BayesSum: Bayesian Quadrature in Discrete Spaces

Sophia Seulkee Kang — *Independent Researcher*
François-Xavier Briol — *University College London*
Toni Karvonen — *Lappeenranta–Lahti University of Technology LUT*
Zonghao Chen — *University College London*

Abstract

This paper addresses the challenging computational problem of estimating intractable expectations over discrete domains. Existing approaches, including Monte Carlo and Russian Roulette estimators, are consistent but often require a large number of samples to achieve accurate results. We propose a novel estimator, *BayesSum*, which is an extension of Bayesian quadrature to discrete domains. It is more sample efficient than alternatives due to its ability to make use of prior information about the integrand through a Gaussian process. We show this through theory, deriving a convergence rate significantly faster than Monte Carlo in a broad range of settings. We also demonstrate empirically that our proposed method does indeed require fewer samples on several synthetic settings as well as for parameter estimation for Conway-Maxwell-Poisson and Potts models.

Proceedings of the 2nd International Conference on Probabilistic Numerics (ProbNum) 2026, Lappeenranta, Finland. PMLR Volume TBA.

1 Introduction

We consider the challenge of computing an expectation over a discrete domain, which can typically be expressed as an intractable sum to be estimated. More precisely, for a discrete domain $\mathcal{X}$, an integrable function $f : \mathcal{X} \to \mathbb{R}$, and a probability measure $\mathbb{P}$ on $\mathcal{X}$ with probability mass function $p : \mathcal{X} \to \mathbb{R}$, the quantity of interest is:

$$I = \mathbb{E}_{X\sim\mathbb{P}}[f(X)] = \sum_{x\in\mathcal{X}} f(x)p(x). \tag{1}$$

In practice, I is typically intractable either because there are countably many terms (e.g., when $\mathcal{X} = \mathbb{N}$), or because there is a finite but prohibitively large number of terms, such as the set of all configurations in a combinatorial structure such as a lattice.

The challenge of estimating I arises for statistical and machine learning models of discrete data which *lack* a closed-form normalization constant, rendering maximum likelihood training, posterior inference, and model evaluation computationally intractable. Prominent examples of unnormalized likelihood models include the Ising/Potts model for protein sequences (Balakrishnan et al., 2011) and statistical physics (Baxter, 2016), spatial point processes for ecological and epidemiological data (Moller and Waagepetersen, 2003), multivariate count data (Piancastelli et al., 2023), and exponential random graph models for social networks (Robins et al., 2007). In Bayesian inference, the intractable constant of these likelihood models (e.g., a Potts model with large state spaces) along with the intractable posterior normalizing constant leads to the notorious *doubly intractable* posterior distributions (Murray et al., 2006).

Despite its broad relevance, surprisingly few methods exist in the literature for estimating I (in (1)) as compared to its counterpart in the continuous domain. The most straightforward approach is the Monte Carlo (MC) method (Rubinstein and Kroese, 2016), where N i.i.d. samples $x_1, \ldots, x_N$ are drawn from $\mathbb{P}$ and the intractable sum is approximated by the empirical average $\hat{I}_{\text{MC}} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$. Under mild conditions, $\hat{I}_{\text{MC}}$ is consistent, however, its convergence rate remains relatively slow, as MC does not exploit any structural properties of the integrand f such as smoothness or sparsity. In the continuous domain, a family of approaches has been proposed that leverage such properties of f and consequently result in faster convergence rate. Prominent examples include quasi Monte Carlo (Dick, 2011), Stein variational gradient descent (SVGD) (Liu and Wang, 2018), classical cubature methods (Davis and Rabinowitz, 2007), kernel herding and Stein points (Chen et al., 2010, 2018) kernel thinning and its variants (Dwivedi and Mackey, 2024, 2022; Li et al., 2024; Carrell et al., 2025), stationary points (Chen et al., 2025a), and Bayesian quadrature (Briol et al., 2019)[1]. In contrast, analogous developments for the discrete domain are far more limited.

In this paper, we propose a novel approach for discrete domains called *BayesSum*. The name comes from the fact that our approach extends the Bayesian quadrature algorithm (O'Hagan, 1991; Diaconis, 1988; Rasmussen and Ghahramani, 2003; Briol et al., 2019) to the computation of intractable expectations taking the form of sums over a discrete domain $\mathcal{X}$. By leveraging the structural properties of the integrand f through a prior Gaussian process (Williams and Rasmussen, 2006) over $\mathcal{X}$, BayesSum achieves a fast rate of convergence, i.e., we observe in our experiments that it requires fewer samples than competing methods to reach the same approximation accuracy. BayesSum falls within the framework of Bayesian probabilistic numerical methods (Cockayne et al., 2019; Hennig et al., 2022) in that it provides finite-sample Bayesian uncertainty quantification.

In addition to the basic BayesSum algorithm, we in-

[1]The superior numerical integration performance of points produced by SVGD and Stein points has been observed empirically, without theoretical guarantees.

troduce the following variants that further improve its applicability and efficiency:

1. *BayesSum + Bayesian quadrature.* BayesSum can be seamlessly combined with Bayesian quadrature, resulting in a novel algorithm for estimating intractable expectations over mixed domains with both discrete and continuous components. Computing expectations over mixed domains remains largely underexplored in the literature, with only a few baselines available.
2. *Active BayesSum.* We propose active BayesSum, which selects future query locations by maximizing a utility function, such as expected information gain. This active learning strategy further enhances the sample efficiency of BayesSum by only performing function evaluations at locations most useful for computing the expectation.
3. *Stein BayesSum.* Similarly to Bayesian quadrature, a central ingredient of BayesSum is the availability of closed-form kernel mean embeddings. To this end, we provide in Table 1 a comprehensive dictionary of kernel–distribution pairs that admit such closed-form expressions in the discrete setting. For cases where closed-form embeddings are not available (e.g., where $\mathbb{P}$ is known only up to a normalization constant), we propose Stein BayesSum, which leverages discrete Stein kernels (Yang et al., 2018) to bypass the need for explicit kernel mean embeddings.

We empirically verify the effectiveness of all of these variants of BayesSum over several synthetic examples where we have access to the ground truth of (1) which allows extensive benchmarking. We then also test BayesSum in more realistic settings, including to perform parameter estimation for a Conway–Maxwell–Poisson model of count data, and for a large Potts model on a set of sequences resembling biological proteins.

2 Background and Related Work

In this section, we begin by reviewing existing methods for estimating intractable expectations over discrete domains. Notably, none of these methods exploit structural information about the integrand f. Next, we present Bayesian quadrature, a widely used framework for numerical integration in continuous spaces. Finally, we introduce Gaussian processes and reproducing kernels on discrete domains $\mathcal{X}$, which provide the technical foundations for BayesSum.

2.1 Existing methods

Importance sampling: Beyond the standard Monte Carlo estimator introduced above, alternative sampling distributions are often helpful. Importance sampling estimates the expectation in (1) by rewriting the sum under an easy-to-sample proposal distribution $\mathbb{Q}$ with mass function q that shares the same support as $\mathbb{P}$. Specifically, $I = \sum_{x\in\mathcal{X}} f(x)\frac{p(x)}{q(x)}q(x) = \mathbb{E}_{X\sim\mathbb{Q}}[f(X)\frac{p(X)}{q(X)}]$. This formulation allows us to approximate I by drawing samples from $\mathbb{Q}$ and averaging the alternative integrand $f(x)p(x)/q(x)$ (Owen, 2013). The choice of $\mathbb{Q}$ is critical to the variance of the final estimator.

Russian roulette: A specific approach to estimate (1) over a discrete domain without exhaustively summing over all configurations in $\mathcal{X}$ is to use an unbiased random truncation scheme, often called the *Russian roulette* estimator. The key idea is to evaluate the summand only on a randomly selected subset $\mathcal{X}' \subset \mathcal{X}$, and to compensate for the missing terms through appropriately re-weighting so that the final estimator remains unbiased. In practice, the Russian roulette estimator is known to suffer from high variance (Lyne et al., 2015; Hendricks and Booth, 2006). Although it bears similarity to importance sampling, it differs fundamentally by summing over only a random subset $\mathcal{X}'$ instead of the entire $\mathcal{X}$.

Stratified sampling: Another approach to estimating (1) is stratified sampling, where the domain is partitioned into M disjoint subsets and $\lfloor N/M \rfloor$ samples are drawn from the conditional distribution within each subset, thereby enforcing more uniform coverage of the domain (Owen, 2013, Chapter 10). In continuous settings, quasi–Monte Carlo (QMC) methods can be viewed as a refinement of stratified sampling, using increasingly finer partitions that yield low-discrepancy point sets (Caflisch, 1998; Owen, 2003). However, QMC methods are not directly applicable to discrete domains, as it is unclear how to construct low-discrepancy point sets in general discrete spaces. Another limitation of stratified sampling is that it requires a tractable partition of $\mathcal{X}$ and efficient sampling from each partition, which becomes challenging in domains of high dimensionality.

Kernel thinning: A family of methods closely related to the estimation of intractable expectations, in both continuous and discrete domains, is kernel thinning (Dwivedi and Mackey, 2024). Although originally proposed for compression, i.e., for selecting a small set of representative samples from a larger collection, the methods are motivated by numerical integration, reducing to a two-stage procedure of oversampling followed by thinning. In particular, KT-Split (Dwivedi and Mackey, 2022), KT-Split-Compress (Shetty et al., 2022), Gram–Schmidt Thinning and Gram–Schmidt Compress algorithms (Carrell et al., 2025) all achieve faster-than-Monte-Carlo rates for integration error in discrete spaces. The main distinction, between thinning and quadrature considered in this paper, is that our setting assumes access to only N sample points, whereas kernel thinning begins with a much larger and oversampled candidate set and subsequently thins it down to N points.

