`posteriordb`: Testing, Benchmarking and Developing Bayesian Inference Algorithms

Måns Magnusson
Uppsala University

Jakob Torgander
Uppsala University

Paul-Christian Bürkner
TU Dortmund University

Lu Zhang
University of Southern California

Bob Carpenter
Flatiron Institute

Aki Vehtari
Aalto University

Abstract

The general applicability and robustness of posterior inference algorithms is critical to widely used probabilistic programming languages such as Stan, PyMC, Pyro, and Turing.jl. When designing a new inference algorithm, whether it involves Monte Carlo sampling or variational approximation, the fundamental problem is evaluating its accuracy and efficiency across a range of representative target posteriors. To solve this problem, we propose `posteriordb`, [1] a database of models and data sets defining target densities along with reference Monte Carlo draws. We further provide a guide to the best practices in using `posteriordb` for algorithm evaluation and comparison. To provide a wide range of realistic posteriors, `posteriordb` currently comprises 120 representative models with data, and has been instrumental in developing several inference algorithms.

1 INTRODUCTION

The `posteriordb` repository is developed to address the problem of evaluating Bayesian posterior inference algorithms.

Probablistic Programming Languages (PPLs) are (often embedded) domain-specific programming languages for probabilistic modelling. PPLs have attracted hundreds of thousands of users over the past three decades. These frameworks enable users to flexibly specify models with unknown parameters and provide posterior inference conditioned on data (e.g., parameter estimation, event probability estimation, predictive inference). Based on academic paper citations, PPLs are used in almost every corner of applied statistics and machine learning, including the physical, biological and social sciences, medicine, engineering, education, finance, and entertainment. The most widely used PPLs today (according to Štrumbelj et al., 2024) are Stan (Carpenter et al., 2017), Tensorflow Probability (Dillon et al., 2017), PyMC (Salvatier et al., 2016), Pyro (Bingham et al., 2019), JAGS (Plummer, 2003), and Turing.jl (Tarek et al., 2020).

PPLs support both the development and programming of statistical models and also provide inference algorithms. For most of the PPLs listed above, this is carried out in a "black box" fashion that relies only on the model's log density and gradients, not on the specific structure of the model. Given a (not necessarily normalized) joint probability model $p(y, \theta)$ for the unknown parameters θ and observed data y, the main interest is to estimate expectations based on θ such as parameter estimates, event probabilities, or predictions. Starting out, we specify a joint model, $p(\theta, y)$ using PPL syntax. Given data y, Bayes' theorem says the posterior of interest is proportional to the joint density, which can be unpacked into the likelihood and the prior, $p(\theta \mid y) \propto p(\theta, y) = p(y \mid \theta) \cdot p(\theta)$. Using the posterior density, we can calculate the posterior predictive distribution $p(\tilde{y} \mid y)$ for new data $\tilde{y}$, estimate event probabilities $\Pr[\theta \in E]$, and estimate parameters as $\mathbb{E}[\theta \mid y]$ (Gelman et al., 2013).

In most settings, computing $p(\theta \mid y)$ is analytically intractable. PPLs instead use approximate inference algorithms, some of which are asymptotically exact. Recently, interest has been focused on "black-box"

[1] https://github.com/stan-dev/posteriordb/tree/v1.0

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

inference algorithms applicable across diverse models. Several such inference algorithms have been proposed, such as Hamiltonian Monte Carlo (HMC, Neal, 2011; Hoffman et al., 2014), variational inference approaches (VI, Jordan et al., 1999; Ranganath et al., 2014; Blei et al., 2017), and Laplace approximations (LA, Tierney and Kadane, 1986; Rue et al., 2009) along with various adaptations and improvements of these approaches (e.g. see Bales et al., 2019; Dhaka et al., 2021; Wu and Goodman, 2022; Modi et al., 2024; Wang et al., 2024). Inference algorithms have different properties. HMC, as other Markov chain Monte Carlo (MCMC) methods, will, in many settings, converge to the posterior in total variation distance (Tierney, 1994). MCMC algorithms are costly due to the many sequential iterations needed to reliable approximate the posterior. On the other hand, VI and LA can be less computationally costly but often introduce a bias in estimating the posterior expectations (Wang and Blei, 2018), introducing a tradeoff between accuracy and computational cost.

posteriordb focus on the evaluation of posterior inference algorithms. Typically, new or improved inference algorithms are evaluated on a small number of posteriors. When developing and maintaining PPLs and inference algorithms, we want to *test* that they work as intended for a range of posteriors. When developing new algorithms, we also want to *assess performance* to gain insights on which posteriors the algorithms work well and where they fail. Finally, we want to *benchmark* proposed algorithms to assess how they compare to existing approaches.

We introduce `posteriordb`, a database to aid in algorithm and PPL development, such as testing, performance assessment and benchmarking. To facilitate this, `posteriordb` contains a collection of hundreds of posteriors, models, datasets, and reference posteriors in a simple repository structure (see Figures 1 and 2). The database also includes references to papers, details about the posteriors, and metadata on models and data to simplify performance analysis. `posteriordb` is a fully open posterior database, and we encourage contributors to share their posteriors and models with the repository, especially more complicated posteriors. `posteriordb` has already been used in multiple studies on posterior approximations (e.g., see Dhaka et al., 2021; Welandawe et al., 2024; Baudart et al., 2021; Liang et al., 2022).

Previous work on collecting models and datasets has been focused on particular subclasses of models. Examples of such collections are causal structure graph models (Rios et al., 2021) and Bayesian neural networks (Vadera et al., 2022). In addition, most of the popular PPLs currently provide example models for comparison

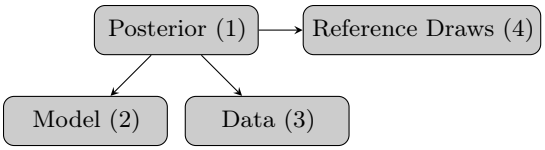

Figure 1: *The conceptual* `posteriordb`.

```
posterior_database
* bibliography
* posteriors
* models
  + stan
  + pymc3
  + info
* data
  + data-raw
  + data
  + info
* reference_posteriors
  + draws
    - draws
    - info
* summary_statistics
  + mean_value
    - mean_value
    - info
  + mean_value_squared
    ...
```

Figure 2: *Directory structure of* `posteriordb`.

and evaluation purposes, such as Stan (Stan Development Team, 2021), BUGS and JAGS (Lunn et al., 2000; Plummer, 2003), PyMC(3) (Salvatier et al., 2016), (Num)Pyro (Bingham et al., 2019; Phan et al., 2019), Turing.jl (Tarek et al., 2020), ADMB/TMB (Monnahan and Kristensen, 2018), and NIMBLE (de Valpine et al., 2017), to name a few, not to mention black-box MCMC packages that are not embedded in PPLs (e.g., emcee (Foreman-Mackey et al., 2013) and Blackjax (Cabezas et al., 2024)). Some small examples of sets of posteriors for benchmarking are `Inference Gym` (Sountsov et al., 2020) and `PPLbench` (PPL bench Developers, 2022; Kulkarni et al., 2020). `PPLbench` contains five models. Of these models, one is already in posteriordb (logistic regression), and two are similar to models already included (linear robust regression and the topic model). `Inference Gym` includes 19 posteriors. Some, such as eight schools, are identical to models in `posteriordb`. Others are similar, such as the radon models, item-response models, and logistic regression models.

Section 2 introduces the main use cases of posterior repositories. Section 3 introduces `posteriordb` and how it can be used. Section 4 describes an example of using `posteriordb` in evaluating the Pathfinder algorithm (Zhang et al., 2022), and Section 5 concludes.

2 USE CASES

The primary goal in Bayesian inference is estimating posterior expectations. Let $\hat{p}(\theta \mid y)$ be any approximation of the posterior $p(\theta \mid y)$. For example, with variational inference, $\hat{p}$ will be a member of the variational family, with Laplace approximation, $\hat{p}$ will be multivariate normal, and with sampling, $\hat{p}$ will be the discrete empirical distribution. Further, we assume

that it is possible to generate a set $\{\theta^{(s)}\}_{s=1}^{S}$ of S draws $\theta^{(s)} \sim \hat{p}(\theta \mid y)$ for $s \in \{1, \ldots, S\}$ with which to compute expectations and quantiles for empirical evaluation. While many different expectations might be of interest, the focus is commonly on means, variances, and tail quantiles of parameters and predictive variables defined as transforms of parameters and data (e.g., posterior predictions and event probability forecasts).

