Parallel MCMC Without Embarrassing Failures

de Souza, Daniel Augusto

Journal of Machine Learning Research
2022-02-22

de Souza , D A , Mesquita , D , Kaski , S & Acerbi , L 2022 , Parallel MCMC Without Embarrassing Failures . in Proceedings of The 25th International Conference on Artificial Intelligence and Statistics . Proceedings of Machine Learning Research (PMLR) , vol. 151 , Journal of Machine Learning Research , pp. 1786-1804 , International Conference on Artificial Intelligence and Statistics , Valencia , Spain , 30/03/2022 . < https://proceedings.mlr.press/v151/de-souza22a.html >

http://hdl.handle.net/10138/354045

unspecified
publishedVersion

Parallel MCMC Without Embarrassing Failures

Daniel Augusto de Souza[1], **Diego Mesquita**[2,3], **Samuel Kaski**[2,4], **Luigi Acerbi**[5]
[1]University College London [2]Aalto University [3]Getulio Vargas Foundation
[4]University of Manchester [5]University of Helsinki
daniel.souza.21@ucl.ac.uk, diego.mesquita@fgv.br, samuel.kaski@aalto.fi, luigi.acerbi@helsinki.fi

Abstract

Embarrassingly parallel Markov Chain Monte Carlo (MCMC) exploits parallel computing to scale Bayesian inference to large datasets by using a two-step approach. First, MCMC is run in parallel on (sub)posteriors defined on data partitions. Then, a server combines local results. While efficient, this framework is very sensitive to the quality of subposterior sampling. Common sampling problems such as missing modes or misrepresentation of low-density regions are amplified – instead of being corrected – in the combination phase, leading to catastrophic failures. In this work, we propose a novel combination strategy to mitigate this issue. Our strategy, Parallel Active Inference (PAI), leverages Gaussian Process (GP) surrogate modeling and active learning. After fitting GPs to subposteriors, PAI (i) shares information between GP surrogates to cover missing modes; and (ii) uses active sampling to individually refine subposterior approximations. We validate PAI in challenging benchmarks, including heavy-tailed and multi-modal posteriors and a real-world application to computational neuroscience. Empirical results show that PAI succeeds where previous methods catastrophically fail, with a small communication overhead.

1 INTRODUCTION

Markov Chain Monte Carlo (MCMC) methods have become a gold standard in Bayesian statistics (Gelman et al., 2013; Carpenter et al., 2017). However, scaling MCMC methods to large datasets is challenging due to their sequential nature and that they typically require many likelihood evaluations, implying repeated sweeps through the data. Various approaches that leverage distributed computing have been proposed to mitigate these limitations (Angelino et al., 2016; Robert et al., 2018). In general, we can split these approaches between those that incur constant communication costs and those requiring frequent interaction between server and computing nodes (Vehtari et al., 2020).

Embarrassingly parallel MCMC (Neiswanger et al., 2014) is a popular class of methods which employs a divide-and-conquer strategy to sample from a target posterior, requiring only a single communication step. For dataset $\mathcal{D}$ and model parameters $\theta \in \mathbb{R}^D$, suppose we are interested in the Bayesian posterior $p\left(\theta|\mathcal{D}\right) \propto p(\theta)p(\mathcal{D}|\theta)$, where $p(\theta)$ is the prior and $p(\mathcal{D}|\theta)$ the likelihood. Embarrassingly parallel methods begin by splitting the data $\mathcal{D}$ into K smaller partitions $\mathcal{D}_1, \ldots, \mathcal{D}_K$ so that we can rewrite the posterior as

$$p\left(\theta|\mathcal{D}\right) \propto \prod_{k=1}^{K} p(\theta)^{1/K} p(\mathcal{D}_k|\theta) \equiv \prod_{k=1}^{K} p_k(\theta). \qquad (1)$$

Next, an MCMC sampler is used to draw samples $\mathcal{S}_k$ from each *subposterior* $p_k(\theta)$, for $k = 1 \ldots K$, in parallel. Then, the computing nodes send the local results to a central server, for a final aggregation step. These local results are either the samples themselves or approximations $q_1, \ldots, q_K$ built using them.

Works in embarrassingly parallel MCMC mostly focus on combination strategies. Scott et al. (2016) employ a weighted average of subposterior samples. Neiswanger et al. (2014) propose using multivariate-normal surrogates as well as non-parametric and semi-parametric forms. Wang et al. (2015) combine subposterior samples into a hyper-histogram with random partition trees. Nemeth and Sherlock (2018) leverage density values computed during MCMC to fit Gaussian process (GP) surrogates to log-subposteriors. Mesquita et al. (2019) use subposterior samples to fit normalizing flows and apply importance sampling to draw from their product.

Despite these advances, parallel MCMC suffers from an unaddressed limitation: its dependence on high-quality subposterior sampling. This requirement is especially

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

difficult to meet when subposteriors are multi-modal or heavy-tailed, in which cases MCMC chains often visit only a subset of modes and may underrepresent low-density regions. Furthermore, the surrogates $(q_k)_{k=1}^K$ built only on local MCMC samples might match poorly the true subposteriors if not carefully tuned.

Outline and contributions. We first discuss the failure modes of parallel MCMC (Section 2). Drawing insight from this discussion, Section 3 proposes a novel GP-based solution, Parallel Active Inference (PAI). After fitting the subposterior surrogates, PAI shares a subset of samples between computing nodes to prevent mode collapse. PAI also uses active learning to refine low-density regions and avoid catastrophic model mismatch. Section 4 validates our method on challenging benchmarks and a real-world neuroscience example. Finally, Section 5 reviews related work and Section 6 discusses strengths and limitations of PAI.

2 EMBARRASSINGLY PARALLEL MCMC: HOW CAN IT FAIL?

We recall the basic structure of a generic embarrassingly parallel MCMC algorithm in Algorithm 1. This schema has major failure modes that we list below, before discussing potential solutions. We also illustrate these pathologies in Fig 1. In this paper, for a function f with scalar output and a set of points $\mathcal{S} = \{s_1, \ldots, s_N\}$, we denote with $f(\mathcal{S}) \equiv \{f(s_1), \ldots, f(s_N)\}$.

Algorithm 1 Generic embarrassingly parallel MCMC

```
Input: Data partitions $\mathcal{D}_1, \ldots, \mathcal{D}_K$; prior $p(\theta)$; likelihood function $p(\mathcal{D}|\theta)$.
1: parfor $1 \ldots K$ do                                   ▷ Parallel steps
2:     $\mathcal{S}_k \leftarrow$ MCMC samples from $p_k(\theta) \propto p(\theta)^{1/K} p(\mathcal{D}_k|\theta)$
3:     build subposterior model $q_k(\theta)$ from $\mathcal{S}_k$
4: end parfor
5: Combine: $q(\theta) \propto \prod_{k=1}^K q_k(\theta)$            ▷ Centralized step
```

2.1 Failure modes

I: Mode collapse. It is sufficient that *one* subposterior q_k misses a mode for the combined posterior q to lack that mode (see Fig 1A). While dealing with multiple modes is an open problem for MCMC, here the issue is exacerbated by the fact that a single failure propagates to the final solution. A back-of-the-envelope calculation shows that even if the chance of missing a mode is small, $\varepsilon > 0$, the probability of mode collapse in the combined posterior is $\approx (1 - \varepsilon)^K$ making it a likely occurrence for sufficiently large K.

Insight 1: For multimodal posteriors, mode collapse is almost inevitable unless the computing nodes can exchange information about the location of important posterior regions.

II: Catastrophic model mismatch. Since the q_k are approximations of the true subposteriors p_k, small deviations between them are expected – this is not what we refer to here. Instead, an example of *catastrophic* model mismatch is when a simple parametric model such as a multivariate normal is used to model a multimodal posterior with separate modes (see Section 4). Even nonparametric methods can be victims of this failure. For example, GP surrogates are often used to model nonparametric deviations of the log posterior from a parametric 'mean function'. While these models can well represent multimodal posteriors, care is needed to avoid grossly mismatched solutions in which a q_k 'hallucinates' posterior mass due to an improper placement of the GP mean function (Fig 1B).

Insight 2: We cannot take subposterior models at face value. Reliable algorithms should check and refine the q_k's to avoid catastrophic failures.

III: Underrepresented tails. This effect is more subtle than the failure modes listed above, but it contributes to accumulating errors in the estimate of the combined posterior. The main issue here is that, by construction, MCMC samples and subposterior models based on these samples focus on providing information about the high-posterior-mass region of the subposterior. However, different subposteriors may overlap only in their tail regions (Fig 1C), implying that the tails and the nearby 'deep' regions of each subposterior might actually be the most important in determining the exact shape of the combined posterior.

Insight 3: Subposterior models built only from MCMC samples (and their log density) can miss important information about the tails and nearby regions of the subposterior which would also contribute to the combined posterior.

2.2 Past solutions

Since there is no guarantee that q approximates well the posterior p, Nemeth and Sherlock (2018) refine q with an additional parallel step, called Distributed Importance Sampler (DIS). DIS uses q as a proposal distribution and draws samples $\mathcal{S} \sim q$, that are then sent back for evaluation of the log density $\log p_k(\mathcal{S})$ at each parallel node. The true log density $\log p(\mathcal{S}) = \sum_k \log p_k(\mathcal{S})$ is then used as a target for *importance sampling/resampling* (Robert and Casella, 2013). Technically, this step makes the algorithm not 'embarrassingly parallel' anymore, but the prospect of fixing catastrophic failures outweighs the additional communication cost. However, DIS does not fully solve the issues raised in Section 2.1. Notably, importance sampling will not

