# Stacking for non-mixing Bayesian computations: the curse and blessing of multimodal posteriors

Yuling Yao * Aki Vehtari † Andrew Gelman ‡

**Abstract**

When working with multimodal Bayesian posterior distributions, Markov chain Monte Carlo (MCMC) algorithms can have difficulty moving between modes, and default variational or mode-based approximate inferences will understate posterior uncertainty. And, even if the most important modes can be found, it is difficult to evaluate their relative weights in the posterior. Here we propose an approach using parallel runs of MCMC, variational, or mode-based inference to hit as many modes or separated regions as possible and then combine these using Bayesian stacking, a scalable method for constructing a weighted average of distributions so as to minimize cross validation prediction error. The result from stacking is not necessarily equivalent, even asymptotically, to fully Bayesian inference, but it serves many of the same goals. Under misspecified models, stacking can give better predictive performance than full Bayesian inference, hence the multimodality can be considered a blessing rather than a curse. We explore theoretical consistency with an examples where the stacked inference can approximate the true data generating process from the misspecified model and a non-mixing sampler. We consider practical implementation in several model families: latent Dirichlet allocation, Gaussian process regression, hierarchical regression, horseshoe variable selection, and neural networks.

**Keywords**: Bayesian stacking, Markov chain Monte Carlo, model misspecification, multimodal posterior, parallel computation, postprocessing.

## 1. The curse of multimodal posteriors

Bayesian computation becomes difficult when posterior distributions are multimodal or, more generally, meta-stable, as then it is generally impossible to compute moments analytically or to directly draw simulations. Variational and mode-based approximations can be poor fits to the posterior, and it can be difficult to identify all the modes in the first place. General-purpose Markov chain Monte Carlo algorithms can have problems moving between modes. And even if different modes are found, it is difficult to compute their relative weights in the posterior distribution, as this requires integration over the posterior density within each mode.

Should we just run longer and longer chains? This is inefficient when effective sample size per iteration (Vehtari et al., 2020a) is low. The state-of-the-art Hamiltonian Monte Carlo sampler for a bimodal density mixes as poorly as random-walk Metropolis (Mangoubi et al., 2018), and even optimal tuning and Riemannian metrics do not help. When the posterior distribution has unconnected masses, the Markov chain is not irreducible and will never converge to its target.

In some cases it is possible to collapse multimodality using reparameterization (Papaspiliopoulos et al., 2007; Johnson et al., 2012; Betancourt and Girolami, 2015; Gorinova et al., 2019), but this is not automated for general problems. Several schemes have been proposed for sampling from distributions with isolated modes by increasing the temperature to enhance the transition probability between modes; these methods include annealing (Kirkpatrick et al., 1983; Bertsimas and Tsitsiklis, 1993), parallel tempering (Hansmann, 1997; Earl and Deem, 2005), simulated tempering (Marinari and Parisi, 1992; Neal, 1993), Wang-Landau algorithm (Wang and Landau, 2001), and path sampling (Gelman and Meng, 1998; Yao et al., 2020). These merthods enlarge the sampling space by

*Department of Statistics, Columbia University. yy2619@columbia.edu.

†Helsinki Institute of Information Technology; Aalto University, Department of Computer Science.

‡Department of Statistics and Political Science, Columbia University.

an auxiliary tempeture and often require to estimate the partition function: a high dimensional integral. They are useful for many statistical physics and molecular biology problems, in which the only goal is to sample from a given density. However, tempering-based methods are sensitive to implementation and tuning, and their theoretical mixing rates drop quickly in high dimensions (Bhatnagar and Randall, 2004). Based on simulations in Yao et al. (2020), methods based on simulated tempering are unlikely to work in large statistical models with the scale that we consider in the present paper.

Moreover, the metastability of sampling (Figure 1) comes from both the energetic (two modes are distinct) and entropic (two regions are connected through a narrow neck) barriers. Increasing the temperature does not ease the entropic barrier, which is a common problem with hierarchical models.

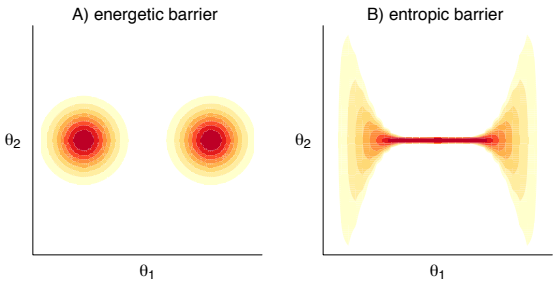



Figure 1: *Both the energetic and entropic barriers lead to sampling metastability, and the entropic barrier cannot be eliminated by increasing the temperature.*

Despite these computational difficulties, we would like to aim for full Bayesian inference, or some approximation, because of the general benefits of using probability to quantify uncertainty in inferences.

One way to explore a multimodal space is to run a large number of chains of MCMC or variational inference from dispersed starting points, but then the question arises of how to average over resulting inferences if they have not mixed. There can be benefits to averaging non-mixing MCMC chains even with uniform weighting (Hoffman and Ma, 2020), but it should be possible do better: equal weighting is convenient but is not in general appropriate and can lead to a strong dependence on initial values.

*Stacking* (Wolpert, 1992; Breiman, 1996; LeBlanc and Tibshirani, 1996; Clarke, 2003) and its Bayesian variants (Clyde and Iversen, 2013; Le and Clarke, 2017; Yao et al., 2018a) use cross validation to average over a discrete set of fitted models.

The present paper extends stacking to combine multiple chains fitting the same model. Although this might seem like a small step, the practical implications of this idea are large, because multimodality, meta-stability, and poor mixing are common problems when fitting complex multilevel models.

Applying Bayesian stacking for non-mixing computations involves two challenges. The computational challenge is to approximate the results of cross validation without the need to repeatedly re-fit the model. This can be done using Pareto-smoothed importance sampling (Vehtari et al., 2017, 2019b). The conceptual challenge is that the goal of minimizing prediction error is not the same as minimizing Kullback-Leibler divergence to or approximating a posterior distribution. We can prove some theoretical results regarding the benefits of cross validation in this setting, but ultimately the effectiveness of stacking is best demonstrated by applying it to a series of challenging problems that represent different sorts of posterior distributions that arise in applied statistics.

The contribution of this paper is to provide a practical solution to yield a combined inference from non-mixing computations, and to evaluate it on a diverse set of statistical examples. Our procedure is scalable with negligible postprocessing cost and constructed to minimize cross-validated prediction error.

In Section 2 we discuss various types of posterior multimodality, some of which can arise from model misspecification. Section 3 details our method and practical implementation to deal with non-mixing chains for Bayesian computation. We use a theoretical example in Section 4 to show that stacked-chain inference can achieve higher predictive efficiency than exact posterior density

asymptotically. In Section 5 we demonstrate the proposed method in applied examples, including hyperparameter bimodality and slow mixing in Gaussian process regression, latent Dirichlet allocation, unstable variational inference in horseshoe regression, and Bayesian neural networks.

## 2. The folk theorem of statistical computing

*When you have computational problems, often there's a problem with your model.* This "folk theorem" (Gelman, 2008; Gelman et al., 2020) can be understood by thinking of a statistical model or family of distributions as a set of possible probabilistic explanations for a dataset. If the data come from some distribution in the model class, then with identification and reasonable sample size we can expect to distinguish among these explanations, and with a small sample size and continuous model, we would hope to find a continuous range of plausible explanations and thus a well behaved posterior distribution. But if the data do not fit the model, so that none of the candidate explanations work, then the posterior distribution represents a mixture of the best of bad choices, and it can have poor geometry in the same way that the seafloor can look rough if the ocean is drained.

Poor data fit, or conflict between the prior and likelihood, do not necessarily lead to awkward computation. For example, the normal-normal model yields a log-concave posterior density with constant curvature for any data. But if a model is flexible enough to fit different qualitative explanations of data, then poorly fitting data can be interpreted by the model as ambiguity, as indicated by posterior multimodality.

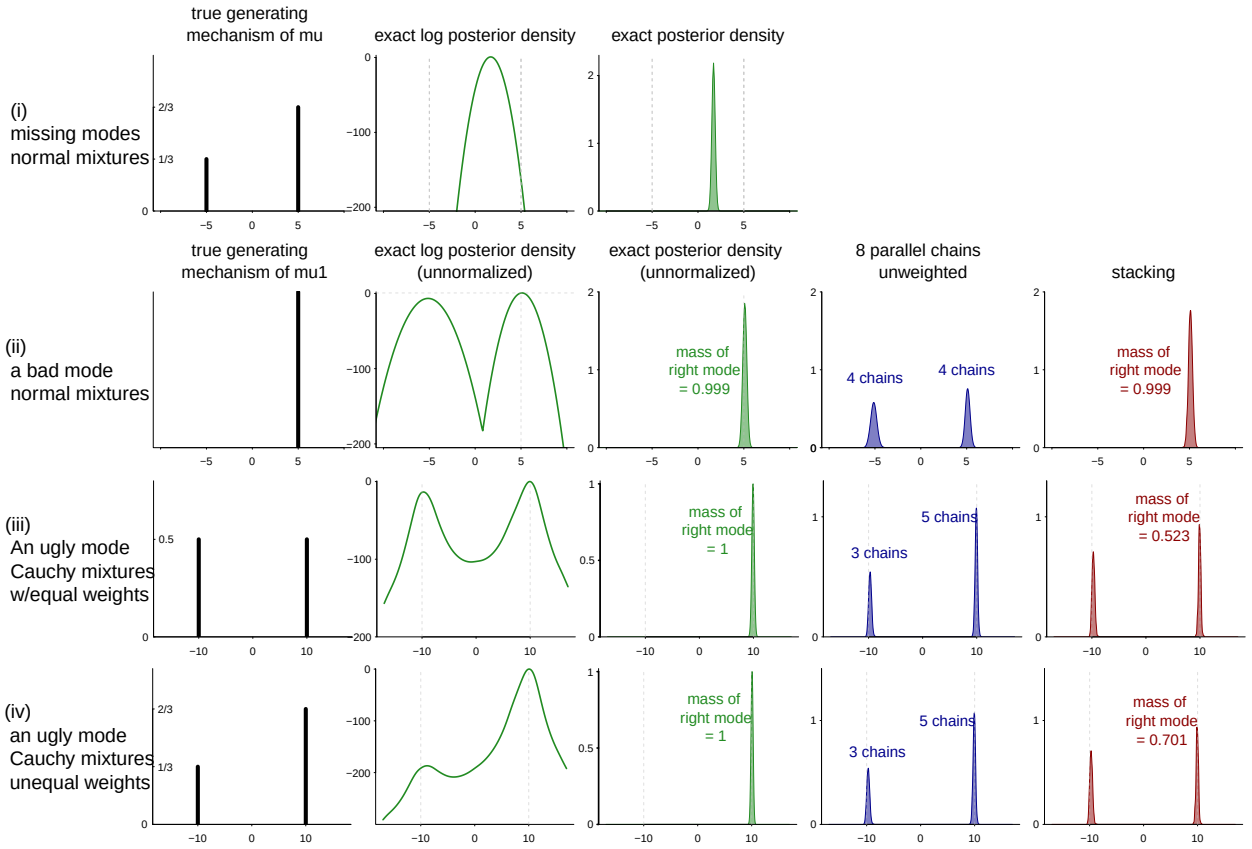
The other way a model can be difficult to fit is if its parameters are only weakly constrained by the posterior. With a small sample size (or, in a hierarchical model, a small number of groups), uncertainty in the hyperparameters can yield a posterior distribution of widely varying curvature, which leads to slowly mixing MCMC. In practice, we can often fix the geometry by putting stronger priors on these hyperparameters. However, a strong prior constraint is not always desired—sometimes we are interested in fitting a model that is legitimately difficult to compute, because we want to allow for different possible explanations of the data, and a too strong prior implies an adhoc selection. These are settings where the stacking approach discussed in this paper can be useful.

Under the correct model and reasonable priors, Bayesian posteriors often attain asymptotic normality and leave little room for several distinct and non-vanishing modes. That ensures rapid mixing for random-walk Metropolis, scaling as $\mathcal{O}(d)$ (Roberts et al., 1997; Cotter et al., 2013; Dwivedi et al., 2018), and Hamilton Monte Carlo, scaling as $\mathcal{O}(d^{1/4})$ (Beskos et al., 2013; Bou-Rabee et al., 2018; Mangoubi and Smith, 2017, 2019). From this perspective, multimodal posteriors should be unlikely with a large enough sample size. With smaller sample sizes, lack of convergence to the asymptotic normality can arise from various sources.

### 2.1. Modes: the good, the bad, and the ugly

To demonstrate the behavior of exact inference, uniformly weighted parallel chains, and stacking weighted parallel chains in case of different types posteriors, we design four mixture examples, visualized in Figure 2:

(i) A missing mode: We draw $n$ points $y_i, \dots, y_n$ independently from the mixture, $\frac{2}{3}\text{normal}(5, 1) + \frac{1}{3}\text{normal}(-5, 1)$. We fit the model $y_i|\mu \sim \text{iidnormal}(\mu, 1)$ with a flat prior on $\mu$. The true data generating process (DG) is expressed by $\mu \sim \frac{2}{3}\delta(5) + \frac{1}{3}\delta(-5)$, but the Bayesian posterior $p(\mu|y) = \text{normal}(\bar{y}, 1/\sqrt{n})$ is unimodally concentrated at $\mu = \bar{y} \approx 5/3$ and cannot catch the two modes in data.




| **Summary:** example | # modes in DG | # modes in posterior | posterior → DG as $n \to \infty$? | unweighted chains approx. DG? | stacking approx. DG? |
|---|---|---|---|---|---|
| i | 2 | 1 | ✗ | ✗ | ✗ |
| ii | 1 | 2 | ✓ | ✗ | ✓ |
| iii | 2 | 2 | ✗ | ✗ | ✓ |
| iv | 2 | 2 | ✗ | ✗ | ✓ |

Figure 2: *Under a multimodal data generating mechanism, the exact Bayesian posterior can miss the modes in (i) or over-concentrate at one mode (iii–iv). Stacking, our proposed method, approximates the data generating process well in (ii–iv). The sample size* $n = 30$ *in (i–ii) and* $n = 100$ *in (iii–iv).*

(ii) A bad mode: With the same data $y$ above, now we fit a two-component normal model $y \sim \frac{2}{3}\text{normal}(\mu_1, 1) + \frac{1}{3}\text{normal}(\mu_2, 1)$ with known mixture probability and a flat prior on $\mu_1, \mu_2$. The model is identifiable, but the resulting posterior is bimodal, centered around $(\mu_1, \mu_2) = (5, -5)$ and $(-5, 5)$ respectively. Asymptotically $(n \to \infty)$ the posterior converges to the first mode, thereby the data generating process, but the existence of a second artifact mode both challenges the sampling and compromises the prediction with finite data sample size. In Figure 2 we simulate $n = 30$ data points and run eight parallel chains. Four chains are trapped in the "wrong" mode.

(iii) An ugly mode: We generate data $y_1, \ldots, y_n$ iid from $\frac{1}{2}\text{Cauchy}(10, 1) + \frac{1}{2}\text{Cauchy}(-10, 1)$. We fit a one-component model $y \sim \text{Cauchy}(\mu, 1)$ with a flat prior. The true data generating process is expressed by $\mu \sim \frac{1}{2}\delta(10) + \frac{1}{2}\delta(-10)$. In the limit $(n \to \infty)$, the posterior density will be concentrated at one of two points $\mu \approx \pm 9.8$. In the simulation with $n = 100$, the right-side posterior mode contains almost 100% mass (up to the precision $10^{-6}$). The induced predictive

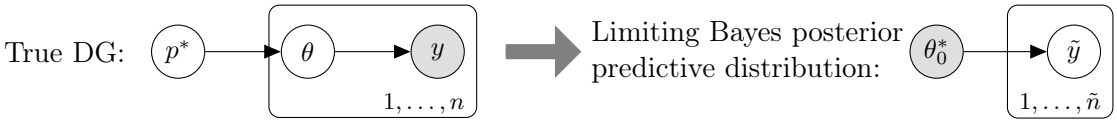



Figure 3: *When the parameter $\theta$ is randomly drawn from a distribution $p^*$ in the data generating process* (3)*, the limiting posterior inference $p(\theta|y)$ almost surely converges to a point estimate $\theta_0^*$.*

model then only describes half of data. *Stacking*, as implemented in this paper, assigns a weight of 0.52 to the right-side mode, achieving a much better prediction compared to the data generating process.

(iv) Another ugly mode: We draw $n$ data points $y_i, \ldots, y_n$ independently from the mixture model, $\frac{2}{3}\text{Cauchy}(10, 1)+\frac{1}{3}\text{Cauchy}(-10, 1)$ and again fit a one-component model $y \sim \text{Cauchy}(\mu, 1)$ with a flat prior. The posterior $p(\theta|y)$ carries almost all masses on the right-side mode $p(\theta|y) \approx \delta(\theta-10)$, while our proposed method still approximates the true data generating process.

These examples represent various sources of posterior multimodality. In example (ii), one of the modes is purely an artifact. Not drawing a posterior sample around it improves finite sample-predictions. Such artifact-type modes are found in cases of prior-data conflict, label-switching, aliasing (Bafumi et al., 2005), mixture and cluster-based models (Stephens, 2000; Blei et al., 2003), and hierarchical models (Liu and Hodges, 2003).

In other examples, the data generating process (DG) can be expressed via a bimodal distribution on $\mu$. In example (i), the Bayesian posterior $p(\mu|y)$ converges to some middle point. In (iii–iv), the posterior overconfidently concentrates at one of the modes and ignores the other, even when these two modes have equal density at the modal point. We will revisit the Cauchy mixture in Section 4 and prove its limiting behavior, in which Bayesian inference almost surely occasions overconfident concentration, but the blessing of bimodality enables stacking to recover the *true* data generating process from the *wrong* model and *wrong* inference.

### 2.2. The curse of multiverse data generating mechanism: Bayes can be overconfident, too.

Given data $y_1, \ldots, y_n$ generated independently and identically distributed from an unknown data generating process: $p_{\text{true}}(y)$, and a potentially misspecified model $y|\theta \sim f(y|\theta)$ and prior $p(\theta), \theta \in \Theta$, when the sample size $n$ goes to infinity and regularization conditions apply, the limiting Bayesian posterior will be almost surely supported on the set of global modes (Berk, 1966; Kleijn and Van der Vaart, 2012): $\mathcal{A} = \left\{\theta^* \in \Theta : \text{E}_{\tilde{y}\sim p_{\text{true}}} \log f(\tilde{y}|\theta^*) = \max_{\theta\in\Theta} \text{E}_{\tilde{y}\sim p_{\text{true}}} \log f(\tilde{y}|\theta)\right\}$.