Assessing the performance of an inference algorithm is non-trivial. We can evaluate inference algorithms in three ways, all of which may be measured using `posteriordb` targets.

Accuracy How well does the algorithm approximate the target density (e.g., KL-divergence, squared error, Wasserstein distance, etc.)?

Efficiency What is the algorithm's computational cost (in time, gradient evaluations, memory, power consumption, etc.)?

Generality Which posteriors can the algorithm approximate and with what accuracy and efficiency?

2.1 Testing Algorithms and Their Implementations

Testing posterior inference algorithms poses more problems than standard software testing (Dutta et al., 2018) and shares similarities with functional testing (Kaner et al., 1999). When testing posterior inference algorithms, especially asymptotically unbiased algorithms, such as MCMC and HMC, the focus is usually on testing the posterior expectations. Let $\epsilon^2_{\hat{p},g} = \left(\mathbb{E}_p(g(\theta) \mid y) - \mathbb{E}_{\hat{p}}(g(\theta) \mid y)\right)^2$ be the squared approximation error for a given expectation. Then, the marginal means and variances of the posterior have the benefit that, if they are finite, the Markov chain central limit theorem can be used to assess the inference algorithm (Jones, 2004). If the algorithm works as expected, the approximation error $\epsilon_{\hat{p},g}$ will decrease over iterations at the rate $\mathcal{O}(1/\sqrt{n})$. Hence, we can use a high-quality reference posterior approximation for testing purposes; see Section 3.3 for details.

A *testably correct algorithm* generates draws whose marginal distribution follows the target density and thus can be used to evaluate inference algorithms. Independent draws, e.g. for close-form posteriors, are testably correct, as are MCMC methods run for finite amounts of time given some verifiable assumptions, such as geometric ergodicity (Roberts and Rosenthal, 1997). However, with challenging posterior density geometry, computational limitations can result in low accuracy for finite runs of MCMC. For example, random-walk Metropolis, Gibbs, and HMC all fail to sample the funnel density (Neal, 2003) in finite time, despite asymptotic guarantees, because of the poor and varying condition in the mouth and neck (Papaspiliopoulos et al., 2007; Modi et al., 2023). Nevertheless, we have two ways out of this dilemma. First, we can reparameterize, which allows us to take independent draws for the funnel example. Second, we can assess a poorly mixing or even asymptotically biased algorithm in terms of how well it estimates expectations in finite time.

In terms of evaluation reliability, the best we can do is analytical expectations, which exist in many cases. The next best thing to do is to take independent samples, the standard error for which is known. The last resort is to take MCMC draws and attempt to verify the results are correct (e.g., with simulation-based calibration (Talts et al., 2018; Modrák et al., 2023)) and then thin them until roughly independent.

Even though posterior expectations are the main statistic of interest, various discrepancies among distributions can be used for a more holistic assessment of the properties of the posterior, such as Wasserstein distance (Villani et al., 2009; Craig, 2016), maximum mean discrepancy (Gretton et al., 2012), or the Pareto-$\hat{k}$ diagnostic (Vehtari et al., 2024). To assess accuracy, we recommend the following:

1. RMSE of posterior moments of interest compared to a reference posterior with analytic moments or high accuracy estimates based on a trusted algorithm,
2. Wasserstein distance between the approximate and true posteriors,
3. Maximum Mean Discrepancy (MMD) between the approximate and the true posteriors, and
4. Pareto-$\hat{k}$ diagnostic for the density ratio indicating whether importance sampling can be used to adjust $\hat{p}(\theta \mid y)$ to better approximate $p(\theta \mid y)$ (see Vehtari et al., 2024, for details).

Some inference algorithms, such as variational inference or Laplace approximations, are biased in most applications (i.e., they have non-zero expected error); see, e.g., (Margossian and Saul, 2023). They can also blow up the problem's dimensionality by directly modeling covariance. Hence, accuracy becomes more important to assess how well the true posterior is approximated for these algorithms. Again, we can judge the accuracy using posterior expectations or more holistic approaches.

When testing a posterior inference algorithm for correctness and accuracy, a large set of posteriors that are easy to run simplifies the task. With posteriors of different shapes, sizes, and geometries, and hence difficulty, `posteriordb` allows developers to get a handle on an algorithm's performance in a wide range of realistic settings.

We also recommend evaluating the estimates of param-

eters squared is good practice, as they are required for estimating variance (i.e., $\text{var}[\theta] = \mathbb{E}[\theta^2] - \mathbb{E}^2[\theta]$). With algorithms like HMC that can produce anti-correlated draws, it is possible to estimate parameter expectations well and estimate squared parameters poorly.

2.2 Development of New Algorithms

The second use case of a repository of posterior distributions is the development of new posterior approximation algorithms. When developing new inference methods, some algorithms may work for certain posteriors and fail for others. For example, HMC has difficulties with funnels, and Laplace approximation methods work best for approximately multivariate normal posteriors. We want to find out for which type of posteriors a new algorithm works well and when it fails. Hence, many different posteriors can be used both to find unknown failure cases and demonstrate expected difficulties.

When developing inference algorithms, an important tradeoff is accuracy vs. cost. Assessing the computational performance of posterior approximation algorithms can be implementation *independent* or *dependent*. An implementation-dependent quantity is wall time or energy consumed; implementation-independent quantities are floating-point operations, log density evaluations, or gradient evaluations. Typically, computation is dominated by log density and/or gradient calculations; with automatic differentiation, the log density and gradient are computed simultaneously (Griewank and Walther, 2008).

We can compare the accuracy after a fixed amount of computation when developing algorithms, whether they are biased or asymptotically exact. Examples of implementation-dependent measures are log density evaluation per second (LDE/s), gradient evaluations per second (GE/s), and the effective sample size per second (ESS/s). For algorithms producing draws, ESS/s can be estimated from the posterior standard deviation and standard error over several runs as $\text{ESS} = (\text{sd}/\text{se})^2$. This is the central implementation-dependent performance metric since it measures the approximation precision achievable within a given practical time budget (Bürkner et al., 2023).

2.3 Benchmarking of Existing Algorithms

The computation-accuracy tradeoff shows the importance of comparing and benchmarking algorithms. We want to make informed decisions on which algorithms to implement and use as part of methods development. This also applies to minor but important improvements of existing algorithms, such as tweaking adaptation or accelerating computation.

Development and benchmarking require challenging models and posteriors for which we don't necessarily have efficient algorithms yet. Benchmarking a large number of posteriors is crucial to appropriately evaluate the breadth of posteriors that can be approximated with high accuracy and assess the associated computational costs. Even if an asymptotically unbiased algorithm and implementation are correct and work well for certain posterior geometries, they might fail spectacularly for others. For example, dynamic integration-time HMC, e.g. NUTS HMC (Hoffman et al., 2014), with fixed step size integrator fails to reach the asymptotic regime for many funnel-shaped posteriors in feasible time (Betancourt and Girolami, 2015). On the other hand, Laplace and variational approximations combined with importance sampling can reach the asymptotic regime for some of these same funnel-shaped posteriors if they are sufficiently low dimensional (Yao et al., 2018).

A large set of posterior distributions, such as funnel-shaped posteriors, multimodal posteriors, discrete and discrete-continuous-mixed posteriors, high-dimensional posteriors, finite and infinite posteriors (Dirichlet Processes), large data posteriors, and simple, analytically tractable posteriors, enables the assessment of algorithm generality. A posterior approximation algorithm can be useful if it works well for some models and can be diagnosed when it doesn't work. Hence, we want to know the types of errors and problems to assess the generality of benchmarked algorithms.

2.4 Development and Maintenance

The process of testing algorithms includes multiple steps. Employing the same rigorous approaches used for previously well-tested algorithms is common practice when developing a new algorithm. This ensures that the new implementation meets the expected standards of functionality and reliability. Similarly, when maintaining existing software, testing serves the dual purpose of verifying that changes haven't compromised the integrity of the inference algorithm and that the algorithm's performance remains unaffected. Regression testing, as it's known in computer science, compares algorithm outputs over the development lifecycle to catch any behaviour or performance "regressions."