2.2 Bayesian quadrature

The task of estimating I in the continuous setting, i.e., $I = \int f(x)p(x)\mathrm{d}x$, using weighted averages of function evaluations is known as quadrature (Gautschi, 1981). Quadrature has a long history, and numerous sophisticated methods have been developed (Novak, 2006; Dick, 2011; Bardenet and Hardy, 2020; Sun et al., 2023; Li and Sun, 2023; Chopin and Gerber, 2024; Chen et al., 2025a). Of particular relevance to our work is *Bayesian quadrature* (BQ) (O'Hagan, 1991; Diaconis, 1988; Rasmussen and Ghahramani, 2003; Briol et al., 2019), in which the estimator is obtained by integrating the posterior mean of a Gaussian process (GP). It has been shown both theoretically and empirically, in the continuous

domain, that BQ estimators attain a faster convergence rate than standard Monte Carlo (Kanagawa et al., 2016; Briol et al., 2019; Kanagawa and Hennig, 2019; Karvonen et al., 2020; Wynne et al., 2021; Chen et al., 2025b); in other words, for a given error tolerance, BQ requires significantly fewer samples. This improvement in sample efficiency arises from exploiting the smoothness of the integrand f encoded through the GP prior covariance. Remarkably, BQ continues to achieve superior performance even when the integrand f is less smooth than samples drawn from the GP prior, i.e., in the misspecified setting (Kanagawa et al., 2020). BQ has been used, for instance, in estimation of normalization constants (Osborne et al., 2012), in reinforcement learning (Paul et al., 2018) and simulation-based inference (Bharti et al., 2023). It has also been extended to conditional and nested expectations (Chen et al., 2024, 2025b) as well as multi-output (Xi et al., 2018) and multi-fidelity settings (Li et al., 2023).

2.3 Gaussian Processes and Reproducing kernel Hilbert Space on Discrete Domains

Here we briefly recall Gaussian processes and kernels defined on discrete domains, which we will use in BayesSum to encode prior knowledge about the integrand f.

For a positive semi-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, its unique associated reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ is a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$ and norm $\|\cdot\|_{\mathcal{H}_k}$ (Aronszajn, 1950) such that: (i) $k(x, \cdot) \in \mathcal{H}_k$ for all $x \in \mathcal{X}$, and (ii) the reproducing property holds, i.e. for all $h \in \mathcal{H}_k, x \in \mathcal{X}$, $h(x) = \langle h, k(x, \cdot) \rangle_{\mathcal{H}_k}$.

Although kernels defined on continuous domains, e.g., Gaussian RBF kernels, remain valid (i.e., symmetric and positive semi-definite) when restricted to discrete domains (Steinwart and Christmann, 2008, Theorem 4.16), it is unclear whether the corresponding RKHS retains the same structural (e.g smoothness) properties as in the continuous case. For example, on an open subset $\mathcal{X} \subseteq \mathbb{R}^d$, the RKHS of a Matérn kernel on $\mathcal{X} \subseteq \mathbb{R}^d$ coincides with a Sobolev space, consisting of functions with square-integrable derivatives up to a certain order (Wendland, 2004, Corollary 10.48). However, when the domain $\mathcal{X}$ is discrete, it is unclear whether such smoothness remains true. An exception is the Brownian motion kernel $k_0(x, y) = \min(x, y)$, which gives the minimum of the two arguments. On the discrete domain $\mathcal{X} = \mathbb{N}$, its RKHS $\mathcal{H}_{k_0}$ consists of functions $f : \mathbb{N} \to \mathbb{R}$ satisfying $\sum_{n \in \mathbb{N}} |f(n+1) - f(n)|^2 < \infty$ (Jorgensen and Tian, 2015), similar to its continuous analogue. If the domain contains 0, however, then $k_0(0, y) = 0$ for every y, so any Gram matrix whose design contains 0 has a zero row and column. We therefore use the shifted Brownian motion kernel $k_c(x, y) = c + \min(x, y)$ with $c > 0$. Indeed, for pairwise distinct points $0 \leq x_1 < \cdots < x_N$, its Gram matrix $\mathbf{K}_c$ satisfies $\det(\mathbf{K}_c) = (c + x_1) \prod_{i=2}^{N} (x_i - x_{i-1}) > 0$ and is thus invertible. The shift adds a constant-function component while preserving the increment-based smoothness encoded by k_0.

Fortunately, there exist several reproducing kernels which are explicitly designed for discrete domains. The most widely used among them is the *Hamming distance kernel*: for $\mathcal{X} = \{0, 1\}^L$, the kernel is $k(x, y) = d_H(x, y)$ where $d_H(x, y) = \sum_{i=1}^{L} 1\{x_i \neq y_i\}$ represents the Hamming distance between two sequences of length L. This kernel encodes the prior knowledge that sequences are more similar if they differ in fewer positions, making it useful for biological sequences. Another alternative is the *exponential Hamming kernel* proposed in Amin et al. (2023) where $k(x, y) = A \exp(-\lambda d_H(x, y))$ with A and λ being the kernel amplitude and lengthscale. This kernel encodes similar prior knowledge as above and it achieves better empirical performances in kernel ridge regression (Amin et al., 2023), making it the default kernel in `Botorch` for categorical inputs (Balandat et al., 2020). There also exist other kernels defined over discrete domains which are tailored to specific applications; see Gartner (2008) and Jorgensen and Tian (2015) for surveys.

Apart from RKHSs, another popular framework built upon positive semi-definite kernels is Gaussian processes (GPs) (Williams and Rasmussen, 2006), denoted as $\mathcal{GP}(m, k)$, which is a stochastic process over functions $f : \mathcal{X} \to \mathbb{R}$ such that every finite collection $[f(x_1), \ldots, f(x_n)]^\top$ follows a multivariate normal distribution with mean $[m(x_1), \ldots, m(x_n)]^\top$ and covariance $[k(x_i, x_j)]_{i,j} \in \mathbb{R}^{n \times n}$. One particular application of Gaussian processes over discrete domains $\mathcal{X}$ (Fortuin et al., 2021) is their use as probabilistic surrogate models in Bayesian optimization when the black-box function to be optimized is defined on discrete domains (Balandat et al., 2020; Ru et al., 2020; Daulton et al., 2022).

3 Methodology

We are now ready to introduce *BayesSum*, a Bayesian estimator for the intractable expectation in Eq. (1), with N samples $x_{1:N} := [x_1, \ldots, x_N]^\top$ and corresponding function evaluations $f(x_{1:N}) := [f(x_1), \ldots, f(x_N)]^\top$.

3.1 BayesSum

We begin by positing a GP prior on f. We denote this prior $\mathcal{GP}(m, k)$, with $m : \mathcal{X} \to \mathbb{R}$ a mean function and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive semi-definite kernel function as reviewed above. Without loss of generality, here we take $m \equiv 0$ as any nonzero mean can be absorbed by centering the observations (i.e., subtracting the empirical mean). The kernel function k encodes prior knowledge about the notion of similarity or distance on $\mathcal{X}$ (e.g., the Hamming distance for biological strings). In this section, we assume that $\mathbb{E}_{X \sim \mathbb{P}}[\sqrt{k(X, X)}] < \infty$.

Once a GP prior has been selected and conditioned on *noiseless* function values $f(x_{1:N}) = [f(x_1), \cdots, f(x_N)]^\top$, we obtain a posterior GP on f which induces a univariate Gaussian posterior distribution $\mathcal{N}(\hat{I}, \hat{\sigma}^2)$ on I with:

$$\hat{I} := \mu_{\mathbb{P}}(x_{1:N})^\top \mathbf{K}^{-1} f(x_{1:N}), \qquad \text{(BayesSum)}$$
$$\hat{\sigma}^2 := \mathbb{E}_{X, X' \sim \mathbb{P}}[k(X, X')] - \mu_{\mathbb{P}}(x_{1:N})^\top \mathbf{K}^{-1} \mu_{\mathbb{P}}(x_{1:N}).$$

Here, $\mu_{\mathbb{P}}(x) = \mathbb{E}_{X \sim \mathbb{P}}[k(X, x)]$ is the kernel mean embedding, $\mathbb{E}_{X, X' \sim \mathbb{P}}[k(X, X')]$ is often called the initial error, and $\mathbf{K} = k(x_{1:N}, x_{1:N}) \in \mathbb{R}^{N \times N}$ is the associated Gram matrix with entries $\mathbf{K}_{i,j} = k(x_i, x_j)$. When the kernel

is finite-rank, so that the associated Gram matrix is singular (e.g., the linear or Tanimoto kernel), a small diagonal regularization term is added to ensure numerical stability. Beyond the posterior mean, BayesSum also provides a posterior variance at only marginal additional cost, which quantifies the epistemic uncertainty due to the reliance on a finite set of function evaluations—a perspective rooted in the framework of probabilistic numerics (Hennig et al., 2022). We refer to the proposed estimator as *BayesSum*[2].