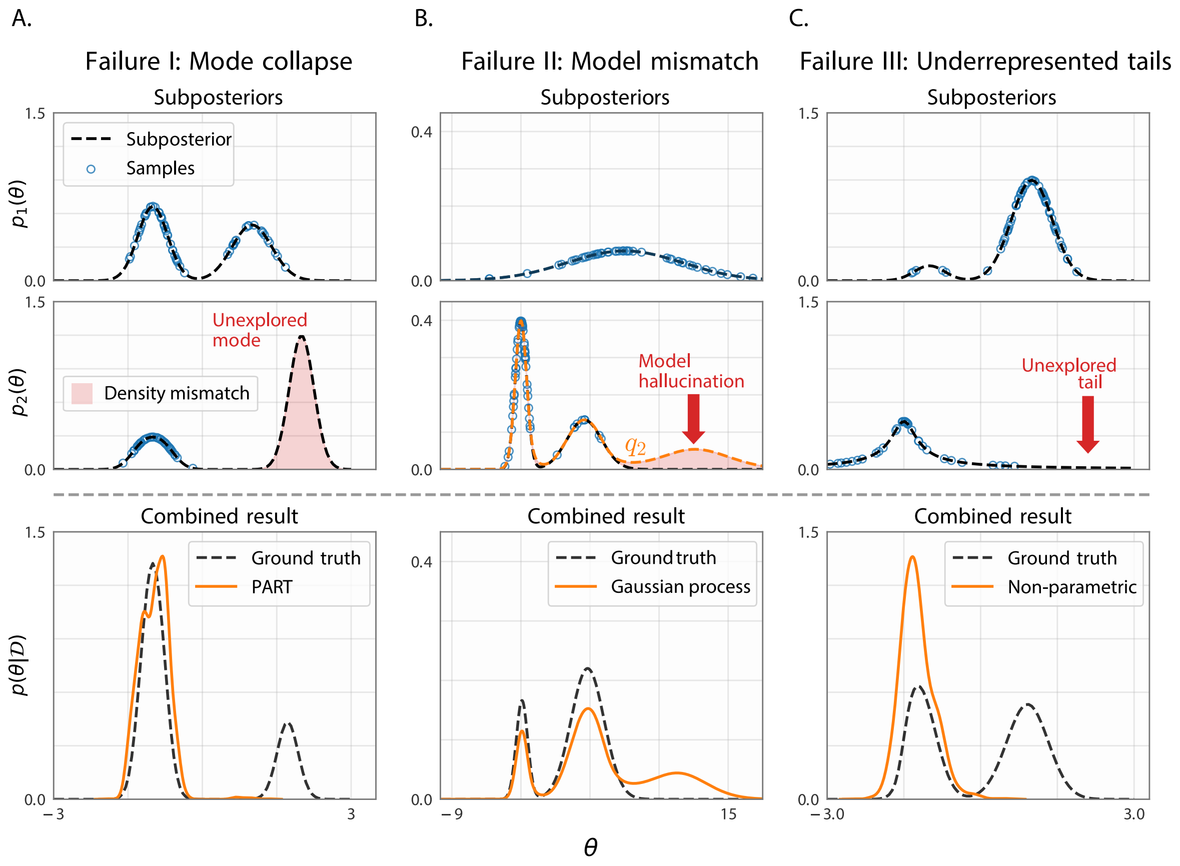


Figure 1: **Failure modes of embarrassingly parallel MCMC. A–C**. Each column illustrates a distinct failure type described in Section 2.1. For each column, the top rows show two subposteriors $p_k(\theta)$ (black dashed line: ground truth; blue circles: MCMC samples), and the bottom row shows the full posterior $p(\theta|\mathcal{D})$ with the approximate posterior combined using the method specified in the legend (orange line). These failure modes are general and not unique to the displayed algorithms (see Appendix A for details and further explanations).

help recover the missing regions of the posterior if q does not cover them in the first place. DIS can help in some model mismatch cases, in that 'hallucinated' regions of the posterior will receive near-zero weights after the true density is retrieved.

2.3 Proposed solution

Drawing from the insights in Section 2.1, we propose two key ideas to address the blind spots of embarrassingly parallel MCMC. Here we provide an overview of our solution, which is described in detail in Section 3. The starting point is modeling subposteriors via Gaussian process surrogates (Fig 2A).

Sample sharing. We introduce an additional step in which each node shares a selected subset of MCMC samples with the others (Fig 2B). This step provides sufficient information for local nodes to address mode collapse and underrepresented tails. While this communication step makes our method not strictly 'embarrassingly' parallel, we argue it is necessary to avoid posterior density collapse. Moreover, existing methods already consider an extra communication step (Nemeth and Sherlock, 2018), as mentioned in Section 2.2.

Active learning. We use *active learning* as a general principle whenever applicable. The general idea is to select points that are informative about the shape of the subposterior, minimizing the additional communication required. Active learning is used here in multiple steps: when selecting samples from MCMC to build the surrogate model (as opposed to thinning or random subsampling); as a way to choose which samples from other nodes to add to the current surrogate model of each subposterior q_k (only informative samples are added); to actively sample *new* points to reduce uncertainty in the local surrogate q_k (Fig 2C). Active learning contributes to addressing both catastrophic model mismatch and underrepresented tails.

Combined, these ideas solve the failure modes discussed previously (Fig 2D).

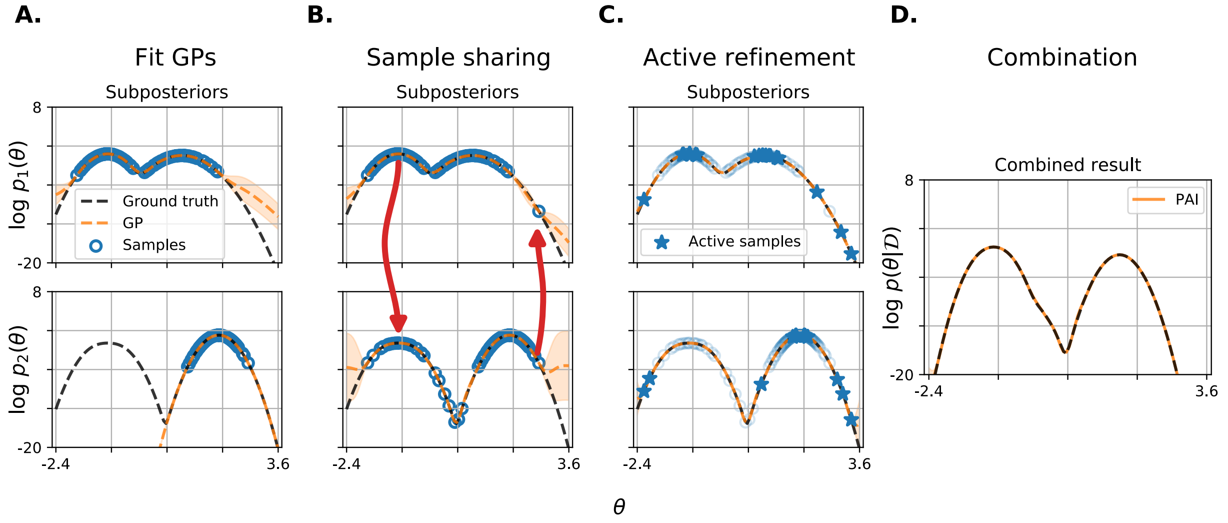


Figure 2: **Parallel active inference (PAI). A.** Each log subposterior $\log p_k(\theta)$ (black dashed line) is modeled via Gaussian process surrogates (orange dashed line: mean GP; shaded area: 95% confidence interval) trained on MCMC samples $\mathcal{S}_k$ (blue circles) and their log-density $\log p_k(\mathcal{S}_k)$. Here, MCMC sampling on the second subposterior has missed a mode. **B.** Selected subsets of MCMC samples are shared across nodes, evaluated locally and added to the GP surrogates. Here, sample sharing helps finding the missing mode in the second subposterior, but the GP surrogate is now highly uncertain outside the samples. **C.** Subposteriors are refined by actively selecting new samples (stars) that resolve uncertainty in the surrogates. **D.** Subposteriors are combined into the full approximate log posterior (orange line); here a perfect match to the true log posterior (black dashed line).

3 PARALLEL ACTIVE INFERENCE

In this section, we present our framework, which we call Parallel Active Inference (PAI), designed to address the issues discussed in Section 2. The steps of our method are schematically explained in Fig 2 and the detailed algorithm is provided in Appendix C.

3.1 Subposterior modeling via GP regression

As per standard embarrassingly parallel algorithms, we assume each node computes a set of samples $\mathcal{S}_k$ and their log density, $\log p_k(\mathcal{S}_k)$, by running MCMC on the subposterior p_k. We model each log-subposterior $\mathcal{L}_k(\theta) \equiv \log q_k(\theta)$ using GP regression (Fig 2A; see Rasmussen and Williams (2006); Nemeth and Sherlock (2018); Görtler et al. (2019) and Appendix B for more information). We say that a GP surrogate model is trained on $\mathcal{S}_k$ as a shorthand for $(\mathcal{S}_k, \log p_k(\mathcal{S}_k))$.

When building the GP model of the subposterior, it is not advisable to use all samples $\mathcal{S}_k$ because: (1) exact inference in GPs scales cubically in the number of training points (although see Wang et al. (2019)); (2) we want to limit communication costs when sharing samples between nodes; (3) there is likely high redundancy in $\mathcal{S}_k$ about the shape of the subposterior. Nemeth and Sherlock (2018) simply choose a subset of samples by 'thinning' a longer MCMC chain at regular intervals. Instead, we employ *active subsampling* as follows.

First, we pick an initial subset of n_0 samples $\mathcal{S}_k^{(0)} \subset \mathcal{S}_k$, that we use to train an initial GP (details in Appendix C.1). Then, we iteratively select points $\theta^\star$ from $\mathcal{S}_k$ by maximizing the *maximum interquantile range* (MAXIQR) acquisition function (Järvenpää et al., 2021):

$$\theta^\star = \arg\max_\theta \left\{ e^{m(\theta;\mathcal{S}_k^{(t)})} \sinh\left(u \cdot s(\theta;\mathcal{S}_k^{(t)})\right) \right\}, \quad (2)$$

where $m(\theta;\mathcal{S}_k^{(t)})$ and $s(\theta;\mathcal{S}_k^{(t)})$ are, respectively, the posterior latent mean and posterior latent standard deviation of the GP at the end of iteration $t \geq 0$; and $\sinh(z) = (\exp(z) - \exp(-z))/2$ for $z \in \mathbb{R}$ is the hyperbolic sine. Eq. 2 promotes selection of points with high posterior density for which the GP surrogate is also highly uncertain, with the trade-off controlled by $u > 0$, where larger values of u favor further exploration. In each iteration $t+1$, we greedily select a batch of n_{batch} points at a time from $\mathcal{S}_k \setminus \mathcal{S}_k^{(t)}$ using a batch version of MAXIQR (Järvenpää et al., 2021). We add the selected points to the current training set, $\mathcal{S}_k^{(t+1)}$, and retrain the GP after each iteration (see Appendix C.1). After T iterations, our procedure yields a subset of points $\mathcal{S}'_k \equiv \mathcal{S}_k^{(T)} \subseteq \mathcal{S}_k$ that are highly informative about the shape of the subposterior.

3.2 Sample sharing

In this step, each node k shares the selected samples $\mathcal{S}'_k$ with all other nodes (Fig 2B). Thus, node k gains access to the samples $\mathcal{S}'_{\setminus k} \equiv \bigcup_{j \neq k} \mathcal{S}'_j$. Importantly, $\mathcal{S}'_{\setminus k}$ might contain samples from relevant subposterior regions that node k has has not explored. As discussed in Section 3.1, for efficiency we avoid adding *all* points $\mathcal{S}'_{\setminus k}$ to the current GP surrogate for subposterior k. Instead, we add a sample $\theta^\star \in \mathcal{S}'_{\setminus k}$ to the GP training set only if the prediction of the current GP surrogate deviates from the true subposterior $\log p_k(\theta^\star)$ in a significant way (see Appendix C.2 for details). After this step, we obtain an expanded set of points $\mathcal{S}''_k$ that includes information from all the nodes, minimizing the risk of mode collapse (see Section 2.1).

3.3 Active subposterior refinement

So far, the GP models have been trained using selected subsets of samples from the original MCMC runs. In this step, we refine the GP model of each subposterior by sampling *new* points (Fig 2C). Specifically, each node k actively selects new points by optimizing the MAXIQR acquisition function (Eq. 2) over $\mathcal{X} \subseteq \mathbb{R}^D$ (see Appendix C.3). New points are selected greedily in batches of size n_{batch}, retraining the GP after each iteration. This procedure yields a refined set of points $\mathcal{S}'''_k$ which includes new points that better pinpoint the shape of the subposterior, reducing the risk of catastrophic model mismatch and underrepresented tails. The final log-subposterior surrogate model $\mathcal{L}_k$ is the GP trained on $\mathcal{S}'''_k$.