3 `posteriordb`

`posteriordb` is a comprehensive repository containing posteriors, models, data, and reference posteriors. The primary objective is to leverage this set of posteriors to test, assess, benchmark, develop and maintain PPLs and posterior approximation algorithms. The database contains both more difficult/complex posteri-

ors, such as Covid-19 epidemic models (Flaxman et al., 2020), Bayesian neural networks (Lampinen and Vehtari, 2001), and simpler, standard posteriors, such as the eight schools example (Rubin, 1981; Gelman et al., 2013). All posteriors, data and models are stored in the same format, simplifying the estimation of many posteriors for generality and benchmarking purposes.

3.1 The `posteriordb` Components

The `posteriordb` contains four main types of objects (see Figures 1 and 2 for an overview).

The *posterior* (1) object summarizes all information about a specific posterior in the collection. A posterior object points to a (not necessarily normalized) joint model $p(y, \theta)$, data y, and a reference posterior (if any). The purpose of separating models from data is that some models use the same data, which is relevant for model comparison diagnostics, and some models can be used for multiple datasets, enabling comparison of the performance of the same model for different data. Finally, the posterior object points to reference posterior draws if such exist.

The *model* (2) object in `posteriordb` stores an (unnormalized) joint model, $p(y, \theta)$, in the form of PPL code and JSON information files. While most models are currently written in Stan or PyMC, the structure allows us to easily include code from other PPLs.

The *data* (3) objects, y, are stored as compressed JSON files for ease of use. Each data file also contain an information JSON file. The `data-raw` folder contains code and information on processing the data in case processing has been done.

The *reference posterior* (4) object (in the JSON sense of "object") represents the true posterior distribution, usually in the form of posterior draws, if it is possible to compute such a representation. A set of draws must be of very high quality to serve as a reference posterior, as detailed further in Section 3.3. Depending on the size and the possibility of computing the posterior distribution, the reference posterior draws themselves and/or the corresponding posterior expectations are stored in the reference posterior object as compressed JSON files. The benefit of a true, or well approximated, reference posterior is that we can assess, to a given tolerance and a specific computational budget, if the output of an algorithm is correct concerning a true underlying posterior distribution. An information JSON file includes exact details of how the posterior draws are computed.

As an example, the `eight_schools-eight_schools_centered` posterior points to the data set `eight_schools` and the centered parameterization of the eight school model `eight_schools_centered` (Betancourt and Girolami, 2015). Further, the posterior object includes the posterior dimension and points to the reference posterior `eight_schools-eight_schools_noncentered`. The centered parametrization is well-known to have difficulties due to the funnel geometry of the posterior. Hence, the non-centered parameterization is used as a reference posterior for the centered model.

The choice of models is motivated by the goal of including a large and diverse set of posteriors. We also focus on including posteriors where data and models have been published openly. References enable users of `posteriordb` to study specific models and data in more detail. A table with all posteriors can be found in the supplementary material.

`posteriordb` can be accessed in two ways. The first is to directly access the content of `posteriordb` from the repository `https://github.com/stan-dev/posteriordb`. The folder `posteriord_database` in the repository contains the data in a folder structure shown in Figure 2. Data and reference posterior draws are compressed as zip archives. It is also possible to interact with the `posteriordb` through the R package (`https://github.com/stan-dev/posteriordb-r`) and the Python library (`https://github.com/stan-dev/posteriordb-python`) to simplify quick access. All posteriors, data, models, reference posteriors, and software are version-controlled using semantic versioning.

3.2 Additional Metadata

We include potential relevant information in all four objects (e.g., the number of parameters for posterior objects and keywords to group the posteriors). We also include keywords for posteriors to enhance the capacity to assess diverse performance aspects, enabling a comprehensive understanding of algorithmic behaviors and for benchmarking purposes, for example, developing new algorithms. This also aids in diagnosing issues with new algorithms or in benchmarking settings. Where available, posteriors, models, and data also contain bibliographical entries to provide background.

3.3 Reference Posterior Distributions

A key component of `posteriordb` is the *reference posterior* (RP) object. The RP object consists of (approximately) independent and identically distributed Monte Carlo draws from the corresponding posterior *model* object and serves as a representation of the *true* underlying posterior distribution. Depending on the form of the posterior $p(\theta \mid y)$, the draws of the RP object are created by independent sampling or MCMC.

For some simple models, it is possible to sample independently from an analytic posterior. We expand the set of reference posteriors by including models for which we have high confidence that we can obtain draws that are approximately independent draws from the posterior. We use MCMC, specifically Stan's dynamic HMC variant, to obtain reference posteriors for well-behaved models from which we cannot independently sample. First, we compute a set of draws, $\{\theta^{(s)}\}$ and include them in `posteriordb`. Second, we compute the posterior parameter expectations (either analytically or with the draws). The posterior means support direct error evaluation, and the draws allow more holistic evaluation, for example, using Wasserstein distances. The inclusion of draws further aids in identifying areas and specific types of posteriors that exhibit suboptimal performance and regions of difficulty.

Even with the 10 000 roughly independent draws supplied by `posteriordb`, there will be an upper bound on the accuracy. For instance, estimating the mean of a standard normal distribution will have a standard error of 0.01. This imposes an upper bound on the accuracy of a system being evaluated before the error in the reference dominates the estimated error.

We define a reference posterior, or expectations thereof, as 10 000 draws from the true posterior distribution. In practice, this is only possible in a limited number of analytical settings. In the case of MCMC, the chains should be thinned so that the draws are approximately independent to make further comparisons easier. However, we also use MCMC to generate draws from the true posterior distribution. To consider draws generated using MCMC as a reference posterior, we require

1. 10 000 draws per parameter,
2. approximately independent draws, that is, all parameters have a mean autocorrelation at lag 1 over the chains that is less than 0.05 in absolute value,
3. an $\hat{R}$ below 1.01 for all parameters (see Vehtari et al., 2021),
4. if HMC is used, all expected fraction of missing information (E-FMI) is below 0.2, i.e. avoid situations with poor exploration of the energy level (see Betancourt, 2017), and
5. there are no divergent transitions (see Betancourt, 2016).

To get reference posterior draws, we use Stan's implementation of NUTS. Other approaches, such as model-specific algorithms, can also be used in special circumstances if it is clear that it is necessary (e.g., for discrete parameter models).

Our repository also includes interesting and challenging posteriors for which we cannot compute a reference draws (e.g., for combinatorially multimodal posteriors such as latent Dirichlet allocation (Blei et al., 2003) or Bayesian neural networks).

3.4 The Current Scope of `posteriordb`

`posteriordb` currently contains 147 posteriors, 120 models, 91 datasets and 46 reference posterior draws. Of these, roughly a third are easy cases, whereas the remaining two-thirds are more challenging. Table 1 contains examples of posteriors that can be sampled using NUTS, but where default settings result in a large number of divergent transitions or excessive numbers of leapfrog steps, indicating more complex posterior geometries. Smaller step sizes and more leapfrog iterations can reduce the divergence rates. A complete summary of the posteriors featured in `posteriordb` can be found in Section 2 of the supplementary materials.

We can see that some posteriors, such as `soil_carbon-soil_incubation`, produce divergent transitions, indicating large changes in curvature. In contrast, `synthetic_grid_RBF_kernels-kronecker_gp` needs many leapfrog steps to explore the posterior.

Incorporating complex posteriors without reference draws also includes scenarios where, for example, Stan's implementation of NUTS is too inefficient. Including these posteriors fosters the development of new inference algorithms and serves as a valuable resource describing current algorithm limitations.

4 CASE STUDY: PATHFINDER

The content of `posteriordb` has already been used in multiple settings to evaluate and develop new algorithms (e.g. see Dhaka et al., 2020; Lange et al., 2022; Wang et al., 2024). Here, we present a refined example, selectively condensed to show how `posteriordb` facilitates algorithm comparison and evaluation, with a focus on the assessment of the Pathfinder variational inference algorithm (Zhang et al., 2022). In addition, we introduce a novel test for log-concavity that further illustrates how `posteriordb` enhances the understanding of algorithm performance and provides key insights into the strengths and weaknesses.