Such limiting behavior has two undesired properties. First, when data are generated from one parameter $\theta_0$ in the model (an $\mathcal{M}$-closed view), $p_{\text{true}} = f(\cdot|\theta_0)$, the posterior will be asymptotically concentrated at $\theta_0$. But otherwise, the limiting predictive distribution should be interpreted as the closest distribution to data generating process in terms of Kullback–Leibler (KL) divergence, as we can rewrite $\mathcal{A}$ as

$$\mathcal{A} = \arg\min_{\theta\in\Theta} \text{KL}\Big(p_{\text{true}}(\cdot) \;||\; f(\cdot|\theta)\Big), \quad \forall\eta > 0,\ \Pr_{\text{bayes}}(||\theta - \mathcal{A}|| < \eta \mid y_{1\ldots,n}) \xrightarrow{n\to\infty,\ a.s.} 1, \qquad (1)$$

The asymptotic predictive distribution is equivalent to some *point* estimate $f(\cdot|\theta^*), \theta^* \in \mathcal{A}$. What makes a method Bayesian is the use of probability to quantify uncertainties. Ideally we would fully use the expressibility of the model and find the optimal *probabilistic* inference $p_{\text{optimal}}(\theta)$ from

some space $\mathcal{F}$ that renders the best prediction for future unseen data according to a user-specified divergence $D(\cdot||\cdot)$,

$$p_{\text{optimal}} = \arg\min_{\tilde{p}\in\mathcal{F}} \mathrm{D}\Big(p_{\text{true}}(\cdot) \;||\; \int_\Theta f(\cdot|\theta)\tilde{p}(\theta)d\theta\Big). \tag{2}$$

In particular, if the model is expressive enough (see Figure 3), then there is a density $p^*(\cdot) \in \mathcal{F}$ such that

$$p_{\text{true}}(\tilde{y}) = \int_\Theta f(\tilde{y}|\theta)p^*(\theta)d\theta. \tag{3}$$

and we can choose any divergence $D$ corresponding to a strictly proper scoring rule (Gneiting and Raftery, 2007). Then the minimum of (2) is attained at $p^*$, hence the "correct inference."

The limiting Bayesian posterior (1) is a special solution to (2) where we fix D to be KL divergence and restrict the distribution family $\mathcal{F}$ to be formed from Dirac delta functions: $\mathcal{F} = \{\delta(\theta_0) \mid \theta_0 \in \Theta\}$.

To rephrase the folk theorem, challenges in sampling and difficulties in modeling are confounded. There is no general algorithm to sample from truly multimodal distributions, and even if this could be done, the posterior multimodality can signify that the true data are unlikely generated to have been from any single parameter in the model, and so the Bayesian posterior itself, which over-concentrates in the limit, is not appropriate.

With enough data, model misspecifications can be detected using posterior predictive checks (Gelman et al., 1996), and (3) can be expanded to a hierarchical model by replacing $\theta$ by $n$ copies $\theta_{1,\ldots,n}$, with hyperpriors $\theta_i|\tau \sim p(\theta_i|\tau)$:

$$y_i|\theta_i = f(y_i|\theta_i), \quad \theta_i|\tau \sim p(\theta_i|\tau), \quad \tau \sim p(\tau), \quad i = 1, \ldots, n. \tag{4}$$

But it enlarges the model $n$ times bigger, hurting both computational scalability and finite-sample convergence rate. Meanwhile, the hyperprior $p(\theta_i|\tau)$ can still be misspecified.

## 3. An approach to inference from non-mixed computation: parallel approximation and stacking

### 3.1. Proposed method

To start, we assume we have some existing computer program that attempts to draw samples from a posterior distribution $p(\theta|y)$ but might get trapped in a single mode or, more generally, a small part of the distribution. We will typically be using Hamiltonian Monte Carlo with the no-U-turn sampler (HMC/NUTS) in Stan (Stan Development Team, 2020). For the present paper, all that is necessary is that the algorithm produces *some* set of posterior draws, which might also be obtained for example by variational inference (Blei et al., 2017) or mode-based approximation such as Laplace's method or expectation propagation (Vehtari et al., 2020b).

*Step 1: Parallel evaluation.* We run our program $M$ times from different starting points to have a chance to explore many modes or areas of the target distribution. We also recommend an overdispersed initialization. Multiple starting points is not a new idea in statistical computation, but we emphasize that our goal here is *exploration*, without the expectation that the chains will mix with each other, nor that all modes and separated regions are reached. It could, for example, make sense to run the simulation algorithm in parallel on a large number of processors in a cluster. If the algorithm is iterative, follow the usual protocol to discard the initial transient states; for example when running MCMC we typically discard, as warmup, the first half of each simulated chain.

In practice, within-chain convergence is easier than mixing among parallel chains. This is especially true for distribution with isolated modes and high between-mode energy barriers. To monitor the within-chain convergence, we use split-$\widehat{R}$ (Vehtari et al., 2020a). For most simulation we experimented, it is fairly easy to have split-$\widehat{R} \approx 1$ for most chains.

*Step 2 (optional): Clustering.* We can use a between-chain mixing measure such as $\widehat{R}$ (Gelman and Rubin, 1992; Vehtari et al., 2020a) to partition the $M$ parallel simulations into $K$ clusters, each of which approximately captures the same part of the target distribution. Label the simulations from cluster $k$ as $(\theta_{ki}, i = 1, \ldots, S_k)$, with the total number of draws being $S = \sum_{k=1}^{K} S_k$. This step is optional and recommended if the number of parallel runs $M$ is large.

To keep notation coherent, when the clustering step is skipped, we denote $K = M$ and $\theta_{ks}$ as the $s$-th sample in the $k$-th chain. Throughout the paper, we use $1 \le i \le n$ to index outcome observations, $1 \le k \le K$ to index clusters (chains, optimization runs), and $1 \le s \le S$ to index posterior draws.

*Step 3: Reweighing non-mixing chains using stacking.* From the previous two steps, we assume $\theta_{ki}$ come from a stationary distribution $p_k(\theta|y)$, which in general do not mix, nor do they match the exact posterior $p(\theta|y)$.

To take into account between-chain heterogeneity, we consider a generalized form of Monte Carlo estimate for any integral function $h(\theta)$ from chain-wise weights $w_1, w_2, \ldots, w_K$:

$$\mathrm{E}[h(\theta)] \approx \sum_{k=1}^{K} \sum_{s=1}^{S_k} w_k S_k^{-1} h(\theta_{ks}). \tag{5}$$

The usual Monte Carlo estimate is a special case with $w_1 = \cdots = w_K = 1/K$.

We optimize weights in (5) to maximize the leave-one-out cross validation performance of the distribution formed from the weighted average of the simulation draws. This first requires estimation of the pointwise leave-one-out (loo) log predictive density (lpd, Gelman et al., 2014; Vehtari et al., 2017) from the $k$-th cluster (chain):

$$\log p_k(y_i|y_{-i}) = \log \int_{\theta\in\Theta} p(y_i|\theta) p_k(\theta|y_{1,\ldots,i-1,i+1,\ldots,n}) d\theta, \quad i = 1, \ldots, n,\ k = 1, \ldots, K. \tag{6}$$

Second, we solve

$$w^*_{1,\ldots,K} = \arg\max_{w\in\mathbb{S}(K)} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p_k(y_i|y_{-i}) + \log p_{\text{prior}}(w), \tag{7}$$

where $\mathbb{S}(K)$ is the space of $K$-dimensional simplex,

$$\mathbb{S}(K) = \{w : 0 \le w_k \le 1, \forall 1 \le k \le K;\ \sum_{k=1}^{K} w_k = 1\},$$

and $p_{\text{prior}}(w)$ is prior regularization.

In Section 3.2, we further approximate all $\log p_k(y_i|y_{-i})$ terms by importance sampling—it suffices to fit all the full data once in each chain. We will also discuss the choice of priors $p_{\text{prior}}(w)$.

We view this optimization as a finite-sample estimate in (2), where we construct the distribution family as a mixture from sampled posterior clusters, $\mathcal{F} = \left\{\sum_{k=1}^{K} w_k p_k(\theta|y) : w \in \mathbb{S}(K)\right\}$.

Finally, plugging the stacking wights $w_1^*, \ldots, w_K^*$ into (5), we obtain the chain-weighted Monte Carlo estimates. The resulting approximation of the target distribution uses $\sum_{k=1}^K S_k$ draws, with each $\theta_{ks}$ having weight $w_k^*/S_k$.

Notably, we are by default using the logarithmic scoring rules in predictive evaluation, for it is strictly proper. It is straightforward to adopt a user-specified prediction utility by replacing the log predictive densities in the optimization step (7).

*Step 4: Monitoring convergence.* After $K$ parallel runs, we cannot exclude the possibility that another local mode or separated posterior region has been overlooked. We could use capture-recapture methods to estimate the number of unseen modes. When there is a discrete combinatorial explosion, it is essentially impossible to capture the full support of the distribution. So we are implicitly assuming that we have a rough sense of the support of most of the posterior mass, or, conversely, that we were previously willing to approximate the target distribution using a single mode, in which case we would hope a multimodal average to be an improvement.

On the other hand, there is no need to capture all modes that are predictively identical. We monitor the weighted log predictive density as a function of how many components are added in stacking. Ideally we should test it over an independent hold-out test data set, and stop when the log predictive density of the stacked posterior reaches the maximum. Alternatively we can use cross validation. For each $K' \leq K$, obtain stacking weights $w_k^{K'}$ from chain $1, \ldots, K'$, and monitor the their stacked lpd as a function of number of chains $K'$, which typically monotonically increases:

$$\mathrm{lpd}_{\mathrm{loo}}(K') = \sum_{i=1}^n \log \sum_{k=1}^{K'} w_k^{K'} p_k(y_i|y_{-i}),\ 1 \leq K' \leq K. \tag{8}$$

We terminate if $\mathrm{lpd}_{\mathrm{loo}}(K')$ becomes relatively stable. Otherwise we sample extra chains and repeat steps 1–4 on all chains.

### 3.2. Practical implementation

*Leave-one-out posterior distributions.* Let $p_k(\theta|y)$ to be the stationary distribution from which the $k$-th cluster (chain) is drawn. Working with the exact leave-one-out distributions $p_k(\theta|y_{-i})$ in (6) is not only computationally intensive (requiring the model to be fit $n$ times) but also conceptually ambiguous: Using full data and given initialization, the sampler obtains $\theta_{k1}, \ldots, \theta_{kS_k}$ from the $k$-th region, but what if after $y_i$ is removed from the same initialization reaches another mode, or what if there is a phase transition and there are no longer $K$ clusters?

The usual leave-one-out model can be written as, $p(\theta|y_{-i}) \propto p(\theta|y_{-i})p(\theta) = p(\theta|y)p(\theta)/p(y_i|\theta) = p(\theta|y)/p(y_i|\theta)$. We avoid the ambiguity by defining $p_k(\theta|y_{-i})$ to be

$$p_k(\theta|y_{-i}) \coloneqq \frac{p_k(\theta|y)/p(y_i|\theta)}{\int_{\theta\in\Theta} p_k(\theta|y)/p(y_i|\theta)}. \tag{9}$$

*Efficient approximation of leave-one-out distributions.* We use Pareto smoothed importance sampling (PSIS, Vehtari et al., 2017, 2019b) to compute (9). It suffices to only fit the full model once per chain. For each chain $k$, we obtain the raw leave-one-out importance ratios $1/p(y_i|\theta_{ks})$, $i = 1, \ldots, n$ and stabilize these by replacing the largest ratios by the expected order statistics in a fitted generalized Pareto distribution and followed by right truncation. Labeling the Pareto-smoothed importance

ratio as $r_{iks}$, we approximate $p_k(y_i|y_{-i})$ by

$$p_k(y_i|y_{-i}) \approx \frac{\sum_{s=1}^{S_k} p_k(y_i|\theta_{ks}) r_{iks}}{\sum_{s=1}^{S_k} r_{iks}},\ k = 1, \ldots, K,\ i = 1, \ldots, n. \tag{10}$$

This is asymptotically ($S_k \to \infty$) unbiased and consistent. The finite-sample reliability and convergence rate can be assessed using the estimated shape parameter $\hat{k}$ of the fitted generalized Pareto distribution. We refer to Vehtari et al. (2017, 2019a) and Appendix B of this paper for detailed algorithms and software implementation.

In summary, after parallel sampling, the extra computation costs of stacking only involve summations in (10) and a length-$K$-vector optimization in (7), which are negligible compared with the cost of sampling.

*Prior on stacking weights.* Extra priors beyond a simplex constraint in model averaging have been considered (Le and Clarke, 2017; Yao et al., 2018a) but seldom applied in practice. Under a flat prior $p_{\text{prior}}(w) = 1$, the optimum in (7) is nonidentified and numerically unstable if two simplexes $w' \neq w''$ entail the identical prediction $\sum_k w'_k p_k(\cdot|y) = \sum_k w''_k p_k(\cdot|y)$. We need an informative prior for the *predictive power* versus *Monte Carlo error* tradeoff.

If all chains are distributed identically, and within chain sampling is independent, the variance of (5) will be $\text{Var}\left(\sum_{k=1}^{K}\sum_{s=1}^{S_k} w_k {S_k}^{-1} h(\theta_{ks})\right) = \sum_{k=1}^{K} {w_k}^2 {S_k}^{-1} \text{Var}\left(h(\theta)\right)$, whose minimum is attained at $w_k = S_k / \sum_{k'} S_{k'}$. This justifies the uniform weights $1/K$ in the usual multi-chain Monte Carlo scheme where, after complete mixing, any weighting yields unbiased estimates.

Further, when the $k$-th chain has an effective sample size $S_{\text{eff},k}$ (Vehtari et al., 2020a), we approximate the variance of the Monte Carlo estimate (5) to be $\text{Var}\left(\sum_{k=1}^{K}\sum_{s=1}^{S_k} w_k {S_k}^{-1} h(\theta_{ks})\right) = \sum_{k=1}^{K} {w_k}^2 {S_{\text{eff},k}}^{-1} \text{Var}\left(h(\theta)\right)$, whose minimum will be attained at $w_k = S_{\text{eff},k} / \sum_{k'} S_{\text{eff},k'}$. This also suggests we can estimate the the effective sample size of $w$-weighted samples by:

$$\hat{S}_{\text{eff}} := \left(\sum_{k=1}^{K} {w_k}^2 {S_{\text{eff},k}}^{-1}\right)^{-1}.$$

To reduce Monte Carlo error, we partially pool stacking weights using a Dirichlet prior with a tuning scale parameter $\lambda > 0$ that controls the amount of partial pooling,

$$p_{\text{prior}}(w_{1,\ldots,K}) = \text{Dirichlet}\left(\frac{\lambda S_{\text{eff},1}}{\sum_{k'=1}^{K} S_{\text{eff},k'}}, \ldots, \frac{\lambda S_{\text{eff},K}}{\sum_{k'=1}^{K} S_{\text{eff},k'}}\right). \tag{11}$$

We add this regularization term into (7). If $\lambda = 1$ and $S_{\text{eff},k}$ is equal for all $k$, it becomes the unregularized Bayesian stacking. If $\lambda \to \infty$ and $S_{\text{eff},k} \propto S_k$, it results in the usual Monte Carlo estimate $w_k/S_k = 1/S$. Ideally $\lambda$ can be further tuned using hold-out data or extra cross validation. In later experiments of this paper, we simply use $\lambda = 1.001$ as a rule-of-thumb value.

Stacking using a flat prior is always convex, and therefore adding a small $\lambda$ breaks the tie and makes it strictly convex. If all chains are already mixed, stacking with an informative prior does not hurt, and we will recover the approximately uniform weighting.

*Thinning and importance resampling.* For settings where it is inconvenient to work with weighted simulation draws, we can perform thinning to obtain a set of $S_{\text{thin}}$ simulation draws approximating

the weighted mixture of $K$ distributions. We further adopt quasi Monte Carlo strategy to reduce variance. Given weights $\{w_k\}_{k=1}^K$ for $K$ clustered simulation draws $\{\theta_{ks}\}_{k=1,s=1}^{K,\ S_k}$, and an integer $S_{\text{thin}} \leq \inf_k(S_k/w_k)$, we first draw a fixed-sized $S_k^* = \lfloor S_{\text{thin}} w_k \rfloor$ sample randomly without replacement from the $k$-th cluster, and then sample the remaining $S_{\text{thin}} - \sum_{k=1}^K S_k^*$ without replacement with the probability proportional to $(w_k - S_k^*/S_{\text{thin}})$ from cluster $k$.

Lastly, we have implemented all related functions together to facilitate PSIS-loo based chain-stacking in an `R` package `loo`. It works seamlessly with Stan. See Appendix B for an example.

### 3.3. Comparison to importance sampling and Bayesian model (chain) averaging

We compare stacking with other possible chain-combination methods. First we can use importance sampling to adjust for differences between non-mixed chains and the target distribution, with the hope of recovering the exact Bayesian posterior. The importance ratio for the $k$-th cluster/chain is

$$\text{(Importance sampling)}: \ \alpha_k \propto \frac{1}{S_k}\sum_{s=1}^{S_k} p(\theta_{ks}, y) \approx \int_\Theta p_k(\theta|y)d\theta, \ k = 1, \ldots, K. \tag{12}$$

Under the ideal assumption that each chain is well-separated, and all regions of the posterior distributions have been fully explored,

$$\Theta = \bigcup_{k=1}^K \Theta_k; \quad \forall k' \neq k, \ p_k\left(\Theta_{k'}\right) \approx 0; \quad s.t. \ \forall \theta \in \theta_K, \ p(\theta|y) \approx \alpha_k p_k(\theta|y). \tag{13}$$

Then the importance ratio $p(\theta|y)/p_k(\theta|y) \approx$ constant $\alpha_k$ under $p_k(\cdot|y)$. Hence, the importance resampled draws match the exact posterior. Under the same assumption, the importance ratio $\alpha_k$ is proportional to the marginal likelihood. To see this, rewrite (12) as

$$\text{(BMA)}: \ \alpha_k \propto \int_\Theta p(y|\theta)p_{\text{prior}}(\theta)\mathbb{1}(\Theta_k)d\theta = P(y|\Theta_k), \ k = 1, \ldots, K.$$

Under a flat prior $p_{\text{prior}}(\Theta_k) = 1/K$, this leads to $\Pr(\Theta_k|y) \propto \Pr(y|\Theta_k) \propto \alpha_k$. Thus, using the importance ratio (12) in the weighted Monte Carlo (5) is exactly Bayesian model averaging (BMA, Madigan et al., 1996; Hoeting et al., 1999) on a discrete space $\{\Theta_k : 1 \leq k \leq K\}$. When assumption (13) does not hold, importance sampling (12) can still be viewed as BMA on models that are implicitly constructed from each chain. In all experiments later in the paper, we refer BMA to (12).

Yao et al. (2018a) introduced a pseudo-BMA weighting for model averaging. In our context, the pseudo-BMA weight for cluster $k$ is

$$\text{(pseudo}-\text{BMA)}: \ \alpha_k \propto \exp\left(\sum_{i=1}^n \log p(y_i|y_{-i}, \Theta_k)\right) \approx \exp\left(\sum_{i=1}^n \log \sum_{s=1}^{S_k} \frac{r_{iks}p(y_i|\theta_{ks})}{r_{iks}}\right),$$

where $r_{iks}$ is the same leave-one-out importance ratio in (10).