3.4 Combining the subposteriors

Finally, we approximate the full posterior $\log p\left(\theta|\mathcal{D}\right) = \sum_{k=1}^{K} \log p_k(\theta)$ by combining all subposteriors together (Fig 2D). Since each log-subposterior is approximated by a GP, the approximate full log-posterior is a sum of GPs and itself a GP, $\mathcal{L}(\theta) = \sum_{k=1}^{K} \mathcal{L}_k(\theta)$. Note that $\mathcal{L}(\theta)$, being a GP, is still a distribution over functions. We want then to obtain a point estimate for the (unnormalized) posterior density corresponding to $\exp \mathcal{L}(\theta)$. One choice is to take the posterior mean, which leads to the expectation of a log-normal density (Nemeth and Sherlock, 2018). We prefer a robust estimate and use the posterior median instead (Järvenpää et al., 2021). Thus, our estimate is

$$q(\theta) \propto \exp\left\{ \sum_{k=1}^{K} m_k(\theta; \mathcal{S}'''_k) \right\}. \tag{3}$$

In low dimension ($D = 1, 2$), Eq. 3 can be evaluated on a grid. In higher dimension, one could sample from $q(\theta)$ using MCMC methods such as NUTS (Hoffman and Gelman, 2014), as done by Nemeth and Sherlock (2018). However, $q(\theta)$ is potentially multimodal which does not lend itself to easy MCMC sampling. Alternatively, Acerbi (2018) runs variational inference on $q(\theta)$ using as variational distribution a mixture of Gaussians with a large number of components. Finally, for moderate D, importance sampling/resampling with an appropriate (adaptive) proposal would also be feasible.

As a final optional step, after combining the subposteriors into the full approximate posterior $q(\theta)$, we can refine the solution using *distributed importance sampling* (DIS) as proposed by Nemeth and Sherlock (2018) and discussed in Section 2.2.

3.5 Complexity analysis

Similarly to conventional embarrassingly parallel MCMC, we can split the cost of running PAI into two main components. The first consists of local costs, which involve computations happening at individual computing nodes (i.e., model fitting and active refinement). The second are global (or aggregation) costs, which comprise communication and sampling from the combined approximate posterior.

3.5.1 Model fitting

After sampling their subposterior, each computing node k has to fit the surrogate model on the subset of their samples, $\mathcal{S}'_k$. These subsets are designed such that their size is $\mathcal{O}(D)$ (see Appendix C). Thus, the cost of fitting the surrogate GP models in each of the K computing nodes is $\mathcal{O}(D^3)$. The same applies for $\mathcal{S}''_k$ and $\mathcal{S}'''_k$.

3.5.2 Communication costs

Traditional embarrassingly parallel MCMC methods only require two global communication steps: (1) the central server splits the N observations among K computing nodes; (2) each node sends S subposterior samples of dimension D back to the server, assuming nodes draw the same number of samples. Together, these steps amount to $\mathrm{O}(N + KSD)$ communication cost.

PAI imposes another communication step, in which nodes share their subposterior samples and incurring $\mathrm{O}(KSD)$ cost. Furthermore, supposing PAI acquires A active samples to refine each subposterior, the cost of sending local results to servers is increased by $\mathrm{O}(KAD)$. PAI also incur a small additional cost for sending back the value of the GP hyperparameters $\mathrm{O}(KD)$. In sum, since usually $A \ll S$, the asymptotic communication cost of PAI is equivalent to traditional methods.

3.5.3 Active refinement costs

Active learning involves GP training and optimization of the acquisition function, but only a small number of likelihood evaluations. Thus, under the embarrassingly parallel MCMC assumption that likelihood evaluations are costly (e.g., due to large datasets), active learning is relatively inexpensive (Acerbi, 2018). More importantly, as shown in our ablation study in Appendix D.1, this step is crucial to avoid the pathologies of embarrassingly parallel MCMC.

3.5.4 Sampling complexity

Sampling from the aggregate approximate posterior $q(\theta)$ only requires evaluating the GP predictive mean for each subposterior and does not require access to the data or all samples. The sampling cost is linear in the number of subposteriors K and the size of each GP $\mathcal{O}(D)$. Even if K is chosen to scale as the size of the actual data, each GP only requires a small training set, making them comparably inexpensive.

4 EXPERIMENTS

We evaluate PAI on a series of target posteriors with different challenging features. Subsection 4.1 shows results for a posterior with four distinct modes, which is prone to mode collapse (Fig 1A). Subsection 4.2 targets a posterior with heavy tails, which can lead to underrepresented tails (Fig 1C). Subsection 4.3 uses a rare event model to gauge how well our method performs when the true subposteriors are drastically different. Finally, Subsection 4.4 concludes with a real-world application to a model and data from computational neuroscience (Acerbi et al., 2018). We provide implementation details in Appendix B.3 and source code is available at `https://github.com/spectraldani/pai`.

Algorithms. We compare basic PAI and PAI with the optional distributed importance sampling step (PAI-DIS) against six popular and state-of-the-art (SOTA) embarrassingly parallel MCMC methods: the parametric, non-parametric and semi-parametric methods by Neiswanger et al. (2014); PART (Wang et al., 2015); and two other GP-surrogate methods (Nemeth and Sherlock, 2018), one using a simple combination of GPs (GP) and the other using the distributed importance sampler (GP-DIS; see Section 2.2).

Procedure. For each problem, we randomly split the data in equal-sized partitions and divide the target posterior into K subposteriors (Eq. 1). We run MCMC separately on each subposterior using Stan with multiple chains (Carpenter et al., 2017). The same MCMC output is then processed by the different algorithms described above, yielding a combined approximate posterior for each method. To assess the quality of each posterior approximation, we compute the mean marginal total variation distance (MMTV), the 2-Wasserstein (W2) distance, and the Gaussianized symmetrized Kullback-Leibler (GsKL) divergence between the approximate and the true posterior, with each metric focusing on different features. For each problem, we computed ground-truth posteriors using numerical integration (for $D \leq 2$) or via extensive MCMC sampling in Stan (Carpenter et al., 2017). For all GP-based methods (PAI, PAI-DIS, GP, GP-DIS), we sampled from the potentially multimodal combined GP (Eq. 3) using importance sampling/resampling with an appropriate proposal. We report results as mean ± standard deviation across ten runs in which the entire procedure was repeated with different random seeds. For all metrics, lower is better, and the best (statistically significant) results for each metric are reported in bold. See Appendix D.2 for more details.

4.1 Multi-modal posterior

Setting. In this synthetic example, the data consist of $N = 10^3$ samples $y_1, \ldots, y_N$ drawn from the following hierarchical model:

$$\theta \sim p(\theta) = \mathcal{N}(0, \sigma_p^2 \mathbb{I}_2)$$

$$y_1, \ldots, y_N \sim p(y_n|\theta) = \sum_{i=1}^{2} \frac{1}{2} \mathcal{N}\left(P_i(\theta_i), \sigma_l^2\right)$$

where $\theta \in \mathbb{R}^2$, $\sigma_p = \sigma_l = 1/4$ and P_i's are second-degree polynomial functions. By construction, the target posterior $p(\theta|y_1, \ldots, y_N) \propto p(\theta) \prod_{n=1}^{N} p(y_n|\theta)$ is multimodal with four modes. We run parallel inference on $K = 10$ equal-sized partitions of the data. We provide more details regarding P_1, P_2 in Appendix D.3.

Results. Fig 3 shows the output of each parallel MCMC method for a typical run, displayed as samples from the approximate combined posterior overlaid on top of the ground-truth posterior. Due to MCMC occasionally missing modes in subposterior sampling, the combined posteriors from all methods except PAI lack at least one mode of the posterior (*mode collapse*, as seen in Fig 1A). Other methods also often inappropriately distribute mass in low-density regions (as seen in Fig 1B). In contrast, PAI accurately recovers all the high-density regions of the posterior achieving a near-perfect match. Table 1 shows that PAI consistently outperforms the other methods in terms of metrics.

Table 1: **Multi-modal posterior.**

Model	**MMTV**	**W2**	**GsKL**
Parametric	$0.89_{\pm 0.12}$	$1.08_{\pm 0.33}$	$8.9_{\pm 11} \times 10^{2}$
Semi-param.	$0.81_{\pm 0.09}$	$1.08_{\pm 0.12}$	$5.6_{\pm 1.3} \times 10^{1}$
Non-param.	$0.81_{\pm 0.09}$	$1.12_{\pm 0.09}$	$5.0_{\pm 1.8} \times 10^{1}$
PART	$0.55_{\pm 0.09}$	$1.06_{\pm 0.33}$	$7.3_{\pm 14} \times 10^{2}$
GP	$0.93_{\pm 0.16}$	$1.01_{\pm 0.36}$	$1.2_{\pm 1.3} \times 10^{4}$
GP-DIS	$0.87_{\pm 0.18}$	$1.04_{\pm 0.34}$	$4.8_{\pm 14} \times 10^{16}$
PAI	$\mathbf{0.037}_{\pm 0.011}$	$\mathbf{0.028}_{\pm 0.011}$	$\mathbf{1.6}_{\pm 1.7} \times \mathbf{10^{-4}}$
PAI-DIS	$\mathbf{0.034}_{\pm 0.019}$	$\mathbf{0.026}_{\pm 0.008}$	$\mathbf{3.9}_{\pm 2.4} \times \mathbf{10^{-5}}$

Large datasets. To illustrate the computational benefits of using PAI for larger datasets, we repeated the same experiment in this section but with 10^5 data points in each of the $K = 10$ partitions. Remarkably, even for this moderate-sized dataset, we notice a 6× speed-up – decreasing the total running time from 6 hours to 57 minutes, (50 for subposterior sampling + 7 from PAI; see Appendix D.4). Overall, PAI's running

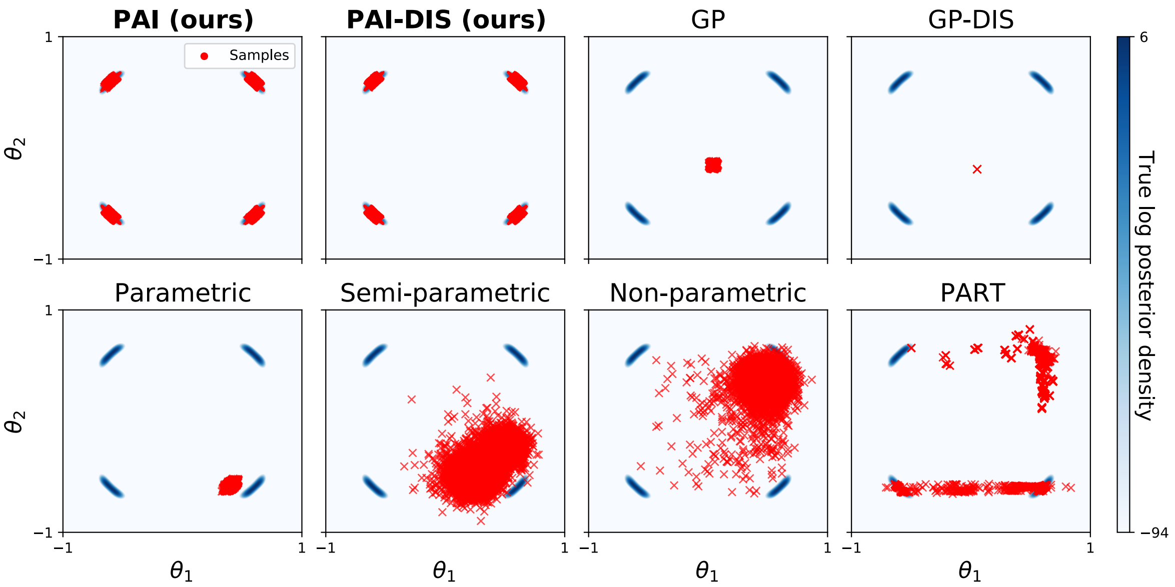


Figure 3: **Multi-modal posterior.** Each panel shows samples from the combined approximate posterior (red) against the ground truth (blue). With exception of PAI, all methods completely miss at least one high-density region. Moreover, PAI is the only method that does not assign mass to regions without modes.