4.1 Pathfinder Evaluation

Pathfinder is compared to ADVI (Kucukelbir et al., 2017) and to short MCMC runs, using Stan's implementation of ADVI and Stan's dynamic HMC (Stan Development Team, 2021). The latter procedure corresponds to the first stage of MCMC warmup or a variational inference algorithm in its own right, following (Hoffman and Ma, 2020). The approximation

Table 1: *Examples of posteriors with difficult geometries for HMC. Ellipses (...) are used used to shorten long names.*

Posterior	Steps	Diverg.	Posterior	Steps	Diverg.	Posterior	Steps	Diverg.
diamonds-diamonds	970	0.00	...covid19...v2	300	0.20	...covid19...v3	260	0.22
hmm_gaussian...	600	0.25	mcycle_gp...	660	0.13	mcycle_splines...	1020	0.00
mnist_100-nn...	1020	0.00	pilots-pilots...	290	0.24	...prophet	1000	0.00
...RBF...kronecker..gp	1020	0.00	soil_carbon...	110	0.18	uk..state_space...	790	0.05

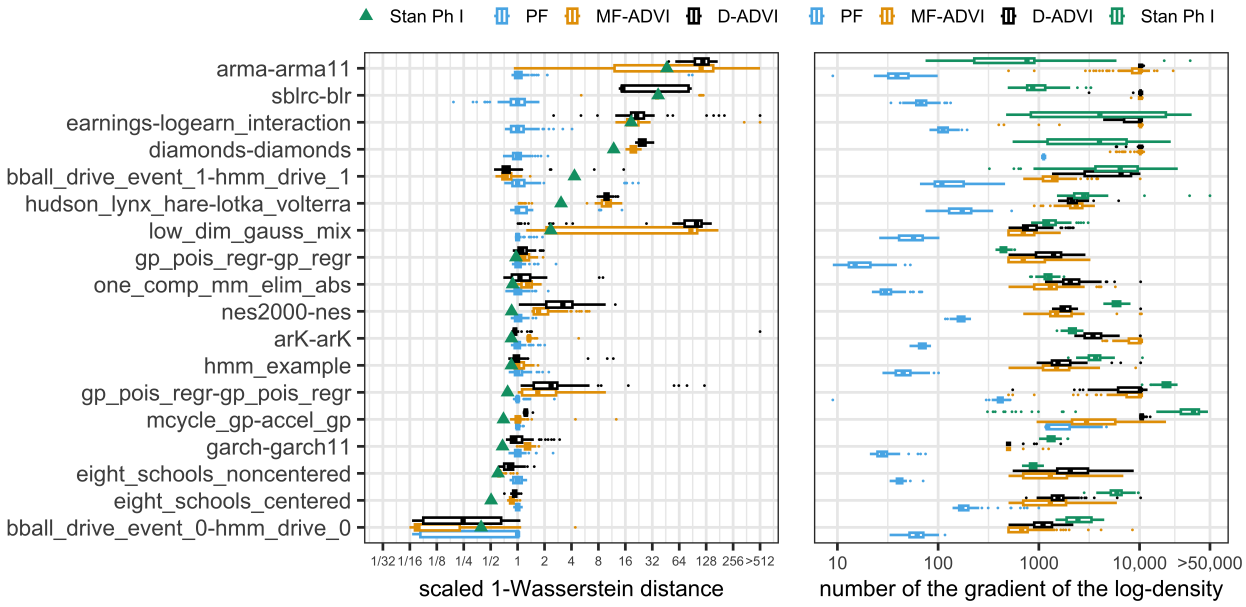

Figure 3: *Left panel: Box plots of 1-Wasserstein (1-W) distances between the reference posterior samples and approximate draws from Pathfinder (PF) and ADVI for 18 posteriors in* `posteriordb`*. Each box plot displays the 1-W distances of 100 independent runs of PF, mean-field ADVI (MF-ADVI), and dense ADVI (D-ADVI). The 1-W distance for the Stan phase I sampler is calculated using 100 approximate draws from the last iteration of 100 runs of Stan's phase I warmup (adaptive HMC). Distances for each model are scaled by the median of the 1-W distances for PF. Right panel: Box plots of the number of gradient evaluations required by all algorithms. Each box plot summarizes the cost for 100 independent runs.*

performance is evaluated through the discrete form of the 1-Wasserstein (1-W) distance and the counts of log density and gradient evaluations. The Pathfinder algorithm is tested on 18 selected posteriors from `posteriordb`. This set of posteriors is chosen to have varying posterior geometries, such as low-dimensional and high-dimensional, low correlation and high correlation, close to normal and highly non-normal, unimodal and multimodal, and log-concave and non-log-concave. Second, the set includes models and posteriors that are common in practice, such as hierarchical models with funnels, normal and generalized linear models (GLM), Gaussian processes (GP), mixture models, hidden Markov models (HMM), time series (AR) models, and heteroskedastic models. We recommend this set as the minimal starting point for any algorithm comparisons.

For each posterior, Pathfinder are run 100 times and is compared to the result of 100 runs of 1) Stan phase I adaptation: Stan's warmup adaptation using dynamic HMC sampler, 2) dense ADVI: ADVI with a dense covariance matrix, and 3) mean-field ADVI: ADVI with a diagonal covariance matrix. The right panel of Figure 3 compares computational efficiency using the number of GE. The number of LDE and GE assessed the implementation-independent computational costs. The experiment shows that Pathfinder required the lowest computational cost among the evaluated algorithms. In most cases, the cost of Stan phase I sampler is lower than that of mean-field ADVI, and dense ADVI is the most computationally expensive. Significant differences in computational costs is evident across the models and algorithms. The left panel illustrates a comparison of Pathfinder, ADVI, and Stan's phase I sampler through 1-W distances. Overall, Pathfinder produce lower 1-W distances than ADVI variants and showed stable 1-W distances across challenging posteriors compared to Stan phase I adaptation. Notably, for the hidden Markov model `bball_drive_event_0-hmm_drive_0`, mean-field ADVI achieve a median 1-W

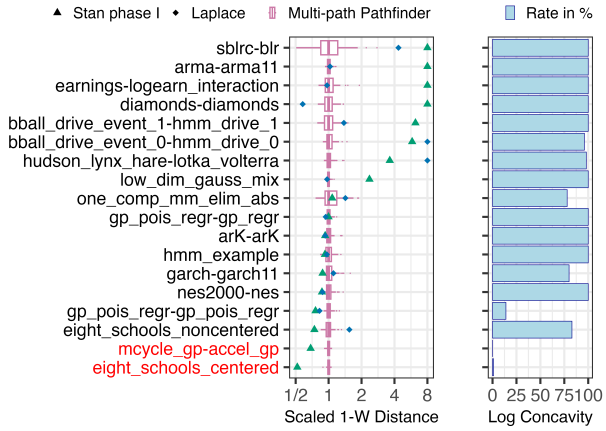

Figure 4: *Box plots of scaled 1-W distances between reference posterior samples and approximate draws from multi-path Pathfinder for 18 posteriors in* `posteriordb`*. Each box plot summarizes 100 independent runs of multi-path Pathfinder. Distances are scaled by the median 1-W distance for Pathfinder. The Laplace approximation and Stan phase I are included for comparison, except for the two models in red, which do not have a Laplace approximation. Models are ordered by the ratio of the 1-W distance for Stan phase I sampler to the median distance for Pathfinder. Right panel: Success rate of the log-concavity test (in %).*

distance less than one-tenth of Pathfinder's. This model has multiple meaningful posterior modes, and the noise inherent in the stochastic gradient descent approach used by ADVI allows it to escape minor modes that can trap the L-BFGS optimizer used by Pathfinder. For this model, multi-path Pathfinder, which combines multiple runs of (single-path) Pathfinder with different initials, will perform better than (single-path) Pathfinder.

4.2 Log-concavity Diagnostics

To further investigate the variability in Pathfinder's performance, we design an additional diagnostic to assess the log-concavity of the target posterior using `posteriordb`. We randomly select 100 reference posterior draws for each model, computed the Cholesky decomposition of the numerical Hessian of the negative log density at each point, and recorded the success rate. Figure 4 compares the performance of multi-path Pathfinder (with 20 initials), Stan phase I adaptation, the Laplace approximation and the results of the log-concavity test. The plot shows that Pathfinder performs worse on models with a lower success rate in the log-concavity test. This insight leads us to identify the underlying cause: the L-BFGS algorithm used by Pathfinder enforces positive definiteness in the Hessian approximation, resulting in a Gaussian approximation that underperforms for posteriors that lack log-concavity.