Remark 1. *The BayesSum posterior mean and variance mirror those of Bayesian quadrature* exactly. *The only distinction is that our domain $\mathcal{X}$ is discrete rather than continuous. For this reason, referring to the method as quadrature would be misleading, since quadrature concerns numerical integration over continuous domains. Given this equivalence, the BayesSum posterior mean can be written as a weighted sum of function evaluations $\sum_{i=1}^N w_i f(x_i)$, where the weights $w_{1:N} = \mu_{\mathbb{P}}(x_{1:N})^\top \mathbf{K}^{-1}$ are* optimal *in the associated RKHS (Huszár and Duvenaud, 2012; Briol et al., 2019).*

Computational cost: The computational cost of our BayesSum with N samples is $\mathcal{O}(N^3)$, which is much larger than the $\mathcal{O}(N)$ cost of Monte Carlo. Numerous approaches have been proposed in the literature to reduce the cost of Bayesian quadrature in the continuous domain: for instance, one can use geometric properties of the point set (Karvonen and Sarkka, 2018; Karvonen et al., 2019; Kuo et al., 2025), Nyström approximations (Hayakawa et al., 2022, 2023), or randomly pivoted Cholesky (Epperly and Moreno, 2023). Among these methods, Nyström approximations and randomly pivoted Cholesky can also be applied to the discrete domain. Moreover, since the BayesSum weights do not depend on the integrand f, they can be precomputed once and reused, reducing the computational cost to $\mathcal{O}(N)$ (Chen et al., 2025b, Appendix F.1). See Section 4 for details on how this is implemented in our experiments. Furthermore, in scenarios where the dominant cost arises from function evaluations (e.g., expensive computer simulations) or from obtaining samples (e.g., Markov chain Monte Carlo), the superior sample efficiency of BayesSum allows it to achieve accurate estimates with far fewer evaluations than MC, thereby reducing the overall computational cost in practice.

Hyperparameter selection: The choice of kernel hyperparameters, such as amplitude A and lengthscale λ for the exponential Hamming distance kernel $k(x,y) = A\exp(-\lambda d_H(x,y))$, is crucial for both the accuracy of the BayesSum estimator and the calibration of its posterior variance. In our experiments, these are selected via empirical Bayes, which involves choosing the values that maximize the log marginal likelihood (Williams and Rasmussen, 2006) from a predefined candidate set, e.g., $\{1.0, 10.0, 100.0, 1000.0\}$ for the amplitude A and $\{0.1, 0.3, 1.0, 3.0, 10.0\}$ for the lengthscale λ.

Convergence guarantees: The following theorem shows that the BayesSum estimator $\hat{I}$ converges to the ground truth I if $f \in \mathcal{H}_k$. To optimize the rate of convergence, one should use samples at which most of the probability mass is concentrated.

Theorem 1. *Without loss of generality, let $\mathcal{X}$ admit an enumeration $\mathcal{X} = \{x_i\}_{i=1}^\infty$ where the first N samples $\{x_i\}_{i=1}^N$ correspond to the observed samples. Suppose that $\mathbb{E}_{X\sim\mathbb{P}}[\sqrt{k(X,X)}] < \infty$, and that the N samples $\{x_i\}_{i=1}^N$ are non-repetitive. Then, for any $f \in \mathcal{H}_k$, $|I - \hat{I}| \to 0$ as $N \to \infty$. Moreover, if there exists a constant $C > 0$ satisfying $\sup_{x\in\mathcal{X}} \sqrt{k(x,x)} \le C$, then $|I - \hat{I}| \le C\|f\|_{\mathcal{H}_k}\left(1 - \sum_{i=1}^N p(x_i)\right)$.*

Proof. BayesSum is a special case of Bayesian quadrature, which is worst-case optimal in $\mathcal{H}_k$. From this it follows that, for any weights $w_1, \ldots, w_N \in \mathbb{R}$,

$$\hat{\sigma}^2 \le \mathbb{E}_{X,X'\sim\mathbb{P}}[k(X,X')] - 2\sum_{i=1}^N w_i \mu_{\mathbb{P}}(x_i) + \sum_{i,j=1}^N w_i w_j k(x_i,x_j). \tag{2}$$

We refer to Section 2 in Briol et al. (2019) for this result. Select $w_i = p(x_i)$. Then it is easy to compute that the right-hand side in (2) equals $\sum_{i,j>N} k(x_i,x_j)p(x_i)p(x_j)$ and it can be further upper bounded with

$$\hat{\sigma}^2 \le \sum_{i,j>N} \sqrt{k(x_i,x_i)}\sqrt{k(x_j,x_j)}p(x_i)p(x_j) = \left(\sum_{i>N} \sqrt{k(x_i,x_i)}p(x_i)\right)^2 \to 0 \text{ as } N \to \infty.$$

The convergence follows because $\sum_{i=1}^\infty \sqrt{k(x_i,x_i)}p(x_i) < \infty$. The claim of the theorem follows from Equation (5) in Briol et al. (2019) that $|I - \hat{I}| \le \|f\|_{\mathcal{H}_k}\hat{\sigma}$. In the bounded-kernel case, $|I - \hat{I}| \le C\|f\|_{\mathcal{H}_k}\left(1 - \sum_{i=1}^N p(x_i)\right) \to 0$ as $N \to \infty$. □

The example in Appendix C shows that the upper bound in Theorem 1 is relatively tight. It also shows that $|I - \hat{I}| = \mathcal{O}(N^{-\alpha+1})$ for a probability mass function with tail exponent α, which improves upon the standard Monte Carlo rate $\mathcal{O}(N^{-1/2})$ whenever $\alpha > 3/2$.

Remark 2 (Non-repetitive points)**.** *We would like to highlight that our theory requires non-repetitive samples. This is a distinctive feature in the discrete domain, since in continuous domains the probability of drawing identical samples is zero. We observe empirically that repetitive sampling leads to notably larger errors in the experiments (see Figure 6). We attribute this degradation in performance to two factors: (i) the resulting Gram matrix becomes singular, and (ii) conditioning the GP prior on repetitive samples leaves the BayesSum posterior unchanged, providing no additional information.*

3.2 Estimate intractable expectations over mixed domains

BayesSum can be seamlessly combined with Bayesian quadrature (BQ), resulting in a novel estimator for intractable expectations over *mixed domains*, i.e., domains that involve both continuous and discrete parameters. Consider a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which takes two

[2]Despite the similar terminology, this should not be confused with kernel Bayes rule (Fukumizu et al., 2011) or kernel sum rule (Nishiyama et al., 2020).

Table 1: Closed-form kernel mean embeddings and availability of closed-form initial errors. A check mark in the final column indicates that the initial error admits a closed-form expression; a cross indicates that it does not. The available formulas and derivations are given in Appendix B. Q denotes regularized Gamma function (Silverman and Lebedev, 1972), I_q denotes regularized incomplete Beta function (Silverman and Lebedev, 1972), T_n denotes Touchard polynomial function (Touchard, 1939), M_n denotes the generalized Touchard polynomial function, and B_n denotes complete exponential Bell polynomial function (Bell, 1934). For the shifted Brownian motion kernel, $c > 0$ denotes the constant offset. We omit kernel amplitude A, as it merely rescales the kernel mean embeddings.

$\mathcal{X}$	$\mathbb{P}$	Kernel	Kernel mean embedding $\mu_{\mathbb{P}}(y)$	Initial error
$\mathbb{N}_0$	Poisson(η)	$k_c(x,y) = c + x \wedge y$	$c + \sum_{n=0}^{y} n \frac{e^{-\eta}\eta^n}{n!} + y(1 - Q(y+1, \eta))$	✓
$\mathbb{N}_0$	NB(τ, q)	$k_c(x,y) = c + x \wedge y$	$c + y - \sum_{n=0}^{y-1} I_q(\tau, n+1)$	✓
$\mathbb{N}_+$	Log(q)	$k_c(x,y) = c + x \wedge y$	$c + y + \frac{1}{\ln(1-q)} \sum_{n=1}^{y-1} \frac{(y-n)q^n}{n}$	✗
$\mathbb{N}_0$	Poisson(η)	$k(x,y) = (xy+1)^r$	$\sum_{n=0}^{r} \binom{r}{n} y^n T_n(\eta)$	✓
$\mathbb{N}_0$	NB(τ, q)	$k(x,y) = (xy+1)^r$	$\sum_{n=0}^{r} \binom{r}{n} y^n M_n(\tau, q)$	✓
$\mathbb{Z}$	Skellam(λ_1, λ_2)	$k(x,y) = (xy+1)^r$	$\sum_{n=0}^{r} \binom{r}{n} y^n B_n(\lambda_1 - \lambda_2, \ldots, \lambda_1 + (-1)^n \lambda_2)$	✓
$\{-1, +1\}^d$	Uniform	$k(x,y) = \exp(-\lambda d_H(x,y))$	$\exp(-0.5\lambda d) \left[\cosh\left(\frac{\lambda}{2}\right)\right]^d$	✓
$\{0, \ldots, m\}^d$	Uniform	$k(x,y) = \exp(-\lambda d_H(x,y))$	$\left(\frac{1+me^{-\lambda}}{m+1}\right)^d$	✓
$\{0,1\}^d$	Uniform	$k(x,y) = \frac{x^\top y}{\|x\|^2 + \|y\|^2 - x^\top y}$	See Appendix B	✓

inputs: $x \in \mathcal{X}$ is a discrete input (e.g. natural numbers) and $y \in \mathcal{Y}$ is a continuous input (e.g. real vectors). Let $\mathbb{P}$ be a probability measure over this mixed domain and the target of interest in this setting is an intractable expectation over the mixed domain:

$$I = \mathbb{E}_{(X,Y)\sim\mathbb{P}}[f(X,Y)]. \tag{3}$$

Following the same framework as BayesSum, we place a zero-mean Gaussian process prior $\mathcal{GP}(0, k)$ on f with a kernel function $k : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$. One valid option to construct a kernel k over the mixed domain is to take the product of two kernels $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Conditioned on N noiseless evaluations $f(x_{1:N}, y_{1:N}) = [f(x_1, y_1), \ldots, f(x_N, y_N)]^\top$, the GP posterior induces a univariate Gaussian posterior on I with the following mean and variance:

$$\hat{I} := \mu_{\mathbb{P}}(x_{1:N}, y_{1:N})^\top \mathbf{K}^{-1} f(x_{1:N}, y_{1:N}), \tag{4}$$

$$\hat{\sigma}^2 := \mathbb{E}_{(X,Y),(X',Y')\sim\mathbb{P}}[k_{\mathcal{X}}(X,X') k_{\mathcal{Y}}(Y,Y')] - \mu_{\mathbb{P}}(x_{1:N}, y_{1:N})^\top \mathbf{K}^{-1} \mu_{\mathbb{P}}(x_{1:N}, y_{1:N}). \tag{5}$$