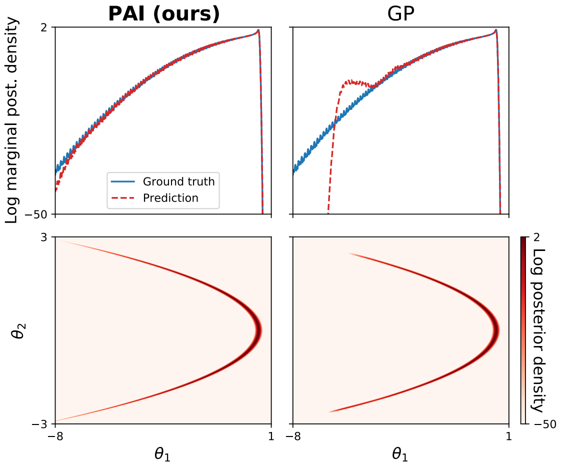


Figure 4: **Warped Student's t.** Top: Log marginal posterior for θ_1. Bottom: Log posterior density. Thanks to active sampling, PAI better captures details in the depths of the tails.

time is in the same order of magnitude as the previous SOTA (e.g. Wang et al., 2015; Nemeth and Sherlock, 2018). However, only PAI returns correct results while other methods fail.

4.2 Warped Student's t

Setting. We now turn to a synthetic example with heavy tails. Consider the following hierarchical model:

$$\theta \sim p(\theta) = \mathcal{N}(0, \sigma_p^2 \mathbb{I}_2)$$
$$y_1, \ldots, y_N \sim p(y_n|\theta) = \mathrm{StudentT}\left(\nu, \theta_1 + \theta_2^2, \sigma_l^2\right)$$

where $\theta \in \mathbb{R}^2$, $\nu = 5$ is the degrees of freedom of the Student's t-distribution, $\sigma_p = 1$, and $\sigma_l = \sqrt{2}$. This model is a heavy-tailed variant of the Warped Gaussian model studied in earlier work, e.g., Nemeth and Sherlock (2018); Mesquita et al. (2019). As before, we generate $N = 10^3$ samples and split the data into $K = 10$ partitions for parallel inference.

Results. Fig 4 shows the full posterior and the marginal posterior for θ_1 obtained using the two best-performing methods without DIS refinement, PAI and GP (see Table 2). While PAI(-DIS) is very similiar to GP(-DIS) in terms of metrics, Fig 4 shows that, unlike GP(-DIS), PAI accurately captures the far tails of the posterior which could be useful for downstream tasks, avoiding failure mode III (Fig 1C).

Table 2: **Warped Student's t.**

Model	**MMTV**	**W2**	**GsKL**
Parametric	$0.51{\scriptstyle \pm 0.01}$	$0.71{\scriptstyle \pm 0.07}$	$1.9{\scriptstyle \pm 0.1} \times 10^{0}$
Semi-param.	$0.50{\scriptstyle \pm 0.03}$	$0.57{\scriptstyle \pm 0.05}$	$1.1{\scriptstyle \pm 0.2} \times 10^{1}$
Non-param.	$0.51{\scriptstyle \pm 0.02}$	$0.59{\scriptstyle \pm 0.03}$	$1.2{\scriptstyle \pm 0.2} \times 10^{1}$
PART	$0.66{\scriptstyle \pm 0.07}$	$0.78{\scriptstyle \pm 0.09}$	$1.2{\scriptstyle \pm 0.7} \times 10^{2}$
GP	$\mathbf{0.015}{\scriptstyle \pm \mathbf{0.003}}$	$\mathbf{0.003}{\scriptstyle \pm \mathbf{0.002}}$	$4.5{\scriptstyle \pm 10.5} \times 10^{-4}$
GP-DIS	$0.018{\scriptstyle \pm 0.004}$	$\mathbf{0.002}{\scriptstyle \pm \mathbf{0.001}}$	$6.6{\scriptstyle \pm 5.8} \times 10^{-5}$
PAI	$\mathbf{0.015}{\scriptstyle \pm \mathbf{0.003}}$	$\mathbf{0.002}{\scriptstyle \pm \mathbf{0.001}}$	$\mathbf{1.2}{\scriptstyle \pm \mathbf{0.8}} \times \mathbf{10^{-5}}$
PAI-DIS	$0.018{\scriptstyle \pm 0.003}$	$\mathbf{0.002}{\scriptstyle \pm \mathbf{0.001}}$	$3.8{\scriptstyle \pm 3.4} \times 10^{-5}$

4.3 Rare categorical events

Setting. To evaluate how our method copes with heterogeneous subposteriors, we run parallel inference for a synthetic example with Categorical likelihood and $N = 10^3$ discrete observations split among three classes. To enforce heterogeneity, we make the first two classes rare (true probability $\theta_1 = \theta_2 = 1/N$) and the remaining class much more likely (true probability $\theta_3 = (N-2)/N$). Since we split the data into $K = 10$ equal-sized parts, some of them will not contain even a single rare event. We perform inference over $\theta \in \Delta^2$ (probability 2-simplex) with a symmetric Dirichlet prior with concentration parameter $\alpha = 1/3$.

Results. Fig 5 shows the samples from the combined approximate posterior for each method. In this example, PAI-DIS matches the shape of the target posterior extremely well, followed closely by GP-DIS (see also Table 3). Notably, even standard PAI (without the DIS correction) produces a very good approximation of the posterior – a further display of the ability of PAI of capturing fine details of each subposterior, particularly important here in the combination step due to the heterogeneous subposteriors. By contrast, the other methods end up placing excessive mass in very-low-density regions (PART, Parametric, GP) or over-concentrating (Non-parametric, Semi-parametric).

Table 3: **Rare categorical events.**

Model	**MMTV**	**W2**	**GsKL**
Parametric	$0.26_{\pm 0.14}$	$0.15_{\pm 0.19}$	$1.1_{\pm 1.4} \times 10^{0}$
Semi-param.	$0.49_{\pm 0.21}$	$0.27_{\pm 0.23}$	$3.5_{\pm 3.4} \times 10^{0}$
Non-param.	$0.43_{\pm 0.17}$	$0.19_{\pm 0.25}$	$2.8_{\pm 3.9} \times 10^{0}$
PART	$0.31_{\pm 0.14}$	$0.08_{\pm 0.13}$	$8.6_{\pm 10} \times 10^{-1}$
GP	$0.16_{\pm 0.09}$	$0.04_{\pm 0.07}$	$3.5_{\pm 4.8} \times 10^{-1}$
GP-DIS	$0.011_{\pm 0.002}$	$6.3_{\pm 0.9} \times 10^{-4}$	$1.1_{\pm 1.5} \times 10^{-4}$
PAI	$0.028_{\pm 0.027}$	$0.001_{\pm 0.002}$	$8.0_{\pm 16} \times 10^{-3}$
PAI-DIS	$\mathbf{0.009}_{\pm \mathbf{0.002}}$	$\mathbf{5.4}_{\pm \mathbf{0.8}} \times \mathbf{10^{-4}}$	$\mathbf{4.3}_{\pm \mathbf{2.1}} \times \mathbf{10^{-5}}$

4.4 Multisensory causal inference

Setting. *Causal inference* (CI) in multisensory perception denotes the process whereby the brain decides whether distinct sensory cues come from the same source, a commonly studied problem in computational and cognitive neuroscience (Körding et al., 2007). Here we compute the posterior for a 6-parameter CI model given the data of subject S1 from (Acerbi et al., 2018) (see Appendix D.3 for model details). The fitted model is a proxy for a large class of similar models that would strongly benefit from parallelization due to likelihoods that do not admit analytical solutions, thus requiring costly numerical integration. For this experiment, we run parallel inference over $K = 5$ partitions of the $N = 1069$ observations in the dataset.

Results. Table 4 shows the outcome metrics of parallel inference. Similarly to the rare-events example, we find that PAI-DIS obtains an excellent approximation of the true posterior, with GP-DIS performing about equally well (slightly worse on the GsKL metric). Despite lacking the DIS refinement step, standard PAI performs competitively, achieving a reasonably good approximation of the true posterior (see Appendix D.3). All the other methods perform considerably worse; in particular the GP method without the DIS step is among the worst-performing methods on this example.

Table 4: **Multisensory causal inference.**

Model	**MMTV**	**W2**	**GsKL**
Parametric	$0.40_{\pm 0.05}$	$4.8_{\pm 0.6}$	$1.2_{\pm 0.4} \times 10^{1}$
Semi-param.	$0.68_{\pm 0.07}$	$9.7_{\pm 9.6}$	$5.6_{\pm 3.2} \times 10^{1}$
Non-param.	$0.26_{\pm 0.02}$	$0.52_{\pm 0.14}$	$5.3_{\pm 0.3} \times 10^{0}$
PART	$0.24_{\pm 0.04}$	$1.5_{\pm 0.5}$	$8.0_{\pm 5.4} \times 10^{0}$
GP	$0.49_{\pm 0.25}$	$17_{\pm 23}$	$6.3_{\pm 8.9} \times 10^{1}$
GP-DIS	$\mathbf{0.07}_{\pm \mathbf{0.03}}$	$\mathbf{0.16}_{\pm \mathbf{0.07}}$	$8.7_{\pm 14} \times 10^{-1}$
PAI	$0.16_{\pm 0.04}$	$0.56_{\pm 0.21}$	$2.0_{\pm 1.7} \times 10^{0}$
PAI-DIS	$\mathbf{0.05}_{\pm \mathbf{0.04}}$	$\mathbf{0.14}_{\pm \mathbf{0.13}}$	$\mathbf{2.9}_{\pm \mathbf{3.6}} \times \mathbf{10^{-1}}$

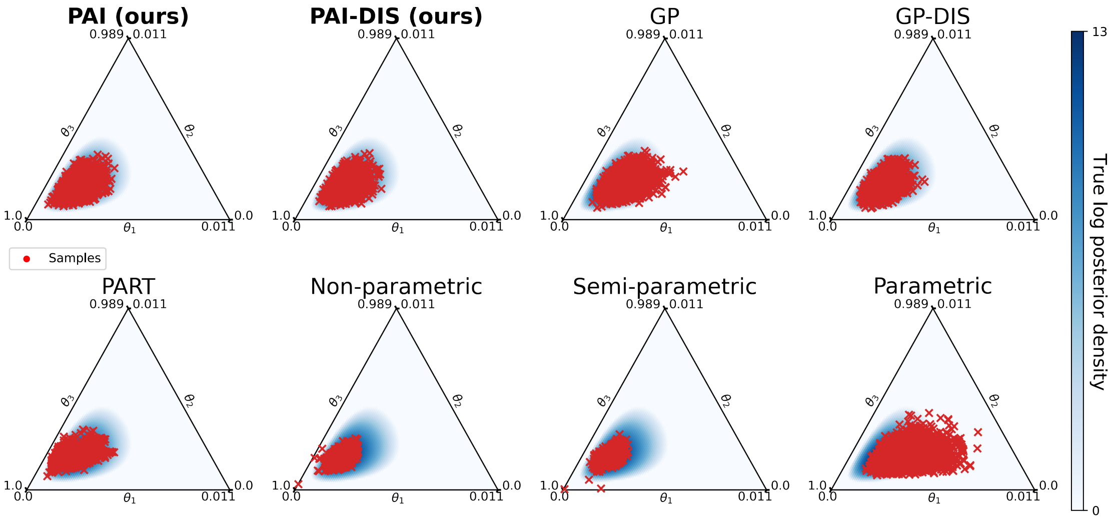
5 RELATED WORKS

While the main staple of embarrassingly parallel MCMC is being a divide-and-conquer algorithm, there are other methods that scale up MCMC using more intensive communication protocols. For instance, Ahn et al. (2014) propose a distributed version of stochatic gradient Langevin dynamics (SGLD, Welling and Teh, 2011) that constantly passes around the chain state to computing nodes, making updates only based on local data. However, distributed SGLD tends to diverge from the posterior when the communications are limited, an issue highlighted by recent work (El Mekkaoui et al., 2021; Vono et al., 2022). Outside the realm of MCMC, there are also works proposing expectation propagation as a framework for inference on partitioned data (Vehtari et al., 2020; Bui et al., 2018).