5 DISCUSSION AND CONCLUSIONS

We present `posteriordb`, a collection of models, data, posteriors, and reference draws to evaluate posterior inference algorithms. During the construction and use of `posteriordb`, we have gained valuable insights about curating a database of posteriors for benchmarking samplers. First, we added many relatively simple posteriors that can be estimated easily using NUTS. In hindsight, more difficult posteriors are more relevant, especially for developing algorithms. Second, labels and information on the posteriors are more important than we first thought. Some posteriors are too complex (multimodal or with weak identifiability), leading to very slow computations, while others are too easy. When we use `posteriordb` for benchmarking and algorithm development, we realize we need to pick appropriate posteriors for the experimental goal (e.g., posteriors not log-concave may be excluded or be the target of interest). Third, an important conclusion is to separate the model, data, and posteriors to facilitate a broader use and reuse of the components.

5.1 Limitations and Future Work

The majority of `posteriordb` models are coded in Stan, with some PyMC contributions. For the Stan models, the package `bridgestan` (Roualdes et al., 2023) provides access to these models' log gradients and densities in many different languages, including Python, R, Julia, and Rust. In addition, some posteriors are so challenging that a reference posterior is lacking, making these posteriors difficult to use for benchmarking at the moment.

We intend to add a wider range of posteriors, and specifically more challenging ones. Second, we want to incorporate posterior model code from additional PPLs. Third, the database should be augmented with predictive distributions or functionality to compute predictive distributions, which would simplify comparisons and diagnostics based on predictive distributions, such as uncertainty calibration. Fourth, we realize the need to identify geometries empirically from draws, for example, funnel-shaped posteriors, posteriors with non-positive-definite Hessians within the set of draws, or posteriors with multiple modes.

Acknowledgments

We acknowledge the support of the Research Council of Finland Flagship programme, Finnish Center for Artificial Intelligence, and Research Council of Finland projects 313122 and 340721. In addition, the research was funded by the Swedish Research Council through

grant 2022-03381.

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

Ben Bales, Arya Pourzanjani, Aki Vehtari, and Linda Petzold. Selecting the metric in Hamiltonian Monte Carlo. *arXiv preprint arXiv:1905.11916*, 2019.

Guillaume Baudart, Javier Burroni, Martin Hirzel, Louis Mandel, and Avraham Shinnar. Compiling Stan to generative probabilistic languages and extension to deep probabilistic programming. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pages 497–510, 2021.

Michael Betancourt. Diagnosing suboptimal cotangent disintegrations in Hamiltonian Monte Carlo. *arXiv preprint arXiv:1604.00695*, 2016.

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Michael Betancourt and Mark Girolami. Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79(30): 2–4, 2015.

Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(28):1–6, 2019.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112 (518):859–877, 2017.

Paul-Christian Bürkner, Maximilian Scholz, and Stefan T. Radev. Some models are useful, but how do we know which ones? Towards a unified Bayesian model taxonomy. *Statistics Surveys*, 17:216–310, 2023. doi: 10.1214/23-SS145.

Alberto Cabezas, Adrien Corenflos, Junpeng Lao, and Rémi Louf. Blackjax: Composable Bayesian inference in JAX. *arXiv*, 2402.10797, 2024.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76, 2017.

Katy Craig. The exponential formula for the Wasserstein metric. *ESAIM: Control, Optimisation and Calculus of Variations*, 22(1):169–187, 2016.

Perry de Valpine, Daniel Turek, Christopher J Paciorek, Clifford Anderson-Bergman, Duncan Temple Lang, and Rastislav Bodik. Programming with models: writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics*, 26(2):403–413, 2017.

Akash Kumar Dhaka, Alejandro Catalina, Michael R Andersen, Måns Magnusson, Jonathan Huggins, and Aki Vehtari. Robust, accurate stochastic optimization for variational inference. *Advances in Neural Information Processing Systems*, 33:10961–10973, 2020.

Akash Kumar Dhaka, Alejandro Catalina, Manushi Welandawe, Michael R Andersen, Jonathan Huggins, and Aki Vehtari. Challenges and opportunities in high dimensional variational inference. *Advances in Neural Information Processing Systems*, 34:7787–7798, 2021.

Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv*, 1711.10604, 2017.

Saikat Dutta, Owolabi Legunsen, Zixin Huang, and Sasa Misailovic. Testing probabilistic programming systems. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 574–586, 2018.

Seth Flaxman, Swapnil Mishra, Axel Gandy, H Unwin, H Coupland, T Mellan, H Zhu, T Berah, J Eaton, P Perez Guzman, et al. Report 13: Estimating the number of infections and the impact of non-pharmaceutical interventions on COVID-19 in 11 European countries. Technical report, Imperial College London, 2020.

Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. emcee: the MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, 2013.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets, 2021. URL https://arxiv.org/abs/1803.09010.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

Matthew Hoffman and Yian Ma. Black-box variational inference as a parametric approximation to Langevin dynamics. *Proceedings of Machine Learning Research*, 119:4324–4341, 2020.

Matthew D Hoffman, Andrew Gelman, et al. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15 (1):1593–1623, 2014.

Galin L Jones. On the Markov chain central limit theorem. *Probability Surveys*, 1:299–320, 2004.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

Cem Kaner, Jack Falk, and Hung Q Nguyen. *Testing computer software*. John Wiley & Sons, 1999.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(1):430–474, 2017.

Sourabh Kulkarni, Kinjal Divesh Shah, Nimar Arora, Xiaoyan Wang, Yucen Lily Li, Nazanin Khosravani Tehrani, Michael Tingley, David Noursi, Narjes Torabi, Sepehr Akhavan Masouleh, and Eric Meijer. PPL bench: Evaluation framework for probabilistic programming languages. *arXiv*, 2010.08886, 2020.

Jouko Lampinen and Aki Vehtari. Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274, 2001.

Richard D Lange, Ari S Benjamin, Ralf M Haefner, and Xaq Pitkow. Interpolating between sampling and variational inference with infinite stochastic mixtures. In *Uncertainty in Artificial Intelligence*, pages 1063–1073. PMLR, 2022.

Feynman Liang, Michael Mahoney, and Liam Hodgkinson. Fat–tailed variational inference with anisotropic tail adaptive flows. In *International Conference on Machine Learning*, pages 13257–13270. PMLR, 2022.

David J Lunn, Andrew Thomas, Nicky Best, and David Spiegelhalter. WinBUGS—a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, 10:325–337, 2000.

Charles C Margossian and Lawrence K Saul. The shrinkage-delinkage trade-off: An analysis of factorized Gaussian approximations for variational inference. In *Uncertainty in Artificial Intelligence*, pages 1358–1367. PMLR, 2023.

Chirag Modi, Alex Barnett, and Bob Carpenter. Delayed rejection Hamiltonian Monte Carlo for sampling multiscale distributions. *Bayesian Analysis*, 1 (1):1–28, 2023.

Chirag Modi, Robert Gower, Charles Margossian, Yuling Yao, David Blei, and Lawrence Saul. Variational inference with Gaussian score matching. *Advances in Neural Information Processing Systems*, 36, 2024.

Martin Modrák, Angie H Moon, Shinyoung Kim, Paul Bürkner, Niko Huurre, Kateřina Faltejsková, Andrew Gelman, and Aki Vehtari. Simulation-based calibration checking for Bayesian computation: The choice of test quantities shapes sensitivity. *Bayesian Analysis*, 1(1):1–28, 2023.

Cole C Monnahan and Kasper Kristensen. No-U-turn sampling for fast Bayesian inference in ADMB and TMB: Introducing the adnuts and tmbstan R packages. *PloS One*, 13(5):e0197954, 2018.

Radford M Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

Radford M Neal. *MCMC using Hamiltonian dynamics*, pages 113–162. Chapman and Hall/CRC, 2011.

Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73, 2007.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in NumPyro. *arXiv*, 1912.11554, 2019.

Martyn Plummer. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, volume 124, 125.10, pages 1–10. Vienna, Austria, 2003.

PPL bench Developers. PPL bench, 2022. URL https://pplbench.org/. Available online (2024-10-04): https://pplbench.org/.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

Felix L Rios, Giusi Moffa, and Jack Kuipers. Benchpress: A scalable and versatile workflow for benchmarking structure learning algorithms. *arXiv preprint arXiv:2107.03863*, 2021.

