Hierarchical Bayesian Quadrature

Tim Weiland[1] Toni Karvonen[2] Philipp Hennig[1]

[1]Tübingen AI Center, University of Tübingen, Tübingen, Germany
[2]School of Engineering Sciences, Lappeenranta–Lahti University of Technology LUT, Lappeenranta, Finland

Abstract

Numerical integration is a cornerstone of various scientific computing applications, such as engineering simulations and model evidence computations in probabilistic machine learning. Bayesian Quadrature uses Gaussian process surrogates that explicitly encode structural assumptions about the integrand to obtain integral estimates with quantified uncertainty. These surrogates are predominantly based on stationary covariance functions, which results in model misspecification for integrands exhibiting nonstationary behavior. We tackle this issue through an adaptively growing, tree-based partition of the integration domain into local stationary models. Our method recombines the local integral estimates through a hierarchy of GP conditioning that reintroduces cross-subdomain correlations, while model selection criteria control the tree growth to avoid unnecessary partitioning. The resulting algorithm is simple, requires no MCMC, and adapts its evaluation budget to local integrand complexity. On benchmark integration problems and a model evidence computation for an epidemiological model, Hierarchical Bayesian Quadrature achieves substantial gains over standard Bayesian Quadrature on nonstationary integrands while matching its performance on stationary ones.

1 INTRODUCTION

Many tasks in fields like scientific computing and probabilistic machine learning require the computation of a definite integral of a function f over some domain $\Omega \subset \mathbb{R}^d$:

$$I = \int_\Omega f(\boldsymbol{x}) d\boldsymbol{x}. \tag{1}$$

For example, the integral could represent a model evidence or the normalizing constant in Bayes' formula [Osborne et al., 2012, Adachi et al., 2022], or summary statistics for a computer simulation [Li et al., 2023]. In practical applications, closed-form expressions for Eq. (1) are often either unavailable or too expensive to evaluate.

Numerical integration is the task of computing an approximation to Eq. (1). Such methods typically consider the integrand f to be a black box by only assuming access to function evaluations $f(\boldsymbol{x})$ for $\boldsymbol{x} \in \Omega$.

There are various classes of algorithms for numerical integration that mainly differ in the extent to which they make prior assumptions about the integrand. Monte Carlo and quasi-Monte Carlo methods [Evans and Swartz, 2000, Dick et al., 2013] make minimal assumptions about the integrand and are particularly prevalent for high-dimensional integration. In contrast, classical quadrature rules [Davis and Rabinowitz, 1984], such as Gaussian quadratures, exploit integrand smoothness to achieve faster convergence rates.

Bayesian quadrature (BQ) [O'Hagan, 1991, Mahsereci and Karvonen, 2026] approaches numerical integration through a Gaussian process (GP) surrogate model of the integrand. Conditioning on point evaluations $f(x)$ yields a GP posterior. As Gaussians are closed under linear operations, the integral of the posterior is a Gaussian random variable. Thus, BQ naturally produces not only a point estimate for the integral, but also a calibrated error estimate. For suitable choices of the kernel function of the GP, the associated computations may be performed in closed-form [Briol et al., 2025]. The ability of BQ to flexibly express assumptions such as smoothness and periodicity about the integrand through the GP prior is a major strength of the method. Well-specified priors yield faster convergence, well-calibrated uncertainty estimates, and are more robust [Briol et al., 2019].

However, BQ methods predominantly employ stationary covariance functions. This results in **model misspecification** for integrands best described by nonstationary processes whose variability depends on location. While BQ

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:7273–7289.

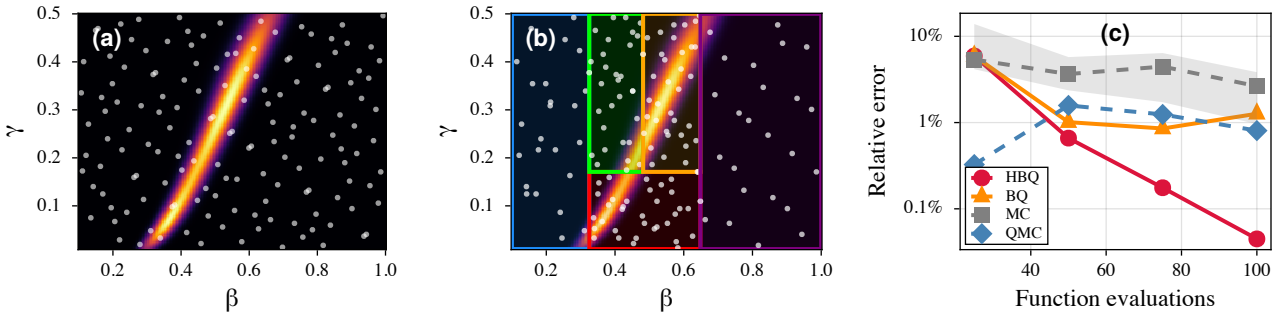

Figure 1: **Adaptive budget allocation for model evidence computation.** We consider an SIR epidemiological model with infection rate β and recovery rate γ; each evaluation requires solving an ODE system. The integrand (normalized likelihood times prior) exhibits a curved ridge due to the β/γ correlation. **(a)** BQ with $N = 150$ Sobol points: uniform coverage regardless of local complexity. **(b)** HBQ with $N = 150$: adaptive partitioning concentrates evaluations on the ridge. **(c)** Relative error in $\log Z$; MC shows median and interquartile range over 20 runs.

is known to exhibit certain robustness to smoothness misspecification [Kanagawa et al., 2020, Mahsereci and Karvonen, 2026], nonstationarity remains an unsolved challenge. Nonstationary integrands arise naturally for example when computing posterior expectations over peaked likelihood surfaces, as well as in engineering simulations [Gramacy and Lee, 2008]. Figure 1 illustrates this on a model evidence computation for an SIR epidemiological model, where the integrand exhibits a sharp ridge surrounded by flat regions. A single stationary kernel must then compromise between resolving the ridge and efficiently covering flat regions, causing inefficient allocation of the evaluation budget.

Contributions: We present **Hierarchical Bayesian Quadrature** (**HBQ**), a BQ method that automatically detects nonstationary integrand behavior and adaptively invests its compute budget accordingly. We achieve this through a tree-based partition of the domain where the leaf surrogates are stationary GPs with varying lengthscale and smoothness. We present an algorithm to adaptively and robustly grow such a tree on-the-fly via marginal likelihood optimization and model selection techniques.

Our method requires no MCMC and introduces little computational overhead; in fact, domain splitting can accelerate GP inference by reducing the size of each local Gram matrix. On integrands exhibiting nonstationary behavior, HBQ achieves substantial performance gains over standard BQ, while matching its performance on stationary integrands. Indeed, our method is closely related to the Genz–Malik rule, a state-of-the-art deterministic adaptive quadrature method.

2 BACKGROUND

2.1 BAYESIAN QUADRATURE

Bayesian Quadrature (BQ) models the integrand f through a Gaussian process (GP) prior with mean function μ and kernel function k: $f \sim \mathcal{GP}(\mu, k)$. Without loss of generality, we assume $\mu = 0$. Through its smoothness and other properties the kernel function encodes assumptions about the function f. See Rasmussen and Williams [2006] and Mahsereci and Karvonen [2026] for introductions to GPs and BQ, respectively.

We collect evaluations $\mathbf{y}$ of the integrand at locations $\boldsymbol{X} := (\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$, where $\boldsymbol{x}_i \in \Omega$. The posterior under the evaluations is again a GP: $f \mid (f(\boldsymbol{X}) = \mathbf{y}) \sim \mathcal{GP}(\mu^*, k^*)$ with

$$\mu^*(\boldsymbol{x}) = \mathbf{k}(x, \boldsymbol{X})\mathbf{K}^{-1}\mathbf{y}, \tag{2}$$

$$k^*(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \mathbf{k}(\boldsymbol{x}, \boldsymbol{X})^\top \mathbf{K}^{-1}\mathbf{k}(\boldsymbol{X}, \boldsymbol{x}'), \tag{3}$$

where $\mathbf{K} := [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^N$ is the Gram matrix and $\mathbf{k}(\boldsymbol{x}, \boldsymbol{X}) := \mathbf{k}(\boldsymbol{X}, \boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_i)]_{i=1}^N$.

Bounded linear transformations of Gaussian processes remain Gaussian with closed-form mean and covariance [Pförtner et al., 2024]. Consequently, we have

$$\int_\Omega f(\boldsymbol{x})\, d\boldsymbol{x} \;\Big|\; (f(\boldsymbol{X}) = \mathbf{y}) \;\sim\; \mathcal{N}\Big(\hat{I},\, \hat{\sigma}^2\Big), \tag{4}$$

with

$$\hat{I} = \mathbf{z}^\top \mathbf{K}^{-1}\mathbf{y}, \tag{5}$$

$$\hat{\sigma}^2 = \int_\Omega \int_\Omega k(\boldsymbol{x}, \boldsymbol{x}')\, d\boldsymbol{x}\, d\boldsymbol{x}' - \mathbf{z}^\top \mathbf{K}^{-1}\mathbf{z}, \tag{6}$$

where $\mathbf{z} := [\int_\Omega k(\boldsymbol{x}, \boldsymbol{x_i})\, d\boldsymbol{x}]_{i=1}^N$ is the vector of kernel mean embeddings.

For suitable choices of kernel functions (see Briol et al. [2025]), we can compute the quantities in Eqs. (5) and (6) — and thus the integral of the posterior — in closed-form. We obtain a point estimate $\hat{I}$ of the true integral Eq. (1) with quantified uncertainty $\hat{\sigma}^2$ that shrinks as the number of point evaluations N grows. In this sense, BQ is a prototypical probabilistic numerical method. These are methods that cast numerical tasks as learning problems to be solved with the tools of Bayesian inference [Hennig et al., 2015, 2022].