Our method, PAI, builds on top of related work on GP-based surrogate modeling and active learning for log-likelihoods and log-densities. Prior work used GP models and active sampling to learn the intractable marginal likelihood (Osborne et al., 2012; Gunter et al., 2014) or the posterior (Kandasamy et al., 2015a; Wang and Li, 2018; Järvenpää et al., 2021). Recently, the framework of Variational Bayesian Monte Carlo (VBMC) was introduced to simultaneously compute both the posterior and the marginal likelihood (Acerbi, 2018, 2019, 2020). PAI extends the above works by dealing with partitioned data in the embarrassingly parallel setting, similarly to Nemeth and Sherlock (2018), but with the key addition of active learning and other algorithmic improvements.

6 DISCUSSION

In this paper, we first exposed several potential major failure modes of existing embarrassingly parallel MCMC methods. We then proposed a solution with


PAI (ours)

PAI-DIS (ours)

GP

GP-DIS

PART

Non-parametric

Semi-parametric

Parametric

Figure 5: **Rare categorical events.** Each ternary plot shows samples from the combined approximate posterior (red) on top of the true posterior (blue). Note that the panels are zoomed in on the relevant corner of the probability simplex. Of all methods, PAI is the one that best captures the shape of the posterior.

our new method, *parallel active inference* (PAI), which incorporates two key strategies: sample sharing and active learning. On a series of challenging benchmarks, we demonstrated that 'vanilla' PAI is competitive with current state-of-the-art parallel MCMC methods and deals successfully with scenarios (e.g., multi-modal posteriors) in which all other methods catastrophically fail. When paired with an optional refinement step (PAI-DIS), the proposed method consistently performs on par with or better than state-of-the-art. Our results show the promise of the proposed strategies to deal with the challenges arising in parallel MCMC. Still, the solution is no silver bullet and several aspects remain open for future research.

6.1 Limitations and future work

The major limitation of our method, a common problem to surrogate-based approaches, is scalability to higher dimensions. Most GP-based approaches for Bayesian posterior inference are limited to up to ~ 10 dimensions, see e.g. Acerbi, 2018, 2020; Järvenpää et al., 2021. Future work could investigate methods to scale GP surrogate modeling to higher dimensions, for example taking inspiration from high-dimensional approaches in Bayesian optimization (e.g., Kandasamy et al., 2015b).

More generally, the validity of any surrogate modeling approach hinges on the ability of the surrogate model to faithfully represent the subposteriors. Active learning helps, but model mismatch in our method is still a potential issue that hints at future work combining PAI with more flexible surrogates such as GPs with more flexible kernels (Wilson and Adams, 2013) or deep neural networks (Mesquita et al., 2019). For the latter, obtaining the uncertainty estimates necessary for active learning would involve Bayesian deep learning techniques (e.g., Maddox et al., 2019).

As discussed before, our approach is not 'embarrassingly' parallel in that it requires a mandatory global communication step in the sample sharing part (see Section 3.2). The presence of additional communication steps seem inevitable to avoid catastrophic failures in parallel MCMC, and has been used before in the literature (e.g., the DIS step of Nemeth and Sherlock, 2018). Our method affords an optional final refinement step (PAI-DIS) which also requires a further global communication step. At the moment, there is no automated diagnostic to determine whether the optional DIS step is needed. Our results show that PAI already performs well without DIS in many cases. Still, future work could include an analysis of the GP surrogate uncertainty to recommend the DIS step when useful.

Acknowledgments

This work was supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI and grants 328400, 325572) and UKRI (Turing AI World-Leading Researcher Fellowship, EP/W002973/1). We also acknowledge the computational resources provided by the Aalto Science-IT Project from Computer Science IT.

References

L. Acerbi. Variational Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

L. Acerbi. Variational Bayesian Monte Carlo with noisy likelihoods. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

L. Acerbi, K. Dokka, D. E. Angelaki, and W. J. Ma. Bayesian comparison of explicit and implicit causal inference strategies in multisensory heading perception. *PLoS Computational Biology*, 14(7):e1006110, 2018.

Luigi Acerbi. An exploration of acquisition and mean functions in Variational Bayesian Monte Carlo. *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference (PMLR)*, 96:1–10, 2019.

S. Ahn, B. Shahbaba, and M. Welling. Distributed stochastic gradient MCMC. In *International Conference on Machine Learning (ICML)*, 2014.

E. Angelino, M. J. Johnson, and R. P. Adams. Patterns of scalable Bayesian inference. *Foundations and Trends in Machine Learning*, 9(2-3):119–247, 2016.

M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. Wilson, and E. Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

T. Bui, C. Nguyen, S. Swaroop, and R. Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *ArXiv:1811.11206*, 2018.

B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 2017.

K. El Mekkaoui, D. Mesquita, P. Blomstedt, and S. Kaski. Federated stochastic gradient Langevin dynamics. In *Uncertainty in Artificial Intelligence (UAI)*, 2021.

J. Gardner, G. Pleiss, D. Bindel, K. Weinberger, and A. Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. CRC press, 2013.

J. Görtler, R. Kehlbeck, and O. Deussen. A visual exploration of Gaussian processes. *Distill*, 4(4):e17, 2019.

Tom Gunter, Michael A Osborne, Roman Garnett, Philipp Hennig, and Stephen J Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. *Advances in Neural Information Processing Systems*, 27: 2789–2797, 2014.

M. D. Hoffman and A. Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

M. Järvenpää, M. U. Gutmann, A. Vehtari, and P. Marttinen. Parallel Gaussian process surrogate Bayesian inference with noisy likelihood evaluations. *Bayesian Analysis*, 16(1):147–178, 2021.

K. Kandasamy, J. Schneider, and B. Póczos. Bayesian active learning for posterior estimation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3605–3611, 2015a.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, pages 295–304. PMLR, 2015b.

Konrad P Körding, Ulrik Beierholm, Wei Ji Ma, Steven Quartz, Joshua B Tenenbaum, and Ladan Shams. Causal inference in multisensory perception. *PLoS one*, 2(9): e943, 2007.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 13153–13164, 2019.

Wesley J Maddox, Sanyam Kapoor, and Andrew Gordon Wilson. When are iterative Gaussian processes reliably accurate? 2021.

D. Mesquita, P. Blomstedt, and S. Kaski. Embarrassingly parallel MCMC using deep invertible transformations. In *Uncertainty in Artificial Intelligence (UAI)*, 2019.

W. Neiswanger, C. Wang, and E. P. Xing. Asymptotically exact, embarrassingly parallel MCMC. In *Uncertainty in Artificial Intelligence (UAI)*, 2014.

C. Nemeth and C. Sherlock. Merging MCMC subposteriors through Gaussian-process approximations. *Bayesian Analysis*, 13(2):507–530, 2018.

Michael Osborne, David K Duvenaud, Roman Garnett, Carl E Rasmussen, Stephen J Roberts, and Zoubin Ghahramani. Active learning of model evidence using Bayesian quadrature. *Advances in Neural Information Processing Systems*, 25:46–54, 2012.

H. Park and C. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36 (2):3336–3341, March 2009. ISSN 0957-4174.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

C. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Science & Business Media, 2013.

C. P. Robert, V. Elvira, N. Tawn, and C. Wu. Accelerating MCMC algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(5):e1435, 2018.

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

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

A. Vehtari, A. Gelman, T. Sivula, P. Jylänki, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich, and C. P. Robert. Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *Journal of Machine Learning Research*, 21(17):1–53, 2020.

M. Vono, V. Plassier, A. Durmus, A. Dieuleveut, and E. Moulines. QLSD: Quantised Langevin Stochastic Dynamics for Bayesian federated learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2022.

Hongqiao Wang and Jinglai Li. Adaptive Gaussian process approximation for Bayesian inference with expensive likelihood functions. *Neural Computation*, pages 1–23, 2018.

K. A. Wang, G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson. Exact Gaussian processes on a million data points. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

X. Wang, F. Guo, K. A. Heller, and D. B. Dunson. Parallelizing MCMC with random partition trees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

M. Welling and Y. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*, 2011.

A. Wilson and R. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning (ICML)*, pages 1067–1075. PMLR, 2013.

Supplementary Material: Parallel MCMC Without Embarrassing Failures

In this Supplement, we include extended explanations, implementation details, and additional results omitted from the main text.

Code for our algorithm and to generate the results in the paper is available at: `https://github.com/spectraldani/pai`.

A Failure modes of embarrassingly parallel MCMC explained

In Section 2.1 of the main text we presented three major failure modes of embarrassingly parallel MCMC (Markov Chain Monte Carlo) methods. These failure modes are illustrated in Fig 1 in the main text. In this section, we further explain these failure types going through Fig 1 in detail.

A.1 Mode collapse (Fig 1A)

Fig 1A illustrates *mode collapse.* In this example, the true subposteriors p_1 and p_2 both have two modes (see Fig 1A, top two panels). However, while in p_1 the two modes are relatively close to each other, in p_2 they are farther apart. Thus, when we run an MCMC chain on p_2, it gets stuck into a single high-density region and is unable to jump to the other mode ('unexplored mode', shaded area). This poor exploration of p_2 is fatal to any standard combination strategy used in parallel MCMC, erasing entire regions of the posterior (Fig 1A, bottom panel). While in this example we use PART (Wang et al., 2015), Section 4.1 shows this common pathology in several other methods as well. Our proposed solution to this failure type is sample sharing (see Section C.2).

A.2 Model mismatch (Fig 1B)

Fig 1B draws attention to *model mismatch* in subposterior surrogates. In this example, MCMC runs smoothly on both subposteriors. However, when we fit regression-based surrogates – here, Gaussian processes (GPs) – to the MCMC samples, the behavior of these models away from subposterior samples can be unpredictable. In our example, the surrogate q_2 hallucinates a mode that does not exist in the true subposterior p_2 ('model hallucination', shaded area in Fig 1B). Our proposed solution to correct this potential issue is to explore uncertain areas of the surrogate using active subposterior sampling (see Section C.3).