Gareth Roberts and Jeffrey Rosenthal. Geometric Ergodicity and Hybrid Markov Chains. *Electronic Communications in Probability*, 2(none):13 – 25, 1997. doi: 10.1214/ECP.v2-981. URL https://doi.org/10.1214/ECP.v2-981.

Edward A. Roualdes, Brian Ward, Bob Carpenter, Adrian Seyboldt, and Seth D. Axen. BridgeStan: Efficient in-memory access to the methods of a Stan model. *Journal of Open Source Software*, 8(87):5236, July 2023. doi: 10.21105/joss.05236. URL https://joss.theoj.org/papers/10.21105/joss.05236.

Donald B Rubin. Estimation in parallel randomized experiments. *Journal of Educational Statistics*, 6(4): 377–401, 1981.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71(2):319–392, 2009.

John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016.

Pavel Sountsov, Alexey Radul, and contributors. Inference Gym, 2020. URL https://pypi.org/project/inference_gym. Available online (2024-10-04): https://pypi.org/project/inference_gym.

Stan Development Team. Stan Reference Manual, 2021. URL https://mc-stan.org/docs/2_26/reference-manual/index.html.

Erik Štrumbelj, Alexandre Bouchard-Côté, Jukka Corander, Andrew Gelman, Håvard Rue, Lawrence Murray, Henri Pesonen, Martyn Plummer, and Aki Vehtari. Past, present and future of software for Bayesian inference. *Statistical Science*, 39(1):46–61, 2024.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian inference algorithms with simulation-based calibration. *arXiv*, 1804.06788, 2018.

Mohamed Tarek, Kai Xu, Martin Trapp, Hong Ge, and Zoubin Ghahramani. DynamicPPL: Stan-like speed for dynamic probabilistic models. *arXiv preprint arXiv:2002.02702*, 2020.

Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

Meet Vadera, Jinyang Li, Adam Cobb, Brian Jalaian, Tarek Abdelzaher, and Benjamin Marlin. URSABench: A system for comprehensive benchmarking of Bayesian deep neural network models and inference methods. *Proceedings of Machine Learning and Systems*, 4:217–237, 2022.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2):667–718, 2021.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25 (72):1–58, 2024.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Congye Wang, Ye Chen, Heishiro Kanagawa, and Chris J Oates. Stein Pi-importance sampling. *Advances in Neural Information Processing Systems*, 36, 2024.

Yixin Wang and David M Blei. Frequentist consistency of variational Bayes. *Journal of the American Statistical Association*, 2018.

Manushi Welandawe, Michael Riis Andersen, Aki Vehtari, and Jonathan H Huggins. A framework for improving the reliability of black-box variational inference. *Journal of Machine Learning Research*, 25 (219):1–71, 2024.

Mike Wu and Noah Goodman. Foundation posteriors for approximate probabilistic inference. *Advances in Neural Information Processing Systems*, 35:5335–5347, 2022.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In *International Conference on Machine Learning*, pages 5581–5590. PMLR, 2018.

Lu Zhang, Bob Carpenter, Andrew Gelman, and Aki Vehtari. Pathfinder: Parallel quasi-Newton variational inference. *Journal of Machine Learning Research*, 23(306):1–49, 2022.

A CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]

(c) Source code, with specification of all dependencies, including external libraries. [Yes, in the github repository]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes, see Section B]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes, see Section B]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

B DATASHEET FOR DATASETS

Below is a datasheet with information on the compilation and setup of the dataset (see Gebru et al., 2021).

B.1 Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

The dataset was created to simplify testing, benchmarking and the development of Bayesian inference algorithms and diagnostics.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The authors created the dataset, and over time, contributors to the database have made additional contributions with more models.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

See Acknowledgements.

B.2 Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

The dataset contains four main components (see the main paper for details). Posteriors, data, models, and reference posteriors. The posterior object points to a joint model $p(y, \theta)$, data y, and a reference posterior (if any). The model object stores a joint model as a PPL code. The data object is stored as compressed JSON files and datasets that can be simulated or publicly open datasets. Finally, the reference posterior represents the "true" posterior distribution, usually in the form of posterior draws, if it is possible to compute such a representation.

How many instances are there in total (of each type, if appropriate)?

There are currently 147 posteriors, 120 models, 91 datasets and 46 reference posterior draws objects.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set

(e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

It is difficult to discuss what constitutes a representative sample of posterior distributions. The repository currently contains many different types of posteriors, such as funnel-shaped, classical linear and logistical regression models, multimodal posteriors, epidemiological, and population models. The target is to have as diverse a set of posteriors as possible.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

See the description above.

Is there a label or target associated with each instance? If so, please provide a description.

Models, posteriors, reference posteriors and data can have labels and keywords associated with them. These keywords can contain information on whether a posterior is included in the Stan benchmark or whether a posterior has a specific know geometry (e.g. funnel or multimodal).

Is any information missing from individual instances? If so, please provide a description explaining why this information is missing (e.g. because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

Not applicable.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

Yes. Each posterior explicitly points to a model, data and a reference object (if any).

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

There are currently labels or tags included for models that, e.g. indicate if a posterior is part of the Stan benchmark set to facilitate benchmarking.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

There are MC standard errors in our reference values and draws due to MCMC errors. In terms of redundancy, we have multiple models for the same data and multiple data for the same model, but none of it is redundant.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

The dataset is self-contained in that it doesn't rely on other sources. However, each posterior (and, to some degree, data and models) points to a URL where more information can be found or includes bibliographical references to the main paper where the posterior is used or introduced. This is to facilitate a more in-depth analysis of individual posteriors.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)? If so, please provide a description.

No. Only publicly open and simulated data are included to minimise privacy risks. We don't think any data sets have granular enough spatial identifiers or anything specific to be a problem with privacy. Also, since the data is already public, we're not introducing further privacy risks.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

No.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

Some datasets might relate to people, although `posteriordb` includes only anonymous public and open data to minimize privacy risks.

Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

A few datasets might have information that could identify subpopulations, such as age, gender, and height.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so,

please describe how.

We do only include anonymous data from public and open datasets.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.

We do only include anonymous data from public and open datasets.

B.3 Collection Process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The data has been collected from open sources and previously published studies.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

The data was collected manually by the paper's authors, students and collaborators contributing posteriors.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

Not applicable.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

The data was collected manually by the paper's authors, students, and collaborators who contributed posteriors. Some posterior were collected as a part of the Google Summer of Code (see https://summerofcode.withgoogle.com/), some posteriors were collected by students at Aalto University that were compensated according to the standards of the university for student work.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.

The data has been collected in different batches since 2019.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

No. It is not applicable since only open and public data and models are used where persons are anonymous.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

Not explicitly. Some posteriors come from social science applications.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

Data has only been collected through third parties where the data and model have been made open and published.

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

This is not applicable since no direct collection has been made.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

This is not applicable since no direct collection has been made. However, if an individual dataset is questioned for ethical reasons, we would remove that dataset.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

This is not applicable since no direct collection has been made. However, if an individual dataset becomes questioned for ethical reasons, we would remove that dataset.

Has an analysis of the potential impact of the dataset

and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

This is not applicable since no direct collection has been made.

B.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

The posteriors are manually collected and curated. For many of them, reference posteriors have also been computed.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.

No. Although data, models, etc., are version-controlled using semantic versioning.

Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.

No, most work has been done manually.

B.5 Uses

Has the dataset been used for any tasks already? If so, please provide a description.

Multiple studies have used the dataset in the development of Bayesian inference algorithms. See the main paper for details.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

This is part of the `posteriordb` objects. If a posterior has been part by a published paper a bibliography entry has been included.

What (other) tasks could the dataset be used for?

There are multiple use cases of the posteriordb (see the main paper for details). We summarize them as

1. testing inference algorithms and their implementations,
2. the development of new inference algorithms,
3. benchmarking of existing algorithms, and
4. the development and benchmarking of PPLs.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?

Not applicable.

Are there tasks for which the dataset should not be used? If so, please provide a description.

There is no obvious bad data usage due to the data itself. However, sound judgment is still needed when evaluating inference algorithms.

B.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

No.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub) Does the dataset have a digital object identifier (DOI)?

The database will be distributed at GitHub (anonymous). (The dataset will also get a DOI on the release of version 1.0.

When will the dataset be distributed?