2.2 HYPERPARAMETER SELECTION

In order to obtain both an accurate point estimate and calibrated uncertainty, we require a well-specified prior. We consider a stationary kernel function of the form

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \sigma_f^2 \, \phi_\nu \left(\frac{\|\boldsymbol{x} - \boldsymbol{x'}\|}{\ell} \right), \tag{7}$$

where σ_f^2 is the output scale, $\ell > 0$ is the lengthscale, and ϕ_ν is a positive definite function parameterized by a smoothness parameter ν. In this work, we focus in particular on the Matérn family. A Matérn-ν kernel is defined by

$$\phi_\nu(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\, r \right)^\nu K_\nu \left(\sqrt{2\nu}\, r \right), \tag{8}$$

where K_ν is the modified Bessel function of the 2nd kind.

The hyperparameters $\theta = (\sigma_f^2, \ell, \nu)$ are typically selected by maximizing the marginal likelihood $p(\mathbf{y} \mid \boldsymbol{X}, \theta)$. The output scale σ_f^2 can be concentrated out analytically, leaving ℓ (and optionally ν) as the only parameters to optimize.

The lengthscale ℓ controls the assumed regularity of the integrand: small ℓ allows rapid variation, while large ℓ implies slow variation. The smoothness parameter ν controls the differentiability of the sample paths. For a nonstationary integrand, any single choice of ℓ and ν is a compromise — too smooth in some regions, too rough in others — resulting in the model misspecification discussed in Section 1.

3 HIERARCHICAL BAYESIAN QUADRATURE

3.1 DOMAIN DECOMPOSITION AND LOCAL BQ

Nonstationary kernels. The natural response to the aforementioned misspecification would be a nonstationary kernel, such as the ones proposed by Gibbs [1997], Paciorek and Schervish [2003], and Remes et al. [2017]. Unfortunately, closed-form kernel mean embeddings are generally unavailable for nonstationary kernels, requiring costly approximation and negating much of BQ's computational appeal.

Our pragmatic response is to express nonstationary behavior through a hierarchy of stationary models on subdomains. In doing so, we adapt to nonstationarity without sacrificing the benefits of stationary models.

Domain decomposition. Assume we have a partition of the domain Ω into K non-overlapping subdomains Ω_i:

$$\Omega = \cup_{i=1}^K \Omega_i, \tag{9}$$

where the subdomains Ω_i are chosen to encapsulate heterogeneous variation and smoothness properties of the integrand. We can then fit a local stationary GP $f_i \sim \mathcal{GP}(0, k_i)$ to each subdomain and use BQ to obtain integral estimates

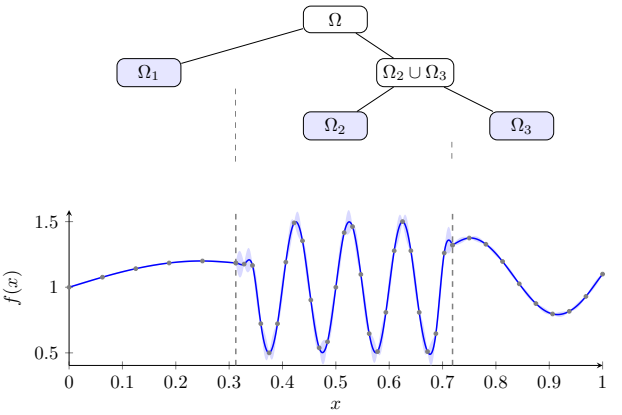

Figure 2: HBQ decomposes the domain into subdomains via a tree of axis-aligned splits. Each leaf fits a local stationary GP with its own lengthscale, adapting to local integrand complexity. Shaded regions show posterior uncertainty.

$$\int_{\Omega_i} f_i(\boldsymbol{x}) d\boldsymbol{x} \sim \mathcal{N}(\hat{I}_i, \hat{\sigma}_i). \tag{10}$$

To keep all computations closed-form, we impose a few restrictions on the kernel class and the domain decomposition.

Kernel choice. While HBQ can in principle use any kernels, we assume a half-integer Matérn or Wendland kernel as a base kernel. Both are of the form in Eq. (7) and adapt to function behavior via their lengthscale and smoothness parameters. Kernel mean embeddings over intervals $[a, b]$ are available in closed form [Briol et al., 2025]. For $d > 1$, we form a tensor product kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \prod_{j=1}^d k_j(x_j, x_j')$.

Domain choice. We further restrict the subdomains Ω_i to axis-aligned hyperrectangles, i.e. products of intervals. Due to the tensor product structure, the kernel mean embeddings are then closed-form as products of one-dimensional kernel mean embeddings over intervals. Starting from the hyperrectangle Ω, we partition via axis-aligned binary splits. This naturally produces hyperrectangular subdomains.

The resulting partition forms a binary tree (Fig. 2). What remains is to recombine leaf integral estimates into a global estimate and to decide adaptively when and where to split.

3.2 INTEGRAL RECOMBINATION

Given the leaf integral estimates from Eq. (10), how do we obtain the global integral? Clearly, the global integral is

$$\int_\Omega f(\boldsymbol{x}) d\boldsymbol{x} = \sum_{i=1}^K \int_{\Omega_i} f(\boldsymbol{x}) d\boldsymbol{x}. \tag{11}$$

The simplest approach is to assume independence of the local GPs. Then Eq. (11) is a sum of independent Gaussian

random variables, and we obtain

$$\int_\Omega f(\boldsymbol{x})\,d\boldsymbol{x} \;\Big|\; (f(\boldsymbol{X}) = \boldsymbol{y}) \sim \mathcal{N}\left(\sum_{i=1}^K \hat{I}_i, \sum_{i=1}^K \hat{\sigma}_i^2\right). \quad (12)$$

We refer to this as the *independent sum* estimate. It is simple and introduces no overhead, but the underlying independence assumption is likely to be wrong in practice.

Tree conditioning. To alleviate this issue, we exploit the tree structure of the partition. When a leaf node is split, it becomes a branch node with children. Rather than discarding the GP that was fitted to the leaf before the split, we retain it as the branch node's prior k_p. Since this kernel was estimated on the full parent domain $\Omega_\mathrm{p} = \cup_{j=1}^m \Omega_{\mathrm{p},j}$, it encodes correlations across the split boundary.

Under this prior, the sub-integrals $I_j = \int_{\Omega_{\mathrm{p},j}} f(\boldsymbol{x})\,d\boldsymbol{x}$ are jointly Gaussian with covariance

$$C_{ij} = \int_{\Omega_{\mathrm{p},i}} \int_{\Omega_{\mathrm{p},j}} k_\mathrm{p}(\boldsymbol{x}, \boldsymbol{x}')\,d\boldsymbol{x}\,d\boldsymbol{x}'. \quad (13)$$

Each child provides an integral estimate $(\hat{I}_j, \hat{\sigma}_j^2)$ from its own local GP. We treat these as noisy observations of the linear functionals $L_j[f] = \int_{\Omega_{\mathrm{p},j}} f(\boldsymbol{x})\,d\boldsymbol{x}$, with observation noise $\hat{\sigma}_j^2$. Define the cross-covariance between the total integral and each sub-integral as

$$c_j = \int_{\Omega_\mathrm{p}} \int_{\Omega_{\mathrm{p},j}} k_\mathrm{p}(\boldsymbol{x}, \boldsymbol{x}')\,d\boldsymbol{x}\,d\boldsymbol{x}' = \sum_{i=1}^m C_{ij}, \quad (14)$$

and let $\boldsymbol{\Sigma} = \mathrm{diag}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_m^2)$. Conditioning the parent GP on the children's estimates gives

$$\int_{\Omega_\mathrm{p}} f(\boldsymbol{x})\,d\boldsymbol{x} \;\Big|\; \left(L_j[f] \approx \hat{I}_j\right)_{j=1}^m \sim \mathcal{N}\left(\hat{I}_\mathrm{p}, \hat{\sigma}_\mathrm{p}^2\right), \quad (15)$$

with

$$\hat{I}_\mathrm{p} = \boldsymbol{c}^\top (\boldsymbol{C} + \boldsymbol{\Sigma})^{-1} \hat{\boldsymbol{I}}, \quad (16)$$

$$\hat{\sigma}_\mathrm{p}^2 = \int_{\Omega_\mathrm{p}} \int_{\Omega_\mathrm{p}} k_\mathrm{p}(\boldsymbol{x}, \boldsymbol{x}')\,d\boldsymbol{x}\,d\boldsymbol{x}' - \boldsymbol{c}^\top (\boldsymbol{C} + \boldsymbol{\Sigma})^{-1} \boldsymbol{c}, \quad (17)$$

where $\hat{\boldsymbol{I}} = (\hat{I}_1, \ldots, \hat{I}_m)^\top$. This is analogous to Eqs. (5) and (6), with the children's integral estimates playing the role of observations and $\boldsymbol{c}$ replacing the kernel mean embedding vector $\mathbf{z}$.

If a child is itself a branch, its estimate $(\hat{I}_j, \hat{\sigma}_j^2)$ is obtained by applying Eqs. (16) and (17) recursively. The global integral is thus computed bottom-up: leaf estimates propagate through the tree, with each branch reintroducing cross-subdomain correlations via its retained prior.

This construction also yields a multi-scale decomposition: the parent kernel k_p captures coarse-scale structure, while the children's kernels model finer-scale variation within each subdomain.

3.3 CONVERGENCE ANALYSIS

We next provide some theory for tree conditioning. The following theorem states that the child with largest error controls the error of its parent. Note that this theorem holds trivially also for (12).