Here, $\mu_{\mathbb{P}}(x_{1:N}, y_{1:N}) = \mathbb{E}_{\mathbb{P}}[k_{\mathcal{X}}(X, x_{1:N}) k_{\mathcal{Y}}(Y, y_{1:N}))]$ is the kernel mean embedding and $\mathbf{K} = k_{\mathcal{X}}(x_{1:N}, x_{1:N}) \odot k_{\mathcal{Y}}(y_{1:N}, y_{1:N})$ where $\odot$ denotes the element-wise product of matrices.

3.3 Closed-form kernel mean embeddings

Despite the claimed advantages, a key limitation of BayesSum is the need for closed-form kernel mean embeddings $\mu_{\mathbb{P}}$ and initial error. While a dictionary of such embeddings and initial error is available in continuous domains (Briol et al., 2025), we now provide a complementary (non-exhaustive) dictionary for pairs of kernels and distributions over the discrete domain in Table 1. Beyond the distributions listed in Table 1, closed-form embeddings and initial error can also be derived when $\mathbb{P}$ can be expressed as the push-forward of a distribution using the standard 'change-of-variables' technique (Chen et al., 2025b). In addition, one can also construct kernels with closed-form embeddings for a wider class of distribution $\mathbb{P}$—including those that are known up to its normalizing constant—via the *Stein reproducing kernel* (Yang et al., 2018). We call BayesSum with Stein reproducing kernels *Stein BayesSum*.

Let $\mathcal{X}$ be a finite discrete domain. Given a base kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the *Stein* reproducing kernel associated with the mass function p can be constructed via a discrete Stein operator (Shi et al., 2022). We focus on a particular discrete Stein kernel based on the difference Stein operators (Yang et al., 2018):

$$k_p(x,x') = s_p(x)^\top k(x,x') s_p(x') - s_p(x)^\top \Delta^*_{x'} k(x,x') - \Delta^*_x k(x,x')^\top s_p(x') + \operatorname{trace}\left(\Delta^*_{x,x'} k(x,x')\right). \tag{6}$$

Here, Δ and Δ^* are partial difference operators defined with respect to a cyclic permutation and an inverse cyclic permutation, and $s_p(x) = \frac{\Delta_x p(x)}{p(x)}$ is the *difference score* function associated with the mass function p. Notably, s_p is accessible even when p is known only up to its normalization constant. See Appendix A for detailed definitions. The Stein kernel k_p is a positive semi-definite reproducing kernel from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$.

A nice property of the Stein kernel by its construction is that the kernel mean embedding $\mu_{\mathbb{P}}(x) = 0$ for any $x \in \mathcal{X}$. However, this means our GP prior on f encodes beliefs that the intractable sum in (1) has mean zero when the GP prior mean $m \equiv 0$. To weaken this, we place a hierarchical flat prior on the GP prior mean (Karvonen et al., 2018), which results in the following estimator, referred to as *Stein BayesSum*.

$$\hat{I} := (1^\top \mathbf{K}^{-1} 1)^{-1} 1^\top \mathbf{K}^{-1} f(x_{1:N}), \quad \text{(Stein BayesSum)}$$
$$\hat{\sigma}^2 := (1^\top \mathbf{K}^{-1} 1)^{-1}.$$

Here, $\mathbf{K} = k_p(x_{1:N}, x_{1:N}) \in \mathbb{R}^{N \times N}$ is the associated Gram matrix of the Stein kernel and $1 \in \mathbb{R}^N$ is a vector with all entries equal to one. A similar form of the estimator for numerical integration in the continuous domain has been adopted in Karvonen et al. (2018, Section 2.3) and Oates et al. (2017, Lemma 3). Previous Bayesian quadrature methods based on Stein kernels, e.g. Chen et al. (2024, Appendix B.2), introduce an additional additive constant c into the kernel. This constant is treated as a kernel hyperparameter and optimized by maximizing the log-marginal likelihood, which results in a higher computational cost compared to our estimator.

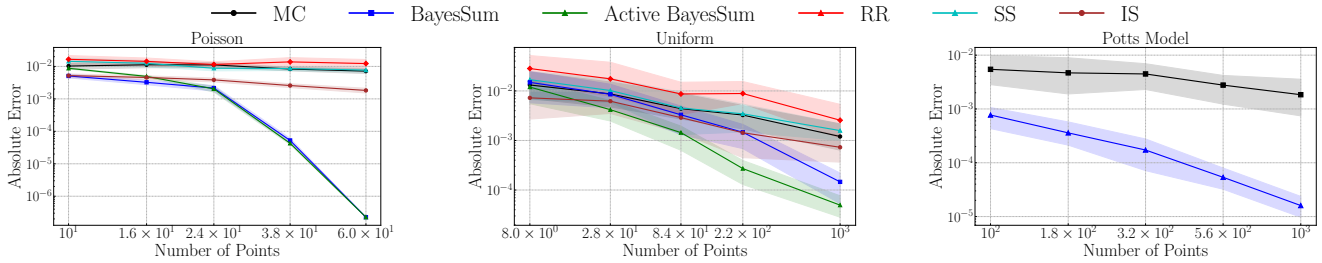

Figure 1: Comparison of BayesSum / active BayesSum against baselines: Monte Carlo (MC), Russian roulette (RR), importance sampling (IS) and stratified sampling (SS). The reported results are absolute error on (a) Poisson distribution (**Left**), (b) uniform distribution over $\mathcal{X} = \{0, 1, 2\}^L$ (**Middle**) and (c) un-normalized distribution characterized by a Potts model (**Right**). Results are averaged over 50 independent runs, while the shaded regions give the 25%-75% quantiles.

3.4 Active BayesSum

Rather than relying on a fixed set of samples, active learning can provide a principled framework for *sequential* sample selection, where future query locations are chosen to maximize a user-specified *utility* function. A natural choice for this utility is the expected information gain, which has been shown to substantially improve sample efficiency over naive randomized sampling in Bayesian quadrature (Gunter et al., 2014; Osborne et al., 2012; Gessner et al., 2020). In our setting, the same utility can in principle be used to guide sample selection. The primary challenge, however, lies in solving the resulting optimization problem over a discrete domain.

Let $\mathcal{D} := \{x_i, f(x_i)\}_{i=1}^N$ denote the set of observed samples. Under a GP prior with mean $m \equiv 0$ and covariance kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the posterior GP conditioned on $\mathcal{D}$ has some mean $\tilde{m} : \mathcal{X} \to \mathbb{R}$ and covariance $\tilde{k}(x, x') := k(x, x') - k(x, x_{1:N})k(x_{1:N}, x_{1:N})^{-1}k(x_{1:N}, x')$. Recall from Section 3 that the BayesSum estimator induces a univariate Gaussian posterior distribution $\mathcal{N}(\hat{I}, \hat{\sigma}^2)$. As a result, the mutual information between I and a new observation $f(x^*)$ admits the following closed-form expression (Gessner et al., 2020):

$$\alpha(x^*) = -\frac{1}{2}\log\left(1 - \frac{(\mathbb{E}_{X\sim\mathbb{P}}[\tilde{k}(X, x^*)])^2}{\hat{\sigma}^2\tilde{k}(x^*, x^*)}\right).$$

The main challenge is to identify the optimal next query $x^* \in \arg\max_{x\in\mathcal{X}} \alpha(x)$ over the discrete domain $\mathcal{X}$, where standard gradient-based optimization is not applicable and exhaustive evaluation of the utility across the entire $\mathcal{X}$ is expensive. In our experiments, we select next query by maximizing $\alpha(x^*)$ over a randomly chosen small subset $\mathcal{X}_{\text{sub}} \subset \mathcal{X}$ through exhaustive evaluation. Although straightforward, this approach may miss the optimal next query. Another widely used alternative is to maximize the expected utility under an auxiliary distribution on $\mathcal{X}$, parameterized by continuous variables ϑ (Daulton et al., 2022; Swersky et al., 2020).

4 Experiments

We now illustrate BayesSum across a range of applications, including several synthetic examples, a Conway–Maxwell–Poisson model trained on count data, and a large Potts model trained on a set of sequences resembling biological proteins. The code to reproduce all our experiments is available at `https://github.com/seulkang0518/BayesSum`.

4.1 Synthetic data

First, we consider toy synthetic settings with the following pairs of function f and distribution $\mathbb{P}$:

(a): $\mathbb{P}$ is a Poisson distribution over $\mathcal{X} = \mathbb{N}_0$ with rate parameter $\eta = 30$ and $f(x) = \exp(-(x-15)^2/8)$.