In comparison, BMA is fully Bayesian under assumption (13) and correct model specification. In many approximate inferences, $p_k(\cdot|y)$ is underdispersed and BMA loses mass; Even when using multi-chain MCMC, $\Theta_k$ are often duplicate (without clustering) or overlapped. As a result, BMA is sensitive to initialization and priors. Furthermore, Yao et al. (2018a) noted that BMA and pseudo-BMA can perform disastrously when there are many similar weak models in the list of candidate models. Similiary, BMA, importance sampling, and pseudo-BMA overweight "bad" modes when

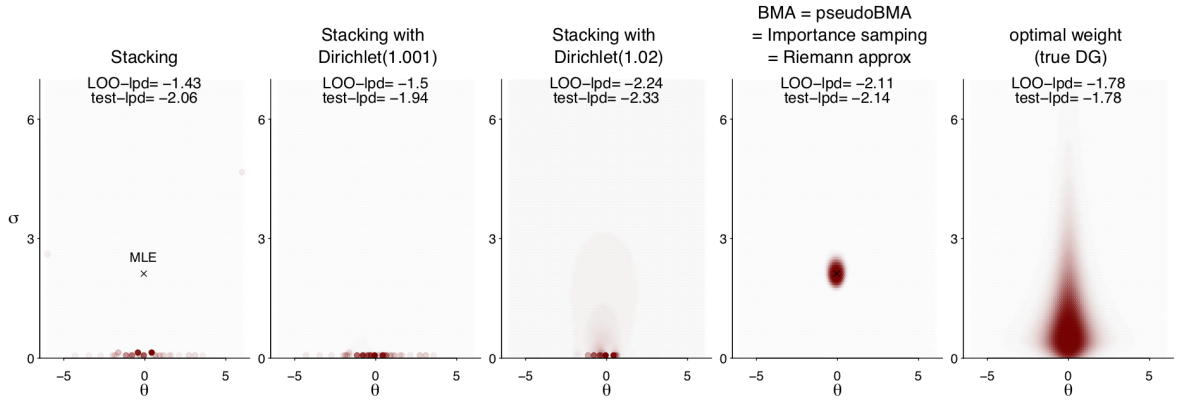



Figure 4: *We generate random $\theta$ and $\sigma$ and further $y$ from normal$(\theta, \sigma)$, and stack $9 \times 10^4$ points of $(\theta, \sigma)$ from a uniform grid. Stacking overfits due to the non-identification and finite sample size $n$. BMA, pseudo BMA, and importance sampling are identical and overconfidently concentrate.*

they are oversampled. As discussed by Geyer (1992) a simple unweighted average over chains helps when the starting distribution is close to the target density and chains mix slowly—the scenario in which other naive methods will work, too. In contrast, stacking is invariant to chain duplication and is less sensitive to initial values.

### 3.4. Understanding "overfitting"

Overfitting is a combination of model and inference. Given a model, it is possible some regions of posterior distribution more significantly overfits the data than others. Therefore, in (7) we use leave-one-out log predictive densities (loo lpd) to evaluate the expected log predictive density (elpd) for each chain even though they come from the same model.

For a fixed $K$ and weights $w$, in the limit when $n \to \infty$, the mean loo lpd of aggregated chains: $n^{-1} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p_k(y_i|y_{-i})$ converges to elpd: $\mathrm{E}_{\tilde{y} \sim p_{\text{true}}} \log \sum_{k=1}^{K} w_k p_k(\tilde{y}|\theta) p_k(\theta|y) d\theta$ (see Theorem 1 for a rigorous statement) . However, stacking uses the data twice, once in the parallel sampling and once in the aggregation of chains. In the aggregation step, we optimize over weights $w$. If both $n$ and $K$ go to infinity, the aggregated mean loo lpd is no longer an asymptotically unbiased or consistent estimate of elpd.

Figure 4 constructs an extreme example. The data $\{y_i\}_{i=1}^{100}$ are generated from normal$(\theta_i, \sigma_i)$, where $\theta_i$ and $\sigma_i > 0$ are generated from a 2-dimensional (half) Student-$t$ distribution centered at 0 and 2 respectively (see the last column).

Now to fit the model $y_i \sim$ iid normal$(\theta, \sigma), i = 1, 2, \ldots, 100$, we draw $K = 9 \times 10^4$ samples of $(\theta, \sigma)$ from uniform$(-7, 7) \times (0, 7)$. We view these as $K$ chains, with one iteration per chain, and compute stacking weights. We evaluate the leave-one-out and test data lpd (using holdout data of size 1000) of stacking, Bayesian posterior, and the true data generating process.

In this setting BMA, pseudo-BMA, importance sampling using a uniform proposal, and Riemann approximation are the same method, all over-concentrated at the posterior mode. Overfitting is revealed by a large gap between leave-one-out lpd and test data lpd. What's worse, the overconfidence of exact Bayesian inference and BMA will be amplified by a larger $n$. In contrast, stacking overfits because of nonidentification, and is asymptotically optimal for a fixed $K$ and $n \to \infty$. A Dirichlet prior with small $\lambda$ reduces overfitting (Figure 5) in stacking. It is in agreement with our recommendation in Section 3.2. This example also suggests we can use stacking to re-aggregate a

*unimodal* posterior distribution and achieve better prediction than from exact Bayesian inference.

### 3.5. Related work

*Scalable MCMC.* Bayesian inference can be more scalable in advent of parallel distributed computation. Various subsampling methods have been introduced that distribute data batches to parallel nodes and aggregate the resulting inference (Huang and Gelman, 2005; Welling and Teh, 2011; Angelino et al., 2016; Mesquita et al., 2019; Quiroz et al., 2019). These methods typically rely on certain approximations to rescale the subsampled posteriors.

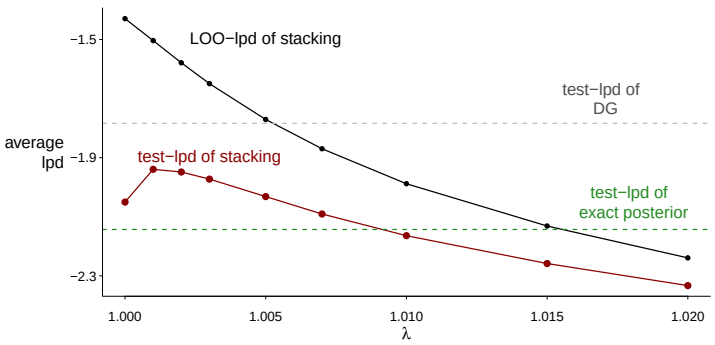



Figure 5: *Overfitting is revealed the gap between leave-one-out and test data lpd. A Dirichlet prior with a small $\lambda$ reduces overfitting in stacking, even better than the exact Bayesian posterior.*

Our approach is a divide-and-conquer strategy. It allows embarrassingly parallelization and eliminates between-chain communication, which often dominates the budget of parallel computations (Scott et al., 2016). Arguably, stacking does not speed up at all if the posterior is unimodal and tail-log-concave, when usual HMC mixes fast. We are aiming here for complicated models and pathological posterior geometry. We believe that the bottleneck of modern Bayesian computation is sometimes not the input dimension, but the slow mixing rate arising from awkward geometry of metastable distributions.

*Multiple starting points.* Gelman and Rubin (1992) used multiple sequences and importance resampling to approximate the posterior distribution, where each individual chain was iteratively constructed from a locally Student-$t$ approximation at posterior mode. However, a poor initial point can still slow convergence (Geyer, 1992) because of the use of importance sampling. In our approach, we are less concerned about starting points and only prefer it to be overdispersed.

Raftery and Lewis (1992a,b) suggested to abandon poor initial points coming with slow convergence rate and high autocorrelation by restarting. In the context of multimodality, it is hard to tell if this represents a poor initialization (that sits near the boundary of an attraction region) or a bad mode. A restart may lose the chance to explore some posterior regions.

Our convergence criteria in Section 3.2 are similar to the early approaches on stochastic optimization stopping rules following the capture-recapture model (Good, 1953; Robbins, 1968; Finch et al., 1989). Those analyses were focused on the convergence in parameter space, while ours are directly targeted at the outcome space and are thereby more applicable to models with a large number of disjoint but functionally identical modes.

*Approximate inference using mixtures.* Although our narrative has been focused on MCMC sampling, stacking can be applied to multiple runs of approximate inference; see examples in Section 5.4. Using mixture distributions to enrich the expressiveness of variational Bayes is not new. Earlier works have used mixture of mean-field approximations to match the posterior (Bishop et al., 1998; Jaakkola and Jordan, 1998; Gershman et al., 2012; Ranganath et al., 2016; Gal and Ghahramani, 2016; Miller et al., 2017; Chang et al., 2019). However, a direct application of mixture variational methods can be prohibitively expensive in large models, and weights are often fixed to ease the cost. Stacking does

not need to specify either the parametric form of the mixture or the number of mixture components, both of which adapt to data and prevent extra model misspecification.

## 4. Asymptotic analysis in a theoretical example

In this section, we analyze the asymptotic behavior of stacking. We first prove that in general chain-stacking is no worse than chain-picking. Then we derive a closed-form solution in a theoretical example to show that with model-misspecification and multimodal posterior, chain-stacking can be predictively better than the exact posterior inference.

### 4.1. Optimality of the stacked predictive distribution

The stacking weights are *not* the same as posterior masses of each mode. Even asymptotically, minimizing cross validation errors is different from integrating the target distribution. Theorem 1 affirms that the stacked inference is optimal from a predictive paradigm—it asymptotically maximizes the expected log predictive densities (elpd) among all linearly weighted combination of parallel chains of form (5). Proofs are available in Appendix A.

**Theorem 1.** *Assuming we draw $S$ posterior samples in each chain from their stationary distribution $p_k$, and we approximate the leave-one-out distribution by PSIS as in* (10)*, $p^S_{k,-i}(y_i) = \sum_{s=1}^{S_k} p_k(y_i|\theta_{ks}) r_{iks} / \sum_{s=1}^{S_k} r_{iks}$, then for a fixed number of chains $K$ and a fixed weight vector $w$, when in the limit of both the size of observations $n$ and number of posterior draws $S$, under regularities conditions (see Appendix), the objective function in stacking converges to stacked elpd:*

$$\frac{1}{n}\sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p^S_{k,-i}(y_i) - \mathrm{E}_{\tilde{y}|y_{1:n}} \log \sum_{k=1}^{K} w_k p_k(\tilde{y}|y_{1:n}) \xrightarrow{L_2} 0, \quad n \to \infty,\ S \to \infty.$$

### 4.2. Cauchy example revisit: When can stacking outperform exact Bayes in the limit?

Let's revisit the Cauchy mixture example in Section 2.1 and Figure 2. We consider univariate observations $y_{1,\dots n}$ iid from the data generating process,

$$\mathrm{DG}: \ y_i \sim \mathrm{Cauchy}\left((2z_i - 1)a, 1\right), \quad z_i \sim \mathrm{Bernoulli}(p_0), \quad i = 1, 2, \dots, n.$$

In other words, $y$ is either $\mathrm{Cauchy}(a, 1)$ or $\mathrm{Cauchy}(-a, 1)$ with probabilities $p_0$ and $1-p_0$, where the location $a > 0$ and probability $p_0 \in [0.5, 1]$ are unknown constants (the $0 \le p_0 < 0.5$ counterpart is symmetric and hence omitted). We denoted the density of this data generating process by $p_{\mathrm{true}}(y)$.

We now fit $y$ with the iid Cauchy likelihood with unknown parameter $\mu$ and a prior $p_0(\mu)$ that has full support on $\mathbb{R}$,

$$\mathrm{Model}: \ y_i \sim \mathrm{Cauchy}(\mu, 1), \quad \mu \sim p_0(\mu), \quad \mu \in \mathbb{R}\,.$$

In particular, the data generating process can be expressed from this model if an inference $\theta$ is given by a mixture of two points,

$$\text{express DG in Model}: \ \mu \sim p_0\delta(a) + (1 - p_0)\delta(-a).$$

The following theorems characterize the behavior of modes and the concentration of exact full Bayesian inference in the limit of large $n$. Proofs and related lemmas appear in Appendix A.

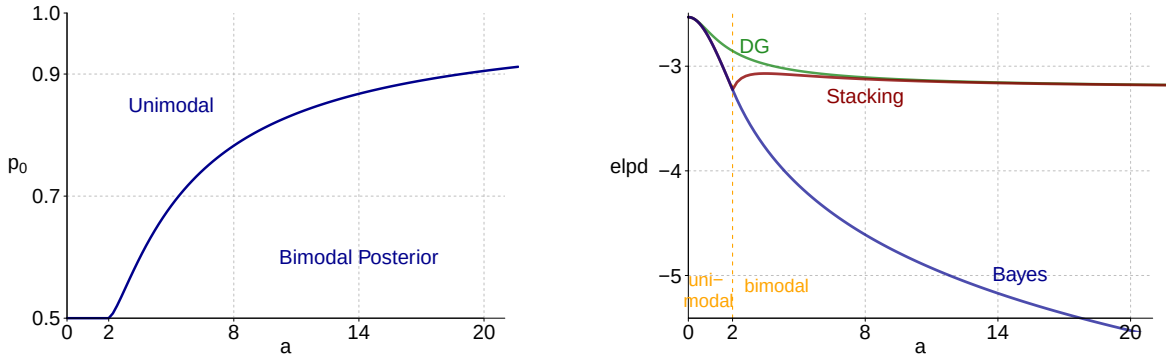



Figure 6: *Left: the deterministic function $\xi(a)$. For any $a > 2$ the posterior is bimodal with a large $n$ if and only if $p_0 < \xi(a)$. Right: the elpd of the true data generating process and the asymptotic ($n \to \infty$) elpd of full Bayes and multi-chain stacking at $p_0 = 0.5$. When $p_0 = 0.5$, $a < 2$ the posterior is unimodally spiked at 0, and stacking is identical to Bayes.*

**Theorem 2.** *We construct a deterministic function $\xi(a)$ as defined in Lemma 9 and Figure 6. It is an increasing function of $a$, with $\xi(2) = 0.5$ and $\xi(\infty) = 1$. The modality of posterior density $p(\mu|y_1, \ldots, y_n)$ has a closed form determination.*

*(a) For any $a > 2$, and $p_0 \geq \xi(a)$, there exists a large $N$, such that for all $n > N$, the posterior is unimodal. The peak is near $\mu = a$ for a large $a$.*

*(b) For any $a > 2$, and $0.5 \leq p_0 < \xi(a)$, there exists a large $N$, such that for all $n > N$, the posterior is bimodal. The two local maximums are near $(-a, a)$ for a large $a$.*

*(c) For any $0 < a < 2$, there exists a large $N$, such that for all $n > N$, the posterior is unimodal with global maximum between 0 and $a$. If further $p_0 = 0.5$, the maximum is at 0.*

*(e) When $a > 2, p_0 = 0.5$ and equipped a symmetric prior $p(\mu) = p(-\mu)$, there exists a large $N$ such that, for all $n > N$, the posterior is always bimodal with two maximums, which asymptotically ($n \to \infty$) converge to $\mu = \pm\sqrt{a^2 - 4}$.*

**Theorem 3.** *(a) For any $a > 2$, and $p_0 > 0.5$, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ converges in distribution to a point mass $\delta(\gamma)$ as $n \to \infty$, where $\gamma = \gamma(p_0, a)$ depends on $p_0$ and $a$.*

*(b) For any $a > 2$, $p_0 = 0.5$, a prior that is symmetric $p(\mu) = p(-\mu)$, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ is asymptotically only charged at two points $\pm\gamma$, with a closed form expression $\gamma = \sqrt{a^2 - 4}$. More precisely, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ is almost surely concentrated at $\pm\sqrt{a^2 - 4}$ with equal probabilities $1/2$.*

*(c) Under the same condition in (b), for any $\eta > 0$, almost surely the following limits hold,*

$$\limsup_{n\to\infty} \Pr\left(\left|\mu - \sqrt{a^2-4}\right| < \eta \mid y_1, \ldots, y_n\right) = \limsup_{n\to\infty} \Pr\left(\left|\mu + \sqrt{a^2-4}\right| < \eta \mid y_1, \ldots, y_n\right) = 1$$

When $a > 2$, if $0.5 < p_0 \leq \xi(a)$, two modes $(\gamma^+, \gamma^-)$ exist, but the exact inference will asymptotically concentrate at the right mode $\gamma = \gamma^+$. Even when $p_0 = 0.5$ so that the two centers $\pm a$ are equally important in the data generating process, the exact inference would still pick one mode asymptotically, with the left and right mode having equal chances of being selected.

**Corollary 4.** *In all cases in Theorem 3, the expected log predictive density (elpd) from the exact*

*Bayesian posterior* $p(\mu|y_1,\ldots,y_n)$ *is*

$$\begin{aligned}\text{elpd}_{\text{bayes}} &= \int_{\mathbb{R}} p_{\text{true}}(\tilde{y}|p_0)\log\int_{\mathbb{R}} p(\tilde{y}|\mu)p(\mu|y_1,\ldots,y_n)d\mu d\tilde{y}\\ &\xrightarrow{n\to\infty} -\left(p_0\log\left(\pi(4+(\gamma-a)^2)\right)+(1-p_0)\log\left(\pi(4+(\gamma+a)^2)\right)\right)\\ &\overset{a \text{ is large}}{\approx} -(1-p_0)\log\left(1+a^2\right)-\log 4\pi.\end{aligned}$$

When $a>2$, and $0.5\le p_0\le\xi(a)$, the two modes $(\gamma^+,\gamma^-)$ are detectable from multi-chain MCMC. In this case, stacking behaves better than exact Bayesian inference. Indeed, the next corollary shows that stacking approximates the data generating process in KL divergence.

**Corollary 5.** *(a) When* $n$ *is large, for any* $a>2$ *and* $0.5<p_0<\xi(a)$*, both modes* $\gamma^-$ $\gamma^+$ *receive asymptotically nonzero weights, and the elpd of the stacking average,*

$$\text{elpd}_{\text{stacking}} = \int_{\mathbb{R}} p_{\text{true}}(\tilde{y}|p_0)\log\int_{\mathbb{R}} p(\tilde{y}|\mu)p_{\text{stacking}}(\mu|y_1,\ldots,y_n)d\mu d\tilde{y},$$

*is strictly larger than* $\text{elpd}_{\text{bayes}}$*.*

*(b) When* $a$ *is large, stacking weights for* $(\gamma^-,\gamma^+)$ *are asymptotically close to* $1-p_0$ *and* $p_0$*. Consequently, the stacked posterior predictive distribution approximates the data generating process,*

$$\text{KL}\left(p_{\text{true}}(\cdot),\ \int_{\mathbb{R}} p(\cdot|\mu)p_{\text{stacking}}(\mu|y_1,\ldots,y_n)d\mu\right)\gtrapprox 0,\quad \text{when } n\to\infty, a \text{ is fixed and large.}$$

When $n$ is large, for $a>2, p_0=0.5$, the stacking weights for two modes $\pm\sqrt{a^2-4}$ are asymptotically equally 0.5. We analytically evaluate the elpd under the true data generating process, the asymptotic ($n\to\infty$) elpd of full Bayes, and multiple chain stacking in the right panel of Figure 6. Stacking is predictively superior to the full Bayes. The elpd difference between the data generating process and stacking vanishes for a large $a$, implying the KL divergence between them approaches 0.

Lastly, our Cauchy example at $p_0=0.5$ might remind readers of the one constructed by Diaconis and Freedman (1986a,b). They used a Dirichlet prior with the parameter measure $\text{Cauchy}(\mu,1)$ to fit observations essentially coming from $y\sim 0.5\delta(a)+0.5\delta(-a)$ with $a>1$. The resulting Bayesian posterior of $\mu$ is concentrated at $\pm\sqrt{a^2-1}$. However, instead emphasizing the inconsistency of this Bayesian procedure, we use our example to praise stacking: it approximates the *true* data generating process given an *misspecified* model, *inconsistent* Bayesian inference, and *non-mixing* samplers.

## 5. Examples

We demonstrate effectiveness of stacking by applying it to a series of challenging problems.

### 5.1. Latent Dirichlet allocation

Latent Dirichlet allocation (LDA, Blei et al., 2003) is a mixed-membership clustering model widely used in natural language processing, computer vision, and population genetics. In the model, the $j$-th document ($1\le j\le J$) is drawn from the $l$-th topic ($1\le l\le L$) with probability $\theta_{jl}$, where the topic is defined by a vector of probability distribution $\phi_l$ over the vocabulary, such that each word in the document from topic $l$ is independently drawn from a multinomial distribution with probability $\phi_l$.