Theorem 1. *Suppose that* $|I_j - \hat{I}_j| \le \varepsilon_j$ *and* $\hat{\sigma}_j^2 \le \delta_j^2$ *for some* $\varepsilon_j, \delta_j \ge 0$ *and all* $j \in \{1, \ldots, m\}$. *Let* $\varepsilon_\mathrm{max} = \max\{\varepsilon_1, \ldots, \varepsilon_m\}$ *and* $\delta_\mathrm{max} = \max\{\delta_1, \ldots, \delta_m\}$. *Then*

$$|I - \hat{I}_\mathrm{p}| = O(\varepsilon_\mathrm{max}) + O(\delta_\mathrm{max}^2) \;\; \textit{and} \;\; \hat{\sigma}_\mathrm{p}^2 = O(\delta_\mathrm{max}^2). \quad (18)$$

as $\varepsilon_\mathrm{max}, \delta_\mathrm{max} \to 0$.

Proof. Expanding $(\boldsymbol{C} + \boldsymbol{\Sigma})^{-1} = (\boldsymbol{I} + \boldsymbol{C}^{-1}\boldsymbol{\Sigma})^{-1}\boldsymbol{C}^{-1}$ as a Neumann series gives

$$\hat{I}_\mathrm{p} = \boldsymbol{c}^\top \boldsymbol{C}^{-1} \hat{\boldsymbol{I}} + \sum_{k=1}^\infty \boldsymbol{c}^\top (-\boldsymbol{C}^{-1}\boldsymbol{\Sigma})^k \boldsymbol{C}^{-1} \hat{\boldsymbol{I}}. \quad (19)$$

The series converges when δ_max is sufficiently small since $\boldsymbol{\Sigma} = \mathrm{diag}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_m^2)$. Eqs. (13) and (14) yield $\boldsymbol{C}^{-1}\boldsymbol{c} = (1, \ldots, 1)^\top$. Therefore $\hat{I}_\mathrm{p} = \sum_{i=1}^m \hat{I}_i + O(\delta_\mathrm{max}^2)$, so that

$$|I - \hat{I}_\mathrm{p}| \le \sum_{j=1}^m |I_j - \hat{I}_j| + O(\delta_\mathrm{max}^2), \quad (20)$$

which gives the claim for $|I - \hat{I}_\mathrm{p}|$. The proof for $\hat{\sigma}_\mathrm{p}^2$ is analogous. □

Convergence and variance contraction rates for BQ are well-understood when the kernel is a stationary Matérn kernel of order ν [Kanagawa and Hennig, 2019, Kanagawa et al., 2020, Mahsereci and Karvonen, 2026]. Here we consider stationary Matérns rather than tensor product Matérns because convergence results for the latter are more complicated [Mahsereci and Karvonen, 2026, Sec. 5.1.2]. Consider the posterior for standard BQ given in Eqs. (4)–(6) and let $\eta > 0$. If Ω is bounded and its boundary is sufficiently regular (e.g., Ω is a rectangle) and f is an element of $H^{\eta+d/2}(\Omega)$, the Sobolev space of order $\eta + d/2$ on Ω, one can prove that

$$|I - \hat{I}| = O(N^{-\min\{\eta,\nu\}/d - 1/2}) \quad (21)$$

and $\hat{\sigma}^2 = O(N^{-2\nu/d-1})$ provided that the locations $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ are quasi-uniform [Mahsereci and Karvonen, 2026, Thm. 5.6]. Quasi-uniformity means that the separation radius and fill-distance,

$$q_N = \frac{1}{2} \min_{1 \le i \ne j \le N} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \text{ and } h_N = \sup_{\boldsymbol{x} \in \Omega} \min_{\boldsymbol{x}_i \in \boldsymbol{X}} \|\boldsymbol{x} - \boldsymbol{x}_i\|,$$

satisfy $q_N = \Theta(h_N)$ as $N \to \infty$. Separation radius measures the smallest distance between any two integration locations while the fill-distance is the radius of the largest ball in Ω that contains none of $\boldsymbol{x}_i$. Applying these asymptotics to the children gives the following corollary of Theorem 1.

Corollary 2. *Suppose that each $\Omega_j \subset \mathbb{R}^d$ is a rectangle and that a stationary Matérn kernel of order ν_j and N_j quasi-uniform observation locations are used to obtain (10). If $f|_{\Omega_j} \in H^{\eta_j + d/2}(\Omega_j)$ for $j \in \{1, \dots, m\}$, then*

$$|I - \hat{I}_{\mathrm{p}}| = O\big(\max\{N_j^{-\min\{\eta_j,\nu_j\}/d - 1/2}\}_{j=1}^m\big)$$

and

$$\hat{\sigma}^2_{\mathrm{p}} = O\big(\max\{N_j^{-2\nu_j/d-1}\}_{j=1}^m\big)$$

as $N_1, \dots, N_m \to \infty$.

This corollary states that the convergence of HBQ is controlled by the smallest local smoothness of f (the smallest η_j) or the roughest local Matérn model (the smallest ν_j).

4 ADAPTIVE TREE CONSTRUCTION

In practice, the hierarchy in Section 3 is not known a priori and must be determined from the integrand evaluations. We grow the partition adaptively: posterior uncertainty guides us where to invest more integrand evaluations, while model selection techniques decide when to partition further.

4.1 SPLITTING CRITERION

We use model selection to decide when a single stationary GP is misspecified and should be replaced by GPs on smaller subdomains.

When to split? Let f be a GP on the domain $\Omega = [a, b]$ conditioned on evaluation points $\boldsymbol{X}$. Select a point $x_{\text{split}} \in \boldsymbol{X} \setminus \{a, b\}$. We obtain two independent GPs: f_1 on the domain $\Omega_1 = [a, x_{\text{split}}]$ conditioned on evaluation points $\boldsymbol{X}_1 = \boldsymbol{X} \cap [a, x_{\text{split}}]$, and f_2 on the domain $\Omega_2 = [x_{\text{split}}, b]$ conditioned on evaluation points $\boldsymbol{X}_2 = \boldsymbol{X} \cap [x_{\text{split}}, b]$. We can evaluate the Bayes factor of the two models:

$$\mathrm{BF}_{\text{split}} = \frac{p(\boldsymbol{y} \mid f_1, f_2)}{p(\boldsymbol{y} \mid f)} = \frac{p(\boldsymbol{y}_1 \mid f_1) p(\boldsymbol{y}_2 \mid f_2)}{p(\boldsymbol{y} \mid f)} \tag{22}$$

Since the Bayes factor requires intractable marginalization over hyperparameters, we approximate it with the Bayesian Information Criterion (BIC) [Schwarz, 1978], which penalizes model complexity using the profile marginal likelihood from Section 4.2: Let $\mathcal{L}_{\text{parent}}$ and $\mathcal{L}_{\text{split}}$ denote the maximized profile log-likelihoods of the single-GP and split models, respectively. We accept a split if

$$\mathcal{L}_{\text{split}} - \mathcal{L}_{\text{parent}} > \frac{\Delta k}{2} \log N, \tag{23}$$

where N is the number of evaluation points and Δk is the number of additional parameters introduced by the split.

With a shared lengthscale, the single-GP model has parameters (σ_f^2, ℓ); the split model has two such pairs plus the split location, giving $\Delta k = 3$. With per-dimension lengthscales (ARD), $\Delta k = d + 2$. The penalty grows with N, preventing over-fragmentation as the evaluation budget increases.

To further guard against pathological splits, we require each child to contain at least 5^d evaluation points. This prevents the creation of leaves with too few points for reliable hyperparameter estimation.

Where to split? Given the criterion in Eq. (23), it remains to select the split location. We restrict candidates to existing evaluation points and perform a grid search along each coordinate axis. For each candidate, we fit local hyperparameters to both children and evaluate the combined log-likelihood $\mathcal{L}_{\text{split}}$. We select the axis and location that maximize $\mathcal{L}_{\text{split}}$, accepting the split only if it satisfies Eq. (23).

4.2 HYPERPARAMETER FITTING

Each leaf's kernel (7) has hyperparameters (σ_f^2, ℓ, ν). The output scale σ_f^2 is concentrated out analytically, leaving ℓ as the sole continuous parameter to optimize via the profile marginal likelihood. We select the smoothness ν by a discrete search over a set of candidate Matérn orders, optimizing ℓ for each and retaining the best pair. This allows different subdomains to use different kernel smoothness, matching local regularity.

Adaptive splitting means that leaves frequently have few evaluation points, causing the profile likelihood for ℓ to plateau near zero. We regularize with a log-normal prior on ℓ anchored at the parent's fitted lengthscale:

$$\log \ell_{\text{child}} \sim \mathcal{N}(\log \bar{\ell}_{\text{parent}},\, \sigma^2_{\text{prior}}), \tag{24}$$

where $\bar{\ell}$ is the geometric mean of the parent's per-dimension lengthscales in the ARD setting. Hyperparameters are re-estimated whenever a leaf is refined. Appendix B gives further details, including the root node prior and candidate sets.

4.3 ADAPTIVE REFINEMENT

Adaptively growing the partition is a sequential decision making problem: At each step, we can take one of two actions — split a leaf to reduce model misspecification, or add a batch of integrand evaluations to a leaf.

Integral variance reduction. Intuitively, at each step we want to act on the leaf that would most improve our integral estimate. Let $\hat{\sigma}^2$ denote the current posterior variance of the global integral estimate, and let $\hat{\sigma}^2_{k+}$ denote the hypothetical variance obtained by adding a batch of M_k integrand evaluations to leaf k. Here M_k may depend on the leaf — for instance, inserting midpoints into an existing grid of N_k points adds $N_k - 1$ new evaluations. We score each leaf by

Algorithm 1 Hierarchical Bayesian Quadrature

Require: integrand f, domain Ω, budget $N_{\max}$, tolerance τ
Ensure: integral estimate $\hat{I} \pm \hat{\sigma}$

```
Initialize root node with GP fitted on initial evaluations
Î, σ̂ ← BQ(root)
while N_used < N_max and σ̂ > τ do
    Select leaf k* with largest expected variance reduction
    if ShouldSplit(k*) then
        Split k*: create children, fit local GPs
        Recursively split children while accepted
    else
        Refine k*: add evaluation points, refit hyperparameters
    end if
    Î, σ̂ ← Recombine(root)
end while
```

its expected variance reduction per evaluation,

$$s_k = \frac{\hat{\sigma}^2 - \hat{\sigma}^2_{k+}}{M_k}, \tag{25}$$

and select the highest-scoring leaf $k^* = \arg\max_k s_k$. This is a batched variant of the integral variance reduction (IVR) criterion used in active BQ [Osborne et al., 2012], extended to the hierarchical setting. The hypothetical variance $\hat{\sigma}^2_{k+}$ is computed cheaply by re-evaluating the BQ posterior variance formula Eq. (6) under the current kernel with the augmented point set, without re-fitting hyperparameters or evaluating f. The cost normalization by M_k prevents bias toward leaves with large grids.