The dataset is already available. We will release version 1.0 of `posteriordb` when the paper is published.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

The licencing will follow the general licence of Stan, that is BSD-3. More information on the licence can be found here: https://opensource.org/license/bsd-3-clause

Some models also have other licences, such as MIT. Then this information is included in the database.

Have any third parties imposed IP-based or other re-

strictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

Some datasets or models that are standard in the community, such as the MNIST dataset, have other open licences than BSD3. For these models or data, a separate licence is included in the object's meta-data information.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

No.

B.7 Maintenance

Who will be supporting/hosting/maintaining the dataset?

Since 2020, the database repository has been maintained and developed by the authors of this paper. Also, additional contributors have added additional models over time. We will continue to support and facilitate adding more posteriors over time.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

The corresponding author, Måns Magnusson, can be contacted through `mans.magnusson@statistik.uu.se`.

Is there an erratum? If so, please provide a link or other access point.

The repository is version-controlled using semantic versioning. When additional posteriors, models, and reference posteriors are added, a release is made with this information.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

The dataset will be continuously updated with new posteriors, relevant metadata (see main paper for details), and potential fixes of errors. When new content is available, new releases will be made on Git Hub following semantic versioning.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.

Not applicable.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

Yes. The whole database will be version-controlled on Git Hub to enable access to older versions.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

We welcome contributors. Details on how to contribute can be found in `doc/CONTRIBUTING.md` in the repository. The main solution to contributions is opening pull requests to the repository. Each contribution will be manually verified. The same release process stated above will be followed when contributions have been made.

C Table of posteriors in `posteriordb`

See next page (Table 2).

Table 2: *Summary of Posteriors In* `posteriordb`. *Showing number of parameter, existence of reference draws, number of data points of the largest data input component, and model description of each posterior featured in the database.*

Posterior name	# of params.	Ref. draws	Max data size	Model description
arK-arK	7	Yes	200	Autoregressive-5 model
arma-arma11	4	Yes	200	Autoregressive Moving Average
bball_drive_event_0-hmm_drive_0	8	Yes	416	Hidden Markov Model
bball_drive_event_1-hmm_drive_1	8	Yes	416	Hidden Markov Model
bones_data-bones_model	13	No	442	Latent Trait Model for Multiple Ordered Categorical Responses
butterfly-multi_occupancy	106	No	560	Multiple Species-Site Occupancy Model
diamonds-diamonds	26	Yes	125000	Multiple Highly Correlated Predictors Log-Log Model
dogs-dogs_hierarchical	2	No	750	Hierarchical Logistic Mixed Effects Model
dogs-dogs_log	2	No	750	Logarithmic Mixed Effects Model
dogs-dogs_nonhierarchical	67	No	750	Non-Hierarchical Logistic Mixed Effects Model
dogs-dogs	3	No	750	Logistic Mixed Effects Model
dugongs_data-dugongs_model	6	No	27	Dugong Age and Length
earnings-earn_height	3	Yes	1192	One Predictor Linear Model
earnings-logearn_height_male	4	Yes	1192	Multiple Predictors Log-linear Model
earnings-logearn_height	3	Yes	1192	One Predictor Log-linear Model
earnings-logearn_interaction_z	5	Yes	1192	Multiple Linearly Transformed Predictors Interacting Log-linear Model
earnings-logearn_interaction	5	Yes	1192	Multiple Predictors Interacting Log-linear Model
earnings-log10earn_height	3	Yes	1192	One Predictor Log10-linear Model
earnings-logearn_logheight_male	4	Yes	1192	Multiple Predictors Log-Log Model
ecdc0401-covid19imperial_v2	51	No	8400	Epidemic model v2 of Flaxman et al (2020)
ecdc0401-covid19imperial_v3	66	No	8400	Epidemic model v3 of Flaxman et al (2020)
ecdc0501-covid19imperial_v2	51	No	8400	Epidemic model v2 of Flaxman et al (2020)
ecdc0501-covid19imperial_v3	66	No	8400	Epidemic model v3 of Flaxman et al (2020)
eight_schools-eight_schools_centered	10	Yes	8	A centered hiearchical model for 8 schools
eight_schools-eight_schools_noncentered	10	Yes	8	A non-centered hiearchical model for 8 schools
election88-election88_full	90	No	11566	Generalized Linear Mixed Effects Model
fims_Aus_Jpn_irt-2pl_latent_reg_irt	531	No	7000	Two-parameter logistic item theory model with latent regression
garch-garch11	4	Yes	200	Generalized Autoregressive Conditional Heteroscedastic Model
GLM_Binomial_data-GLM_Binomial_model	83	No	40	Success Rate of Peregrine broods
GLM_Poisson_Data-GLM_Poisson_model	84	No	40	Poisson GLM for modeling a population of Peregrines
GLMM_data-GLMM1_model	2395	No	2072	Generalized Linear Mixed Model for Peregrine Population Size
GLMM_Poisson_data-GLMM_Poisson_model	125	No	40	Mixed Model to Predict Population Size with Random Site and Year Effects
gp_pois_regr-gp_pois_regr	13	Yes	11	Gaussian Process Poisson Regression
gp_pois_regr-gp_regr	3	Yes	11	Gaussian Process regression
hmm_example-hmm_example	6	Yes	100	Hidden Markov Model
hmm_gaussian_simulated-hmm_gaussian	13	No	500	HMM with Gaussian emission
hudson_lynx_hare-lotka_volterra	8	Yes	40	Lotka-Volterra Error Model
iohmm_reg_simulated-iohmm_reg	19	No	2000	Input Output HMM
irt_2pl-irt_2pl	144	No	2000	Two Parameter Logistic Item Response Theory Model
kidiq_with_mom_work-kidscore_interaction_c	5	Yes	434	Multiple Interacting Predictors Centered Linear Model
kidiq_with_mom_work-kidscore_interaction_c2	5	Yes	434	Multiple Interacting Predictors Conventionally Centered Linear Model
kidiq_with_mom_work-kidscore_interaction_z	5	Yes	434	Multiple Interacting Predictors Standardized Linear Model
kidiq_with_mom_work-kidscore_mom_work	5	Yes	434	Multiple Factor Level Predictors Linear Model
kidiq-kidscore_interaction	5	Yes	434	Interacting Linear Model
kidiq-kidscore_momhs	3	Yes	434	One Predictor Linear Model
kidiq-kidscore_momhsiq	4	Yes	434	Multiple Predictors Linear Model
kidiq-kidscore_momiq	3	Yes	434	One Predictor Linear Model
kilpisjarvi_mod-kilpisjarvi	3	Yes	62	Multiple Highly Correlated Predictors Linear Model
loss_curves-losscurve_sislob	15	No	55	Hierarchical Loss Curve Model
low_dim_gauss_mix_collapse-low_dim_gauss_mix_collapse	5	No	1000	A Two-Dimensional (unordered) Gaussian Mixture Model
low_dim_gauss_mix-low_dim_gauss_mix	5	Yes	1000	A Two-Dimensional Gaussian Mixture Model
lsat_data-lsat_model	1012	No	160	Random Effects (Rasch) Model for True Difficulty of LSAT Questions
M0_data-M0_model	4	No	711	Inferring population size with constant detection probability
Mb_data-Mb_model	1596	No	1590	Inferring population size considering immediate trap-response
mcycle_gp-accel_gp	66	Yes	5320	Gaussian Processes that model the mean and std of acceleration
mcycle_splines-accel_splines	82	No	5054	Splines for time-series data with varying mean and std
mesquite-logmesquite_logva	5	Yes	46	Log-Log Model