In this example, we used a simple GP surrogate approach for illustration purposes. The more sophisticated GP-based Distributed Importance Sampler (GP-DIS; Nemeth and Sherlock (2018)) might seem to provide an alternative solution to model hallucination, in that the hallucinated regions with low true density would be down-weighted by importance sampling. However, the DIS step only works if there are 'good' samples that cover regions with high true density that can be up-weighted. If no such samples are present, importance sampling will not work. As an example of this failure, Fig 3 in the main text shows that GP-DIS (Nemeth and Sherlock, 2018) does not recover from the model hallucination (if anything, the importance sampling step concentrates the hallucinated posterior even more).

A.3 Underrepresented tails (Fig 1C)

Fig 1C shows how neglecting low-density regions (*underrepresented tails*) can affect the performance of parallel MCMC. In the example, the true subposterior p_2 has long tails that are not thoroughly represented by MCMC samples. This under-representation is due both to actual difficulty of the MCMC sampler in exploring the tails, and to mere Monte Carlo noise as there is little (although non-zero) mass in the tails, so the number of samples is low. Consequently, the surrogate q_2 is likely to underestimate the density in this unexplored region, in which the other subposterior p_1 has considerable mass. In this case, even though q_1 is a perfect fit to p_1, the product $q_1(\theta)q_2(\theta)$ mistakenly attributes near-zero density to the combined posterior in said region (Fig 1C, bottom panel). In our method, we address this issue via multiple solutions, in that both sample sharing and active sampling would help uncover underrepresentation of the tails in relevant regions.

For further illustration of the effectiveness of our proposed solutions to these failure modes, see Section D.1.

B Gaussian processes

Gaussian processes (GPs) are a flexible class of statistical models for specifying prior distributions over unknown functions $f : \mathcal{X} \subseteq \mathbb{R}^D \to \mathbb{R}$ (Rasmussen and Williams, 2006). In this section, we describe the GP model used in the paper and details of the training procedure.

B.1 Gaussian process model

In the paper, we use GPs as surrogate models for log-posteriors (and log-subposteriors), for which we use the following model. We recall that GPs are defined by a positive definite covariance, or kernel function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$; a mean function $m : \mathcal{X} \to \mathbb{R}$; and a likelihood or observation model.

Kernel function. For simplicity, we use the common and equivalently-named squared exponential, Gaussian, or exponentiated quadratic kernel,

$$\kappa(x, x'; \sigma_f^2, \ell_1, \dots, \ell_D) = \sigma_f^2 \exp\left[-\frac{1}{2}(x - x')\, \mathbf{\Sigma}_\ell^{-1} (x - x')^\top\right] \qquad \text{with } \mathbf{\Sigma}_\ell = \operatorname{diag}\left[\ell_1^2, \dots, \ell_D^2\right], \tag{S1}$$

where σ_f is the output length scale and $(\ell_1, \dots, \ell_D)$ is the vector of input length scales. Our algorithm does not hinge on choosing this specific kernel and more appropriate kernels might be used depending on the application (e.g., the spectral mixture kernel might provide more flexibility; see Wilson and Adams, 2013).

Mean function. When using GPs as surrogate models for log-posterior distributions, it is common to assume a *negative quadratic* mean function such that the surrogate posterior (i.e., the exponentiated surrogate log-posterior) is integrable (Nemeth and Sherlock, 2018; Acerbi, 2018, 2019, 2020). A negative quadratic can be interpreted as a prior assumption that the target posterior is a multivariate normal; but note that the GP can model deviations from this assumption and represent multimodal distributions as well (see for example Fig 3 in the main text). In this paper, we use

$$m(x; m_0, \mu_1, \dots, \mu_D, \omega_1, \dots, \omega_D) \equiv m_0 - \frac{1}{2}\sum_{i=1}^{D} \frac{(x_i - \mu_i)^2}{\omega_i^2}, \tag{S2}$$

where m_0 denotes the maximum, $(\mu_1, \dots, \mu_D)$ is the location vector, and $(\omega_1, \dots, \omega_D)$ is a vector of length scales.

Observation model. Finally, GPs are also characterized by a likelihood or observation noise model. Throughout the paper we assume exact observations of the target log-posterior (or log-subposterior) so we use a Gaussian likelihood with a small variance $\sigma^2 = 10^{-3}$ for numerical stability.

B.2 Gaussian process inference and training

Inference. Conditioned on training inputs $\mathbf{X} = \{x_1, \dots, x_N\}$, observed function values $\mathbf{y} = f(\mathbf{X})$ and GP hyperparameters ψ, the posterior GP mean and covariance are available in closed form (Rasmussen and Williams, 2006),

$$\begin{aligned} \overline{f}_{\mathbf{X},\mathbf{y}}(x) &\equiv \mathbb{E}\left[f(x)|\mathbf{X}, \mathbf{y}, \psi\right] = \kappa(x, \mathbf{X})\left[\kappa(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbb{I}_D\right]^{-1}(\mathbf{y} - m(\mathbf{X})) + m(x) \\ C_{\mathbf{X},\mathbf{y}}(x, x') &\equiv \operatorname{Cov}\left[f(x), f(x')|\mathbf{X}, \mathbf{y}, \psi\right] = \kappa(x, x') - \kappa(x, \mathbf{X})\left[\kappa(\mathbf{X}, \mathbf{X}) + +\sigma^2 \mathbb{I}_D\right]^{-1} \kappa(\mathbf{X}, x'), \end{aligned} \tag{S3}$$

where ψ is a hyperparameter vector for the GP mean, covariance, and likelihood (see Section B.1 above); and $\mathbb{I}_D$ is the identity matrix in D dimensions.

Training. Training a GP means finding the hyperparameter vector(s) that best represent a given dataset of input points and function observations $(\mathbf{X}, \mathbf{y})$. In this paper, we train all GP models by maximizing the log marginal likelihood of the GP plus a log-prior term that acts as regularizer, a procedure known as *maximum-a-posteriori* estimation. Thus, the training objective to maximize is

$$\begin{aligned} \log p(\psi|\mathbf{X}, \mathbf{y}) = &-\frac{1}{2}(\mathbf{y} - m(\mathbf{X}; \psi))^\top \left[\kappa(\mathbf{X}, \mathbf{X}; \psi) + \sigma^2 \mathbb{I}_D\right]^{-1} (\mathbf{y} - m(\mathbf{X}; \psi)) \\ &+ \frac{1}{2}\log\det\left(\kappa(\mathbf{X}, \mathbf{X}; \psi) + \sigma^2 \mathbb{I}_D\right) + \log p(\psi) + \text{const}, \end{aligned} \tag{S4}$$

where $p(\psi)$ is the prior over GP hyperparameters, described below.

Hyperparameter	Description	Prior distribution
$\log \sigma_f^2$	Output scale	—
$\log \ell^{(i)}$	Input length scale	$\mathrm{Log}\mathcal{N}\left(\log \sqrt{\frac{D}{6}} L^{(i)}, \log \sqrt{10^3}\right)$
m_0	Mean function maximum	$\mathrm{SmoothBox}\left(y_{\min}, y_{\max}, 1.0\right)$
$x_m^{(i)}$	Mean function location	$\mathrm{SmoothBox}\left(B_{\min}^{(i)}, B_{\max}^{(i)}, 0.01\right)$
$\log \omega^{(i)}$	Mean function scale	$\mathrm{Log}\mathcal{N}\left(\log \sqrt{\frac{D}{6}} L^{(i)}, \log \sqrt{10^3}\right)$

Table S1: Priors over GP hyperparameters. See text for more details

Priors. We report the prior $p(\psi)$ over GP hyperparameters in Table S1, assuming independent priors over each hyperparameter and dimension $1 \leq i \leq D$. We set the priors based on broad characteristics of the training set, an approach known as empirical Bayes which can be seen as an approximation to a hierarchical Bayesian model. In the table, $\mathbf{B}$ is the 'bounding box' defined as the D-dimensional box including all the samples observed by the GP so far plus a 10% margin; $\mathbf{L} = \mathbf{B}_{\max} - \mathbf{B}_{\min}$ is the vector of lengths of the bounding box; and $y_{\max}$ and $y_{\min}$ are, respectively, the largest and smallest observed function values of the GP training set. $\mathrm{Log}\mathcal{N}(\mu, \sigma)$ denotes the log-normal distribution and $\mathrm{SmoothBox}(a, b, \sigma)$ is defined as a uniform distribution on the interval $[a, b]$ with a Gaussian tail with standard deviation σ outside the interval. If a distribution is not specified, we assumed a flat prior.

B.3 Implementation details

All GP models in the paper are implemented using GPyTorch[1] (Gardner et al., 2018), a modern package for GP modeling based on the PyTorch machine learning framework (Paszke et al., 2019). For maximal accuracy, we performed GP computations enforcing exact inference via Cholesky decomposition, as opposed to the asymptotically faster conjugate gradient implementation which however might not reliably converge to the exact solution (Maddox et al., 2021). For parts related to active sampling, we used the BoTorch[2] package for active learning with GPs (Balandat et al., 2020), implementing the acquisition functions used in the paper as needed.

We used the same GP model and training procedure described in this section for all GP-based methods in the paper, which include our proposed approach and others (e.g., Nemeth and Sherlock, 2018).

C Algorithm details

Algorithm S1 Parallel Active Inference (PAI)

```
Input: Data partitions $\mathcal{D}_1, \ldots, \mathcal{D}_K$; prior $p(\theta)$; likelihood function $p(\mathcal{D}|\theta)$.
parfor $1 \ldots K$ do                                                        ▷ Parallel steps
    $\mathcal{S}_k \leftarrow$ MCMC samples from $p_k(\theta) \propto p(\theta)^{1/K} p(\mathcal{D}_k|\theta)$
    $\mathcal{S}'_k \leftarrow \textsc{ActiveSubsample}(\mathcal{S}_k)$                ▷ See Sections 3.1 and C.1
    send $\mathcal{S}'_k$ to all other nodes, receive $\mathcal{S}'_{\setminus k} = \bigcup_{j \neq k} \mathcal{S}'_j$        ▷ Communication step
    $\mathcal{S}''_k \leftarrow \mathcal{S}'_k \cup \textsc{SelectSharedSamples}(\mathcal{S}'_k, \mathcal{S}'_{\setminus k})$    ▷ See Sections 3.2 and C.2
    $\mathcal{S}'''_k \leftarrow \mathcal{S}''_k \cup \textsc{ActiveRefinement}(\mathcal{S}''_k)$         ▷ See Sections 3.3 and C.3
    train GP model $\mathcal{L}_k$ of the log subposterior on $(\mathcal{S}'''_k, \log p_k(\mathcal{S}'''_k))$
end parfor
combine subposteriors: $\log q(\theta) = \sum_{k=1}^{K} \mathcal{L}_k(\theta)$          ▷ Centralized step, see Section 3.4
Optional: refine $\log q(\theta)$ with Distributed Importance Sampling (DIS)      ▷ See Section 2.2
```

In this section, we describe additional implementation details for the steps of the Parallel Active Inference (PAI) algorithm introduced in the main text. Algorithm S1 illustrates the various steps. Each function called by the algorithm may involve several sub-steps (e.g., fitting interim surrogate GP models) which are detailed in the following sections.