Split or refine. Given target leaf k^*, we attempt a BIC-guided split (Section 4.1) and recursively try to split the resulting children until no further splits are accepted. If the initial split is rejected, we refine by adding evaluation points and re-fitting hyperparameters. The loop repeats until the budget is exhausted or $\hat{\sigma}$ falls below a tolerance (Algorithm 1).

5 RELATED WORK

Partitioned Gaussian processes. Fitting local stationary GPs on subdomains is a well-established strategy for handling nonstationarity in regression. Treed GP models [Gramacy and Lee, 2008] recursively partition the input space and fit independent GPs per leaf, with the tree structure learned via reversible-jump MCMC. Because the leaf GPs are independent, predictions at subdomain boundaries are discontinuous. Several aggregation schemes address this, e.g. product-of-experts [Tresp, 2000, Deisenroth and Ng, 2015] and boundary pseudo-observations [Park and Apley, 2018]. Nested kriging [Rullière et al., 2018] aggregates local predictions through a second-level GP — the regression analogue of our tree conditioning. Most related to our splitting criterion, Hinne et al. [2022] also use BIC to choose between a merged GP and a split GP with a discontinuity, albeit for regression and discontinuity design rather than integration. None of these approaches have been applied to integration, where the recombination mechanism directly affects integral uncertainty and where kernel mean embeddings impose additional structure.

Local Bayesian optimization. Trust-region methods such as TuRBO [Eriksson et al., 2019] also fit local stationary GPs, but for optimization: acquisition functions bias evaluations toward extrema and abandon regions of low expected improvement. For quadrature every subdomain contributes additively—small-amplitude regions still matter through their volume—so HBQ instead keeps a global partition and recombines the local integral estimates into a single posterior over the integral, with no analogue of TuRBO's trust-region shrinkage. The BQ-specific contribution is thus not local stationary models per se, but their combination with closed-form subdomain integration and hierarchical recombination of integral functionals.

Bayesian quadrature. Bayesian quadrature [O'Hagan, 1991, Mahsereci and Karvonen, 2026] places a GP prior on the integrand and derives a posterior distribution over the integral. Active point selection via integral variance reduction [Osborne et al., 2012], Frank–Wolfe optimization [Briol et al., 2015], and warped models for non-negative integrands [Gunter et al., 2014] have substantially improved practical performance. All of these methods use a single global GP, which limits their ability to adapt to spatially varying integrand complexity. BART-Int [Zhu et al., 2020] replaces the GP with a Bayesian additive regression tree prior, whose sum-of-trees structure implicitly partitions the domain. However, BART-Int's piecewise-constant tree basis limits its effectiveness for smooth integrands, where GP-based BQ retains an advantage. Multilevel BQ [Li et al., 2023] introduces hierarchy, but across fidelity levels rather than spatial subdomains. Our approach combines domain partitioning with GP-based BQ in each subdomain, preserving the smoothness advantages of GP priors while adapting to nonstationarity through the tree structure.

Adaptive cubature. The Genz–Malik algorithm [Genz and Malik, 1980, Berntsen et al., 1991] is the gold standard adaptive cubature method. Like HBQ, it partitions the domain into axis-aligned hyperrectangles via binary splits, concentrating effort where the integrand is most difficult. HBQ can be viewed as its Bayesian counterpart: we replace Genz–Malik's embedded polynomial rules with local GP surrogates, gaining principled uncertainty quantification, and split only when the local model is detectably misspecified (Section 4.1), refining the existing GP otherwise.

Model evidence computation. When the integral is a model evidence, sampling-based estimators such as Warp-

III bridge sampling [Gronau et al., 2019] and the truncated harmonic-mean estimator THAMES [Metodiev et al., 2025] are also widely used. These require posterior samples rather than pointwise black-box evaluations and do not quantify integral uncertainty, but they are effective complementary tools for model comparison.

6 EXPERIMENTS

We provide a Julia implementation of our method.[1] Refer to Appendix A for further details on all experiments.

6.1 GENZ BENCHMARK FUNCTIONS

We evaluate HBQ on the standard Genz test function suite [Genz, 1984], a collection of parameterized integrand families over $[0,1]^d$, each isolating a specific challenge for quadrature methods. We use five of the six families: Oscillatory (high-frequency oscillations), Product Peak (localized peak with product structure), Corner Peak (concentration near the origin), Gaussian Peak (localized Gaussian bump), and Continuous (C^0 kink). Each family is parameterized by vectors $\boldsymbol{a}, \boldsymbol{u} \in \mathbb{R}^d$ controlling difficulty and location, with analytic integrals available for all families.

Setup. We test in $d = 2$ with difficulty $C = \sum_i a_i = 9$, drawing $\boldsymbol{a}$ and $\boldsymbol{u}$ randomly for each of 20 independent trials. We compare four methods at evaluation budgets $N \in \{32, 64, 128, 256, 512\}$: Monte Carlo (MC), quasi-Monte Carlo with Sobol sequences (QMC), standard BQ with a single stationary GP, and HBQ. Both BQ and HBQ select the kernel smoothness automatically from $\{\text{Matérn-}1/2, \text{Matérn-}3/2, \text{Matérn-}5/2\}$ via the profile marginal likelihood, ensuring a fair comparison. BQ uses Sobol points with no adaptive refinement; HBQ uses the full adaptive loop of Algorithm 1.

Results. Figure 3 shows median absolute error with interquartile bands. HBQ substantially outperforms BQ on Corner Peak, where the integrand concentrates sharply near the origin and flattens elsewhere, achieving up to $6\times$ lower error. HBQ's adaptive partitioning concentrates evaluations in the region of high complexity rather than spreading them uniformly. On the remaining four families, HBQ performs comparably to BQ, confirming that the adaptive overhead does not degrade performance when the integrand is well-described by a single stationary GP.

6.2 MODEL EVIDENCE FOR AN EPIDEMIOLOGICAL MODEL

We return to the motivating example from Fig. 1: computing the model evidence $Z = \int p(y \mid \beta, \gamma)\, p(\beta, \gamma)\, d\beta\, d\gamma$ for a susceptible-infected-recovered (SIR) model with two parameters, infection rate β and recovery rate γ. Each likelihood evaluation requires solving an ODE system, making function evaluations expensive. We use a small-population setting ($N_{\text{pop}} = 200$) with two weekly incidence observations under Poisson noise, producing a broad but sharply curved ridge in the likelihood surface (details in the appendix).

Figure 1(c) compares convergence of four methods using a ground truth computed via adaptive cubature. HBQ with independent summation achieves sub-1% relative error at $N = 100$ evaluations, while BQ, MC, and QMC remain above 1%. Panel (b) shows how HBQ concentrates its evaluation budget: at $N = 150$, the tree allocates five leaves with the densest coverage along the likelihood ridge, while the flat regions receive only coarse coverage.

6.3 STRUCTURE RECOVERY AND CALIBRATION

We now test whether HBQ can recover known nonstationary structure from data. We sample integrands from a piecewise Matérn GP on $[0,1]$ with three segments—rough edges (Matérn-$1/2$, $\ell = 0.02$) flanking a smooth center (Matérn-$5/2$, $\ell = 0.1$)—so that the true partition, kernel smoothness, and integral are all known exactly. Since the integrand comes from a piecewise GP, this is an on-model test: any errors are due to the method, not model misspecification. We run 200 trials with $N = 400$ evaluations each.

Structure recovery. HBQ recovers the correct number of segments in 100% of trials and selects the correct kernel orders in 96%. Figure 4 shows a representative trial: BQ selects the roughest available kernel to accommodate the rough edges, but cannot adapt to local smoothness, leaving it overly uncertain in the smooth center. HBQ identifies the piecewise structure and fits each region with an appropriate kernel, yielding $3.3\times$ tighter uncertainty.

Calibration. Figure 5 shows that this tighter uncertainty is earned: z-scores for both methods track $\mathcal{N}(0,1)$ well (BQ: 92% coverage of a nominal 95% CI, $\sigma_z = 1.13$; HBQ: 90%, $\sigma_z = 1.53$).[2] HBQ has four outliers with $|z| > 4$, all caused by imprecise split placement at the sharp smoothness transitions in this synthetic problem; when splits land correctly, HBQ is well calibrated ($\sigma_z \approx 1$).

[1] https://github.com/timweiland/HierarchicalBayesQuad.jl

[2] One of the 200 BQ trials produced a degenerate posterior with $\hat{\sigma} = 0$, i.e. an infinite z-score, and is excluded from BQ's σ_z.

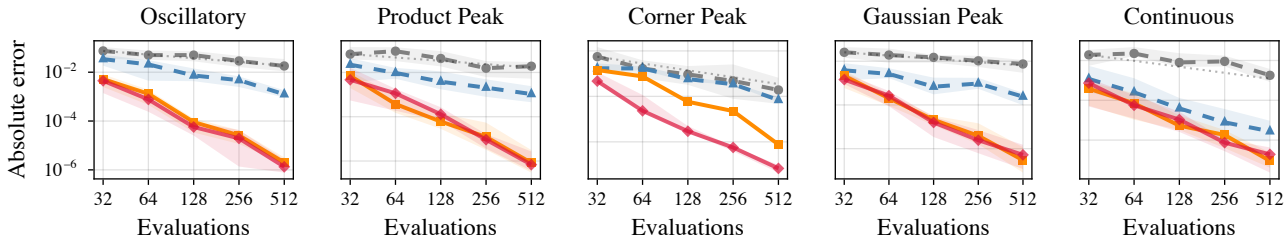

Figure 3: **Convergence on the Genz test function suite** ($d = 2$, difficulty $C = 9$, 20 trials). Lines show median absolute error; shaded regions indicate the interquartile range. The dotted line shows the $N^{-1/2}$ Monte Carlo reference rate. HBQ matches BQ on four families and substantially outperforms it on Corner Peak, where spatially varying complexity rewards adaptive partitioning. MC and QMC shown for reference.