Posterior name	# of params.	Ref. draws	Max data size	Model description
mesquite-logmesquite_logvas	8	Yes	46	Log-Log Model
mesquite-logmesquite_logvash	7	Yes	46	Log-Log Model
mesquite-logmesquite_logvolume	3	Yes	46	Log-Log Model
mesquite-logmesquite	8	Yes	46	Multiple Predictors Log-Log Model
mesquite-mesquite	8	Yes	46	Multiple Predictors Linear Model
Mh_data-Mh_model	1159	No	385	Logistic-Normal Heterogeneity Model
mnist_100-nn_rbm1bJ10	7949	No	78400	A One Layer Restricted Boltzman Machine Neural Network
mnist-nn_rbm1bJ100	79409	No	47040000	A One Layer Restricted Boltzman Machine Neural Network (100 hidden units)
Mt_data-Mt_model	7	No	711	Capture-recapture model where detection probability varies by occasion
Mtbh_data-Mtbh_model	1912	No	730	Capture-recapture model where detection probability varies by occasion
Mth_data-Mth_model	5044	No	1935	Capture-recapture model where detection probability varies by occasion and heterogeneity
nes_logit_data-nes_logit_model	2	No	1179	Logistic Regression Model for Voting Preference based on Income
nes1972-nes	10	Yes	1330	Multiple Predictor Linear Model
nes1976-nes	10	Yes	1184	Multiple Predictor Linear Model
nes1980-nes	10	Yes	701	Multiple Predictor Linear Model
nes1984-nes	10	Yes	1226	Multiple Predictor Linear Model
nes1988-nes	10	Yes	1113	Multiple Predictor Linear Model
nes1992-nes	10	Yes	1350	Multiple Predictor Linear Model
nes1996-nes	10	Yes	1043	Multiple Predictor Linear Model
nes2000-nes	10	Yes	476	Multiple Predictor Linear Model
normal_2-normal_mixture	3	No	1000	Two Component Gaussian Mixture Model
normal_5-normal_mixture_k	15	No	1701	K Component Gaussian Mixture Model
one_comp_mm_elim_abs-one_comp_mm_elim_abs	4	Yes	20	A one compartment model
ovarian-logistic_regression_rhs	3075	No	82944	Logistic Regression with Regularized Horseshoe Prior
pilots-pilots	18	No	40	Linear Mixed Effects Model
prideprejudice_chapter-ldaK5	7714	No	32877	LDA with 5 topics
prideprejudice_paragraph-ldaK5	15570	No	32877	LDA with 5 topics
prostate-logistic_regression_rhs	11935	No	608532	Logistic Regression with Regularized Horseshoe Prior
radon_all-radon_county_intercept	388	No	12573	A county intercept model (no pooling) for the Radon dataset
radon_all-radon_hierarchical_intercept_centered	391	No	12573	A county intercept model with county level covariate for the Radon dataset
radon_all-radon_hierarchical_intercept_noncentered	391	No	12573	A county intercept model with county level covariate for the Radon dataset (non-centered)
radon_all-radon_partially_pooled_centered	389	No	12573	Hiearchical intercept model for the Radon dataset (centered)
radon_all-radon_pooled	3	No	12573	A pooled linear model for the Radon dataset
radon_all-radon_variable_intercept_centered	390	No	12573	Variable intercept hiearchical model (cen tered)
radon_all-radon_variable_intercept_noncentered	390	No	12573	Variable intercept hiearchical model (non-centered)
radon_all-radon_variable_intercept_slope_centered	777	No	12573	Variable intercept and slope hiearchical Radon model (centered)
radon_all-radon_variable_intercept_slope_noncentered	777	No	12573	Variable intercept and slope hiearchical Radon model (noncentered)
radon_all-radon_variable_slope_centered	390	No	12573	Variable slope hiearchical Radon model (centered)
radon_all-radon_variable_slope_noncentered	390	No	12573	Variable slope hiearchical Radon model (non-centered)
radon_mn-radon_county_intercept	87	No	919	A county intercept model (no pooling) for the Radon dataset
radon_mn-radon_hierarchical_intercept_centered	90	No	919	A county intercept model with county level covariate for the Radon dataset
radon_mn-radon_hierarchical_intercept_noncentered	90	No	919	A county intercept model with county level covariate for the Radon dataset (non-centered)
radon_mn-radon_partially_pooled_centered	88	No	919	Hiearchical intercept model for the Radon dataset (centered)
radon_mn-radon_partially_pooled_noncentered	88	No	919	Hiearchical intercept model for the Radon dataset (non-centered)
radon_mn-radon_pooled	3	No	919	A pooled linear model for the Radon dataset
radon_mn-radon_variable_intercept_centered	89	No	919	Variable intercept hiearchical model (centered)
radon_mn-radon_variable_intercept_noncentered	89	No	919	Variable intercept hiearchical model (non-centered)
radon_mn-radon_variable_intercept_slope_centered	175	No	919	Variable intercept and slope hiearchical Radon model (centered)

Posterior name	# of params.	Ref. draws	Max data size	Model description
radon_mn-radon_variable_intercept_slope_noncentered	175	No	919	Variable intercept and slope hiearchical Radon model (noncentered)
radon_mn-radon_variable_slope_centered	89	No	919	Variable slope hiearchical Radon model (centered)
radon_mn-radon_variable_slope_noncentered	89	No	919	Variable slope hiearchical Radon model (non-centered)
radon_mod-radon_county	389	No	12573	Hierarchical Model
Rate_1_data-Rate_1_model	1	No	1	Predicting the rate of correct test covariate question answers
Rate_2_data-Rate_2_model	3	No	1	Difference in success rates
Rate_3_data-Rate_3_model	1	No	1	Common rate of success from two trials
Rate_4_data-Rate_4_model	4	No	1	Success rate of a trial with prior and posterior inference
Rate_5_data-Rate_5_model	3	No	1	Common Rate of Success From Two Trials with Posterior Predictives
rats_data-rats_model	69	No	150	Normal Heirarchical Model to Model Rats' Weight Over Time
rstan_downloads-prophet	62	No	39746	Structural Time Series Model
sat-hier_2pl	670	No	19200	Hierarchical Two-Parameter Logistic Item Response Model
sblrc-blr	6	Yes	500	A Bayesian linear regression model with vague priors
sblri-blr	6	Yes	500	A Bayesian linear regression model with vague priors
science_irt-grsm_latent_reg_irt	408	No	2744	Rating scale and generalized rating scale models with latent regression
seeds_data-seeds_centered_model	47	No	21	Normal Heirarchical Model to Model Rats' Weight Over Time
seeds_data-seeds_model	27	No	21	Random Effect Logistic Regression for Seed Germination Proportion
seeds_data-seeds_stanified_model	26	No	21	Normal Heirarchical Model to Model Rats' Weight Over Time
sesame_data-sesame_one_pred_a	3	No	240	Linear model for the effect of encouragement to watch on actually watching Sesame Street
sir-sir	4	No	20	Simple SIR model
soil_carbon-soil_incubation	6	No	25	Two Pool Linear Model with Feedback
state_wide_presidential_votes-hierarchical_gp	181	No	550	Hierarchical Gaussian Process
surgical_data-surgical_model	28	No	12	Random Effects Model to Rank Hospitals on True Failure Probability
Survey_data-Survey_model	1002	No	5	Inferring the Return Rate and Number of Surveys from Observed Returns
synthetic_grid_RBF_kernels-kronecker_gp	438	No	900	Kronecker Gaussian Process
three_docs1200-ldaK2	12	No	1200	LDA with 2 topics
three_men1-ldaK2	510	No	4999	LDA with 2 topics
three_men2-ldaK2	526	No	4999	LDA with 2 topics
three_men3-ldaK2	516	No	4999	LDA with 2 topics
timssAusTwn_irt-gpcm_latent_reg_irt	530	No	5500	Partial credit and generalized partial credit models with latent regression
traffic_accident_nyc-bym2_offset_only	3845	No	5461	BYM2 model
uk_drivers-state_space_stochastic_level_stochastic_seasonal	389	No	192	Structured Time Series model with stochastic level and stochastic seasonal
wells_data-wells_daae_c_model	6	No	3020	4-Predictor logistic regression model with centered inputs for decision to switch wells
wells_data-wells_dae_c_model	5	No	3020	3-Predictor logistic regression model with centered inputs for decision to switch wells
wells_data-wells_dae_inter_model	7	No	3020	3-Input logistic regression model with interactions and centered inputs for decision to switch wells
wells_data-wells_dae_model	4	No	3020	3-Predictor logistic regression model for decision to switch wells.
wells_data-wells_dist	2	No	3020	Logistic regression model for decision to switch wells
wells_data-wells_dist100_model	2	No	3020	Logistic regression model for decision to switch wells
wells_data-wells_dist100ars_model	3	No	3020	2-Predictor Logistic regression model for decision to switch wells
wells_data-wells_interaction_c_model	3	No	3020	2-Predictor logistic regression model with interactions and centered inputs for decision to switch wells
wells_data-wells_interaction_model	4	No	3020	2-Predictor Logistic regression model with interactions for decision to switch wells