(b): $\mathbb{P}$ is a uniform distribution over the state space $\mathcal{X} = \{0, 1, 2\}^L$ and $f(x) = \exp(-\beta\sum_{i=1}^L h_i x_i - \beta\sum_{i\sim j} J_{ij}(x_i * x_j))$. Here, $x_i * x_j$ denotes an element-wise product between their one-hot encoded vectors and $i \sim j$ denotes adjacent positions on a $L \times L$ grid. The dimensionality of the original configuration space is $d = L$, and it becomes $d = 3L$ after one-hot encoding. We set $h_i = 0.1$, $J_{ij} = 0.1$ and $\beta = 1/2.269$ following Moores et al. (2020).

(c): $\mathbb{P}$ is a distribution specified by a Potts model over the state space $\mathcal{X} = \{0, 1, 2\}^L$ where the probability mass function, known up to its normalization constant, is $p(x) \propto \exp(-\beta\sum_{i=1}^L h_i x_i - \beta\sum_{i\sim j} J_{ij}(x_i * x_j))$ where we set $h_i = 0.1$, $J_{ij} = 0.1$ and $\beta = 1/2.269$ as above. The integrand function $f(x) = (1 + \exp(-\frac{1}{L}\sum_{i=1}^L x_i))^{-1}$.

(d): $\mathbb{P}$ is the product of a uniform distribution over $\mathcal{X} = [-1, 1]$ and a uniform distribution over $\mathcal{Y} = \{-1 + 0.125\,j \;:\; j = 0, 1, \ldots, 16\}^2$. The dimensionality of this problem is $d = 3$. The function $f(x, y) = -20\exp[-0.2|(1 + 0.1y_1 + 0.1y_2)x|] - \exp[\cos(2\pi x(1 + 0.05y_1^2 + 0.05y_2^2))] + 20 + e$.

Cases (a)–(d) cover all the various settings in which the proposed *BayesSum* methods in Section 3 are applicable: (a) intractable expectations over countably infinite domains; (b) expectations over finite but combinatorially large domains; (c) expectations over combinatorially large domains where the underlying distribution is known only up to a normalization constant; and (d) expectations over mixed domains. To enable extensive benchmarking against baseline methods, we set $L = 15$ in cases (b) and (c), ensuring that the total number of possible states $3^{15} \approx 10^7$ remains computationally manageable such that the target expectation in (1) can be computed exactly by explicitly summing over all possible states. For case (a), we truncate the infinite sum at $x = 200$ and treat this value as the ground truth, as both the integrand $f(x) = \exp(-(x-15)^2/8)$ and the Poisson distribution decays exponentially fast. Case (d) admits a closed-form expression for its ground truth.

Regarding kernel choices for our BayesSum and active BayesSum, we employ the shifted Brownian motion kernel $k_c(x, y) = c + \min(x, y)$ with $c = 1$ in (a); an exponential Hamming distance kernel in (b); a Stein

Table 2: Comparison of empirical convergence rate of different methods. For each method, we report the estimated $\alpha > 0$ such that the integration error scales as $O(n^{-\alpha})$. Larger values indicate faster convergence.

Method	Poisson	Uniform	Potts
MC	0.455	0.530	0.540
RR	0.335	0.515	—
SS	0.454	0.552	—
IS	0.515	0.545	—
BayesSum	7.322	1.139	**1.800**
Active BayesSum	**7.706**	**1.251**	—

reproducing kernel built upon the exponential Hamming distance kernel in (c); and a product of the Gaussian kernel over the continuous domain and the exponential Hamming distance kernel over the discrete domain in (d). Closed-form expressions for the corresponding kernel mean embeddings are provided in Table 1. For benchmarking, we compare against Monte Carlo (MC), Russian Roulette (RR), importance sampling (IS), and stratified sampling (SS). In cases (a), (b), and (d), Monte Carlo samples are drawn independently with replacement, whereas BayesSum samples are drawn without replacement. This distinction is intentional: duplicate samples can lead to singular Gram matrices in BayesSum; see Figure 6 for a comparison. By contrast, standard Monte Carlo sampling is performed with replacement to preserve its unbiasedness. In case (c), samples for both BayesSum and MC are drawn from an unnormalized distribution using Metropolis–Hastings (Gelman et al., 1995, Chapter 11). The empirical performance is reported by the mean absolute error with respect to the ground truth. Details can be found in Appendix D.

From Figure 1 (**Left**) (**Middle**) (**Right**) and Figure 2 (**Left**), we observe that BayesSum achieves much smaller absolute errors than baseline methods for the same number of samples over all benchmarks from (a) to (d). This strong performance stems partly from the choice of kernel that encodes prior knowledge of the function f; in (a) the Brownian motion kernel captures smooth variations over the inputs, in (b)(c) the Hamming distance kernel captures smooth variations of f across similar discrete sequences, and in (d), the product kernel jointly characterizes smoothness over the continuous domain and similarity over the discrete domain. We also observe that active BayesSum can further improve the sample efficiency of BayesSum, by selecting the next query point that drives the largest reduction in posterior uncertainty. We also compare in Figure 6 the performance of BayesSum with repetitive versus non-repetitive samples. We observe that repetitive sampling leads to larger errors, consistent with Remark 2.

To further illustrate the sample efficiency, we estimate the empirical convergence rate of all methods by performing a linear regression in log–log space between the number of samples and the mean absolute error. Note that for Poisson distribution, the rate is obtained with many more samples than plotted in Figure 1 (**Left**). We observe that MC, RR, SS, and IS all exhibit similar convergence rates around 0.5. This is consistent with the central limit theorem for i.i.d. samples, as well as for samples drawn from a well-mixed Markov chain (Owen, 2013). In contrast, Table 2 shows that both our BayesSum and Active BayesSum achieve substantially faster rates of convergence, confirming their superiority in sample efficiency.

In Figure 2 (**Middle**), we also study the performance of BayesSum, active BayesSum in case (b) with increasing dimension d. We observe that the performance margin over the baselines gets smaller as d increases. This is expected, since Bayesian quadrature and Gaussian process are known to suffer in high dimensions (Chen et al., 2024; Briol et al., 2019), and a similar phenomenon is anticipated in discrete domains. We also study the calibration of the BayesSum posterior variance in Figure 5. We observe that the posterior variances for the uniform distribution over $\{0,1,2\}^d$ in case (b) and the mixed distribution in case (d) are well-calibrated, whereas the posterior variance for the Poisson distribution in (a) and for the Potts model in (d) are underconfident. This behavior arises because the Brownian kernel and the Stein kernel are unbounded or have large kernel magnitudes, which leads to inflated posterior variances even with carefully tuning.

4.2 Parameter estimation for unnormalized models

Conway–Maxwell–Poisson model for count data: The Conway–Maxwell–Poisson (CMP) distribution is a generalization of the Poisson distribution over $\mathcal{X} = \mathbb{N}$ (Conway, 1961), which has been widely used for count data in a wide range of fields including transport, finance, and retail (Inouye et al., 2017; Benson and Friel, 2021; Sellers and Shmueli, 2010). The CMP distribution has two parameters $(\theta_1, \theta_2) \in (0,\infty) \times [0,\infty)$ and its probability mass function $p(x;\theta_1,\theta_2)$ is known only up to a normalization constant $Z(\theta_1,\theta_2)$:

$$p(x;\theta_1,\theta_2) = \frac{\theta_1^x (x!)^{-\theta_2}}{Z(\theta_1,\theta_2)},$$
$$Z(\theta_1,\theta_2) = e^{\eta_0}\mathbb{E}_{X\sim\mathrm{Poisson}(\eta_0)}\left[\left(\frac{\theta_1}{\eta_0}\right)^X (X!)^{1-\theta_2}\right].$$

Our goal is to model the sales dataset of Shmueli et al. (2005) with a CMP distribution with parameters (θ_1,θ_2). To estimate these parameters via gradient-based maximum likelihood, it is necessary to compute the normalization constant $Z(\theta_1,\theta_2)$. To enable the use of BayesSum, we first rewrite the normalization constant $Z(\theta_1,\theta_2)$ as an expectation under a Poisson distribution with rate parameter η_0: $Z(\theta_1,\theta_2) = e^{\eta_0}\mathbb{E}_{X\sim\mathrm{Poisson}(\eta_0)}[\frac{\theta_1}{\eta_0}(X!)^{1-\theta_2}]$, where η_0 is set to be the empirical mean of the data. The key reason we write the normalization constant as an expectation with respect to a fixed distribution is computational: we only need to draw $N = 10$ samples without replacement and to compute the BayesSum weights $w_{1:N} = \mu_{\mathbb{P}}(x_{1:N})^\top \mathbf{K}^{-1}$ once at initialization. We use the shifted Brownian motion kernel $k_c(x,y) = c + \min(x,y)$ with $c = 1$, which admits a closed-form kernel mean embedding $\mu_{\mathbb{P}}$.