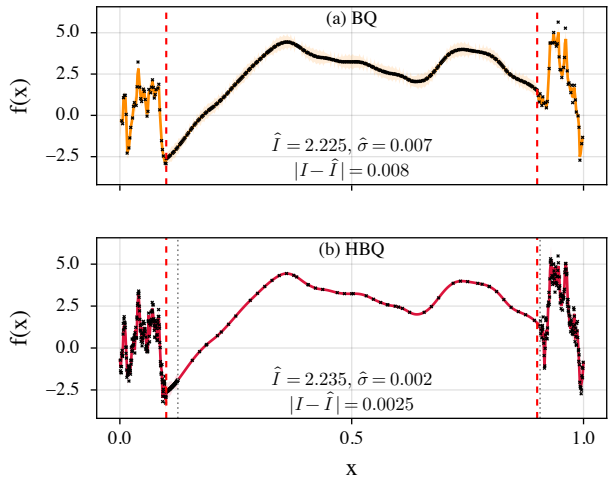

Figure 4: **GP posterior fits on a piecewise integrand.** The integrand is sampled from a piecewise Matérn GP with narrow rough edges (Matérn-$1/2$, $\ell = 0.02$) and a wide smooth center (Matérn-$5/2$, $\ell = 0.1$); true boundaries are shown as dashed red lines. **(a)** BQ fits a single stationary GP. **(b)** HBQ adaptively partitions the domain and recovers the piecewise structure (dotted gray lines), achieving $3.3\times$ tighter uncertainty.

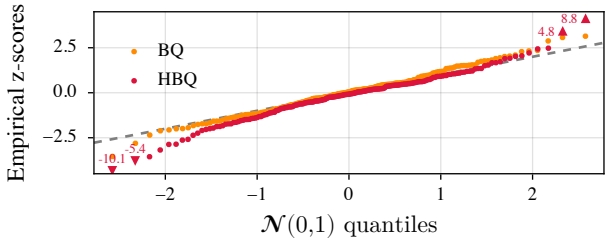

Figure 5: **QQ plot of z-scores** against $\mathcal{N}(0, 1)$ over 200 trials on the piecewise integrand. Both BQ (orange) and HBQ (red) track the diagonal well. HBQ has four outliers (triangles) attributable to imprecise split placement at sharp smoothness transitions.

6.4 REACTION–DIFFUSION PDE

We test HBQ on integrands arising from a one-dimensional reaction–diffusion PDE. The diffusivity follows a sigmoid step, modelling a composite material with a smooth, slowly varying region and a spiky, rapidly varying region. Random Gaussian bump sources and point sources produce sharp local features in the low-diffusivity region. This creates integrands with genuine multi-scale character: a single stationary GP cannot simultaneously capture the smooth and spiky regimes without either oversmoothing or overfitting.

Setup. We draw 20 independent PDE instances with randomized source configurations and compare MC, QMC, BQ, and HBQ at budgets $N \in \{8, 16, 32, 64, 128, 256, 512\}$. Ground-truth integrals are computed via the trapezoidal rule on a fine grid ($N_{\text{fine}} = 8000$).

Results. Figure 6 shows median absolute error with interquartile bands. All methods perform similarly at small budgets ($N \leq 32$), but HBQ pulls away sharply from $N = 64$ onward. At $N = 512$, HBQ achieves a median error of 6.7×10^{-6}, roughly $35\times$ lower than BQ (2.3×10^{-4}). HBQ automatically allocates different lengthscales and point densities to the smooth and spiky regions.

7 DISCUSSION AND CONCLUSION

We presented Hierarchical Bayesian Quadrature, a method that addresses nonstationarity in BQ by adaptively partitioning the domain into subdomains with local stationary GPs. A hierarchy of GP conditioning recombines the local integral estimates while preserving cross-subdomain correlations. Model selection via BIC controls tree growth, preventing unnecessary partitioning: on stationary integrands, HBQ recovers the performance of standard BQ. On nonstationary integrands, HBQ achieves substantial gains by concentrating evaluations where the integrand is most complex.

Our convergence analysis shows that the global error is controlled by the worst-case local smoothness (Theorem 1),

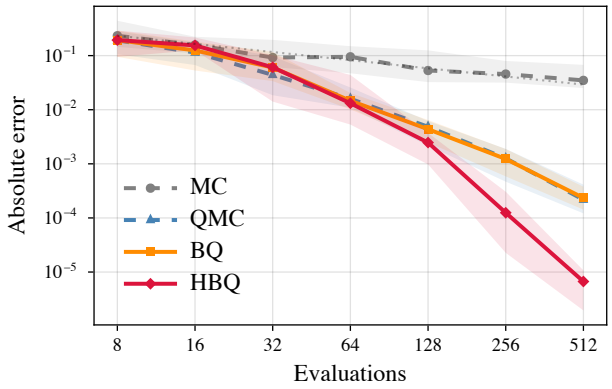

Figure 6: **Convergence on 1D reaction–diffusion PDE integrands.** HBQ separates from BQ at $N = 64$ and reaches $35\times$ lower error at $N = 512$.

providing a theoretical foundation for the adaptive strategy. The structure recovery experiment confirms that HBQ can identify piecewise-stationary structure from data and produce well-calibrated uncertainty estimates.

HBQ is closely related to Genz–Malik adaptive cubature: both partition the domain into axis-aligned hyperrectangles and adaptively subdivide. HBQ replaces polynomial quadrature rules with GP surrogates, gaining principled uncertainty quantification and data-driven split decisions.

7.1 SCOPE AND LIMITATIONS

HBQ targets expensive, low-to-moderate-dimensional integration whose integrand has spatially varying behavior—for example model evidences over peaked or ridged likelihoods, or simulator and PDE responses with multi-scale features. This is the regime where quadrature generally outperforms Monte Carlo, and where adaptive subdivision has been the standard response to nonstationarity since Genz–Malik. Because splitting is BIC-gated, HBQ reduces to standard BQ when a single stationary model suffices, so its gains concentrate on genuinely nonstationary integrands—as the flat performance on four of the five Genz families shows.

It extends to higher dimensions through tensor-product kernels and the d-dependent penalty, and a $d = 3$ benchmark (Appendix C.1) confirms the advantage persists beyond our two-dimensional experiments, though very high dimensions remain hard as axis-aligned partitioning and the 5^d per-leaf requirement scale unfavourably. The main structural limitation is the restriction to axis-aligned hyperrectangles and closed-form-embedding kernels, which is what keeps recombination closed-form; relaxing it would require numerical mean-embedding approximation.

Finally, the convergence analysis is deliberately partial—conditional on the final tree and kernels, not yet covering tree construction or hyperparameter adaptation—which we leave, together with more flexible partitions, to future work.

Author Contributions

Tim Weiland developed the method, implemented the software, designed and ran the experiments, and led the writing. Toni Karvonen proposed the initial project idea, contributed the convergence analysis and its theoretical framing, and co-supervised the work. Philipp Hennig supervised the project and contributed to its conceptual development and presentation.

Acknowledgements

TW and PH gratefully acknowledge co-funding by the European Union (ERC, ANUBIS, 101123955). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. PH is supported by the DFG through Project HE 7114/6-1 in SPP2298/2. PH is a member of the Machine Learning Cluster of Excellence, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. TW and PH also gratefully acknowledge the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ:01IS18039A); and funds from the Ministry of Science, Research and Arts of the State of Baden-Württemberg. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting TW. TK was supported by Research Council of Finland projects 359183 ("Flagship of Advanced Mathematics for Sensing, Imaging and Modelling") and 368086 ("Inference and approximation under misspecification"), and acknowledges the research environment provided by ELLIS Institute Finland.

References

M. Adachi, S. Hayakawa, M. Jørgensen, H. Oberhauser, and M. A. Osborne. Fast Bayesian inference with batch Bayesian quadrature via kernel recombination. In *Advances in Neural Information Processing Systems*, volume 35, pages 16533–16547, 2022.

J. Berntsen, T. O. Espelid, and A. Genz. Algorithm 698: DCUHRE: An adaptive multidemensional integration routine for a vector of integrals. *ACM Transactions on Mathematical Software*, 17(4):452–456, 1991.

F.-X. Briol, C. Oates, M. Girolami, and M. A. Osborne. Frank-Wolfe Bayesian quadrature: Probabilistic integra-

tion with theoretical guarantees. In *Advances in Neural Information Processing Systems*, volume 28, pages 1162–1170, 2015.

F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: A role in statistical computation? (with discussion and rejoinder). *Statistical Science*, 34(1):1–22, 2019.

F.-X. Briol, A. Gessner, T. Karvonen, and M. Mahsereci. A dictionary of closed-form kernel mean embeddings. In *Proceedings of the 1st International Conference on Probabilistic Numerics*, volume 271 of *PMLR*, pages 84–95, 2025.

P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Computer Science and Applied Mathematics. Academic Press, 2nd edition, 1984.

M. Deisenroth and J. W. Ng. Distributed Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *PMLR*, pages 1481–1490, 2015.

J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 5496–5507, 2019.

M. Evans and T. Swartz. *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford University Press, 2000.

G.-A. Fuglstad, D. Simpson, F. Lindgren, and H. Rue. Constructing priors that penalize the complexity of Gaussian random fields. *Journal of the American Statistical Association*, 114(525):445–452, 2019.