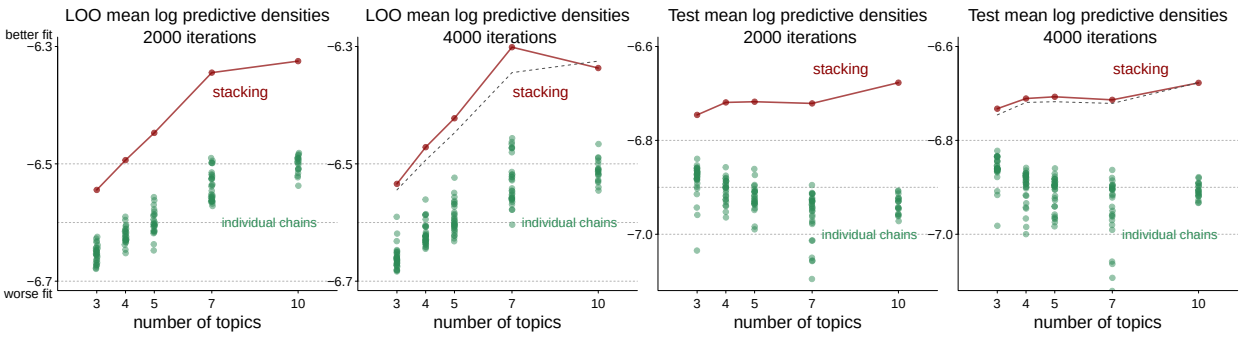



Figure 8: *The mean log predictive densities from 30 randomly initialized chains, and the stacked average of them, evaluated using both leave-one-word-out and independent test data. The number of topics $L$ in the LDA model varies from 3 to 10, and each chain contains 2000 or 4000 iterations. Individual chains do not mix, and the best of them is invariably worse than stacking.*

| chain weight | top words in the topic |
|---|---|
| 0.20 | mr, man, wickham, good, give, young, lydia |
| 0.18 | mr, man, young, bingley, collins, darcy |
| 0.13 | mr, lady, catherine, dear, great, young |
| 0.12 | wickham, elizabeth, mr, darcy, replied, hope |
| 0.09 | elizabeth, darcy, mr, sister, wickham, make |

Figure 7: *Weights of the top 5 chains in the LDA model with $L = 5$, and top words in the topic that the first paragraph belongs to computed from these 5 chains.*

Despite its popularity for data exploration, LDA suffers from computational instability as the inference may not replicate itself from either multiple runs (Mäntylä et al., 2018) or data shuffle (Agrawal et al., 2018). This confuses users as a different result is produced from each new run, and reduces the predictive power of text mining classifiers. Past literature suggests to examine and select one best fit from multiple unstable inference results subjectively or through cross validation, or to conduct extra manual tuning for hyperparameters to get rid of posterior multimodality, which however changes the original model and may further undermine classification efficiency (Tian et al., 2009; Carreño and Winbladh, 2013).

We apply an LDA topic model to texts in the novel *Pride and Prejudice*. After removing frequent and rare words, the book contains 2025 paragraphs and 32877 words, with a total unique vocabulary size of 1495. We randomly split the words in the data into 70% training and 30% test. The dimension of the parameters $\theta$ and $\phi$ grows as a function of the number of topics $L$ by $2025 \times L$ and $L \times 1495$ respectively. We place independent Dirichlet(0.1) priors on $\theta$ and $\phi$. We vary $L$ from 3 to 15, and for each fixed model we sample with Stan using 30 parallel chains initialized at random starting points with 2000 or 4000 iterations per chain.

Due to the multimodal posterior $p(\phi, \theta|y)$, individual chains do not mix after 4000 iterations. As represented by green dots in Figure 8, different chains yield different log predictive densities on test data, suggesting the multimodality is more than label-switching. Figure 7 lists, for five runs, the top words in the topic to which the first paragraph belongs.

Following our stacking approach, the 30-chain-stacked average (red line in Figure 8) improves the model fit compared with even the best of individual chains by orders of magnitude, measured in test data mean log predictive densities. Indeed, the improvement of stacking in mean lpd ($\approx 0.2$) is standardized by sample size and equivalent to roughly an $\exp(10^5)$ outperforming margin in the scale of Bayes factors. There is a mismatch between the trend of loo and test lpd, indicating the inconsistency of single chain loo-selection. This may come from (a) the non-iid nature of textual data, and (b) the parameter size is nearly the same as sample size such that loo has not reached

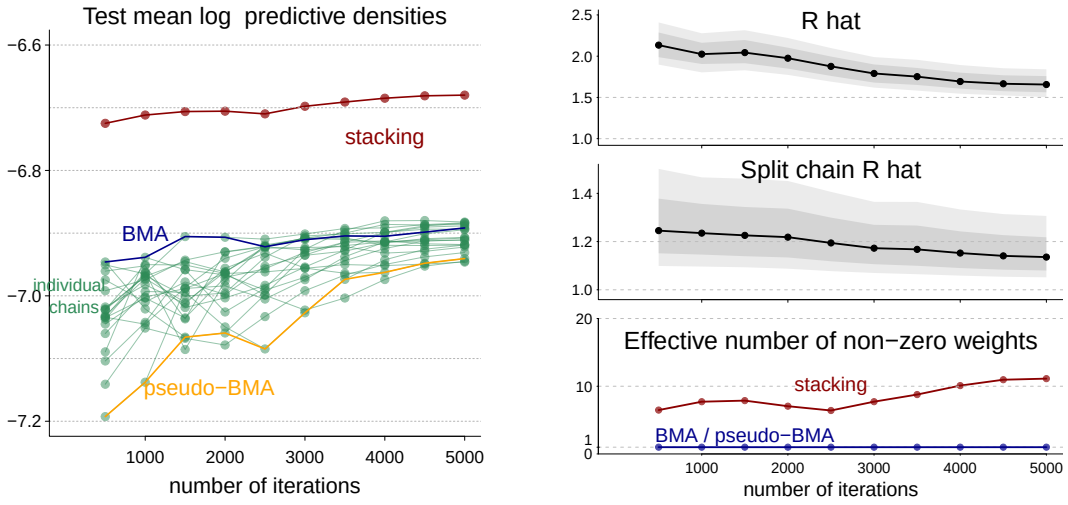



Figure 9: *Stacking benefits from early-stopped MCMC. We run LDA with $L = 10$ topics on 30 chains. As the number of iterations increase from 500 to 5000, the test lpd of individual chains increases, while the stacked average has a flatter slope, indicating we can stop early without losing much predictive power. Monitoring $\widehat{R}$ and split-chain $\widehat{R}$ of all pointwise likelihoods, we find that $\widehat{R}$ is much bigger than split-$\widehat{R}$. The bottom right shows the effective number of nonzero weights. BMA and pseudo-BMA put nearly all weight on one chain.*

its consistency territory. But even so, stacking still performs well in test data and can be combined with other predictive metric such as leave-one-document-out.

The left panel of Figure 9 shows the test data predictive performance using varying number of itearations from 500 to 5000 (with fixed number of topics $L = 10$). As the number of iterations increase, test lpd from inferences using individual chains elevates, while the stacked average has a flatter slope, indicating that we can stop earlier and stack chains without losing much predictive power, even though these chains are not completely mixed. The upper and middle right panel show median, 30% and 50% central interval of $\widehat{R}$ and split-chain $\widehat{R}$ for all pointwise likelihoods. $\widehat{R}$ is much bigger than split chain $\widehat{R}$, suggesting that the non-mixing is mostly due to lack of between-mode transitions. Given that in this problem sampling takes up to 12 hours CPU time per chain per 1000 iterations, such *early stopping of iterations* provides a remarkable opportunity to reduce computation costs. This is also manifested in Figure 8: for all $L \in [3, 10]$, individual chains perform better when per-chain iterations increase from 2000 to 4000, whereas the stacked average remains nearly unchanged (compare the red and dashed grey lines in the second and forth panel).

The bottom right panel of 9 shows the effective number of nonzero weights. In agreement with our theoretical discussion, BMA and pseudo-BMA put nearly all mass onto one chain, and in fact they often do not even select the optimal chain for the test data (left column). Accordingly, it is no surprise that stacking outperforms BMA and pseudo-BMA.

In addition to the benefit of early stopping of iterations, stacking provides an extra bonus of *early stopping of topics*. Usually, the number of topics $L$ involves manual tuning. *Stacking effectively expands the model space*. Therefore, we observe in the right two panels of Figure 8 that the stacked average is less sensitive to $L$ in test data lpd. Stacking compensates the lack of mixture components in the model through additional mixtures of posteriors during chain aggregation.

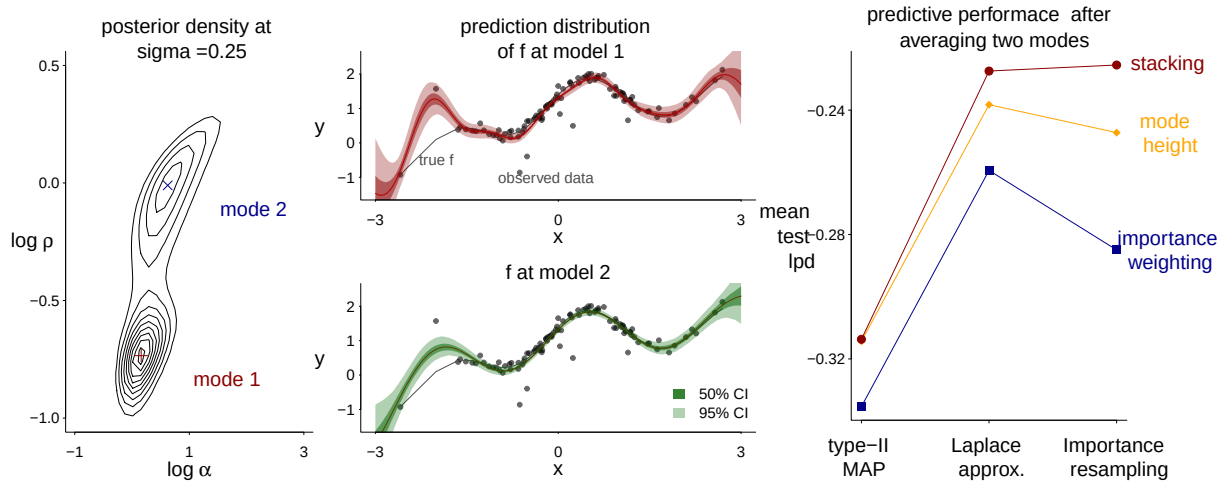



Figure 10: *The posterior distribution of hyperparameters $p(\rho, \alpha, \sigma|y)$ has at least two local modes. The left panel shows contours of the marginal posterior of $\rho$ and $\alpha$ at fixed $\sigma = 0.25$. The middle panel shows draws from the posterior predictive distribution $f|y$ at the two hyperparameter modes. We can either pick these two modes as type-II MAP or locally approximate the posterior of hyperparameters at the modes by Laplace approximation or uniform-grid importance resampling. Then the resulting modes or local approximation can be combined according to stacking, mode height, or importance weighting. The right panel shows that stacking performs the best on test data log predictive densities for all schemes.*

## 5.2. Gaussian process regression

Consider a regression problem with scalar observations $y_i = f(x_i) + \epsilon_i, i = 1, ..., n$, at input locations $X = \{x_i\}_{i=1}^n$, and $\epsilon_i$ are independent noises. We place a Gaussian process prior on latent functions $f$ with zero mean and squared exponential covariance. In the next two experiments, we apply stacking to remedy bimodality in hyperparameter *optimization*, and slow mixing in *sampling*, respectively.

*Combining modes in hyperparameter optimization.* In Gaussian process regression, posterior bimodality can occur even with a normal likelihood:

$$y_i = f(x_i) + \epsilon_i, \;\; \epsilon_i \sim \text{normal}(0, \sigma), \;\; f(x) \sim \mathcal{GP}\left(0, \alpha^2 \exp\left(-\frac{(x - x')^2}{\rho^2}\right)\right). \tag{14}$$

We use data from Neal (1998). The univariate input $x$ is distributed normal$(0, 1)$, and the corresponding outcome $y$ is also Gaussian with standard deviation 0.1. With probability 0.05, the point is considered an outlier and the standard deviation is inflated to 1. In all cases, the true mean of $y|x$ is

$$f_{\text{true}}(x) = 0.3 + 0.4x + 0.5\sin(2.7x) + 1.1/(1 + x^2). \tag{15}$$

Model (14) requires inference on $f(x_i)$ and all hyperparameters $\theta = (\alpha, \rho, \sigma)$. We integrate out all $f(x_i)$ and obtain the marginal posterior distribution

$$\log p(\theta|y) = -\frac{1}{2} y^T \left(K(X, X) + \sigma^2 I\right)^{-1} y - \frac{1}{2}\log|K(X, X) + \sigma^2 I| + \log p(\theta) + \text{constant}, \tag{16}$$

where $p(\theta)$ is the prior for which we choose an elementwise Cauchy$^+(0, 3)$.

In Neal's dataset with training sample size $n = 100$, at least two local maxima of (16) can be found. We visualize the marginal distribution of $p(\rho, \sigma|y)$ at $\sigma = 0.25$ on the leftmost of Figure 10.

Now we consider three standard mode-based approximate inference of $\theta|y$:

*a. Type-II MAP.* The value $\hat{\theta}$ that maximizes the marginal distribution (16) is called the type-II MAP estimate. Using this point estimate of hyperparameters $\theta = \hat{\theta}$, we further draw $f|\hat{\theta}, y$.

*b. Laplace approximation.* We compute $\Sigma$: the inverse of the negative Hessian matrix of (16) at the local mode $\hat{\theta}$, draw $z$ from multi-variate-normal$(0, I_3)$, and use $\theta(z) = \hat{\theta} + \mathrm{V}\Lambda^{1/2}z$ as the approximate posterior samples around the mode $\hat{\theta}$, where the matrices $\mathrm{V}, \Lambda$ are from the eigendecomposition $\Sigma = \mathrm{V}\Lambda^{1/2}\mathrm{V}^T$.

*c. Importance resampling.* Instead of standard Gaussians in the Laplace approximation, we now draw $z$ from uniform$(-4, 4)$, and then resample $z$ without replacement with probability proportional to $p\left(\theta(z)|y\right)$ and use the kept samples of $\theta(z)$ as an approximation of $p(\theta|y)$.

In the existence of two local modes $\hat{\theta}_1, \hat{\theta}_2$, we either obtain two MAPs, or two nearly nonoverlapped draws, $(\theta_{1s})_{s=1}^S, (\theta_{2s})_{s=1}^S$. We then evaluate the predictive distribution of $f$, $p_k(f|y, \theta) = \int p(f|y, \theta)q(\theta|\hat{\theta}_k)d\theta,\ k = 1, 2$, where $q(\theta|\hat{\theta}_k)$ is a delta function at the mode $\hat{\theta}_k$, or the draws from the Laplace approximation and importance resampling that is expanded at $\hat{\theta}_k$. We visualize the predictive distribution of $f$ using two local MAP estimates in the middle panel of Figure 10. The one with the smaller length scale is more wiggling and passes the training data more closely.

For each of these three mode-based inferences, we consider three strategies to combine two modes:

*a. Mode height.* We reweigh the predictive distribution of $f$ according to the height of the marginal posterior density at the the mode: $w_k \propto p(\hat{\theta}_k|y), k = 1, 2$.

*b. Importance weighting.* For approximate posterior draws $(\theta_{1s})_{s=1}^S, (\theta_{2s})_{s=1}^S$, we reweigh them proportional to the mean marginal posterior density $w_k \propto 1/S \sum_{s=1}^S p(\theta_{ks}|y)$. We choose the importance weights of two MAPs using the ones from importance resampling as it approximates the total posterior mass in the surrounding region near the mode.

*c. Stacking.* Our fast approximate loo does not apply to MAP estimation directly. Therefore, we split the data into training $y_{\text{train}}$ and validation data $y_{\text{val}}$. We first obtain either MAPs or approximate hyperparameter draws using training data and optimize their predictions on validation data. Stacking maximizes $\sum_{i=1}^{n_{\text{val}}} \log\left(\sum_{k=1}^K w_k p(y_{\text{val},i}|y_{\text{train}}, \hat{\theta}_k)\right)$ for MAPs or $\sum_{i=1}^{n_{\text{val}}} \log\left(\frac{1}{S}\sum_{k=1}^K w_k \sum_{s=1}^S p(y_{\text{val},i}|y_{\text{train}}, \theta_{ks})\right)$ for Laplace and importance resampling draws.

In the right panel of Figure 10, we evaluate these three weighting strategies by computing the mean expected log predictive density of the combined posterior distribution on hold-out test data ($n_{\text{test}} = 300$). No matter whether we are combining two point estimates or two distinct Laplace/importance resampling draws near the two modes, the stacking weights provide better predictive performance on test data.

*Combining non-mixed chains from Gaussian process regression with a Student-t likelihood.* Neal (1998) originally constructed this example in which noise $\epsilon_i$ in (14) is modeled by a $t$ distribution with mean 0, scale $\sigma$ and degrees of freedom $\nu$:

$$p(y_i|f_i, \sigma, \nu) = \frac{\Gamma((\nu + 1/)2)}{\Gamma(\nu/2)\sqrt{\nu\pi}\sigma}\left(1 + \frac{(y_i - f_i)^2}{\nu\sigma^2}\right)^{-(\nu+1)/2},\ f \sim \mathcal{GP}\left(0, \alpha^2 \exp\left(-\frac{(x - x')^2}{\rho^2}\right)\right).$$

The Student-$t$ model is robust to outlying observations but is computationally challenging, because of (a) lack of closed-form expression for $p(f|y)$, and (b) heavy-tailed posterior densities. Approximate methods exist, such as factorizing variational approximation (Tipping and Lawrence, 2005),

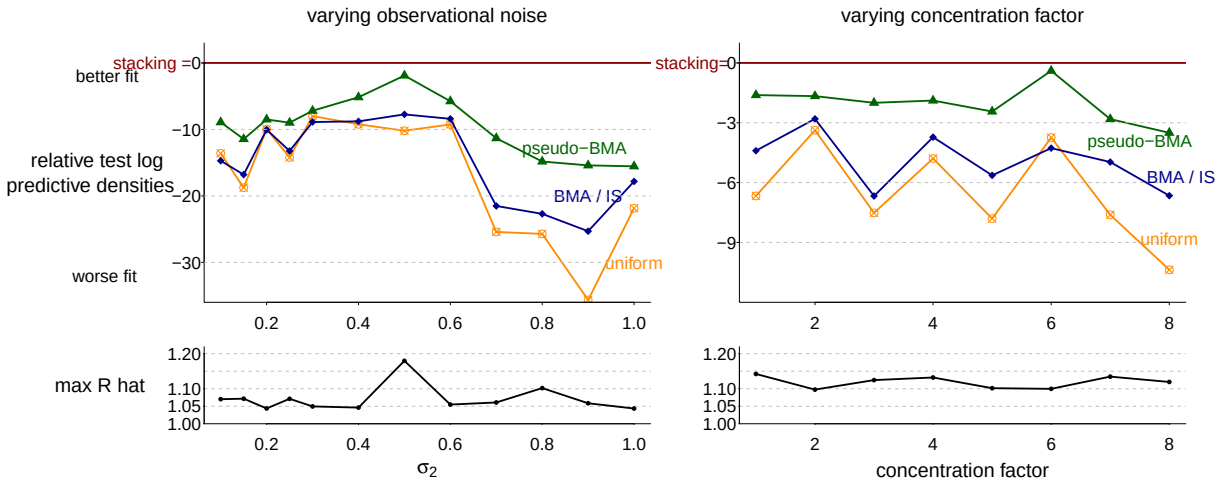



Figure 11: *Left: We fix the concentration factor $C = 5$ and vary the outlier standard deviation $\sigma_2$ from 0.1 to 1 in the data generating mechanism. Right: We fix $\sigma = 0.3$ and vary the concentration factor $C$ from 1 to 8. In each setting, we sample from the posterior distribution using 8 chains with 8000 iterations each, and combine chains using four weighting methods. We report the test log predictive densities (using $n_{\text{test}} = 300$ independent test data) of three other methods subtracting stacking, which are always negative. The lower row reports the maximum $\widehat{R}$ among all parameters.*

Laplace approximation (Vanhatalo et al., 2009), and expectation propagation (Jylänki et al., 2011), but posterior sampling remains difficult.