With this approximation, the parameters (θ_1,θ_2) are then updated via gradient descent on the estimated negative log-likelihood using a learning rate of 1×10^{-3}

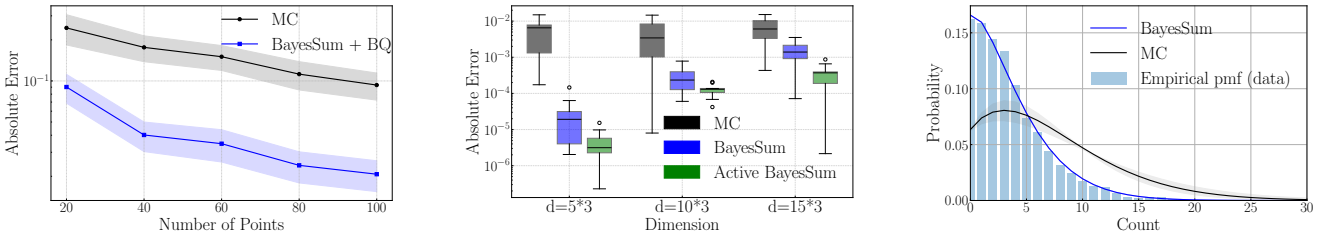

Figure 2: **Left:** Comparison of BayesSum+BQ and MC over the mixed domain. **Middle:** Comparison of BayesSum and active BayesSum and MC with increasing dimension d in case (b). **Right:** Comparison of BayesSum and MC in estimating the normalization constant for maximum likelihood inference of CMP model. BayesSum uses $N = 10$ samples while MC uses $N = 30$ samples.

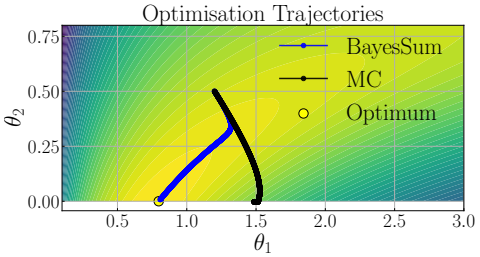

Figure 3: The optimization trajectory of training CMP model with normalization constant estimated via BayesSum and MC.

for 800 iterations where convergence is empirically observed. For comparison, we perform the same training procedure, but with $Z(\theta_1, \theta_2)$ approximated instead by a Monte Carlo estimator $\hat{Z}_{\mathrm{MC}}(\theta_1, \theta_2)$ with $N = 30$ samples at each iteration. As shown in Figure 2 **(Right)**, maximum-likelihood training with $\hat{Z}_{\mathrm{BQ}}$ yields parameters (θ_1, θ_2) that better capture the true distribution of the sales dataset than those obtained with $\hat{Z}_{\mathrm{MC}}$. This improvement arises because our BayesSum provides a more accurate estimate of the normalization constant even with fewer samples, and consequently, of the gradient of the log-likelihood at each iteration. This can be further reflected in Figure 3 where we plot the optimization trajectories of (θ_1, θ_2) for BayesSum and Monte Carlo. The computational time is 4.2s for BayesSum and 4.0s for Monte Carlo.

Potts model for sequence data: The Potts model is a popular model for protein sequences due to its explicit representation of pairwise couplings where mutations at one site are often compensated by changes at another. Formally, the Potts model defines the likelihood of a sequence $x = [x_1, \ldots, x_L] \in \{1, 2, \ldots, S\}^L$ as

$$p(x; \boldsymbol{\theta}) = \frac{\exp\left(\sum_{i=1}^{L} \beta h_i x_i + \sum_{i \sim j} \beta J_{ij}(x_i * x_j)\right)}{Z(\boldsymbol{\theta})}$$

$$Z(\boldsymbol{\theta}) = \sum_{x} \exp\left(\sum_{i=1}^{L} \beta h_i x_i + \sum_{i \sim j} \beta J_{ij}(x_i * x_j)\right). \quad (7)$$

Here, the parameter vector $\boldsymbol{\theta}$ consists of two components: $\mathbf{h} = [h_1, \ldots, h_L]^\top \in \mathbb{R}^L$, which denotes the marginal fields, and $\mathbf{J} = [J_{i,j}] \in \mathbb{R}^{L \times L}$, which encodes the pairwise interaction strengths between x_i and x_j. The symbol $*$ denotes an element-wise product between their one-hot encoded vectors. $\beta = 1/2.269$ represents the inverse temperature. The normalization constant $Z(\boldsymbol{\theta})$ is the sum over S^L terms, which quickly becomes intractable as S and L grow.

We train the Potts model via maximum likelihood using 2000 protein sequences of length $L = 15$ generated as follows: each symbol is independently drawn from the alphabet $\{1, 2, 3\}$ with probabilities $\{0.4, 0.4, 0.2\}$. After one-hot encoding, the total dimension of the problem becomes $d = 15 \times 3 = 45$. As with the CMP distribution above, we perform gradient descent on the log-likelihood and estimate the intractable normalization constant Z at each iteration. To this end, conditioned on a sample x' from the last iteration, we draw $\lceil N/M \rceil$ samples from a product distribution $\prod_{j=1}^{L} \bar{p}_j(x_j \mid x')$ with

$$\bar{p}_j(x_j \mid x'; \mathbf{h}, \mathbf{J}) := \mathrm{softmax}(\beta h_j + \beta \sum_{i \sim j} J_{ij}(x'_i * x_j)).$$

This product distribution is motivated by the pseudo-likelihood approach, which replaces the joint distribution with a product of conditional distributions (Ekeberg et al., 2013). We repeat this procedure for M independent choices of x', yielding a total of N samples. Then, we approximate $Z(\boldsymbol{\theta})$ using our BayesSum estimator $\hat{Z}_{\mathrm{BQ}}$ from these N samples. With this approximation, we are able to optimize the parameters $\mathbf{h}, \mathbf{J}$ by performing gradient descent on the estimated negative log-likelihood, using a learning rate of 10^{-3} for 2000 iterations until convergence is empirically observed. For comparison, we repeat the same procedure as above yet with $Z(\boldsymbol{\theta})$ estimated by a Monte Carlo estimator $\hat{Z}_{\mathrm{MC}}$ from the same N samples as BayesSum at each iteration.

In Figure 4, we compare the two Potts models trained using the above two procedures measured in terms of the squared discrete kernel Stein discrepancy (KSD^2) (Yang et al., 2018) with respect to the true data generating distribution. Figure 4 shows that the final Potts model trained with $\hat{Z}_{\mathrm{BQ}}$ attains much lower KSD^2 than the model trained with $\hat{Z}_{\mathrm{MC}}$, both under an equal computational budget in wall-clock time (**top**) and under an equal number of samples (**bottom**). This result is consistent with case (b) of the synthetic experiments, demonstrating that the benefit of employing a more accurate estimator of the normalization constant would translate into a better-trained model.

5 Conclusions

In this paper, we introduced BayesSum, a novel algorithm for estimating intractable expectations over discrete domains. BayesSum offers improved sample efficiency and provides finite-sample uncertainty quantification for its estimates, making it particularly attractive in scenarios where obtaining samples or function evaluations are expensive.

References

A. N. Amin, E. N. Weinstein, and D. S. Marks. Biological sequence kernels with guaranteed flexibility. *arXiv preprint arXiv:2304.03775*, 2023.

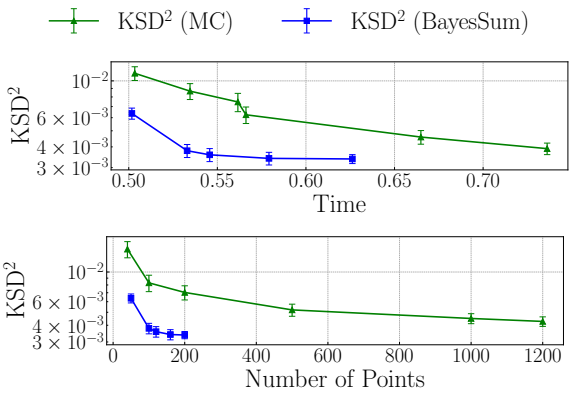

Figure 4: Comparison of a large Potts model trained via maximum log likelihood, with the normalization constant estimated by BayesSum and Monte Carlo, under the same computational time (**top**) and sample size N (**bottom**). Error bars represent the standard error repeated over 30 random seeds.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3): 337–404, 1950.

S. Balakrishnan, H. Kamisetty, J. G. Carbonell, S.-I. Lee, and C. J. Langmead. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078, 2011.

M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: A framework for efficient Monte Carlo Bayesian Optimization. *Advances in Neural Information Processing Systems*, 33:21524–21538, 2020.

R. Bardenet and A. Hardy. Monte Carlo with determinantal point processes. *The Annals of Applied Probability*, 2020.

R. J. Baxter. *Exactly solved models in statistical mechanics*. Elsevier, 2016.

E. T. Bell. Exponential polynomials. *Annals of Mathematics*, 35(2):258–277, 1934.

A. Benson and N. Friel. Bayesian inference, model selection and likelihood estimation using fast rejection sampling: the Conway-Maxwell-Poisson distribution. *Bayesian Analysis*, 16(3):905–931, 2021.

A. Bharti, M. Naslidnyk, O. Key, S. Kaski, and F-X. Briol. Optimally-weighted estimators of the maximum mean discrepancy for likelihood-free inference. In *International Conference on Machine Learning*, pages 2289–2312, 2023.

F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: A role in statistical computation? (with discussion). *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. *Proceedings of the First International Conference on Probabilistic Numerics, PMLR*, 271:84–94, 2025.

R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.

A. M. Carrell, A. Gong, A. Shetty, R. Dwivedi, and L. Mackey. Low-rank thinning. In *Forty-second International Conference on Machine Learning*, 2025.

W. Y. Chen, L. Mackey, J. Gorham, F-X. Briol, and C. J. Oates. Stein points. In *Proceedings of the International Conference on Machine Learning, PMLR 80*, pages 843–852, 2018.

Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 109–116, 2010.

Z. Chen, M. Naslidnyk, A. Gretton, and F.-X. Briol. Conditional Bayesian quadrature. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, pages 648–684, 2024.

Z. Chen, T. Karvonen, H. Kanagawa, F.-X. Briol, C. J. Oates, et al. Stationary MMD points for cubature. *arXiv preprint arXiv:2505.20754*, 2025a.

Z. Chen, M. Naslidnyk, and F.-X. Briol. Nested expectations with kernel quadrature. In *Forty-second International Conference on Machine Learning*, 2025b.

N. Chopin and M. Gerber. Higher-order Monte Carlo through cubic stratification. *SIAM Journal on Numerical Analysis*, 62(1):229–247, 2024.

J. Cockayne, C. J. Oates, T. J. Sullivan, and M. Girolami. Bayesian probabilistic numerical methods. *SIAM Review*, 61(4):756–789, 2019.

L. Comtet. *Advanced Combinatorics: The art of finite and infinite expansions*. Springer Science & Business Media, 2012.

R. W. Conway. A queueing model with state dependent service rate. *Journal of Industrial Engineering*, 12: 132, 1961.

S. Daulton, X. Wan, D. Eriksson, M. Balandat, M. A. Osborne, and E. Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. *Advances in Neural Information Processing Systems*, 35:12760–12774, 2022.

P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

P. Diaconis. Bayesian numerical analysis. *Statistical Decision Theory and Related Topics IV*, 1:163–175, 1988.

J. Dick. Higher order scrambled digital nets achieve the optimal rate of the root mean square error for smooth integrands. *The Annals of Statistics*, pages 1372–1398, 2011.

R. Dwivedi and L. Mackey. Generalized kernel thinning. In *Tenth International Conference on Learning Representations (ICLR 2022)*, 2022.

R. Dwivedi and L. Mackey. Kernel thinning. *Journal of Machine Learning Research*, 25(152):1–77, 2024.

M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 87(1):012707, 2013.

E. Epperly and E. Moreno. Kernel quadrature with randomly pivoted Cholesky. *Advances in Neural Information Processing Systems*, 36:65850–65868, 2023.

V. Fortuin, G. Dresdner, H. Strathmann, and G. Ratsch. Sparse Gaussian processes on discrete domains. *IEEE Access*, 9:76750–76758, 2021.

K. Fukumizu, L. Song, and A. Gretton. Kernel Bayes' rule. *Advances in Neural Information Processing Systems*, 24, 2011.

T. Gartner. *Kernels for structured data*, volume 72. World Scientific, 2008.

W. Gautschi. A survey of Gauss-Christoffel quadrature formulae. In *EB Christoffel: The influence of his work on mathematics and the physical sciences*, pages 72–147. Springer, 1981.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

A. Gessner, J. Gonzalez, and M. Mahsereci. Active multi-information source Bayesian quadrature. In *Uncertainty in Artificial Intelligence*, pages 712–721. PMLR, 2020.

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

S. Hayakawa, H. Oberhauser, and T. Lyons. Positively weighted kernel quadrature via subsampling. *Advances in Neural Information Processing Systems*, 35:6886–6900, 2022.

S. Hayakawa, H. Oberhauser, and T. Lyons. Sampling-based Nyström approximation and kernel quadrature. In *International Conference on Machine Learning*, pages 12678–12699. PMLR, 2023.

J. S. Hendricks and T. E. Booth. MCNP variance reduction overview. In *Monte-Carlo Methods and Applications in Neutronics, Photonics and Statistical Physics*, pages 83–92. Springer, 2006.

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

F. Huszár and D. Duvenaud. Optimally-weighted herding is bayesian quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 377–386, 2012.

D. I. Inouye, E. Yang, G. I. Allen, and P. Ravikumar. A review of multivariate distributions for count data derived from the Poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3): e1398, 2017.

P. Jorgensen and F. Tian. Discrete reproducing kernel Hilbert spaces: sampling and distribution of dirac-masses. *The Journal of Machine Learning Research*, 16(1):3079–3114, 2015.

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

M. Kanagawa, B. Sriperumbudur, and K. Fukumizu. Convergence guarantees for kernel-based quadrature rules in misspecified settings. In *Advances in Neural Information Processing Systems*, pages 3288–3296, 2016.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

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.

T. Karvonen and S. Sarkka. Fully symmetric kernel quadrature. *SIAM Journal on Scientific Computing*, 40(2):A697–A720, 2018.

T. Karvonen, C. J. Oates, and S. Sarkka. A Bayes-Sard cubature method. *Advances in Neural Information Processing Systems*, 31, 2018.

T. Karvonen, S. Sarkka, and C. J. Oates. Symmetry exploits for Bayesian cubature methods. *Statistics and Computing*, 29(6):1231–1248, 2019.

T. Karvonen, C. J. Oates, and M. Girolami. Integration in reproducing kernel Hilbert spaces of Gaussian kernels. *Mathematics of Computation*, 90(331):2209–2233, 2020.

F. Y. Kuo, W. Mo, and D. Nuyens. Constructing embedded lattice-based algorithms for multivariate function approximation with a composite number of points. *Constructive Approximation*, 61(1):81–113, 2025.

K. Li and Z. Sun. Multilevel control functional. *arXiv preprint arXiv:2305.12996*, 2023.

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

L. Li, R. Dwivedi, and L. Mackey. Debiased distribution compression. In *Proceedings of the 41st International Conference on Machine Learning*, pages 27675–27731, 2024.

Q. Liu and D. Wang. Stein variational gradient descent as moment matching. *Advances in Neural Information Processing Systems*, 31, 2018.

A.-M. Lyne, M. Girolami, Y. Atchadé, H. Strathmann, and D. Simpson. On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical Science*, 30(4):443–467, 2015.

J. Moller and R. P. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. CRC Press, 2003.

M. Moores, G. K. Nicholls, A. N. Pettitt, and K. Mengersen. Scalable Bayesian inference for the inverse temperature of a hidden potts model. *Bayesian Analysis*, 15(1):1–27, 2020.

I. Murray, Z. Ghahramani, and D. J. C. MacKay. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 359–366. AUAI Press, 2006.

Y. Nishiyama, M. Kanagawa, A. Gretton, and K. Fukumizu. Model-based kernel sum rule: kernel Bayesian inference with probabilistic models. *Machine Learning*, 109(5):939–972, 2020.

E. Novak. *Deterministic and stochastic error bounds in numerical analysis*, volume 1349. Springer, 2006.

C. J. Oates, M. Girolami, and N. Chopin. Control functionals for monte carlo integration. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(3):695–718, 2017.

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

M. Osborne, R. Garnett, Z. Ghahramani, D. K. Duvenaud, S. J. Roberts, and C. Rasmussen. Active learning of model evidence using Bayesian quadrature. *Advances in Neural Information Processing Systems*, 25, 2012.

A. B. Owen. Quasi-Monte Carlo sampling. *Monte Carlo Ray Tracing: Siggraph*, 1:69–88, 2003.

A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

S. Paul, K. Chatzilygeroudis, K. Ciosek, J.-B. Mouret, M. Osborne, and S. Whiteson. Alternating optimisation and quadrature for robust control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

L. S. C. Piancastelli, N. Friel, W. Barreto-Souza, and H. Ombao. Multivariate Conway-Maxwell-Poisson distribution: Sarmanov method and doubly intractable Bayesian inference. *Journal of Computational and Graphical Statistics*, 32(2):483–500, 2023.

C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. *Advances in Neural Information Processing Systems*, pages 505–512, 2003.

G. Robins, T. Snijders, P. Wang, M. Handcock, and P. Pattison. Recent developments in exponential random graph (p*) models for social networks. *Social Networks*, 29(2):192–215, 2007.

B. Ru, A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*, pages 8276–8285. PMLR, 2020.

R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*. John Wiley & Sons, 2016.

K. F. Sellers and G. Shmueli. A flexible regression model for count data. *The Annals of Applied Statistics*, pages 943–961, 2010.

Abhishek Shetty, Raaz Dwivedi, and Lester Mackey. Distribution compression in near-linear time. In *International Conference on Learning Representations*, 2022.

J. Shi, Y. Zhou, J. Hwang, M. Titsias, and L. Mackey. Gradient estimation with discrete stein operators. *Advances in Neural Information Processing Systems*, 35: 25829–25841, 2022.

G. Shmueli, T. P. Minka, J. B. Kadane, S. Borle, and P. Boatwright. A useful distribution for fitting discrete data: revival of the Conway-Maxwell-Poisson distribution. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 54(1):127–142, 2005.

R. A. Silverman and N. N. Lebedev. *Special functions and their applications*. Courier Corporation, 1972.

I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

Z. Sun, A. Barp, and F.-X. Briol. Vector-valued control variates. In *International Conference on Machine Learning*, pages 32819–32846. PMLR, 2023.

K. Swersky, Y. Rubanova, D. Dohan, and K. Murphy. Amortized Bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pages 769–778. PMLR, 2020.

J. Touchard. Sur les cycles des substitutions. *Acta Mathematica*, 1939.

H. Wendland. *Scattered data approximation*, volume 17. Cambridge University Press, 2004.

C. K. I. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press, Cambridge, MA, 2006.

G. Wynne, F.-X. Briol, and M. Girolami. Convergence guarantees for Gaussian process means with misspecified likelihoods and smoothness. *Journal of Machine Learning Research*, 22(123):1–40, 2021.

