned} y_{ij} &\sim \text{Bin}(N_{ij}, p_{ij}) \\ p_{ij} &= \text{logit}^{-1}(\beta_0 + S(x_i) + \epsilon_{ij}), \end{aligned}$$

where $S(\cdot)$ is a Gaussian process, x_i is the centroid of the ith EA, and ϵ_{ij} are iid Gaussian error terms with standard deviation τ. Following the computation reasoning of Wakefield, Simpson and Godwin (2016) we approximate $S(\cdot)$ using the stochastic partial differential equation approximation (Lindgren, Rue and Lindström, 2011) to a Gaussian process with isotropic covariance function

$$c(x_1, x_2; \rho, \sigma) = \frac{\sqrt{8}\sigma^2}{\rho} \|x_1 - x_2\| K_1 \left(\frac{\sqrt{8}}{\rho} \|x_1 - x_2\| \right),$$

where ρ is the distance at which the spatial correlation between points is approximately 0.1, σ is the pointwise standard deviation, and $K_1(\cdot)$ is a modified Bessel function of the second kind.

To complete the model, we must specify priors on β_0, ρ, σ, and τ. We specify a $\text{N}(-2.5, 1.5^2)$ prior on the logit baseline prevalence β_0. This prior is based on the national HIV prevalence across the world ranges from 0.3% to 20% (Central Intelligence Agency, 2018). We use penalized complexity priors (Simpson et al., 2017; Fuglstad et al., 2017) on the remaining parameters tuned to ensure $\Pr(\rho < 0.1) = \Pr(\sigma > 1) = \Pr(\tau > 1) = 0.1$.

One of the quantities of interest for this model is the average prevalence over a subregion A of Kenya