We generate training data $x_{1:n}$ from uniform$(-3, 3)$, and the outcome $y_i$ has the same mean in (15). $y_i$ either has standard deviation $\sigma_1 = 0.1$, or inflated to $\sigma_2 > 0.1$ with probability proportional to $\exp\left(-\left(C\frac{i-0.4n}{n}\right)^2\right)$, where $C > 0$ is a concentration factor that decides how the outliers are concentrated with each other in $x$-space. In the experiment, we vary $\sigma_2$ from 0.1 to 1 and $C$ from 1 to 8. $n_{\text{test}} = 300$ hold-out test data points $(\tilde{X}_i, \tilde{y}_i)_{i=1}^{n_{\text{test}}}$ are generated from the same mechanism.

We fix the degrees of freedom $\nu = 2$ and sample from the full posterior distribution $p(f_1, \ldots, f_n, \sigma, \alpha, \rho)$ from $K = 8$ parallel chains and 8000 iterations per chain in Stan. We draw initialization from uniform$(-10, 10)$ for unconstrained parameters and set the maximum tree depth to 5 for the NUTS algorihm. In the lower row of Figure 11, we report the maximum $\widehat{R}$ of all sampling parameters among 8 chains: clearly these do not mix in all settings.

We compare four chain-combination strategies: BMA, pseudo-BMA, uniform averaging, and stacking. After each iteration of $(\sigma, \rho, \alpha, f)$, we draw posterior predictive sample of $\tilde{f} = f(\tilde{X})$, and compute the mean test data log predictive densities. Since test performance changes in orders of magnitude under different data-generating settings, in Figure 11 we use stacking as a baseline and compare the test log predictive densities of other methods by subtracting stacking ones. In all cases, stacking outperforms other three approaches.

There are three contributors to the poor mixing in this example. First, chainwise predictions may diverge even when parameters are nearly mixed. Figure 12 display sampling results for a dataset with $n = 20, \sigma_2 = 0.6, C = 5$. In the leftmost column, all $(\sigma, \rho, \alpha, f)$ and transformed parameters have $\widehat{R} < 1.05$. But the log predictive densities are different across chains, shown in the second column (chains have have been re-ordered by test lpd). Stacking detects this difference via leave-one-out cross validation.

Second, the posterior distribution $f|y$ can be multimodal. The rightmost column of Figure 12

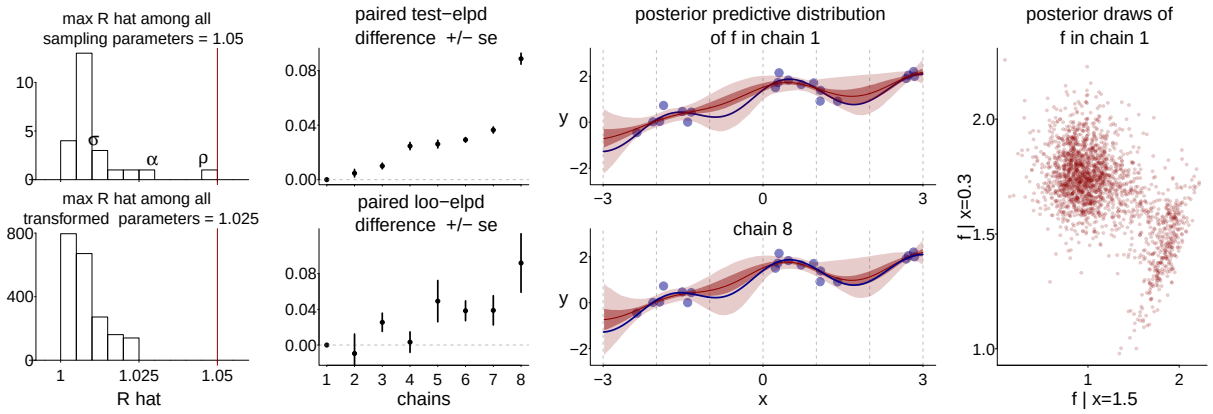



Figure 12: *In an experiment ($n = 20, \sigma_2 = 0.6, C = 5$) even when $\widehat{R}$ of all parameters are smaller than 1.05, 8 chains exhibit different predictive capabilities. The second column shows the estimated log predictive densities subtracting chain 1 and the standard error in test data or loo. Chains have been reordered by test scores. The third column shows the prediction of $f$ in chain 1 and 8. The rightmost column is the joint posterior predictive draw $f$ at $x = 1.5$ and 0.3 in chain 1.*

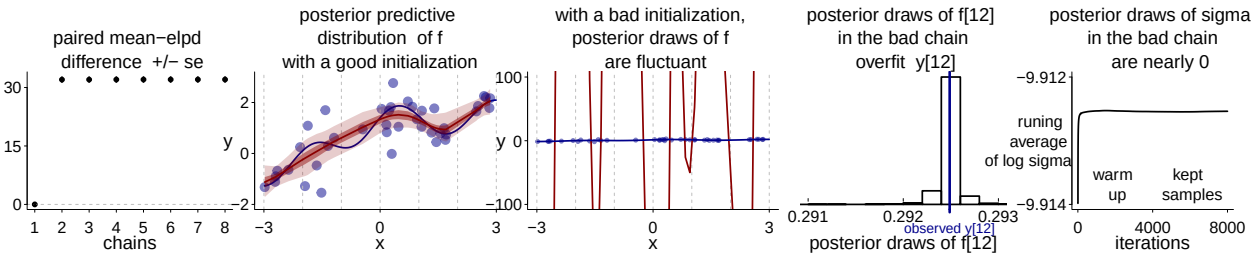



Figure 13: *In an experiment ($n = 40, \sigma_2 = 1, C = 5$), chain 1 is trapped in a bad local mode, where the posterior $f|y$ is narrow and fluctuate. It overfits observed value, and $\sigma$ is trapped near 0 among 8000 iterations. It has a low elpd on both test data and loo, hence abandoned in stacking.*

displays the joint posterior distribution of $f$ conditioning on $x = 0.3$ and 1.5, clearly bimodal. In this example, this is not a sampling concern owing to the small between-mode energy barrier, and HMC/NUTS sampler in Stan is able to move between these two modes rapidly.

Third, some chains may be trapped in bad local modes. In Figure 13, we outline the sampling result from another dataset ($n = 40, \sigma_2 = 1, C = 5$). Chain 1 is trapped in a local mode with $\sigma \approx 0$ and is unable to escape the local trap after 8000 iterations. The posterior prediction $f$ fluctuates and overfits the observations: $f_{12}|y$ is nearly a delta function at $y_{12}$. The strong overfitting of this chain leads to a low elpd on both test data and leave-one-out cross validation, hence it is abandoned by stacking.

### 5.3. Hierarchical models

*When the bimodality occurs and when reparameterization helps.* Consider observations from $J$ exchangeable groups. For simplicity we assume a balanced one-way design, with data $y_{ij}, i = 1, \ldots, N$ from groups $j$. We apply a hierarchical model with parameters $(\theta, \sigma, \mu, \tau)$,

$$\text{centered}: \quad y_{ij}|\theta, \sigma \sim \text{normal}(\theta_j, \sigma),\ \theta_j|\mu, \tau \sim \text{normal}(\mu, \tau),\ 1 \le i \le N,\ 1 \le j \le J. \tag{17}$$

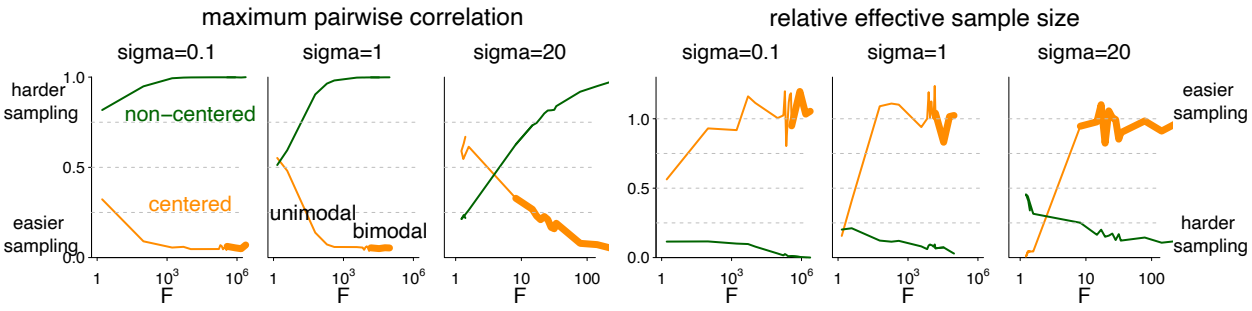



Figure 14: *We fit the hierarchical model on data simulated from by various generating process. When the between group variation is large or the within group variation is small, whose ratio is the sample F statistics, the centered parameterization is more efficient, amid less correlated posterior and large effective sample size. Counterintuitively, this is also when the posterior bimodality occurs.*

Sampling in the space of $(\theta, \sigma, \mu, \tau)$ is called *centered parameterization*. When the likelihood is not strongly informative, the prior dependence between $\tau$ and $\theta$ in (18) can produce a funnel-shaped posterior that is non-log-concave, and slow-to-mix near $\tau = 0$, due to a large entropic barrier.

Alternatively, with *non-centered parameterization*, sampling occurs in the space of $(\xi, \sigma, \mu, \tau)$ through a bijective mapping $\theta_j = \mu + \tau\xi_j$, and the model is equivalently reparameterized by

$$\text{non–centered}: \quad y_{ij} \sim \text{normal}(\mu + \tau\xi_j, \sigma),\ \xi_j \sim \text{normal}(0, 1),\ 1 \le i \le N,\ 1 \le j \le J. \tag{18}$$

When the likelihood is not strongly informative, the non-centered parameterization is preferred (Betancourt and Girolami, 2015; Gorinova et al., 2019), but when the likelihood is strongly informative, then the non-centered parameterized posterior has a funnel shape. The data informativeness can be crudely measured by the inverse of $F$-statistics (between group variance divided by within group variance). But beyond such heuristics and limited classes of models where analytic results can be applied, there is no general guidance on which parameterization to adopt.

Parallel to the slow mixing rate due to the funnel shaped posterior, the posterior in (17) can contain two modes, usually arising when the data indicate a larger between-group variance than does the prior. Liu and Hodges (2003) characterized bimodality of this model under conjugate priors in closed form.

To understand how the posterior bimodality affects sampling efficiency, in the first simulation we generate data from $J = 8$ groups and $N = 10$ observations per group. The true $\tau$ and $\sigma$ vary from 0.1 to 20, with a varying amount of $t$-distributed noise added to $\theta$. We place a conjugate inverse-gamma$(0.1, 0.1)$ prior on both $\tau^2$ and $\sigma^2$. For every realization of data, we sample from the posterior distribution in both centered and non-centered parameterization using 4000 iterations, and analytically determine whether the centered parameterization has two posterior modes.

In Figure 14, we assess the maximum absolute parameterwise correlations (left three columns), and the relative effective sample size (ESS divided by total iterations, right three columns) in posterior samples. Conforming our heuristics, when between-group variation is large and within-group variation is small, the centered parameterization is more efficient, and vice versa.

Surprisingly, in this example metastability and multimodality evolve in opposite directions. In Figure 14 we visualize the occurrence of posterior bimodality in centered parameterization by a thicker line width. When the between-group variation increases, the centered posterior eventually becomes bimodal, but sampling becomes more efficient.

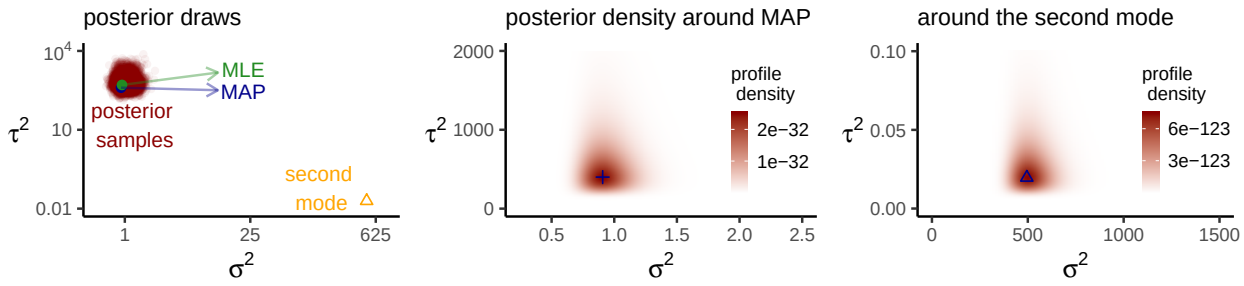



Figure 15: *A grid search finds two posterior modes when data are generated by* $\sigma = 1$ *and* $\tau = 25$. *The second mode in density and prediction ability, ignored by posterior sampling.*

How is this possible? Figure 15 presents an example where the data are generated by $\sigma = 1$ and $\tau = 25$. Both the MAP and MLE are close to the true value. A second local mode explains all variation by a large $\sigma$ (opposite to Figure 13), but it is orders of magnitude lower than the first one in posterior densities, hence ignored by sampling. That's why the centered parameterization runs smoothly in the existence of posterior bimodality. The bad mode also has a low loo elpd, so stacking assigns it zero weight when we combine modes.

*A stacked parameterization and zero-avoiding priors.* Section 5.3 leaves a few open problems: which parameterization to choose in practice, whether the sample has included all local modes, whether the ignored modes are predictively important, and if we should search for them in the first place. The bimodality analysis of Liu and Hodges (2003) applies to conjugate priors. But multimodality readily exists in hierarchical models. To be specific, when the group-level standard deviation $\tau$ has a flat prior, $\tau = 0$ is *always* a mode of the joint posterior distribution. From the modeling perspective, this mode represents complete pooling.

Given that the centered parameterization behaves like an implicit truncation and has sampling difficulty in the small $\tau$ region, we propose a stacking-based solution for reparameterizations. We run $K + 1$ chains. The first chain is complete pooling: restricting $\tau = 0$ and $\theta_j = \theta_1$. The next $K$ parallel chains are centered parameterization with a zero-avoiding prior (Chung et al., 2013) on $\tau$. Finally, we use stacking to average these $K+1$ chains. Intuitively, if $\tau \approx 0$ is predicatively important but missed by the implicitly left truncated centered parameterization, the first chain fills the hole; when $\tau \approx 0$ is incompetent, the centered sampling is boosted by circumventing the computationally intensive region $\tau \approx 0$.

To validate our proposal, we simulate data with dimensions $J = 100$ (number of groups) and $N = 20$ (observations per group). We vary the true within-group standard deviation $\sigma$ from 0.1 to 100 and add between-group noises $Bv_j$ to $\theta_j$, where $B$ is a constant scalar varying from 0 to 50, and each $v_j$ is an independent Student-$t(1)$ noise. We place a zero-avoiding prior $\tau^2 \sim$ inv-gamma$(0.1, 0.1)$. We sample one chain (3000 iterations) from the complete pooling model, eight chains each from centered and non-centered parameterization, stack the complete pooling and centered ones, and evaluate the prediction ability of the posterior inference using mean log predictive densities on $N_{\text{test}} = 300$ independent test data in each group. In the upper row of Figure 16, we place the stacking average as the baseline and extract its elpd from other parameterizations. The complete pooling model almost always has lpd so low that it does not even appear on the graph, and should never be used by itself. Instead of picking between the centered or and non-centered parameterization, the stacking estimate (red line) always has a larger log predictive density than the

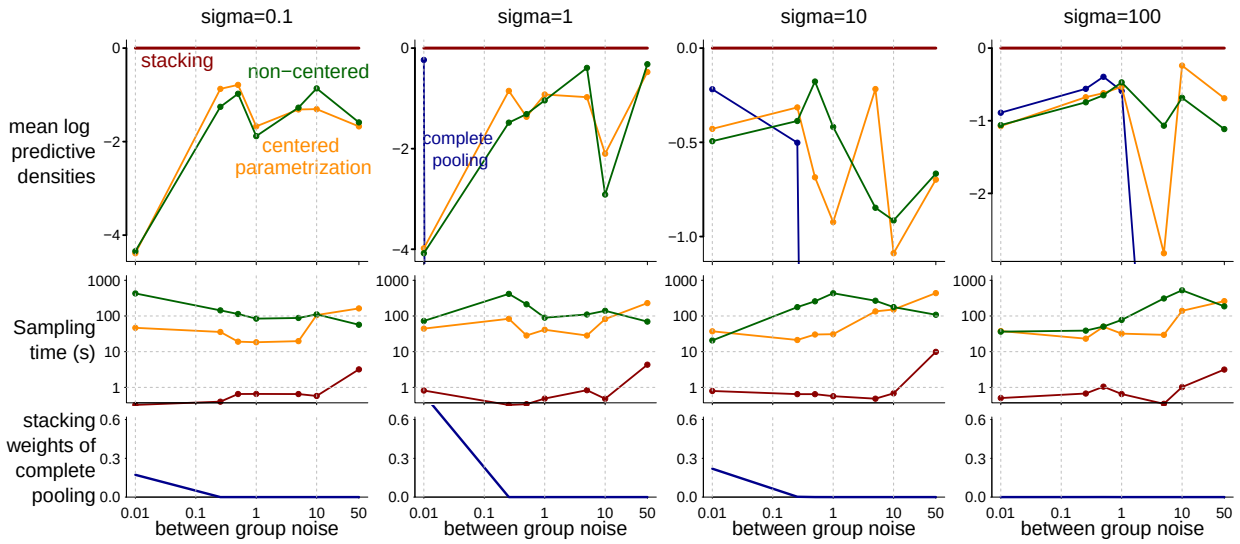



Figure 16: *We stack 8 parallel centered-parameterized chains and 1 complete pooling chain. The stacked average always has better test data performance than both centered and non-centered ones in all data configurations. The additional computation cost of stacking is minimal. Even when the complete pooling chain receives zero weight, stacking still helps remedy slow mixing of remaining chains and achieves better elpd than uniform mixing.*

best of them. Such advantage is achieved at a negligible computation cost compared with sampling time (middle row). These patterns are robust under different prior and data configurations, and we have omitted similar outcomes when we tune $J$ from 10 to 500 and for other zero-avoiding priors.

Lastly, in this example, stacking remedies both the incapability to sample in small $\tau$ regions, and between-chain-non-mixing in the centered parameterization. The last row of Figure 16 monitors stacking weights for the complete pooling chain. Even when it receives zero weight, the stack-weighted draws from centered parameterization are better than the uniform mixing of eight chains.

### 5.4. Stacking multi-run variational inference in a horseshoe regression

The regularized horseshoe prior (Piironen and Vehtari, 2017a,b) is an effective tool for Bayesian sparse regression. Denoting $y_{1:n}$ as a binary outcome and $x_{n\times D}$ as predictors, the logistic regression with a regularized horseshoe prior is,

$$\Pr(y_i=1)=\text{logit}^{-1}(\beta_0+\sum_{d=1}^{D}\beta_d x_{id}),\ i=1,\dots,n,\quad \beta_d|\tau,\lambda,c\sim\text{normal}\left(0,\frac{\tau c\lambda_d}{(c^2+\tau^2\lambda_d^2)^{1/2}}\right),$$
$$c^2\sim\text{Inv–Gamma}(\alpha,\beta),\quad \tau\sim\text{Cauchy}^+(0,1),\quad \lambda_d\sim\text{Cauchy}^+(0,1),\ d=1,\dots,D.$$

Sampling from the exact posterior $p(\beta,\tau,c,\lambda|y)$ is computationally intensive and not scalable to big data. Unfortunately, mean-field variational inference (VI, Blei et al., 2017) which optimizes over the best mean-field Gaussian approximation to the joint posterior measured in KL divergence, behaves poorly on horseshoe regression. In particular, VI cannot capture the posterior multimodality (see examples in Yao et al., 2018b), which is a key aspect of the regularized horseshoe, a continuous counterpart of the spike-and-slab prior.