[1] https://gpytorch.ai/

[2] https://botorch.org/

C.1 Subposterior modeling via GP regression

Here we expand on Section 3.1 in the main text. We recall that at this step each node $k \in \{1, \dots, K\}$ has run MCMC on the local subposterior p_k, obtaining a set of samples $\mathcal{S}_k$ and their log-subposterior values $\log p_k(\mathcal{S})$.

The goal now is to 'thin' the samples to a subset which is still very informative of the shape of the subposterior, so that it can be used as a training set for the GP surrogate. The rationale is that using all the samples for GP training is expensive in terms of both computational and communication costs, and it can lead to numerical instabilities. In previous work, Nemeth and Sherlock (2018) have used *random* thinning which is not guaranteed to keep relevant parts of the posterior.[3]

The main details that we cover here are: (1) how we pick the initial subset of samples $\mathcal{S}_k^{(0)} \subseteq \mathcal{S}_k$ used to train the initial GP surrogate; (2) how we subsequently perform *active subsampling* to expand the initial subset to include relevant points from $\mathcal{S}_k$.

Initial subset: To bootstrap the GP surrogate model, we use a distance clustering method to select the initial subset of samples that we use to fit a first GP model. As this initial subset will be further refined, the main characteristic for selecting the samples is for them to be spread out. For our experiments, we choose a simple k-medoids method (Park and Jun, 2009) with $n_{\text{med}} = 20 \cdot (D + 2)$ medoids implemented in the `scikit-learn-extras` library[4]. The output n_{med} medoids represent $\mathcal{S}_k^{(0)}$.

Active subsampling: After having selected the initial subset $\mathcal{S}_k^{(0)}$, we perform T iterations of active subsampling. In each iteration $t + 1$, we greedily select a batch of n_{batch} points from the set $\mathcal{S}_k \setminus \mathcal{S}_k^{(t)}$ obtained by maximizing a batch version of the maximum interquantile range (MAXIQR) acquisition function, as described by Järvenpää et al. (2021); see also main text. The MAXIQR acquisition function has a parameter u which controls the tradeoff between exploitation of regions of high posterior density and exploration of regions with high posterior uncertainty in the GP surrogate. To strongly promote exploration, after a preliminary analysis on a few toy problems, we set $u = 20$ throughout the experiments presented in the paper. In the paper, we set $n_{\text{batch}} = D$ and the number of iterations $T = 25$, based on a rule of thumb for the total budget of samples required by similar algorithms to achieve good performance at a given dimension (e.g., Acerbi 2018; Järvenpää et al. 2021). The GP model is retrained after the acquisition of each batch. The procedure locally returns a subset of samples $\mathcal{S}'_k = \mathcal{S}_k^{(T)}$.

C.2 Sample sharing

In this section, we expand on Section 3.2 of the main text. We recall that at this step node k receives samples $\mathcal{S}'_{\setminus k} = \bigcup_{j \neq k} \mathcal{S}'_j$ from the other subposteriors. To avoid incorporating too many data points into the local surrogate model (for the reasons explained previously), we consider adding a data point to the current surrogate only if: (a) the local model cannot properly predict this additional point; and (b) predicting the exact value would make a difference. If the number of points that are eligible under these criteria is greater than n_{share}, the set is further thinned using k-medoids.

Concretely, let $\theta^\star \in \mathcal{S}'_{\setminus k}$ be the data point under consideration. We evaluate the true subposterior log density at the point, $y^\star = \log p_k(\theta^\star)$, and the surrogate GP posterior latent mean and variance at the point, which are, respectively, $\mu_\star = \overline{f}(\theta_\star)$ and $\sigma_\star^2 = C(\theta^\star, \theta^\star)$. We then consider two criteria:

a. First, we compute the density of the true value under the surrogate prediction and check if it is above a certain threshold: $\mathcal{N}\left(\log y^\star; \mu_\star, \sigma_\star^2\right) > R$, where $R = 0.01$ in this paper. This criterion is roughly equivalent to including a point only if $|\mu_\star - y^\star| \gtrsim R' \sigma^\star$, for an appropriate choice of R' (ignoring a sublinear term in $\sigma^\star$), implying that a point is considered for addition if the GP prediction differs from the actual value more than a certain number of standard deviations.

b. Second, if a point meets the first criterion, we check if the value of the point is actually relevant for the surrogate model. Let $y_{\max}$ be the maximum subposterior log-density observed at the current node (i.e., approximately, the log-density at the mode). We exclude a point at this stage if both the GP prediction and the true value $y^\star$ are below the threshold $y_{\max} - 20D$, meaning that the point has very low density and the GP correctly predicts that it is very low density (although might not predict the exact value).

[3] Note that instead of thinning we could use sparse GP approximations (e.g., Titsias, 2009). However, the interaction between sparse GP approximations and active learning is not well-understood. Moreover, we prefer not to introduce an additional layer of approximation that could reduce the accuracy of our subposterior surrogate models.

[4] `https://github.com/scikit-learn-contrib/scikit-learn-extra`

Each of the above criteria is checked for all points in $\mathcal{S}'_{\setminus k}$ in parallel (the second criterion only for all the points that pass the first one). Note that the second criterion is optional, but in our experiments we found that it increases numerical stability of the GP model, removing points with very low (log) density that are difficult for the GP to handle (for example, Acerbi (2018); Järvenpää et al. (2021) adopt a similar strategy of discarding very low-density points). More robust GP kernels (see Section B.1) might not need this additional step.

If the number of points that pass both criteria for inclusion is larger than n_{share}, then k-medoids is run on the set of points under consideration with n_{share} medoids, where $n_{\text{share}} = 25D$. The procedure run at node k locally returns a subset of samples $\mathcal{S}''_k$.

C.3 Active subposterior refinement

Here we expand on Section 3.3 of the main text. We recall that up to this point, the local GP model of the log-subposterior at node k was trained only using selected subset of samples from the original MCMC runs (local and from other nodes), denoted by $\mathcal{S}''_k$.

In this step, each node k locally acquires new points by iteratively optimizing the MAXIQR acquisition function (Eq. 2 in the main text) over a domain $\mathcal{X} \subseteq \mathbb{R}^D$. For the first iteration, the space $\mathcal{X}$ is defined as the bounding box of $\bigcup_k \mathcal{S}'_k$ plus a 10% margin. In other words, $\mathcal{X}$ is initially the hypercube that contains all samples from all subposteriors obtained in the sample sharing step (Section C.2) extended by a 10% margin. The limits of $\mathcal{X}$ at the first iteration are computed during the previous stage, without any additional communication cost. In each subsequent iteration, $\mathcal{X}$ is iteratively extended to include the newly sampled points plus a 10% margin, thus expanding the bounding box if a recently acquired point falls near the boundary.

New points are selected greedily in batches of size $n_{\text{batch}} = D$, using the same batch formulation of the MAXIQR acquisition function used in Section C.1. The local GP surrogate is retrained at the end of each iteration, to ensure that the next batch of points targets the regions of the log-subposterior which are most important to further refine the surrogate. For the purpose of this work, we repeat the process for $T_{\text{active}} = 25$ iterations, selected based on similar active learning algorithms (Acerbi et al., 2018; Järvenpää et al., 2021). Future work should investigate an appropriate termination rule to dynamically vary the number of iterations.

The outcome of this step is a final local set of samples $\mathcal{S}'''_k$ and the log-subposterior GP surrogate model $\mathcal{L}_k$ trained on these samples. Both of these are sent back to the central node for the final combination step.

D Experiment details and additional results

In this section, we report additional results and experimental details omitted from the main text for reasons of space.

D.1 Ablation study

As an ablation study, Fig S1 breaks down the effect of each step of PAI in the multi-modal posterior experiment from Section 4.1 of the main text. The first panel shows the full approximate posterior if we were combining it right after active subsampling (Section C.1), using neither sample sharing nor active refinement. Note that this result suffers from Failure mode I (mode collapse; see Section A.1), as active subsampling only on the local MCMC samples is not sufficient to recover the missing modes. The second panel incorporates sample sharing, which covers the missing regions but now suffers from Failure mode II (model mismatch; see Section A.2) with an hallucinated mode in a region where the true posterior has low density. Finally, the third panel shows full-fledged PAI, which further applies active sampling to explore the hallucinated mode and corrects the density around it. The final result of PAI perfectly matches the ground truth (as displayed in the fourth panel).

D.2 Performance evaluation

In this section, we describe in detail the metrics used to assess the performance of the methods in the main text, how we compute these metrics, and the related statistical analyses for reporting our results.

Metrics. In the main text, we measured the quality of the posterior approximations via the mean marginal total variation distance (MMTV), 2-Wasserstein (W2) distance, and Gaussianized symmetrized Kullback-Leibler divergence (GsKL) between true and appoximate posteriors. For all metrics, lower is better. We describe the three metrics and their features below:

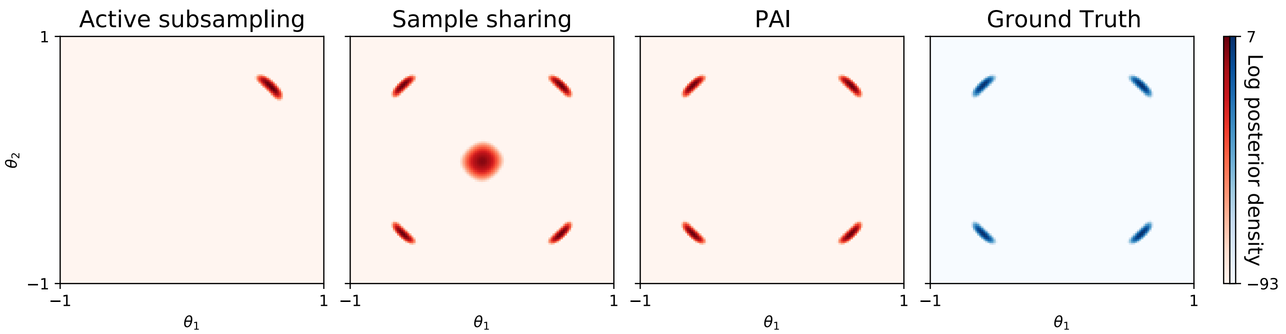


Figure S1: **Ablation study for PAI on the multi-modal posterior.** From left to right: The first panel shows the approximate combined posterior density for our method *without* sample sharing and active learning; the second panel uses an additional step to share samples (but no active refinement); the third shows results for full-fledged PAI. The rightmost plot is the ground truth posterior. Note that both sample sharing and active refinement are important steps for PAI: sample sharing helps account for missing posterior regions while active sampling corrects model hallucinations (see text for details).

- The MMTV quantifies the (lack of) overlap between true and approximate posterior marginals, defined as

$$\mathrm{MMTV}(p, q) = \frac{1}{2D} \sum_{d=1}^{D} \int_{-\infty}^{\infty} \left| p_d^{\mathrm{M}}(x_d) - q_d^{\mathrm{M}}(x_d) \right| dx_d \tag{S5}$$

  where p_d^{M} and q_d^{M} denote the marginal densities of p and q along the d-th dimension. Eq. S5 has a direct interpretation in that, for example, a MMTV metric of 0.5 implies that the posterior marginals overlap by 50% (on average across dimensions). As a rule of thumb, we consider a threshold for a reasonable posterior approximation to be MMTV < 0.2, that is more than 80% overlap.