A. Genz. Testing multidimensional integration routines. In *Proceedings of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, pages 81–94. Elsevier North-Holland, 1984.

A. C. Genz and A. A. Malik. Remarks on algorithm 006: An adaptive algorithm for numerical integration over an N-dimensional rectangular region. *Journal of Computational and Applied Mathematics*, 6(4):295–302, 1980.

M. N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge, 1997.

R. B. Gramacy and H. K. H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

Quentin Gronau, Eric-Jan Wagenmakers, Daniel Heck, and Dora Matzke. A simple method for comparing complex models: Bayesian model comparison for hierarchical multinomial processing tree models using warp-iii bridge sampling. *Psychometrika*, 84(1):261–284, 2019. doi: 10.1007/s11336-018-9648-3.

T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S. J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In *Advances in Neural Information Processing Systems*, volume 27, pages 2789–2797, 2014.

P. Hennig, M. A. Osborne, and M. Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015.

P. Hennig, M. A. Osborne, and H. P. Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.

Max Hinne, David Leeftink, Marcel A. J. van Gerven, and Luca Ambrogioni. Bayesian model averaging for nonparametric discontinuity design. *PLOS ONE*, 17(6):e0270310, 2022. doi: 10.1371/journal.pone.0270310.

M. Kanagawa and P. Hennig. Convergence guarantees for adaptive Bayesian quadrature methods. In *Advances in Neural Information Processing Systems*, volume 32, pages 6237–6248, 2019.

M. Kanagawa, B. K. Sriperumbudur, and K. Fukumizu. Convergence analysis of deterministic kernel-based quadrature rules in misspecified settings. *Foundations of Computational Mathematics*, 20(1):155–194, 2020.

K. Li, D. Giles, T. Karvonen, S. Guillas, and F.-X. Briol. Multilevel Bayesian quadrature. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *PMLR*, pages 1845–1868, 2023.

M. Mahsereci and T. Karvonen. Bayesian quadrature: Gaussian processes for integration. *arXiv:2602.16218v1*, 2026.

Martin Metodiev, Marie Perrot-Dockès, Sarah Ouadah, Nicholas J. Irons, Pierre Latouche, and Adrian E. Raftery. Easily Computed Marginal Likelihoods from Posterior Simulation Using the THAMES Estimator. *Bayesian Analysis*, 20(3):1003 – 1030, 2025. doi: 10.1214/24-BA1422.

A. O'Hagan. Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991.

M. A. Osborne, D. Duvenaud, R. Garnett, C. E. Rasmussen, S. J. Roberts, and Z. Ghahramani. Active learning of model evidence using Bayesian quadrature. In *Advances in Neural Information Processing Systems*, volume 25, pages 46–54, 2012.

C. Paciorek and M. Schervish. Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 16, 2003.

C. Park and D. Apley. Patchwork kriging for large-scale Gaussian process regression. *Journal of Machine Learning Research*, 19(7):1–43, 2018.

M. Pförtner, I. Steinwart, P. Hennig, and J. Wenger. Physics-informed Gaussian process regression generalizes linear PDE solvers. *arXiv:2212.12474v6*, 2024.

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

S. Remes, M. Heinonen, and S. Kaski. Non-stationary spectral kernels. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

D. Rullière, N. Durrande, F. Bachoc, and C. Chevalier. Nested Kriging predictions for datasets with a large number of observations. *Statistics and Computing*, 28(4): 849–867, 2018.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

V. Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

H. Zhu, X. Liu, R. Kang, Z. Shen, S. Flaxman, and F.-X. Briol. Bayesian probabilistic numerical integration with tree-based models. In *Advances in Neural Information Processing Systems*, volume 33, pages 5837–5849, 2020.

Hierarchical Bayesian Quadrature (Supplementary Material)

Tim Weiland[1] Toni Karvonen[2] Philipp Hennig[1]

[1]Tübingen AI Center, University of Tübingen, Tübingen, Germany
[2]School of Engineering Sciences, Lappeenranta–Lahti University of Technology LUT, Lappeenranta, Finland

A EXPERIMENTAL DETAILS

A.1 GENZ BENCHMARK FUNCTIONS

The Genz test suite [Genz, 1984] consists of six parameterized integrand families over $[0,1]^d$. We use the five families that are continuous (omitting Discontinuous, which requires $d \geq 2$ and has a jump discontinuity):

$$\begin{aligned}
\text{Oscillatory:} \quad & f(\mathbf{x}) = \cos\big(2\pi u_1 + \textstyle\sum_i a_i x_i\big), \\
\text{Product Peak:} \quad & f(\mathbf{x}) = \textstyle\prod_i \big(a_i^{-2} + (x_i - u_i)^2\big)^{-1}, \\
\text{Corner Peak:} \quad & f(\mathbf{x}) = \big(1 + \textstyle\sum_i a_i x_i\big)^{-(d+1)}, \\
\text{Gaussian Peak:} \quad & f(\mathbf{x}) = \exp\big(-\textstyle\sum_i a_i^2 (x_i - u_i)^2\big), \\
\text{Continuous:} \quad & f(\mathbf{x}) = \exp\big(-\textstyle\sum_i a_i |x_i - u_i|\big).
\end{aligned}$$

The vectors $\boldsymbol{a} \in \mathbb{R}^d$ and $\boldsymbol{u} \in [0,1]^d$ control difficulty and location, respectively. Analytic integrals are available for all families.

Parameter generation. For each of 20 trials, we sample $\boldsymbol{u} \sim \mathrm{Unif}([0,1]^d)$ and draw $\boldsymbol{a}$ by sampling a weight vector $\boldsymbol{w} \sim \mathrm{Unif}([0,1]^d)$ and setting $a_i = C \cdot w_i / \sum_j w_j$, so that $\sum_i a_i = C$. We use difficulty $C = 9$. This randomization over parameters (rather than fixing $\boldsymbol{a}$ and $\boldsymbol{u}$) tests robustness across a range of integrand configurations.

Method configuration. We test in $d = 2$ with evaluation budgets $N \in \{32, 64, 128, 256, 512\}$.

- **MC**: uniform random points, independent random seed per trial/budget.
- **QMC**: deterministic Sobol sequence (same points across trials for a given N).
- **BQ**: single stationary GP over $[0,1]^2$ with Sobol initial points. Kernel smoothness selected from $\{\text{Matérn-}1/2, 3/2, 5/2\}$ by profile marginal likelihood. No adaptive refinement (points fixed at initialization).
- **HBQ**: full adaptive loop (Algorithm 1) with 5 initial points per dimension (25 total), Sobol refinement adding 20 points per step, tree conditioning, BIC splitting criterion with midpoint splits, and the same kernel candidates. The ARD lengthscale mode is used, with a log-normal child prior anchored at the parent's fitted lengthscale.

Error is measured as $|I - \hat{I}|$ using the analytic ground truth.

A.2 SIR MODEL EVIDENCE

SIR model. We use the standard deterministic SIR (susceptible–infected–recovered) model:

$$\frac{dS}{dt} = -\frac{\beta S I}{N}, \qquad \frac{dI}{dt} = \frac{\beta S I}{N} - \gamma I, \qquad \frac{dR}{dt} = \gamma I,$$

with population $N = 200$, initial infected $I_0 = 5$, and simulation horizon $T = 50$ days. The parameters of interest are the infection rate $\beta \in [0.1, 1.0]$ and recovery rate $\gamma \in [0.01, 0.5]$, with uniform priors over these ranges.

Synthetic data. We generate data at ground truth $(\beta, \gamma) = (0.4, 0.15)$, corresponding to a basic reproduction number $R_0 = 2.67$. Weekly incidence (new infections per week) is observed at weeks 1 and 5. Observations are drawn from $y_k \sim \text{Poisson}(\lambda_k)$, where $\lambda_k = S(7(k-1)) - S(7k)$ is the true weekly incidence computed from the ODE solution.

Integrand. The model evidence is $Z = \int p(y \mid \beta, \gamma)\, p(\beta, \gamma)\, d\beta\, d\gamma$. For numerical stability, we integrate the shifted integrand $g(\beta, \gamma) = \exp(\ell(\beta, \gamma) - \ell_{\max})$, where $\ell(\beta, \gamma) = \log p(y \mid \beta, \gamma)$ is the Poisson log-likelihood and $\ell_{\max}$ is the maximum over a 200×200 grid. The evidence is recovered as $Z = g_0 \cdot e^{\ell_{\max}} / |\Omega|$, where $g_0 = \int g\, d\beta\, d\gamma$.

Ground truth. We compute the reference integral using an implementation of the Genz–Malik method with relative tolerance 10^{-10}, yielding $\log Z \approx -7.01$.

Method configuration. All methods receive the same evaluation budgets $N \in \{25, 50, 75, 100\}$. BQ and HBQ use Sobol initial points; HBQ selects the kernel automatically from $\{\text{Matérn-}1/2, 3/2, 5/2\}$, while BQ uses a fixed Matérn-$3/2$ kernel. HBQ uses the independent sum recombination (Eq. (12)) and the BIC splitting criterion. MC results show the median and interquartile range over 20 independent runs. QMC uses a single deterministic Sobol sequence.

A.3 STRUCTURE RECOVERY AND CALIBRATION

Piecewise GP construction. We define a piecewise Matérn GP on $[0, 1]$ with three segments:

- $[0.0, 0.1]$: Matérn-$1/2$, $\sigma = 5$, $\ell = 0.02$ (rough),
- $[0.1, 0.9]$: Matérn-$5/2$, $\sigma = 1$, $\ell = 0.1$ (smooth), with imposed integral $\int f = 2$,
- $[0.9, 1.0]$: Matérn-$1/2$, $\sigma = 5$, $\ell = 0.02$ (rough).

Narrow rough edges flanking a wide smooth center create a challenging nonstationary structure: the rough regions occupy only 20% of the domain but contribute substantial variability.