In general, the optimization problem in variational inference is not convex. Equipped with stochastic gradient descent, multiple runs of variational inference can return entirely different pa-

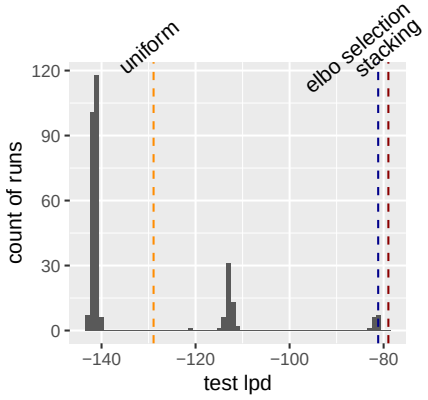



Figure 17: *Test data elpd among 300 runs of variational inference using synthetic data. Stacking over 300 runs achieves better prediction than any single run and also outperforms uniform mixing.*

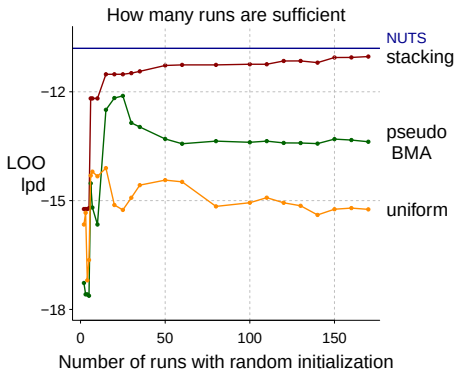



Figure 18: *Monitoring convergence for the leukemia example. Pseudo-BMA and uniform weighting have lower loo-lpd with more runs. Stacking is stable after 10 runs and gives a fit close to NUTS while requiring much less computation time.*

rameters. The common practice is to either select the best run based on the evidence lower bound (elbo) or test data performance. In the presence of posterior multimodality, the best that a normal approximation can do is to pick one mode, which in particular undermines the advantage of altering between no pooling and complete pooling of horseshoe regressions.

In next two experiments, we apply stacking to multiple runs of automatic variational inference (ADVI, Kucukelbir et al., 2017). In the $k$-th run, $k = 1, \ldots, K$, we obtain $S$ posterior approximation draws $\theta_{k1}, \ldots, \theta_{kS}$. We treat these as posterior samples, obtain the leave-one-out predictive densities, and use stacking to derive the optimal combination weights of all $K$ runs.

*Synthetic data.* We first generate data from the model, $\Pr(y_i = 1) = \text{logit}^{-1}\left(\sum_{d=1}^{400} \beta_d x_{id}\right)$, $i = 1, \ldots, n = 40$. The design matrix $X$ is normally distributed with shared featurewise components to increase linear dependence. Of the 400 predictors, only the first three have nonzero coefficients $\beta_{1,2,3} = (3, 2, 1)$; this is the example discussed in Van Der Pas et al. (2014) and Piironen and Vehtari (2017b). We assess the model prediction on hold-out test data with size $n_{\text{test}} = 200$.

Figure 17 presents the test data log predictive densities among 300 ADVI runs with $10^5$ stochastic gradient descent iterations each run. Stacking achieves better prediction than any single run and uniform mixing. Most of the runs have a low lpd, making the uniform reweighing undesired. The elbo selection selects the second best run (in test data lpd).

*Leukemia classification.* We consider regularized horseshoe logistic regression on the leukemia classification dataset. It contains 72 patients $y_i = 0$ or $1, 1 \leq i \leq 72$, and a large set of predictors consisting of 7128 gene features $x_{id}, 1 \leq d \leq 7128$.

In this section, we view HMC/NUTS sampling in Stan as the gold standard, which is slow (several hours per 1000 iterations) but mixes well in this dataset (Piironen and Vehtari, 2017b). We push the limit of variational inference by averaging 200 parallel ADVI runs with $10^5$ stochastic gradient descent iterations, where each run takes less than one minute, but the approximation from any VI run is inaccurate.

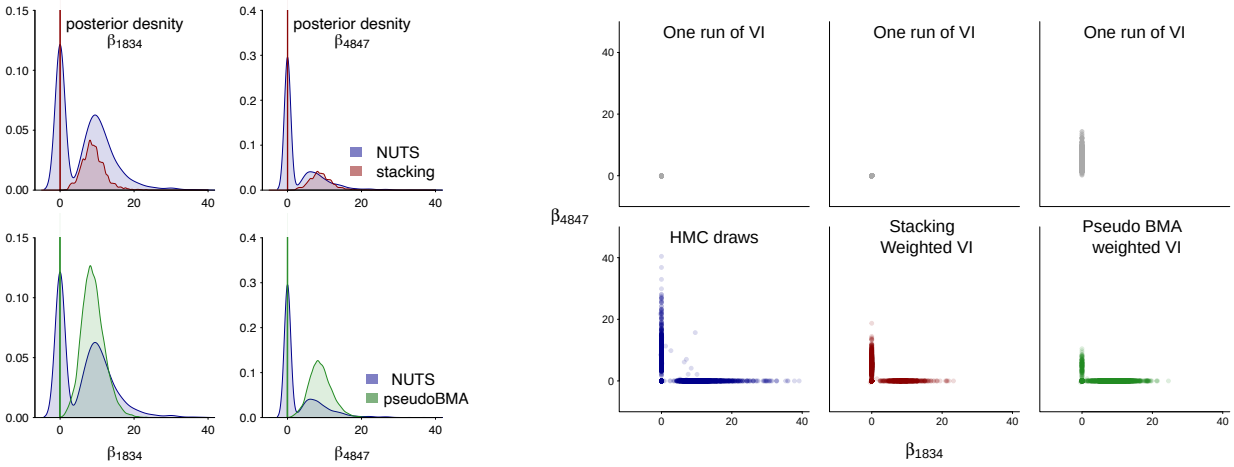



Figure 19: *The stacked VI posterior distribution matches HMC/NUTS draws reasonably well both marginally (left panel) and jointly (right panel) for the leukemia example, although individual runs are inaccurate. The graph displays two parameters $\beta_{1834}$ and $\beta_{4847}$ that have the largest absolute posterior means.*

Figure 18 displays the leave-one-out log predictive density of the combined distribution as a function of the number of runs to average, as previously described in (8). For stacking, there is a first jump at 5 runs, a second jump at roughly 10 runs, and then almost stable afterwards. For pseudo-BMA and uniform weighting, the loo elpd is worse with more runs, because VI is sensitive to initialization, and pseudo-BMA, BMA, and uniform weighting are sensitive to weak but duplicated runs (Yao et al., 2018a). Stacking achieves a much better leave-one-out lpd than all individual chains and other weighting methods, nearly comparable to HMC/NUTS. There is one caveat: because of the optimization procedure, the loo lpd of stacking likely overestimates its expected lpd.

To better evaluate how close the final inference is to the exact sampling, we visualize the stacked posterior VI draws of $\beta_{1834}$ and $\beta_{4847}$ (we pick these two variables which in our computation had the largest absolute posterior means as estimated with HMC/NUTS) in Figure 19. Stacked VI approximates the posterior well both marginally (left two columns) and jointly (right three columns). It captures the main shape: a spike concentrated at 0 and a slab part—a true spike in the stacked distribution might be even more appealing for interpretation. We also plot the joint distributions from three individual runs, all distant from the truth. Stacking recombines these individual mean-field normal approximations, the mixture of enough of which can approximate any continuous distribution.

Finally as a caveat, the PSIS-loo approximation is applicable to VI under assumption that each VI optimum $q_k$ locally matches the exact posterior $p$ (up to a normalization constant $c_k$):

$$\exists \Theta_k \subset \Theta,\ q_k(\Theta_k) \approx 1, \quad s.t.\ \forall \theta \in \Theta_k,\ q_k(\theta) \approx c_k p(\theta|y), \tag{19}$$

which can be assessed by diagnostics in Yao et al. (2018b). In this example, it is implausible that (19) would exactly hold, but PSIS-loo still yields useful results. Alternatively, we can circumvent assumption (19), replace loo by a training-validation split, and perform stacking on the validation set, as shown in Section 5.2.

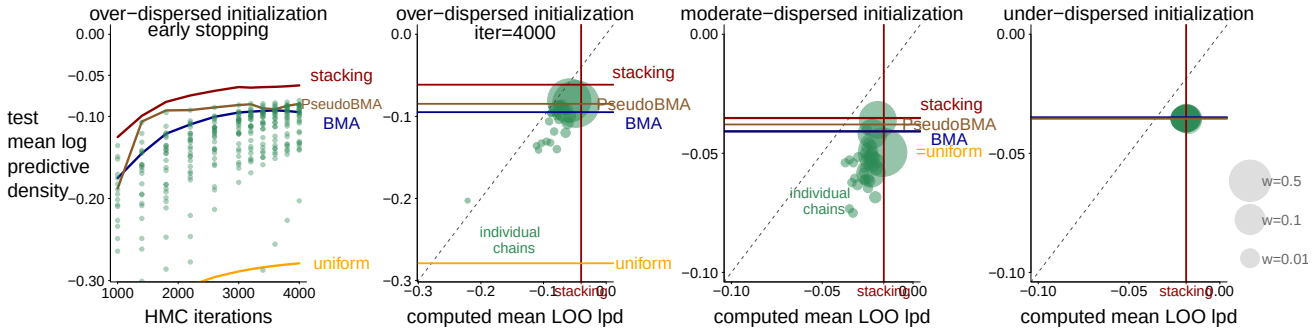



Figure 20: *(1): The test mean log predictive densities of early stopped chains. Stacking performs consistently better than single chains or other weighting methods. (2–4): The mean leave-one-out and test data log predictive densities of 50 individual chains (green dots), their stacking weights (size of the dot), and the test mean lpd of from four weighting strategies when fitting 40-hidden node neural network on MNIST. There were 4000 iterations per chain, and network parameters are initialized from uniform*$(-50, 50)$, $(-5, 5)$, *and* $(-0.001, 0.001)$, *respectively. Some individual changes in the overdispersed setting are out of lower-range and not shown.*

### 5.5. Bayesian neural networks

The posterior distribution of neural network parameters is well known to be often multimodal. We demonstrate stacking for such an example using the MNIST dataset, a collection of images of handwritten digits that are to be classified into their true labels, 0–9. We consider a two-layer neural network with tanh activation function:

$$\Pr(y_i = k) \propto \exp(\sum_{j=1}^{J} h_{ij}\beta_{jk} + \phi_k), \quad h_{ij} = \tanh(\sum_{m=1}^{M} x_{im}\alpha_{mj}), \quad i = 1, \ldots, n,\ k = 0, \ldots, 9.$$

where $n$ is the sample size, $J$ is the number of hidden nodes, and $M = 784$ is the input dimension. Making scalable Bayesian inference remains an open computation problem and beyond the scope of this paper. To simplify the problem while keeping the pathological multimodality in the posterior distribution, we subsample $n = 1000$ training data from the labels $y = 1$ and 2 and set the number of hidden nodes $J = 40$. We use hierarchical priors, $\alpha \sim \text{normal}(0,\sigma_\alpha)$, $\beta \sim \text{normal}(0,\sigma_\beta)$, $\sigma_\alpha, \sigma_\beta \sim \text{normal}^+(0, 3)$. Switching the order of hidden nodes does not change the predictive density. We eliminate the combinatoric non-identification in all other experiments in this section by constraining the order of $\beta$: $\beta_1 \geq \beta_2 \ldots, \geq \beta_J$.

We sample from the posterior distribution $p(\phi, \beta, \alpha|y, x)$ using 50 parallel HMC/NUTS chains in Stan. In the right three panels in Figure 20, we present the posterior predictive performance of individual chains and combinations, evaluated by the mean log predictive densities on both leave-one-out data and an test data with $n_{\text{test}} = 2167$. The test score standard deviation is negligible. The initial values of unconstrained parameters in panels 2–4 are drawn from uniform$(-50, 50)$, $(-5, 5)$, and $(-0.001, 0.001)$, respectively. Each green dot stands for one chain, and the size of the dot reflects the chain weight in stacking (we rescale the size proportional to $w_{\text{stacking}}^{1/5}$ to manifest extremely small weights, see the legend on the right). Under an overdispersed initialization, the posterior inferences considerably diverge, and uniform weighting is jeopardized by "unlucky" chains, while stacking is not affected by a large number of bad chains. The PSIS-loo approximation does not accurately estimate the test performance (detected by large $\hat{k}$ diagnostics), but stacking still

outperforms all other weighting strategies. Under the $(-0.001, 0.001)$ initialization, all 50 chains are essentially identical, and there is no gain from reweighing. In this experiment, a carefully tuned underdispersed initialization is the most efficient. However, choosing optimal starting values in general models remains difficult, whereas stacking is less sensitive to the initialization.

Early stopping is a commonly used ad hoc regularization method in neural networks (Vehtari et al., 2000). The leftmost column in Figure 20 demonstrates that we can stack early stopped chains to achieve a prediction-power and computation-cost tradeoff. In the setting of 40 hidden nodes and overdispersed initialization, stacking is strictly better than the best single chain however early we stop. Stacking with 1500 HMC iterations is better than the best chain at iteration 4000. BMA and pseudo-BMA effectively choose just a single chain, and they and select the wrong chains at times. Uniform weighting is again the worst due to its sensitivity to bad initializations.

The existing literature on neural net ensembles advocate to *uniformly* average over all ensembles constructed by local MAPs found through stochastic gradient descent (Lakshminarayanan et al., 2017), bootstrap resampling (Osband et al., 2019), or varying priors (Pearce et al., 2020). Our experimental results show that inference from uniform weights is highly sensitive to starting points and can be especially disappointing under an overdispersed initialization. The approximate loo-based stacking sheds light on the benefit of post-inference multi-chain-reweighing in modern deeper neural networks. The additional optimization cost is tiny compared to the cost of model training. We leave question of scalability to modern Bayesian deep learning models to future investigation.

## 6. Discussion

### 6.1. Learn better epistemic uncertainty to expiate aleatoric misspecification

Uncertainty comes into inference and prediction through two sources: (a) due to finite amount of data, we learn the *epistemic* uncertainty of unknown parameter $\theta$ through the posterior distribution $p(\theta|y)$, and (b) due to either the stochastic nature of real world, even when $\theta$ is known, we represent the *aleatoric* uncertainty through the probabilistic forecast of next unseen outcome as $p(\tilde{y}|\theta, y)$. The final probabilistic prediction contains both of them via $p(\tilde{y}|y) = \int p(\tilde{y}|\theta, y)p(\theta|y)d\theta$.

Given a model, the epistemic uncertainty is mathematically well-defined though Bayesian inference, but will only be optimal under the true model and when averaging over the prior distribution. By being open-minded to model misspecification, the optimization (2) searches for the "best" probabilistic inference and uncertainty quantification with respect to a given utility function.

Our paper calls attention to post processing and re-calibrating Bayesian epistemic uncertainty. Stacking reweighs separated component in the posterior density, while in general we can consider other transformations of the posterior draws such as location–scale shift, mixtures, and convolutions.

Bayesian inference is known to be poorly-calibrated under model misspecification (Gelman and Shalizi, 2013). In the context of model-selection and averaging, the marginal-likelihood-based "full-Bayes" approach produces over-confident prediction when none of the model is true (Clarke, 2003; Wong and Clarke, 2004; Clyde and Iversen, 2013; Yao et al., 2018a; Yang and Zhu, 2018; Yao, 2019; Oelrich et al., 2020), and therefore is not Bayes optimal (Le and Clarke, 2017).

The suboptimality of Bayesian posteriors does not mean we think Bayesian inference is wrong, but it does imply that there are tensions between a reckless application of Bayes rule under the wrong model and the Bayesian decision theory, and more generally, between Bayesian inference and Bayesian workflow. In the words of Gelman and Yao (2021), such tensions can only be resolved by considering Bayesian logic as a tool, a way of revealing inevitable misfits and incoherences in our model assumptions, rather than as an end in itself.

### 6.2. Chain-stacking as nonparametric inference

A parametric model $y|\theta \sim p(y|\theta),\ \theta \sim p(\theta), \theta \in \Theta$ restricts the data generating mechanisms, which a priori are only supported at $\{p(y|\theta) \mid \theta \in \Theta\}$. The nonparametric Bayesian approach allows more flexible modeling that assigns a prior on a larger space but is subject to other challenges of prior constriction and computation.

Multi-chain stacking enriches Bayesian inference in the same way that nonparametric priors make models flexible. By allowing inference to depart from Bayes' rule, we identify and correct for model misspecification through stacked inference. The nonparametric aspect of stacking is also reflected by the unspecified number of mixture components, as conceptually an infinite mixture of simple distributions can approximate any continuous distribution. Of course, stacking cannot resolve all model misspecification since it uses parametric inferences as building blocks.

### 6.3. Chain-stacking as diagnostics

Besides improving on model predictions, multi-chain stacking can be used as a diagnostic tool.

First, uniformly identical stacking weight implies that parallel chains have mixed in overall predictive performance. In comparison, $\widehat{R}$ is only marginally diagnostic on parameters. One chain can be slightly but constantly better than the other, and such difference will be accumulated across all points. For example, Figure 21 compares the pointwise predictive distributions of chains 1 and 8 in Figure 12. For each point $\tilde{x}_i$, we compute the parameter estimation $\mathrm{E}(f|y, X, \tilde{x}_i) = \int f p(f|y, X, \tilde{x}_i) df d\sigma$ and the log predictive density $\log p(\tilde{y}_i|\tilde{x}_i) = \log \int p(\tilde{y}_i|f, \sigma) p(f|y, X, \tilde{x}_i) df d\sigma$, both using Monte Carlo draws from chain 1 and 8. We compare two Monte Carlo integrals by a $t$ test, and adjust the sample autocorrelation by plugging in the estimated effective sample size (ESS) of the draws. In 84% of the test points $\tilde{x}_i$ uniformly distributed on $(-3, 3)$, chain 8 has a higher pointwise predictive density than chain 1.

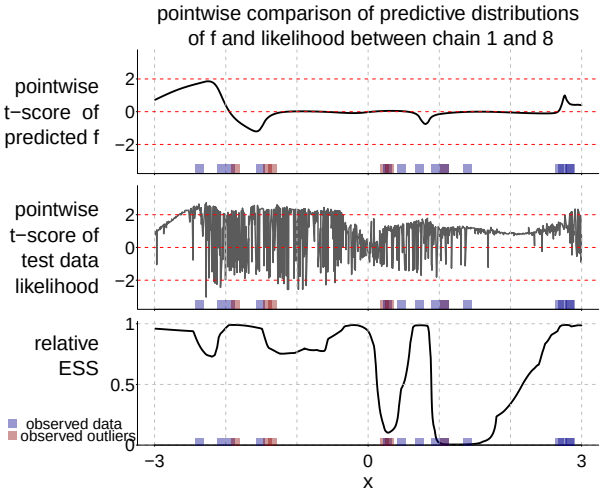



Figure 21: *Pointwise comparison of* $\mathrm{E}[\tilde{f}|y, \tilde{x}]$ *and log predictive densities* $\log p(\tilde{y}|\tilde{x}, y)$ *at each* $\tilde{x}$ *from chain 1 and 8 in Figure 12. We compare two chains by a pointwise* $t$ *test, where we plug in the effective sample size.*

The distribution of log predictive densities often has a thicker tail than the distribution of individual parameters, thereby having a larger Monte Carlo variation and slower mixing rate. Even when all parameters are normally distributed in the posterior, the posterior log likelihood can be $\chi^2$ distributed (see examples in Yao, 2019; Paananen et al., 2019). Compared with $\widehat{R}$, stacking can reveal subtle aspects of poor mixing among chains, offering a diagnostic that is targeted to prediction.