- Wasserstein distances measure the cost of moving amounts of probability mass from one distribution to the other so that they perfectly match – a commonly-used distance metric across distributions. The W2 metric, also known as *earth mover's* distance, is a special case of Wasserstein distance that uses the Euclidean distance as its cost function. The W2 distance between two density functions p and q, with respective supports $\mathcal{X}$ and $\mathcal{Y}$ is given by

$$\mathrm{W2}(p, q) = \left[\inf_{T \in \mathcal{T}} \int_{x \in \mathcal{X}} \int_{y \in \mathcal{Y}} \|x - y\|_2 T(x, y) \, dx \, dy \right]^{\frac{1}{2}}, \tag{S6}$$

  where $\mathcal{T}$ denotes the set of all joint density functions over $\mathcal{X} \times \mathcal{Y}$ with marginals exactly p and q. In practice, we use empirical approximations of p and q to compute the W2, which simplifies Eq. S6 to a linear program.

- The GsKL metric is sensitive to differences in means and covariances, being defined as

$$\mathrm{GsKL}(p, q) = \frac{1}{2} \left[D_{\mathrm{KL}} \left(\mathcal{N}[p] || \mathcal{N}[q] \right) + D_{\mathrm{KL}} (\mathcal{N}[q] || \mathcal{N}[p]) \right], \tag{S7}$$

  where $D_{\mathrm{KL}}\left(p||q\right)$ is the Kullback-Leibler divergence between distributions p and q and $\mathcal{N}[p]$ is a multivariate normal distribution with mean equal to the mean of p and covariance matrix equal to the covariance of p (and same for q). Eq. S7 can be expressed in closed form in terms of the means and covariance matrices of p and q. For reference, two Gaussians with unit variance and whose means differ by $\sqrt{2}$ (resp., $\frac{1}{2}$) have a GsKL of 1 (resp., $\frac{1}{8}$). As a rule of thumb, we consider a desirable target to be (much) less than 1.

Computing the metrics. For each method, we computed the metrics based on samples from the combined approximate posteriors. For methods whose approximate posterior is a surrogate GP model (i.e., GP, GP-DIS, PAI, PAI-DIS in the paper), we drew samples from the surrogate model using importance sampling/resampling (Robert and Casella, 2013). As proposal distribution for importance sampling we used a mixture of a uniform distribution over a large hyper-rectangle and a distribution centered on the region of high posterior density (the latter to increase the precision of our estimates). We verified that our results did not depend on the specific choice of proposal distribution.

Statistical analyses. For each problem and each method, we reran the entire parallel inference procedure ten times with ten different random seeds. The same ten seeds were used for all methods – implying among other things that, for each problem, all methods were tested on the same ten random partitions of the data and using the same MCMC samples on those partitions. The outcome of each run is a triplet of metrics (MMTV, W2, GsKL) with respect to ground truth. We computed mean and standard deviation of the metrics across the ten runs, which are reported in tables in the main text. For each problem and metric, we highlighted in bold all methods whose mean performance does not differ in a statistically significant way from the best-performing method. Since the metrics are not normally distributed, we tested statistical significance via bootstrap ($n_{\text{bootstrap}} = 10^6$ bootstrapped datasets) with a threshold for statistical significance of $\alpha = 0.05$.

D.3 Model details and further plots

We report here additional details for some of the models used in the experiments in the main paper, and plots for the experiment from computational neuroscience.

Multi-modal posterior. In Section 4.1 of the main text we constructed a synthetic multi-modal posterior with four modes. We recall that the generative model is

$$\theta \sim p(\theta) = \mathcal{N}(0, \sigma_p^2 \mathbb{I}_2)$$

$$y_1, \ldots, y_N \sim p(y_n|\theta) = \sum_{i=1}^{2} \frac{1}{2} \mathcal{N}\left(y_n; P_i(\theta_i), \sigma_l^2\right)$$

where $\theta \in \mathbb{R}^2$, $\sigma_p = \sigma_l = 1/4$ and P_i's are second-degree polynomial functions. To induce a posterior with four modes, we chose $P_1(\theta_1)$ and $P_2(\theta_2)$ to be polynomials with exactly two roots, such that, when the observations are drawn from the full generative model in Eq. D.3, each root will induce a local maximum of the posterior in the vicinity of the root (after considering the shrinkage effect of the prior). The polynomials are defined as $P_1(x) = P_2(x) = (0.6 - x)(-0.6 - x)$, so the posterior modes will be in the vicinity of $\theta^\star \in \{(0.6, 0.6), (-0.6, 0.6), (0.6, -0.6), (-0.6, -0.6)\}$.

Multisensory causal inference. In Section 4.4 of the main text, we modeled a benchmark visuo-vestibular causal inference experiment (Acerbi et al., 2018; Acerbi, 2020) which is representative of many similar models and tasks in the fields of computational and cognitive neuroscience. In the modeled experiment, human subjects, sitting in a moving chair, were asked in each trial whether the direction of movement s_{vest} matched the direction s_{vis} of a looming visual field. We assume subjects only have access to noisy sensory measurements $z_{\text{vest}} \sim \mathcal{N}\left(s_{\text{vest}}, \sigma^2_{\text{vest}}\right)$, $z_{\text{vis}} \sim \mathcal{N}\left(s_{\text{vis}}, \sigma^2_{\text{vis}}(c)\right)$, where σ_{vest} is the vestibular noise and $\sigma_{\text{vis}}(c)$ is the visual noise, with $c \in \{c_{\text{low}}, c_{\text{med}}, c_{\text{high}}\}$ distinct levels of visual coherence adopted in the experiment. We model subjects' responses with a heuristic 'Fixed' rule that judges the source to be the same if $|z_{\text{vis}} - z_{\text{vest}}| < \kappa$, plus a probability λ of giving a random response (Acerbi et al., 2018). Model parameters are $\theta = (\sigma_{\text{vest}}, \sigma_{\text{vis}}(c_{\text{low}}), \sigma_{\text{vis}}(c_{\text{med}}), \sigma_{\text{vis}}(c_{\text{high}}), \kappa, \lambda)$, nonlinearly mapped to $\mathbb{R}^6$. In the paper, we fit real data from subject S1 of (Acerbi et al., 2018). Example approximate posteriors for the PAI-DIS and GP-DIS methods, the best-performing algorithms in this example, are shown in Fig S2.

D.4 Scalability of PAI to large datasets

In the main paper, as per common practice in the field, we used moderate dataset sizes (∼10k data points) to easily calculate ground truth. This choice bears no loss of generality because increasing dataset size only makes subposteriors sharper, which does not increase the difficulty of parallel inference (although more data would not necessarily resolve multimodality, e.g. due to symmetries of the model). On the other hand, small datasets make the reporting of run-times not meaningful, as they are dominated by overheads.

To assess the performance of PAI on large datasets, we ran PAI on the model of Section 4.1, but now with *1 million data points* ($K = 10$ partitions). Average metrics for PAI were excellent and similar to what we had before: MTV = 0.009, W2 = 0.005, GKL = 2e-05, while all other methods still failed. Moreover, run-times for this experiment illustrate the advantages of using PAI in practice.

We ran experiments using computers equipped with two 8-core Xeon E5 processors and 16GB or RAM each. Here, the total time for parallel inference was 57 minutes — 50 for subposterior MCMC sampling + 7 for all PAI steps. By contrast, directly running MCMC on the whole dataset took roughly 6 hours.

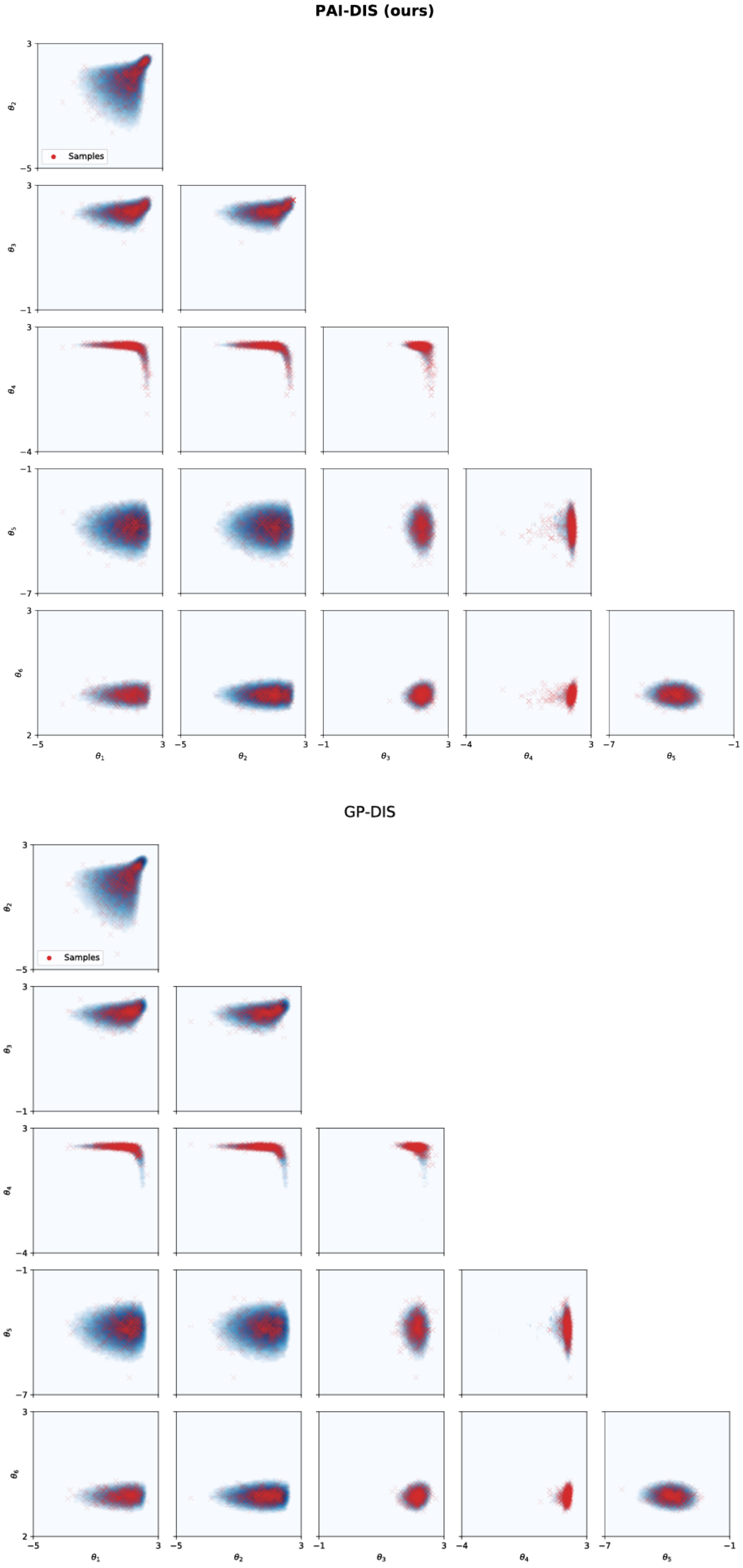


Figure S2: **PAI-DIS and GP-DIS on the multisensory causal inference task.** Each panel shows two-dimensional posterior marginals as samples from the combined approximate posterior (red) against the ground truth (blue). While PAI-DIS (top figure) and GP-DIS (bottom figure) perform similarly in terms of metrics, PAI-DIS captures some features of the posterior shape more accurately, such as the ‘boomerang’ shape of the θ_3 marginals (middle row).