$$\frac{1}{|A|} \int_A \text{logit}^{-1}(\beta_0 + S(x))\, dx.$$

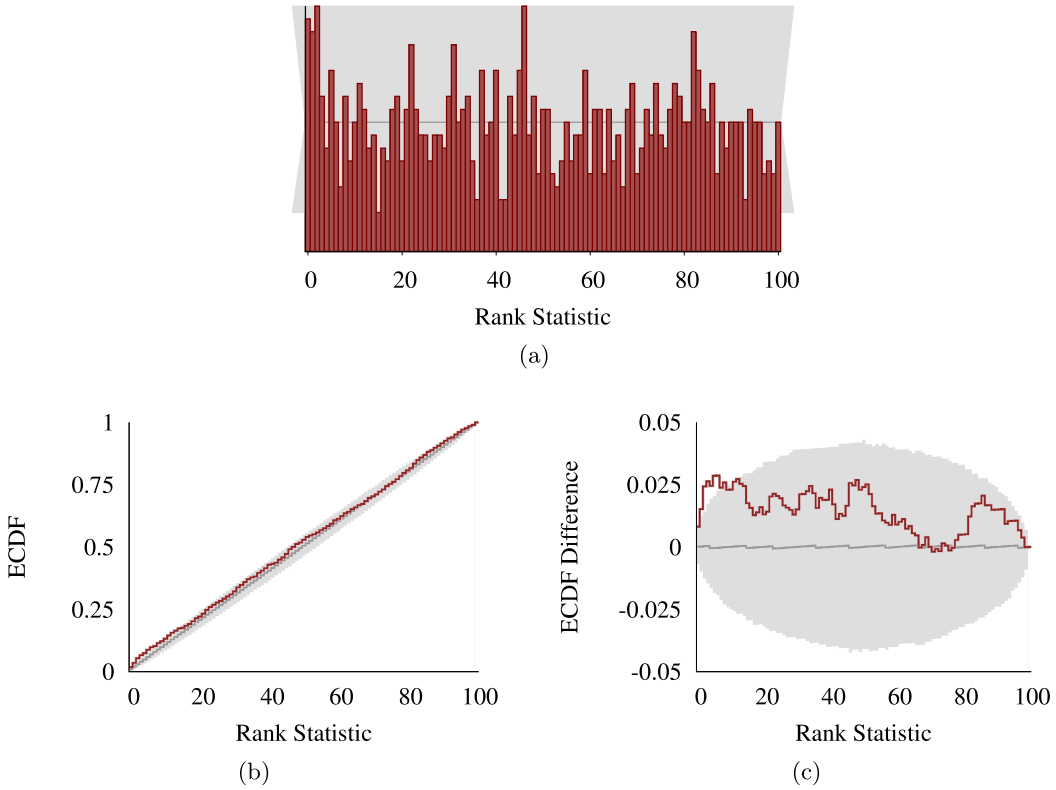

FIG 14. *(a) The SBC histogram for the average prevalence of a spatial model doesn't exhibit any obvious deviations, although the large span of the expected variation (gray) suggests that the SBC histogram may be underpowered. (b) The empirical cumulative distribution function (dark red), however, shows that there is a small deviation at low ranks beyond the variation expected from a uniform distribution (gray). (c) The deviation is more evident by looking at the difference between the empirical cumulative distribution function and the stepwise-linear behavior expected of a discrete uniform distribution.*

Wakefield, Simpson and Godwin (2016) suggested fitting this model using the `R-INLA` package to speed up the computation. As the quantity of interest is a non-linear transformation of a number of parameters, we need to use the `R-INLA`'s approximate posterior sampler, which is a relatively recent feature (Seppä et al., 2017).

Figure 14a shows the SBC histogram for $N = 1000$ replications to which are limited given the relatively high cost to run INLA in this model. The histogram shows that all of the ranks fall within the gray bars, but the large span of the bars indicates that the visual diagnostic may be underpowered. In our tests, we saw that it's common for deviations from a uniform distribution to be sufficiently severe that this histogram will still exhibit the signs of a poorly fitting procedure. Hence for a more fine-scale view of the fit we follow the recommendation in Section 5.2 and consider the ECDF (Figure 14b, c). Here we see

that low ranks are seen slightly more often in the computed ranks than we would expect from a uniform distribution.

It is not surprising that INLA exhibits some bias in this example. Binomial data with low expected counts does not contain much information, which poses some problems for the Laplace approximation. Even though this feature is only present when the observed values of y_{ij}/N_{ij} are close to zero, the SBC procedure is a sufficiently sensitive instrument to identify the problem. Overall, we would view INLA as a good approximation in a country like Kenya where the national prevalence is around 5.4%, while it would be inappropriate in Australia where the prevalence is 0.1% (Central Intelligence Agency, 2018). If we repeated this type of survey in a country with only 0.1% prevalence, however, then we would end up with too many zero observations for the method to be useful.

7. CONCLUSION

In this paper, we introduce SBC, a readily-implemented procedure that can identify sources of poorly implemented analyses, including biased computational algorithms or incorrect model specifications. The visualizations produced by the procedure allow us to not only identify that a problem exists but also learn how the problem will affect resulting inferences. The ability to both identify and interpret these issues makes SBC an important step in a robust Bayesian workflow.

Our reliance on interpreting the SBC diagnostic through visualization, however, can be a limitation in practice, especially when dealing with models featuring a large number of parameters. One immediate direction for future work is to develop reliable numerical summaries that quantify deviations from uniformity of each SBC histogram and provide automated diagnostics that can flag certain parameters for closer inspection.

Global summaries, such as a χ^2 goodness-of-fit test of the SBC histogram with respect to a uniform response, are natural options, but we have found that they do not perform particularly well for this problem. The reason for this is that the deviation from uniformity tends to occur in only a few systematic ways, as discussed in Section 4.2, whereas these tests consider only global behavior and hence do not exploit these known failure modes. A potential alternative is to report a number of summaries that are designed to be sensitive to the specific types of deviation from uniformity we might expect to see.

Another future direction is deriving the expected behavior of the SBC histograms in the presence of autocorrelation and dropping the thinning requirement of SBC. This could even be done empirically, using the output of chains with known autocorrelations to calibrate the deviations in the rank histograms. These calibrated deviations could be used to define a sense of effective sample size for *any* algorithm capable of generating samples, not just Markov chain Monte Carlo.

Finally, the SBC histograms are only able to assess the calibration of one-dimensional posterior summaries. This is a limitation, especially in situations where the quantities of interest are naturally multivariate. An interesting extension of this methodology would be