Second, we can use stacking to diagnose where the posterior geometry cause sampling issues. In the hierarchical model example (Section 5.3), a nonzero complete-pooling chain ($\tau = 0$) indicates that the simulation using centered parameterization has not fully explored the basin around $\tau \approx 0$.

Lastly, stacking can diagnose interactions that have not been included in the model. When stacking reveals strong and persistent differences among chains, it signifies potential model misspecification. In particular, the data can be a mixture of several generating processes corresponding to

different parameters (Kamary et al., 2018). We can expand to a hierarchical model as described in Section 2.2.

### 6.4. Stacking as part of Bayesian workflow

We view stacking of parallel chains as sitting on the boundary between black-box inference and a larger *Bayesian workflow* (Gabry et al., 2019; Gelman et al., 2020).

For an automatic inference algorithm, stacking enables accessible inference from non-mixing chains and a free enrichment of predictive distributions, which is especially relevant for repeated tasks where computation time is constrained.

For Bayesian workflow more generally, we recommend stacking in the model exploration phase, where we need to obtain *some* inference. Parallel computation can be running asynchronously—it may be that only some chains are running slowly—and stopping in the middle frees up computation and human time that can be reallocated to explorations of more models. In addition, non-uniform stacking weights when used in concert with trace plots and other diagnostic tools can help us understand where to focus that effort in an iterative way.

### acknowledgments

We thank the U.S. National Science Foundation, Institute of Education Sciences, Office of Naval Research, and Sloan Foundation for partial support of this work.

## References

Agrawal, A., Fu, W., and Menzies, T. (2018). What is wrong with topic modeling? And how to fix it using search-based software engineering. *Information and Software Technology*, 98:74–88.

Angelino, E., Johnson, M. J., and Adams, R. P. (2016). Patterns of scalable Bayesian inference. *Foundations and Trends in Machine Learning*, 9:119–247.

Bafumi, J., Gelman, A., Park, D. K., and Kaplan, N. (2005). Practical issues in implementing and understanding Bayesian ideal point estimation. *Political Analysis*, 13:171–187.

Berk, R. H. (1966). Limiting behavior of posterior distributions when the model is incorrect. *Annals of Mathematical Statistics*, 37:51–58.

Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, 8:10–15.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19:1501–1534.

Betancourt, M. and Girolami, M. (2015). Hamiltonian Monte Carlo for hierarchical models. In Upadhyay, S. K., Singh, U., Dey, D. K., and Loganathan, A., editors, *Current Trends in Bayesian Methodology with Applications*, pages 79–101. CRC Press.

Bhatnagar, N. and Randall, D. (2004). Torpid mixing of simulated tempering on the Potts model. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 478–487. Society for Industrial and Applied Mathematics.

Bishop, C. M., Lawrence, N. D., Jaakkola, T., and Jordan, M. I. (1998). Approximating posterior distributions in belief networks using mixtures. In *Advances in Neural Information Processing Systems*, pages 416–422.

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

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

Bou-Rabee, N., Eberle, A., and Zimmer, R. (2018). Coupling and convergence for Hamiltonian Monte Carlo. *arXiv preprint arXiv:1805.00452*.

Breiman, L. (1996). Stacked regressions. *Machine Learning*, 24:49–64.

Carreño, L. V. G. and Winbladh, K. (2013). Analysis of user comments: An approach for software requirements evolution. In *35th International Conference on Software Engineering*, pages 582–591. IEEE.

Chang, O., Yao, Y., Williams-King, D., and Lipson, H. (2019). Ensemble model patching: A parameter-efficient variational Bayesian neural network. *arXiv preprint arXiv:1905.09453*.

Chung, Y., Rabe-Hesketh, S., Dorie, V., Gelman, A., and Liu, J. (2013). A nondegenerate penalized likelihood estimator for variance parameters in multilevel models. *Psychometrika*, 78:685–709.

Clarke, B. (2003). Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research*, 4:683–712.

Clyde, M. and Iversen, E. S. (2013). Bayesian model averaging in the $\mathcal{M}$-open framework. In Damien, P., Dellaportas, P., Polson, N. G., and Stephens, D. A., editors, *Bayesian Theory and Applications*, pages 483–498. Oxford University Press.

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28:424–446.

Diaconis, P. and Freedman, D. (1986a). On inconsistent Bayes estimates of location. *Annals of Statistics*, 14:68–87.

Diaconis, P. and Freedman, D. (1986b). On the consistency of Bayes estimates. *Annals of Statistics*, 14:1–26.

Dwivedi, R., Chen, Y., Wainwright, M. J., and Yu, B. (2018). Log-concave sampling: Metropolis-Hastings algorithms are fast! *arXiv preprint arXiv:1801.02309*.

Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7:3910–3916.

Finch, S. J., Mendell, N. R., and Thode Jr, H. C. (1989). Probabilistic measures of adequacy of a numerical search for a global maximum. *Journal of the American Statistical Association*, 84:1020–1023.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society Series A*, 182:389–402.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.

Gelman, A. (2008). The folk theorem of statistical computing. *Statistical Modeling, Causal Inference, and Social Science*. `https://statmodeling.stat.columbia.edu/2008/05/13/the_folk_theore/`.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24:997–1016.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13:163–185.

Gelman, A., Meng, X.-L., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, pages 733–760.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472.

Gelman, A. and Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66:8–38.

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., and Modrák, M. (2020). Bayesian workflow. *arXiv:2011.01808*.

Gelman, A. and Yao, Y. (2021). Holes in Bayesian statistics. *Journal of Physics G: Nuclear and Particle Physics*.

Gershman, S., Hoffman, M., and Blei, D. (2012). Nonparametric variational inference. In *Proceedings of the 29th International Conference on Machine Learning*.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science*, 7:473–483.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264.

Gorinova, M. I., Moore, D., and Hoffman, M. D. (2019). Automatic reparameterisation of probabilistic programs. *arXiv preprint arXiv:1906.03028*.

Hansmann, U. H. (1997). Parallel tempering algorithm for conformational studies of biological molecules. *Chemical Physics Letters*, 281:140–150.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, 14:382–401.

Hoffman, M. and Ma, Y.-A. (2020). Black-box variational inference as distilled langevin dynamics. In *Proceedings of the 37th International Conference on Machine Learning*.

Huang, Z. and Gelman, A. (2005). Sampling for Bayesian computation with large datasets. *Technical Report, Columbia University.* `http://www.stat.columbia.edu/~gelman/research/unpublished/comp7.pdf`.

Jaakkola, T. S. and Jordan, M. I. (1998). Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*, pages 163–173. Springer.

Johnson, L. T., Geyer, C. J., et al. (2012). Variable transformation to obtain geometric ergodicity in the random-walk Metropolis algorithm. *Annals of Statistics*, 40:3050–3076.

Jylänki, P., Vanhatalo, J., and Vehtari, A. (2011). Robust Gaussian process regression with a Student-t likelihood. *Journal of Machine Learning Research*, 12:3227–3257.

Kamary, K., Mengersen, K., Robert, C. P., and Rousseau, J. (2018). Bayesian hypothesis testing as a mixture estimation model. *arXiv preprint arXiv:1412.2044*.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Kleijn, B. J. K. and Van der Vaart, A. W. (2012). The Bernstein-von-Mises theorem under misspecification. *Electronic Journal of Statistics*, 6:354–381.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18:430–474.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6402–6413.

Le, T. and Clarke, B. (2017). A Bayes interpretation of stacking for $\mathcal{M}$-complete and $\mathcal{M}$-open settings. *Bayesian Analysis*, 12:807–829.

LeBlanc, M. and Tibshirani, R. (1996). Combining estimates in regression and classification. *Journal of the American Statistical Association*, 91:1641–1650.

Liu, J. and Hodges, J. S. (2003). Posterior bimodality in the balanced one-way random-effects model. *Journal of the Royal Statistical Society B*, 65:247–255.

Madigan, D., Raftery, A. E., Volinsky, C., and Hoeting, J. (1996). Bayesian model averaging. In *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models*.

Mangoubi, O., Pillai, N. S., and Smith, A. (2018). Does Hamiltonian Monte Carlo mix faster than a random walk on multimodal densities? *arXiv preprint arXiv:1808.03230*.

Mangoubi, O. and Smith, A. (2017). Rapid mixing of Hamiltonian Monte Carlo on strongly log-concave distributions. *arXiv preprint arXiv:1708.07114*.

Mangoubi, O. and Smith, A. (2019). Mixing of Hamiltonian Monte Carlo on strongly log-concave distributions 2: Numerical integrators. *Proceedings of Machine Learning Research*, 89:586–595.

Marinari, E. and Parisi, G. (1992). Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters*, 19:451.

Mesquita, D., Blomstedt, P., and Kaski, S. (2019). Embarrassingly parallel MCMC using deep invertible transformations. *arXiv preprint arXiv:1903.04556*.

Miller, A. C., Foti, N. J., and Adams, R. P. (2017). Variational boosting: Iteratively refining posterior approximations. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2420–2429.

Mäntylä, M. V., Claes, M., and Farooq, U. (2018). Measuring LDA topic stability from clusters of replicated runs. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–4.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.

Neal, R. M. (1998). Regression and classification using Gaussian process priors. In Bernardo, J., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics*, volume 6, pages 475–501. Oxford University Press.

Oelrich, O., Ding, S., Magnusson, M., Vehtari, A., and Villani, M. (2020). When are Bayesian model probabilities overconfident? *arXiv preprint arXiv:2003.04026*.

Osband, I., Van Roy, B., Russo, D., and Wen, Z. (2019). Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20:1–62.

Paananen, T., Piironen, J., Bürkner, P.-C., and Vehtari, A. (2019). Implicitly adaptive importance sampling. *arXiv preprint arXiv:1906.08850*.

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, 22:59–73.

Pearce, T., Zaki, M., Brintrup, A., and Neel, A. (2020). Uncertainty in neural networks: Approximately Bayesian ensembling. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*.

Piironen, J. and Vehtari, A. (2017a). On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*.

Piironen, J. and Vehtari, A. (2017b). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11:5018–5051.

Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. (2019). Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, 114:831–843.

Raftery, A. E. and Lewis, S. (1992a). How many iterations in the Gibbs sampler. In Bernardo, J., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pages 763–773. Oxford University Press.

Raftery, A. E. and Lewis, S. M. (1992b). Comment: One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. *Statistical Science*, 7:493–497.

Ranganath, R., Tran, D., and Blei, D. (2016). Hierarchical variational models. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 324–333.

Robbins, H. E. (1968). Estimating the total probability of the unobserved outcomes of an experiment. *Annals of Mathematical Statistics*, 39:256–257.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7:110–120.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11:78–88.

Stan Development Team (2020). *Stan User's Guide, Version 2.23*.

Stephens, M. (2000). Dealing with multimodal posteriors and non-identifiability in mixture models. *Journal of the Royal Statistical Society B*, 62:795–809.

Tian, K., Revelle, M., and Poshyvanyk, D. (2009). Using latent Dirichlet allocation for automatic categorization of software. In *6th IEEE International Working Conference on Mining Software Repositories*, pages 163–166. IEEE.

Tipping, M. E. and Lawrence, N. D. (2005). Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *Neurocomputing*, 69:123–141.

Van Der Pas, S. L., Kleijn, B. J., Van Der Vaart, A. W., et al. (2014). The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8:2585–2618.

Vanhatalo, J., Jylänki, P., and Vehtari, A. (2009). Gaussian process regression with Student-t likelihood. In *Advances in Neural Information Processing Systems*, pages 1910–1918.

Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., and Gelman, A. (2019a). *loo: Efficient leave-one-out cross-validation and WAIC for Bayesian models*. R package version 2.2.0.

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27:1413–1432.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2020a). Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC. *Bayesian Analysis*.

Vehtari, A., Gelman, A., Sivula, T., Jylänki, P., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D., and Robert, C. (2020b). Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *Journal of Machine Learning Research*, 21:1–53.

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2019b). Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*.

Vehtari, A., Särkkä, S., and Lampinen, J. (2000). On MCMC sampling in Bayesian MLP neural networks. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, pages 317–322.

Wang, F. and Landau, D. P. (2001). Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86:2050–2053.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241–259.

Wong, H. and Clarke, B. (2004). Improvement over bayes prediction in small samples in the presence of model uncertainty. *Canadian Journal of Statistics*, 32:269–283.

Yang, Z. and Zhu, T. (2018). Bayesian selection of misspecified models is overconfident and may cause spurious posterior probabilities for phylogenetic trees. *Proceedings of the National Academy of Sciences*, 115:1854–1859.

Yao, Y. (2019). Bayesian aggregation. *arXiv preprint arXiv:1912.11218*.

Yao, Y., Cademartori, C., Vehtari, A., and Gelman, A. (2020). Adaptive path sampling in metastable posterior distributions. *arXiv:2009.00471*.

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018a). Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis*, 13:917–1003.

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018b). Yes, but did it work?: Evaluating variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5577–5586.

# Appendices

## A. Proofs for asymptotic theories

We sketch the proof for theorems in Section 4.

### A.1. Proofs for Theorem 1

The proof of Theorem 1 is a direct application of the consistency results in Vehtari et al. (2019b) and Le and Clarke (2017).

Assuming samples from the $k$-th chain ($k = 1, 2, \ldots, K$) (not necessarily independently) come from a stationary distribution $p_k(\theta)$,we denote $p_{k,-i}(y_i) = p_k(y_i|y_{-i}) \coloneqq \int_\Theta p(y_i|\theta)p_k(\theta|y_{-i})d\theta$ to be the leave-one-out density.

First, the importance sampling based approximation is pointwise consistent.

**Theorem 6.** *(Theorem 2 and 3 in Vehtari et al., 2019b) Assuming the stationary distribution $p_k(\theta)$ satisfies regularity conditions defined therein, the PSIS-based approximate loo is consistent with a large number of posterior draws. For any fixed chain index $k$, and observation index $i$,*

$$\frac{\sum_{s=1}^S p(y_i|\theta_{ks})r_{iks}}{\sum_{s=1}^S r_{iks}} - p_{k,-i}(y_i) \xrightarrow{L_2} 0,\ S \to \infty.$$

In practice, the convergence rate of approximate PSIS-loo with finite posterior draws can be characterized by the $\hat{k}$ diagnostics (Vehtari et al., 2019b).

Second, Le and Clarke (2017) proved that given set of weights $w_1 \ldots w_K$ and when sample size $n \to \infty$, the leave-one-out logarithmic predictive density (loo lpd), converges to the expected log predictive densities (elpd):

**Theorem 7.** *(Theorem 2.2 in Le and Clarke, 2017) Assuming regularity conditions:*

1. *For each $k = 1, \ldots, K$, there is a function $B_k(\cdot)$ so that*

$$\sup_{y \in \mathbb{R}^n} |\log p_k(\tilde{y}|y)| \le B_k(\tilde{y}) < \infty,$$

*where $B_k$ is independent of other covariates and $\mathrm{E}(g(\tilde{y})) < \infty$ for*

$$g(\tilde{y}) = \max\left\{ \left( \log \sum_{k=1}^K w_k \exp\left(-B_k(\tilde{y})\right) \right)^4, \left( \log \sum_{k=1}^K w_k \exp\left(B_k(\tilde{y})\right) \right)^4 \right\}.$$

2. *For each $k = 1, \ldots, K$, the conditional densities $p_k(y|x, \theta)$ are equicontinuous in $x$ for each $y$ and $\theta \in \Theta_k$, and the predictive densities $p_k(\cdot|y)$ within the are uniformly equicontinuous in $y$.*

*Then we have*

$$\frac{1}{n}\sum_{i=1}^n \log \sum_{k=1}^K w_k p_{k,-i}(y_i) - \mathrm{E}_{\tilde{y}|y} \log \sum_{k=1}^K w_k p_k(\tilde{y}|y) \xrightarrow{L_2} 0,\ n \to \infty.$$

Now return to the objective function in stacking (Equation 7):

$$\max_{w\in\mathbb{S}(K)} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p^S_{k,-i}(y_i) + \log p_{\text{prior}}(w),$$

where the leave-one-out distribution is approximated by importance sampling using $S$ posterior draws each chain,

$$p^S_{k,-i}(y_i) = \frac{\sum_{s=1}^{S} p_k(y_i|\theta_{ks}) r_{iks}}{\sum_{s=1}^{S} r_{iks}}.$$

Combining the previous two consistency results, for a fixed number of chains $K$ and a fixed weight vector $w$, when both the sample size of observations $n$ and the number of posterior draws $S$ go to infinity, under all previous mentioned assumptions, the objective function converges to the elpd of the weighted posterior inference:

$$\frac{1}{n}\sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p^S_{k,-i}(y_i) - \mathrm{E}_{\tilde{y}|y} \log\Big(\sum_{k=1}^{K} w_k p_k(\tilde{y}|y)\Big) \xrightarrow{L_2} 0,$$

which proves Theorem 1.

### A.2. Proofs of Theorems 2 and 3

First, the unnormalized posterior density of $\mu$ is

$$\log p(\mu|y) = \log p_0(\mu) - \sum_{i=1}^{n} \log(1 + (y_i - \mu)^2).$$

Define

$$h(\mu) = -\int_{-\infty}^{\infty} \log(1 + (y-\mu)^2) \left(\frac{1-p_0}{(a+y)^2+1} + \frac{p_0}{(y-a)^2+1}\right) dy,$$

which is always a well-defined and finite integral for all $\mu$.

**Lemma 8.** *$\frac{d}{d\mu}h(\mu)$ has a closed form expression*

$$\begin{aligned}
\frac{d}{d\mu}h(\mu) &= -\int_{-\infty}^{\infty} \frac{d}{d\mu}\log(1 + (y-\mu)^2) \left(\frac{1-p_0}{(a+y)^2+1} + \frac{p_0}{(y-a)^2+1}\right) dy \\
&= -\frac{\pi p_0(\mu - a)}{(a-\mu)^2+4} - \frac{\pi(1-p_0)(a+\mu)}{(a+\mu)^2+4} \\
&= \frac{-\pi\left(4a + a^3 - 8ap - 2a^3p + (4-a^2)u + (-a+2ap)u^2 + u^3\right)}{(a^2 - 2a\mu + \mu^2 + 4)\,(a^2 + 2a\mu + \mu^2 + 4)}.
\end{aligned}$$

*Proof.* Calculus and change of variables. □

We define $\xi(a)$ as the third largest root of the following forth-order equation (as a function of $x$):

$$u_a(x) = x^4\left(a^6 + 4a^4\right) + x^3\left(-2a^6 - 8a^4\right) + x^2\left(a^6 - 8a^4 - 44a^2\right) + x\left(12a^4 + 44a^2\right) - 4a^4 - 8a^2 - 4 = 0.$$

$\xi(a)$ is a bijective and increasing mapping from $[2, \infty)$ to $[0.5, 1)$. $\xi(2) = 0.5$ and $\lim_{a\to\infty}\xi(a) = 1$. We visualize the deterministic function $p_0 = \xi(a)$ in Figure 6.