Sampling procedure. For each trial, we draw $N = 400$ Sobol points on $[0, 1]$, partition them into segments, and sample function values from the piecewise GP. Segments are sampled left to right: the first segment is drawn jointly with its integral from the GP prior; each subsequent segment conditions on the boundary value from its left neighbor to ensure continuity. The middle segment additionally conditions on its imposed integral. This yields a continuous function realization with known per-segment integrals, so the ground-truth integral is $I = \sum_j I_j$.

Method configuration. BQ fits a single stationary GP over the full domain with automatic kernel selection from $\{\text{Matérn-}1/2, 3/2, 5/2\}$. HBQ uses $N = 400$ evaluations with `initdiv` $= 10$ (41 initial evaluation points), up to 2 splits (matching the true number of boundaries), and the same kernel candidates. We run 200 independent trials and compute z-scores $z_i = (I_i - \hat{I}_i)/\hat{\sigma}_i$.

Outlier analysis. Four of 200 HBQ trials have $|z| > 4$. All four recover the correct number of segments and kernel orders; the cause is imprecise split placement. When splits land at 0.125 and 0.875 (outside the rough regions), HBQ achieves $\sigma_z = 0.96$ over $n = 132$ trials. When a split lands at 0.09375 (inside the rough region $[0, 0.1]$), the smooth leaf inherits a sliver of rough territory that its Matérn-$5/2$ kernel cannot capture, producing underestimated uncertainty. This synthetic problem has instantaneous smoothness transitions (Matérn-$1/2$ to Matérn-$5/2$); real-world functions typically exhibit gradual transitions, providing more tolerance for split placement.

A.4 REACTION–DIFFUSION PDE

PDE formulation. We consider the steady-state reaction–diffusion equation on $[0, 1]$:

$$-\frac{d}{dx}\left(D(x)\frac{du}{dx}\right) + r\,u(x) = s(x), \qquad u(0) = u(1) = 0,$$

with constant reaction rate $r = 5$. The diffusivity models a composite material with a sigmoid transition:

$$D(x) = \exp\big(D_0 - \Delta D \cdot \sigma(x; c, \kappa)\big),$$

where $\sigma(x; c, \kappa) = (1 + e^{-\kappa(x-c)})^{-1}$, $D_0 = -3$, $\Delta D = 5$, $c = 0.45$, and $\kappa = 25$.

Source terms. Each trial uses $n_s = 7$ Gaussian bump sources $s(x) = \sum_j h_j \exp(-(x - c_j)^2/w^2)$ with width $w = 0.02$, centers jittered uniformly around equispaced base positions, and heights drawn from $\mathrm{Unif}(5, 11)$. In addition, $n_\delta = 4$ point sources of amplitude $A \sim \mathrm{Unif}(0.2, 0.7)$ are placed randomly in the low-diffusivity region $[0.55, 0.95]$, discretized as A/h on the nearest grid point. These create sharp, spatially localized spikes in the solution that are difficult for a single stationary GP to capture.

Discretization and ground truth. The PDE is solved on a uniform grid with $N_{\text{fine}} = 8{,}000$ points using second-order finite differences with arithmetic-mean diffusivity at cell interfaces. The ground-truth integral $\int_0^1 u(x)\,dx$ is computed via the trapezoidal rule on this fine grid. Function evaluations use piecewise linear interpolation of the fine-grid solution.

Method configuration. All methods receive the same evaluation budgets $N \in \{8, 16, 32, 64, 128, 256, 512\}$. BQ and HBQ use Sobol initial points with automatic kernel selection from {Matérn-$1/2, 3/2, 5/2$}. HBQ uses the tree-conditioning recombination (Eq. (15)) and the BIC splitting criterion. MC uses uniform random points; QMC uses a deterministic Sobol sequence. We run 20 independent trials, each with a different random PDE instance.

Multi-scale character. The characteristic lengthscale of the PDE solution scales as $\ell \sim \sqrt{D/r}$. In the smooth region ($x < 0.45$), $\ell \approx 0.1$; in the spiky region ($x > 0.45$), $\ell \approx 0.008$. This $\sim 12\times$ lengthscale ratio is the source of HBQ's advantage: a single GP must compromise between these scales, while HBQ partitions the domain, fits each region with an appropriate lengthscale and invests integrand evaluations accordingly.

B HYPERPARAMETER FITTING DETAILS

Smoothness selection. We search over the half-integer Matérn orders $\nu \in \{1/2, 3/2, 5/2\}$, corresponding to C^0, C^1, and C^2 sample paths, respectively. For each candidate, we optimize the lengthscale ℓ via the profile marginal likelihood and retain the (ν, ℓ) pair with the lowest penalized negative log-likelihood.

Root node prior. For the root node, we use a penalized complexity (PC) prior [Fuglstad et al., 2019] on the lengthscale, taking the constant function ($\ell \to \infty$) as the base model and calibrating so that $P(\ell < |\Omega|/10) = 0.025$.

Child node prior. When a leaf splits, each child receives the log-normal prior (24) anchored at the parent's fitted lengthscale. In the ARD setting, where each dimension has its own lengthscale ℓ_j, we use the geometric mean $\bar{\ell} = (\prod_j \ell_j)^{1/d}$ as the anchor. This prevents the lengthscale from escaping to degenerate values, which is particularly important in the ARD setting where per-dimension lengthscales along flat directions are otherwise unconstrained.

C FURTHER EXPERIMENTS

These experiments complement the evaluation of Section 6: they examine HBQ in a higher dimension (Appendix C.1), its wall-clock cost against accuracy (Appendix C.2), the effect of the recombination scheme (Appendix C.3), and the reliability of the BIC criterion across penalty strengths and budgets (Appendix C.4). All use the same Julia implementation as the main experiments.

C.1 HIGHER-DIMENSIONAL GENZ BENCHMARK

This experiment extends the two-dimensional Genz benchmark of Section 6.1 to three dimensions to test whether the hierarchical construction retains its advantage in higher dimensions. We integrate two Genz families over the unit cube $[0, 1]^3$: Corner Peak, the canonical nonstationary integrand on which hierarchical partitioning is expected to help, and Product Peak, a localized but separable control. For each family we draw 10 random instances, sampling the difficulty vector as $\boldsymbol{a} = C\boldsymbol{w}/\sum_j w_j$ with $C = 9$ and $\boldsymbol{w}, \boldsymbol{u} \sim \mathrm{Unif}([0, 1]^3)$. We compare the same four estimators as in the main text—MC, QMC, BQ (a single-domain surrogate on Sobol points), and HBQ (adaptive tree refinement run to the budget)—at budgets $N \in \{64, 128, 256, 512, 1024\}$, selecting in both BQ and HBQ among Matérn kernels of smoothness $\nu \in \{1/2, 3/2, 5/2\}$. Figure 7 reports the median absolute error over the 10 instances with interquartile bands. At $N = 1024$, HBQ attains a median error of 1.82×10^{-5} versus 8.81×10^{-5} for BQ on Corner Peak (a $4.8\times$ improvement) and 2.1×10^{-4} versus 3.8×10^{-4} on Product Peak ($1.8\times$), while MC and QMC are one to several orders of magnitude worse. HBQ matches or exceeds BQ at every budget, confirming that its gains extend to $d = 3$ and that it carries no penalty when the hierarchical advantage is small.

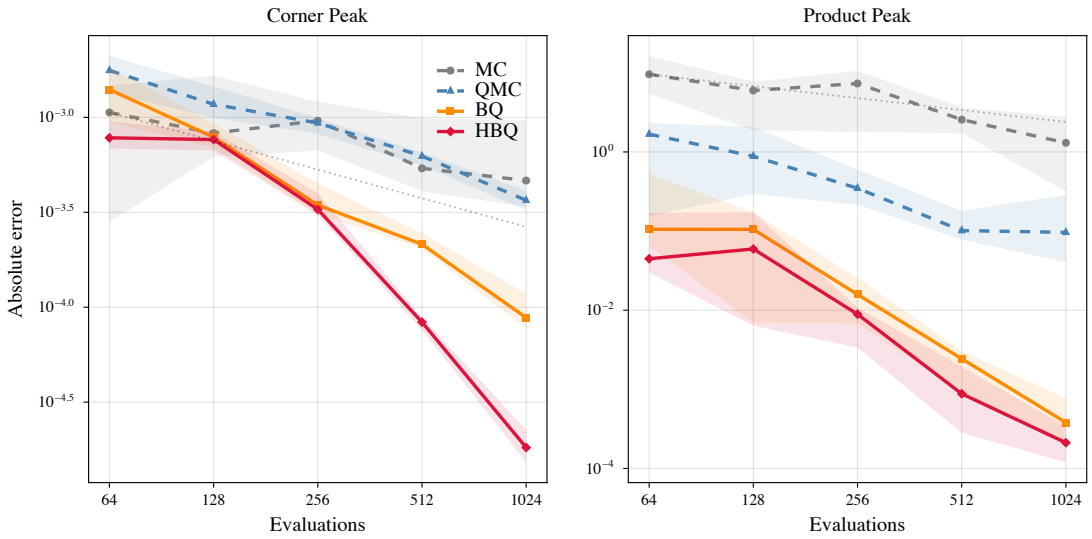

Figure 7: **Higher-dimensional Genz benchmark ($d = 3$).** Median absolute error (interquartile bands) versus the number of function evaluations on the Corner Peak and Product Peak families (10 random instances each). HBQ retains its advantage over BQ at $d = 3$, while MC and QMC are one to several orders of magnitude worse.