**Lemma 9.** *The number of modes in $h(\mu)$ is determined by the relation between $a$ and $p_0$.*

*(a) When $a > 2$ and $p_0 \geq \xi(a)$, $h(\mu)$ only has one global maximum near $a$.*

*(b) When $a > 2$ and $p_0 < \xi(a)$, $h(\mu)$ has two local maximum near $a$ and $-a$ respectively.*

*(c) When $a < 2$, $h(\mu)$ is unimodal with the global maximum between 0 and $a$.*

*Proof.* The denominator in $\frac{d}{d\mu}h(\mu)$ is always positive. Let $g(\mu) = -4a - a^3 + 8ap + 2a^3p + (-4 + a^2)u + (a - 2ap)u^2 - u^3$. It is a cubic polynomial on $\mu$ and has the discriminant:

$$\begin{aligned}\Delta(a, p_0) = &\left(64a^6 + 256a^4\right) p_0^4 + \left(-128a^6 - 512a^4\right) p_0^3 + \left(64a^6 - 512a^4 - 2816a^2\right) p_0^2 \\ &+ \left(768a^4 + 2816a^2\right) p_0 - 256a^4 - 512a^2 - 256.\end{aligned}$$

Solving $\Delta(a, p_0) = 0$ has and only has one root on $a > 2$ and $0.5 < p_0 < 1$: $p_0 = \xi(a)$, where the function $\xi(a)$ is defined in the lemma.

Further, when $p_0 \geq \xi(a)$, $\Delta(a, p_0) \leq 0$, and therefore $g(\mu)$ only has one cross-zero-root. Since $h'(\mu) = g(\mu)$ and $h(\pm\infty) = -\infty$, this unique root is the global maximum of $h(\mu)$. We denote this unique mode by $\gamma(a, p_0)$.

For a large $a$, using the second expression in Lemma 8, $\frac{d}{d\mu}h(\mu)|_{\mu=a} = -\frac{\pi a(1-p_0)}{2a^2+2} \to 0^-$. Therefore the mode $\gamma(a, p_0) \to a^-$, as $a \to \infty$.

In situation (b), when $p_0 < \xi(a)$, $\Delta(a, p_0) > 0$. $g(\mu)$ only has three cross-zero roots. This implies $h(\mu)$ has two local maxima $\gamma^+$ and $\gamma^-$, near but not identical to $\pm a$, and a local minimal (near 0).

Using the second line in Lemma 8, for any $\mu < 0$, $h(-\mu) > h(\mu)$, therefore $h(\gamma^+) > h(-\gamma^-) > h(\gamma^-)$; that is, the right mode is higher than the left mode for $p_0 > 0.5$.

In situation (c), when $0 < a < 2$, $\Delta(a, p_0) < 0$ and therefore $g(\mu)$ only has a cross-zero-root, which is the first root in the following cubic function:

$$u(x) = x^3 + x^2(ap_0 - a) + x\left(4 - a^2\right) - 2a^3p_0 + a^3 - 8ap_0 + 4a = 0.$$

In particular if $p_0 = 0.5$, this root is at $\mu = 0$. □

**Lemma 10.** *For a fixed $p_0$ and $a \to \infty$, the two local modes $(\gamma^+(a, p_0), \gamma^-(a, p_0)) \to (a, -a)$.*

*Proof.* Using the second expression in Lemma 8,

$$\frac{d}{d\mu}h(\mu)|_{\mu=a} = -\frac{\pi a(1 - p_0)}{2a^2 + 2} \to 0^-, \text{ as } a \to \infty,$$

while

$$\frac{d^2}{d\mu^2}h(\mu)|_{\mu=a} = \frac{\pi a(8ap_0 - 8a)}{\left(4a^2 + 4\right)^2} - \frac{\pi\left(-4a^2p_0 - 4\right)}{4\left(4a^2 + 4\right)} \to -\frac{\pi p_0}{4} = O(1).$$

Hence when $a \to \infty$. the mode $\gamma^+(a, p_0) \to a^-$, and likewise $\gamma^-(a, p_0) \to -a^+$. □

The approximation using Lemma 10 is accurate for a moderately large $a$. For example, when $p_0 = 0.6$, and $a = 8$, the right and left modes in $h$ are $(\gamma^+(a, p_0), \gamma^-(a, p_0)) = (7.8, -7.3)$, and at $a$=10 they are $(9.9, -9.7)$.

**Lemma 11.** *When $a > 2, p_0 = 0.5$, $h(\mu)$ has two equally high modes at $\pm\sqrt{a^2 - 4}$.*

*Proof.* This is a special case of the previous lemma in which we can solve $h'(\mu) = 0$ explicitly.

$$\frac{d}{d\mu}h(\mu|a, p_0 = 0.5) = -\frac{2\pi\mu\left(-a^2 + \mu^2 + 4\right)}{-2a^2\left(\mu^2 - 4\right) + a^4 + \left(\mu^2 + 4\right)^2}$$

has three zeros, 0 and $\pm\mu_0$, where $\mu_0 = \sqrt{a^2-4}$. Furthermore we can check $h''(0) > 0$, and $h''(\pm\mu_0) < 0$. Hence $h(\mu)$ has one local minimal at $\mu = 0$ and two global maximum at $\pm\mu_0$. $h(\mu_0) = h(-\mu_0)$ due to symmetry. □

When $a > 2, p_0 > 0.5$, $h(\mu)$ either has a unique mode ((a) in Lemma 9), $\gamma^+ > 0$, or two local modes ((b) in Lemma 9) with unequal heights $h(\gamma^+) > h(\gamma^-)$. The convergence to the right mode is a straightforward application of any usual Bayes consistency result (under model misspecification).

**Lemma 12.** *When $a > 2, p_0 > 0.5$, the posterior $p(\mu|y_1, \ldots, y_n)$ is asymptotically concentrated at the point mass $\gamma_+$. That is, for any $\eta > 0$, when $n \to \infty$,*

$$\Pr(\left|\mu - \gamma^+\right| < \eta | y_1, \ldots, y_n) \to 1, a.s.$$

*Proof.* The weak law of large numbers implies

$$\frac{1}{n} \log C_n p(\mu|y_1, \ldots, y_n) \to h(\mu),$$

where $C_n$ is the normalization constant. Since $h$ is $C^\infty$ smooth, we can choose $\delta = \frac{1}{2}\left(h(\gamma^+) - h(\gamma^-)\right) > 0$, and there exists an $\epsilon$ neighborhood of $\gamma_+$ such that,

$$\inf_{\gamma:|\gamma-\gamma^+|<\epsilon} h(\gamma) > h(\gamma^+) - \delta > \sup_{\gamma:|\gamma-\gamma^+|>\epsilon} h(\gamma),$$

which implies

$$\Pr(\mu \in (\gamma^+ - \epsilon, \gamma^+ + \epsilon) | y_1, \ldots, y_n) \to 1$$

□

Now express the log posterior density of $\mu$ as

$$\begin{aligned}\log p(\mu|y_{1,\ldots,n}) &= \log p_0(\mu) + \sum_{i=1}^{n} -\log(1 + (y_i - \mu)^2) - \log C_n \\ &= \log p_0(\mu) + nh(\mu) + \sqrt{n} G_n(\mu) - \log C_n,\end{aligned}$$

where $\log C_n$ is the log normalization constant, and

$$G_n(\mu) = n^{-1/2} \sum_{i=1}^{n} \left(-\log(1 + (y_i - \mu)^2) - h(\mu)\right),$$

which can also be written as

$$G_n(\mu) = \int -\log(1 - (\mu - y)^2) dB_n(y), \quad B_n(y) = \sqrt{n}(F_n - F).$$

where $F_n$ and $F$ are the empirical distribution of $y_1, \ldots, y_n$ and the distribution function of the data generating process, respectively.

The remaining argument transfers the results from $h(\mu)$ to the posterior. Loosely speaking, the remaining term $G_n(\mu)$ is asymptotically a Gaussian process and bounded by $o(n^{1/2})$, while the main term $nh(\mu)$ outside the neighborhood of the mode of $h(\mu)$ vanishes $O(n)$ quicker than the inside. Therefore, the posterior $p(\mu|y_{1:n})$ will asymptotically carry a mode around the mode in $h(\mu)$. That is Theorem 2. A rigorous proof of Theorem 3 follows from all previous lemmas and Lemma 2.4-2.12 in Diaconis and Freedman (1986a).

### A.3. Proofs for Corollaries 4 and 5

Corollary 4 follows directly from Theorem 3. In specific, for big $a$, we can further approximate the left and right mode near $\pm a$ using Lemma 10. Then the Bayesian posterior is closed to a point mass that is spiked at $a$ for $0.5 < p_0 < \xi(a)$, so the resulting KL divergence is always non-vanishing. Notably, the KL divergence between two Cauchy density $\text{Cauchy}(\mu_1, \sigma)$ and $\text{Cauchy}(\mu_2, \sigma)$ has a closed form expression: $\text{KL}\big(\text{Cauchy}(\mu_1, \sigma) \,||\, \text{Cauchy}(\mu_2, \sigma)\big) = \log\left(1 + \frac{(\mu_1 - \mu_2)}{4\sigma^2}\right)$.

In Corollary 5, we assume the parallel evaluation has captured both modes $\gamma^-$ and $\gamma^+$ and we have classified them into two clusters. Using Theorem 1 , for any $0.5 < p_0 < \xi(a)$, stacking solves

$$\min_{w \in \mathbb{S}(2)} \text{KL}\Big((1-p_0)\,\text{Cauchy}(a,1) + p_0\,\text{Cauchy}(-a,1) \,||\, w_1\,\text{Cauchy}(\gamma^-,1) + w_2\,\text{Cauchy}(\gamma^+,1)\Big).$$

The limiting Bayesian inference is a stacking solution corresponding to a weight of 1 on the right mode. It is easy to check that $w = (0, 1)$ is not the optimum by first order conditions. Using Theorem 1 we see the stacking weights yields a higher elpd.

When $p_0 = 0.5$, $a > 2$, in the $n \to \infty$ limit in Theorem 1, the stacking solution optimizes $\min_{w \in \mathbb{S}(2)} \text{KL}\big(0.5\,\text{Cauchy}(a,1) + 0.5\,\text{Cauchy}(-a,1) \,||\, w_1\,\text{Cauchy}(\sqrt{a^2-4},1) + w_2\,\text{Cauchy}(-\sqrt{a^2-4},1)\big)$, which is attained at $w_1 = w_2 = 0.5$. Direct computation shows tjat the KL divergence above at the optimal $w_1 = w_2 = 0.5$ approaches 0 for big $a$. See Figure 6 for numerical evaluations.

## B. Implementation in `Stan` and `R` package `loo`

We demonstrate the implementation of multiple-chain stacking in the general-purpose Bayesian inference engine Stan (Stan Development Team, 2020). We use the Cauchy mixture model as an example. First save the following Stan file to `cauchy.stan`.

```
data {
int n;
vector[n] y;
}
parameters {
real mu;
}
model {
y ~ cauchy(mu, 1);
}
generated quantities {
vector[n] log_lik;
for (i in 1:n)
log_lik[i] = cauchy_lpdf(y[i]| mu, 1);
}
```

In the `generated quantities` block, we save `log_lik`: the log likelihood of each data point at each posterior draw. We generate data from a Cauchy mixture according to example (iii) in Figure 2, and sample from its posterior densities. Here is the R code:

```
library(rstan)
library(loo)
set.seed(100)
mu = c(-10,10)
n = 100
y = rep(NA, n)
p = 0.5
y[1:(n*p)] = rcauchy(n*(p),mu[1], 1)
y[(n*(p)+1):n] = rcauchy(n*(1-p),mu[2], 1)
K = 8
# Fit the model in stan
set.seed(100)
stan_fit = stan("cauchy.stan", data=list(n=n, y=y), chains=K, seed=100)
mu_sample = extract(stan_fit, permuted=F, pars="mu")[,,"mu"]
print(Rhat(mu_sample))
```

We are using eight parallel chains, and the resulted $\widehat{R} = 1.6$, clearly not mixing.

`chain_stack()` is a function to combine multiple chains in a Stan fit object, returned by `stan()`. It only require the whole model fit once, and save the point wise log likelihood in each iteration, called via `log_lik` here. The `chain_stack()` function uses the Stan optimizer (the default is L-BFGS), and its first time compiling takes up to a few minutes. `lambda` is the tuning parameter that controls the Dirichlet prior on stacking weights.

```
> library(devtools)
> source_url("https://github.com/yao-yl/Multimodal-stacking-code
/blob/master/chain_stacking.R?raw=TRUE")
> stan_model_object = stan_model("stacking_opt.stan")
> stack_obj=chain_stack(fits=stan_fit,lambda=1.0001,log_lik_char="log_lik")

Output:  Stacking 8 chains, with 100 data points and 1000 posterior draws;
using stan optimizer, max iterations = 1e+05
...done.
Total elapsed time for approximate LOO and stacking = 0.87 s
```

We can assess the reliability of the approximate leave-one-out using the $\hat{k}$ diagnostics. In this example, all pointwise $\hat{k}$ estimates (100 observations $\times$ 8 chains $=$ 800 in total) are smaller than 0.5, indicating that the loo approximation is accurate in this example.

```
>  print_k(stack_obj)

Output:                Count Proportion
(-Inf, 0.5] (good)     800   1
(0.5, 0.7]  (ok)       0     0
(0.7, 1]    (bad)      0     0
(1, Inf)    (very bad) 0     0
```

We access the chain wights using

```
> chain_weights = stack_obj$chain_weights
```

Finally, we can use the weighted samples to calculate any posterior integral $\mathrm{E}_{\text{stacking}}(h(\mu)|y)$ as in (5). Here we compute $\Pr(\mu > 0|y)$: the total mass of positive values in the stacked inference.

```
> h = function(mu){mu>0}
> round(chain_weights %*% apply(h(mu_sample), 2, mean), digits=3)
[1] 0.523
```

Alternatively, we provide a quasi Monte Carlo based importance resampling function `mixture_draws()` that draws posterior samples form the stacked inference. This enables us to compute the same integral $\mathrm{E}_{\text{stacking}}[h(\mu) \mid y]$ using usual Monte Carlo methods:

```
> resampling=mixture_draws(individual_draws=mu_sample,weight=chain_weights)
> mean(h(resampling))
[1] 0.523
```

## C. Reproducible code and experiment details

Data and code for this paper are available at `https://github.com/yao-yl/Multimodal-stacking-code`.

*LDA topic models.* In Section 5.1, the text data are all words in the novel Pride and Prejudice. We preprocess the data by removing stop words and rare words. The cleaned data are stored in the posterior database (`https://github.com/MansMeg/posteriordb`), also uploaded as `staninput.RData`. We use the Stan implementation of LDA models (`https://mc-stan.org/docs/2_22/stan-users-guide/latent-dirichlet-allocation.html`) with little modification, as in the file `lda.stan`.

In all experiments, We run parallel inference on Columbia University's shared HPC Terremoto with one chain per core (CPU: Intel Xeon Gold 6126, 2.6 Ghz). When there is no further specification, we use the default starting values: draw all unconstrained parameters from uniform$(-2, 2)$ randomly in each chain.

We pre-specify the maximum running time for 2000 iterations to be 24 hours and 4000 iterations to be 48 hours in all LDA models, and all running-out-of-time chains are discarded.

*Gaussian process regression.* The original data of Neal (1998) can be found in file `odata.txt`. In the first experiment, we use the first half as training data. In the second experiment, we simulate data with varying sample size according to his data generating process. For hyper-parameter optimization, we found two modes by using initialization $(\log\rho, \log\alpha, \log\sigma) = (1, 0.7, 0.1)$ and $(-1, -5, 2)$, respectively. We approximate the posterior by MAP or Laplace approximation and importance resampling around two local mode. The approximate samples have little overlap.

In the full sampling for the $t$ regression, we compare four chain-combination strategies: BMA, pseudo-BMA, uniform averaging, and stacking. After each iteration of $(\sigma, \rho, \alpha, f)$, we draw posterior predictive sample of $\tilde{f} = f(\tilde{X})$, from

$$\tilde{f}|\tilde{X}, X, f \sim \mathrm{MVN}\left(K(\tilde{X}, X)K(X, X)^{-1}f, K(\tilde{X}, \tilde{X}) - K(\tilde{X}, X)K(X, X^{-1})K(X, \tilde{X})\right),$$

and compute the mean test data log predictive densities,

$$1/n_{\text{test}} \sum_{i=1}^{n_{\text{test}}} \log p(\tilde{y}_i|\tilde{f}_i,\sigma)p(\tilde{f}_i,\sigma|X,y)d\tilde{f}_i d\sigma.$$

The full-model specification is in `treg.stan`.

In Figure 21, we compare any two Monte Carlo integral using chains 1 and 8, compute their mean and standard error, and plug in the estimated effective sample size in the $t$-test formula. We view a larger $t$-score as a heuristic indicator of a large across-chain discrepancy.

*Balanced one-way hierarchical model.* There can be entropic barriers in the non-centered parameterization too. The likelihood in (18) is equivalent to $\xi_i|\tau,\mu,y \sim \text{normal}(\frac{1}{\tau}(\bar{y}_{\cdot j}-\mu),\sigma\tau^{-1}J^{-1/2})$, where $\bar{y}_{\cdot j}$ is the sample mean of group $j$. Replacing $\tau$ and $\theta_j$ by plug-in estimates, we derive the conditional variance in the likelihood as $\text{Var}(\xi_i|\mu,\sigma,y) \approx \left(N^{-1}J\sigma^2\right) / \sum_j(\bar{y_{\cdot j}}-\mu)^2$, which forms a funnel between $\mu$ and $\xi$.

In the experiment, the true $\tau$ and $\sigma$ vary from 0.1 to 20. In order to achieve a higher F-statistics so as to manifest posterior bimodality, we additionally add some student $t$-distributed noise added to group mean in the unknown data generating process. $\theta_i := \theta_i + Bz_i$, where $z_i$ is iid $t(1)$ distributed noise, and B varies from 0 to 50. The complete pooling, centered, and non-centered parameterizations are coded in the Stan files `random-effect-zero.stan`, `random-effect.stan` and `random-effect-ncp.stan`.

*Neural networks for MNIST.* We subsample 1000 data points from MNIST as training data, with subsampling details in `readmnist.R` and the saved test and training data in `input.RData`. The model is adapted from Bob Carpenter's Stan code `https://github.com/stan-dev/example-models/blob/master/knitr/neural-nets/nn-simple.stan` with a few modifications as in `2classnn.stan`.

In the experiment, we considered two choices of priors: (a) a fixed-scale elementwise normal$(0,3)$ prior on all unknown parameters $\phi \in \mathbb{R}, \beta \in \mathbb{R}^{40}$, and $\alpha \in \mathbb{R}^{784\times 40}$; and (b) $\alpha \sim \text{normal}(0,\sigma_\alpha)$, $\beta \sim \text{normal}(0,\sigma_\beta)$, $\sigma_\alpha,\sigma_\beta \sim \text{normal}^+(0,3)$. For the experiment we are running, these two sets of priors yield nearly identical posterior sampling results and the same results after chain averaging.

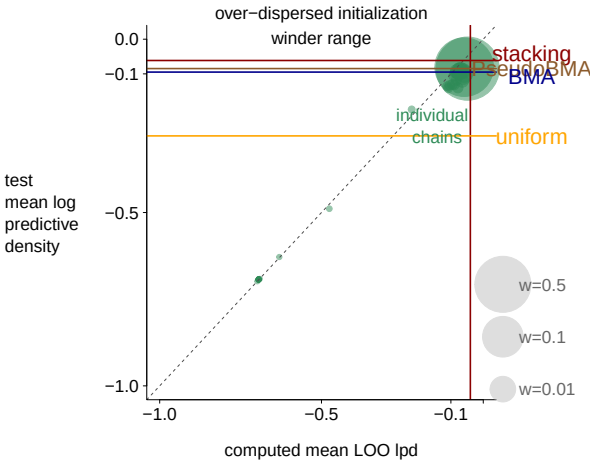



Figure 22: *Some individual changes in the overdispersed setting are out of lower-range and not shown in Figure 20. This is the same graph with wider ranges.*