C.2 WALL-CLOCK COST VERSUS ACCURACY

We measure wall-clock cost against accuracy on the Genz Corner Peak family in $d = 2$, using the parameter protocol and budget grid of Section 6.1 ($C = 9$, $N \in \{32, 64, 128, 256, 512\}$). Because the Genz integrand is analytically cheap, we inject a fixed $50\,\mathrm{ms}$ delay into every integrand evaluation to emulate an expensive simulator, so that total run time is dominated by the number of evaluations rather than by method bookkeeping. Two implementation optimizations—inheriting the parent's selected kernel for split candidates, and a prescreen that fully refits only the winning split candidate—reduce HBQ's algorithmic overhead by up to roughly $20\times$ with no material change in accuracy (paired validation). We compare MC, QMC, BQ, and HBQ on 10 random instances, timing each run end-to-end after a warm-up pass and a garbage-collection call. Because the injected cost dominates, all four methods incur essentially identical wall-clock at a given budget (about $13.6\,\mathrm{s}$ at $N = 256$ and $27.5\,\mathrm{s}$ at $N = 512$), so the comparison reduces to accuracy at matched compute (Fig. 8). At $N = 256$, HBQ reaches a median absolute error of 5.1×10^{-5} versus 4.2×10^{-4} for BQ—about eightfold more accurate at matched wall-clock—and at $N = 512$, 2.0×10^{-5} versus 7.9×10^{-5}, about fourfold; MC and QMC remain well above HBQ at moderate-to-large budgets.

C.3 TREE CONDITIONING VERSUS INDEPENDENT SUM

To quantify the effect of the recombination scheme, we re-ran the $d = 2$ Genz experiment under both the tree-conditioning and independent-sum modes with all other settings held fixed, so that the two differ only in the final recombination step (Section 3.2). We used Corner Peak—where HBQ shows its largest gain, and hence exercises the recombination machinery most—and Product Peak as a control whose tree rarely splits, at $C = 9$, drawing 20 random instances per family and running each at budgets $N \in \{32, 64, 128, 256, 512\}$. Comparing the two modes per instance and taking medians over trials, the ratio of median absolute error between tree conditioning and independent sum was 1.000 in every family–budget cell (two families $\times$ five budgets; maximum deviation below 0.05%), and the posterior standard deviations matched to the same precision (Fig. 9). On these families the cross-subdomain correlations reintroduced by tree conditioning therefore leave both the estimate and its reported uncertainty essentially unchanged, so the independent sum serves as an inexpensive ablation while tree conditioning remains the primary method at no accuracy cost.

C.4 BIC THRESHOLD SENSITIVITY

To assess whether the BIC criterion remains reliable at the small per-leaf sample sizes produced by adaptive splitting, we ran a sensitivity study on the piecewise-GP recovery task of Section 6.3, whose ground truth comprises three segments—

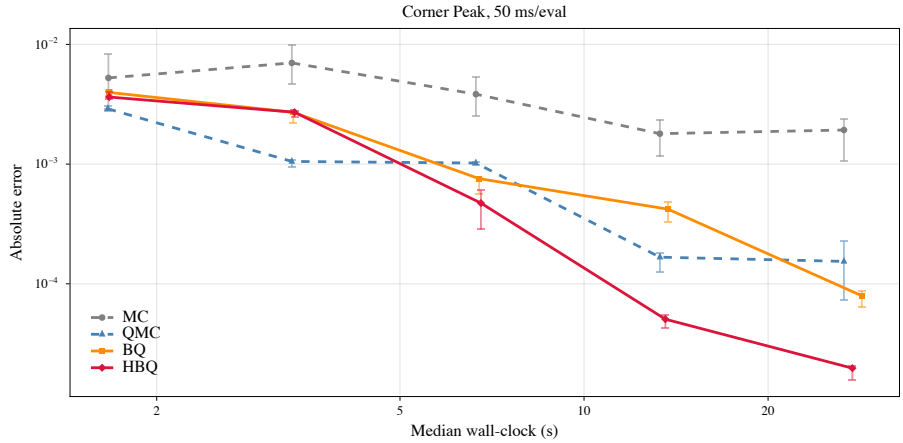

Figure 8: **Wall-clock cost versus accuracy.** Median absolute error against median wall-clock on Genz Corner Peak ($d = 2$, 10 instances) with a $50\,\mathrm{ms}$ delay injected per evaluation. Because the evaluation cost dominates, all methods share essentially the same wall-clock at each budget, so HBQ's per-evaluation accuracy advantage translates directly into accuracy at matched compute.

two narrow rough Matérn-$1/2$ edges flanking a wide smooth Matérn-$5/2$ center over $[0, 1]$. We swept the BIC penalty multiplier over $\{0.5, 1, 2, 4\}$, applied to the threshold of Eq. (23) (with $1\times$ the default), and the evaluation budget over $N \in \{100, 200, 400\}$, running 100 trials per cell—1200 in total—with trial seeds shared across multipliers for paired comparison. For each cell we recorded the structure-recovery rate (fraction of trials recovering the correct number of leaves), the over- and under-split rates, the absolute integration error, empirical 95% coverage, and the per-leaf evaluation counts (Fig. 10). The criterion never over-split: the over-split rate was 0% across all 1200 trials, and at the default penalty recovery was 99% at $N = 100$ and 100% at $N = 200$ and 400, even though the median per-leaf sample count was only about 34 at $N = 100$. The sole failure mode was conservative under-splitting under an aggressive penalty at the tightest budget (40% recovery, 60% under-split at $4\times$ and $N = 100$), which resolved as the budget grew (94% at $N = 200$, 100% at $N = 400$). Empirical 95% coverage stayed close to nominal (87–96%) in all cells except the aggressive-penalty, tightest-budget cell ($4\times$, $N = 100$), where under-splitting reduced it to 75%. In this regime the BIC approximation is thus empirically robust, erring if at all toward too few leaves rather than spurious partitions.

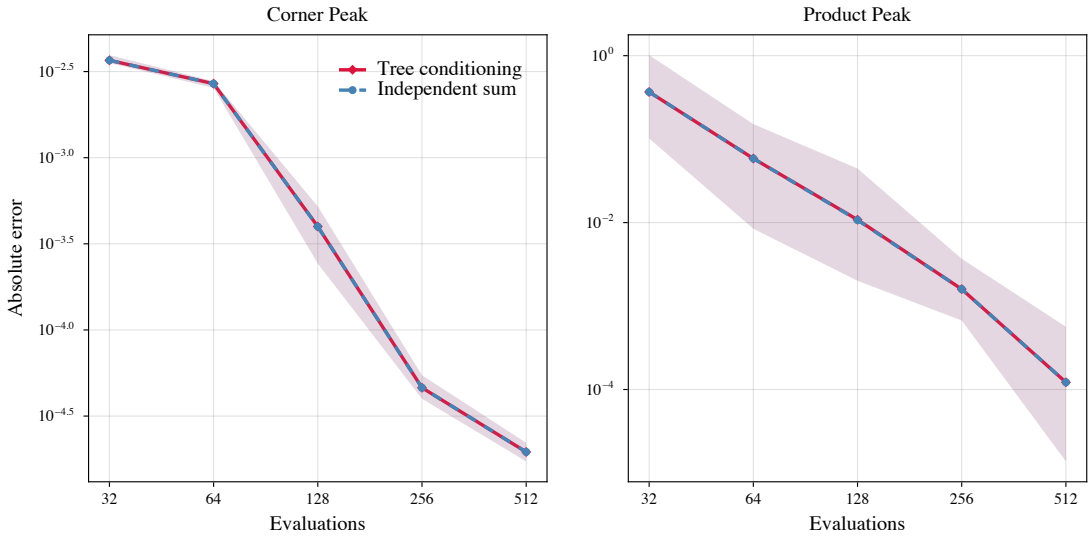

Figure 9: **Tree conditioning versus independent sum.** Convergence under the two recombination modes on Genz Corner Peak and Product Peak ($d = 2$, 20 instances). The two modes coincide to within 0.05% in both median absolute error and posterior standard deviation across all budgets.

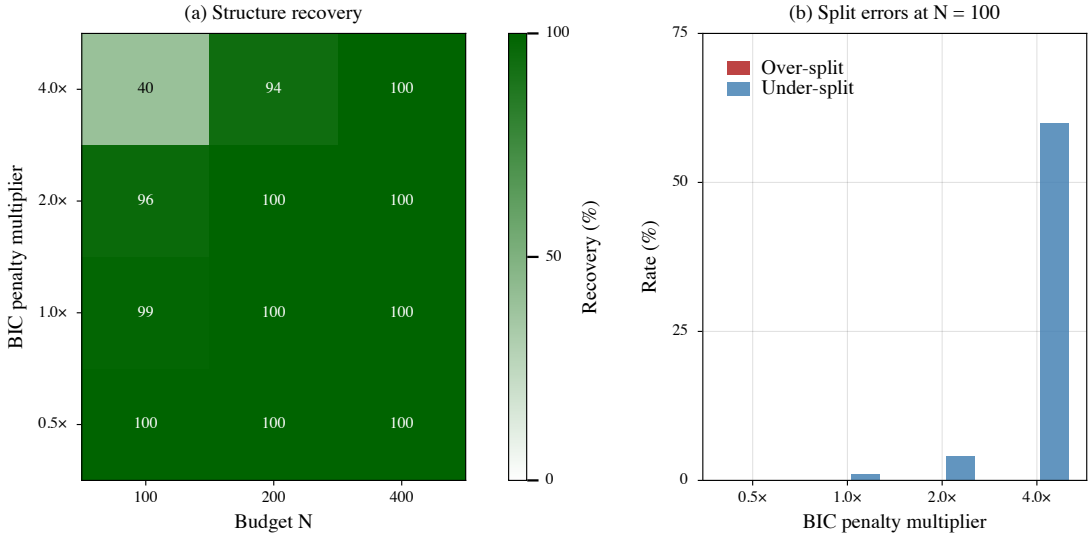

Figure 10: **BIC threshold sensitivity. (a)** Structure-recovery rate over the BIC penalty multiplier and evaluation budget (1200 trials). **(b)** Over- and under-split rates at $N = 100$. The over-split rate is 0% throughout; the only failure mode is conservative under-splitting at the most aggressive penalty and tightest budget.