X. Xi, F.-X. Briol, and M. Girolami. Bayesian quadrature for multiple related integrals. In *International Conference on Machine Learning*, pages 5373–5382. PMLR, 2018.

J. Yang, Q. Liu, V. Rao, and J. Neville. Goodness-of-fit testing for discrete distributions via Stein discrepancy. In *International Conference on Machine Learning*, pages 5561–5570. PMLR, 2018.

Supplementary Material

A Details on Stein reproducing kernel over the discrete domain

Let $\mathcal{X}$ be a finite discrete domain with an ordering $x^{[1]}, x^{[2]}, \ldots, x^{[|\mathcal{X}|]}$. Define the cyclic permutation operator $\neg : \mathcal{X} \to \mathcal{X}$ and its inverse operator $\vdash : \mathcal{X} \to \mathcal{X}$ by $\neg x^{[i]} = x^{[(i+1) \bmod |\mathcal{X}|]}$ and $\vdash x^{[i]} = x^{[(i-1) \bmod |\mathcal{X}|]}$ for any $i = 1, 2, \ldots, |\mathcal{X}|$. It follows immediately that these operators are inverses of each other, i.e. $\neg(\vdash x) = \vdash(\neg x) = x$ for any $x \in \mathcal{X}$.

Given a cyclic permutation $\neg$ on $\mathcal{X}$, for any vector $x = [x_1, \ldots, x_h]^\top \in \mathcal{X}^h$, define $\neg_i$ as a cyclic permutation of the i-th location $\neg_i x := [x_1, \ldots, x_{i-1}, \neg x_i, x_{i+1}, \ldots, x_h]^\top$. For any function $f : \mathcal{X}^h \to \mathbb{R}$, define the partial difference operator as $\Delta_{x_i} f(x) := f(x) - f(\neg_i x), \quad i = 1, \ldots, d$, and write $\Delta_x f(x) = [\Delta_{x_1} f(x), \ldots, \Delta_{x_h} f(x)]^\top \in \mathbb{R}^h$. The difference score function is defined as $s_p(x) := \frac{\Delta_x p(x)}{p(x)}$. As such, we assume that p has full support on $\mathcal{X}^h$, that is, $p(x) > 0$ for all $x \in \mathcal{X}^h$.t Similarly, for the inverse cyclic permutation $\vdash$, define the associated partial difference operator as: $\Delta^*_{x_i} f(x) := f(x) - f(\vdash_i x), \quad i = 1, \ldots, h$, and write $\Delta^*_x f(x) = [\Delta^*_{x_1} f(x), \ldots, \Delta^*_{x_h} f(x)]^\top \in \mathbb{R}^d$. The superscript $*$ arises because Δ^*_x is the adjoint operator of Δ_x with respect to the standard ℓ_2 inner product over $\mathcal{X}^h$.

B Derivations of closed-form Kernel Mean Embeddings

In this section, we provide detailed derivations of all the closed-form kernel mean embeddings in Table 1.

Poisson distribution and Brownian motion kernel Consider a Poisson distribution whose probability mass function $p(x) = \frac{\eta^x e^{-\eta}}{x!}$ for any $x \in \mathbb{N}_0$, and the shifted Brownian motion kernel $k_c(x, y) = c + x \wedge y$ with $c > 0$. The kernel mean embedding can be computed as: for any $y \in \mathbb{N}_0$,

$$\begin{aligned}
\mu_{\mathbb{P}}(y) &= \sum_{x=0}^{\infty} (c + \min(x, y)) \frac{e^{-\eta}\eta^x}{x!} \\
&= c + \sum_{x=0}^{y} x \cdot \frac{e^{-\eta}\eta^x}{x!} + \sum_{x=y+1}^{\infty} y \cdot \frac{e^{-\eta}\eta^x}{x!} \\
&= c + \sum_{x=0}^{y} x \cdot \frac{e^{-\eta}\eta^x}{x!} + y \cdot \sum_{x=y+1}^{\infty} \frac{e^{-\eta}\eta^x}{x!} \\
&= c + \sum_{x=0}^{y} x \cdot \frac{e^{-\eta}\eta^x}{x!} + y \cdot P(X > y) \\
&= c + \sum_{x=0}^{y} x \cdot \frac{e^{-\eta}\eta^x}{x!} + y \cdot (1 - Q(y+1, \eta)),
\end{aligned}$$

where Q is the regularized gamma function and it admits efficient computation in python `SciPy` package.

The initial error can be computed as:

$$\begin{aligned}
\mathbb{E}_{X,X' \sim \mathbb{P}}[k_c(X, X')] &= c + \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \min(m, n) \cdot P(X = m) \cdot P(X' = n) \\
&= c + \sum_{r=0}^{\infty} P(\min(X, X') > r) \\
&= c + \sum_{r=0}^{\infty} P(X > r, X' > r) \\
&= c + \sum_{r=0}^{\infty} P(X > r)^2 = c + \sum_{r=0}^{\infty} (1 - Q(r+1, \eta))^2,
\end{aligned}$$

which would involve an infinite sum over regularized gamma function and does not admit a closed-form expression. In the experiments, we truncate the above infinite sum with first 100 terms.

Negative binomial distribution and Brownian motion kernel Consider a negative binomial distribution $\mathrm{NB}(\tau, q)$ whose probability mass function $P(X = x) = \binom{x+\tau-1}{x}(1-q)^x q^\tau$ for any $x \in \mathbb{N}_0$, and the shifted Brownian motion kernel $k_c(x, y) = c + x \wedge y$ with $c > 0$. The kernel mean embedding can be computed as: for any $y \in \mathbb{N}_0$,

$$\mu_{\mathbb{P}}(y) = c + \sum_{x=0}^{\infty} \min(x, y) \cdot P(X = x) = c + \sum_{k=1}^{y} P(X \geq k) = c + y - \sum_{k=0}^{y-1} I_q(\tau, k+1).$$

Here, I_q denotes the regularized incomplete Beta function, which can be computed efficiently with the Python package `SciPy`. The initial error can be computed as:

$$\mathbb{E}_{X,X' \sim \mathbb{P}}[k_c(X, X')] = c + \sum_{m=0}^{\infty} \left(1 - I_q(\tau, m+1)\right)^2.$$

which unfortunately is another infinite sum that does not admit a closed-form expression. In the experiments, we truncate the above infinite sum with the first 100 terms.

Logarithmic distribution and Brownian motion kernel Consider a logarithmic distribution $\mathrm{Log}(q)$ whose probability mass function is given by $P(X = x) = -\frac{1}{\ln(1-q)} \cdot \frac{q^x}{x}$ for any $x \in \mathbb{N}_+$, and the shifted Brownian motion kernel $k_c(x, y) = c + \min(x, y)$ with $c > 0$. The kernel mean embedding can be computed as: for any $y \in \mathbb{N}$,

$$\begin{aligned}
\mu_{\mathbb{P}}(y) &= c + \sum_{x=1}^{\infty} \min(x, y) \cdot P(X = x) = c + \sum_{x=1}^{y} x \cdot P(X = x) + y \sum_{x=y+1}^{\infty} P(X = x) \\
&= c - \frac{1}{\ln(1-q)} \left(\sum_{x=1}^{y} q^x + y \sum_{x=y+1}^{\infty} \frac{q^x}{x} \right) \\
&= c + y + \frac{1}{\ln(1-q)} \sum_{n=1}^{y-1} \frac{(y-n)q^n}{n}.
\end{aligned}$$

Unfortunately, the initial error $\mathbb{E}_{X,X' \sim \mathbb{P}}[k_c(X, X')]$ does not admit a closed-form expression.

Poisson distribution and polynomial kernel Consider a Poisson distribution whose probability mass function $p(x) = \frac{\eta^x e^{-\eta}}{x!}$ for any $x \in \mathbb{N}_0$, and a polynomial kernel $k(x, y) = (xy + 1)^r$. The kernel mean embedding can be computed as: for any $y \in \mathbb{N}_0$,

$$\mu_{\mathbb{P}}(y) = \sum_{n=0}^{r} \binom{r}{n} y^n \mathbb{E}[X^n] = \sum_{n=0}^{r} \binom{r}{n} y^n T_n(\eta).$$

where $T_n(\eta)$ is the Touchard polynomial (Touchard, 1939) and it admits an efficient computation via dynamical programming (Comtet, 2012). The initial error can be computed as $\mathbb{E}_{X,X' \sim \mathbb{P}}[k(X, X')] = \sum_{n=0}^{r} \binom{r}{n} T_n(\eta)^2$.

Negative binomial distribution and polynomial kernel Consider a negative binomial distribution $\mathrm{NB}(\tau, q)$ whose probability mass function $P(X = x) = \binom{x+\tau-1}{x}(1-q)^x q^\tau$ for any $x \in \mathbb{N}_0$, and a polynomial kernel $k(x, y) = (xy + 1)^r$. The kernel mean embedding can be computed as: for any $y \in \mathbb{N}_0$,

$$\mu_{\mathbb{P}}(y) = \sum_{n=0}^{r} \binom{r}{n} y^n \mathbb{E}[X^n] = \sum_{n=0}^{r} \binom{r}{n} y^n M_n(\tau, q),$$

where $M_n(\tau, q) = \sum_{j=0}^{n} S(n, j) j! \binom{\tau+j-1}{j} (\frac{1-q}{q})^j$ with $S(n, j)$ being the Stirling numbers (Comtet, 2012). The initial error can be computed as $\mathbb{E}_{X,X' \sim \mathbb{P}}[k(X, X')] = \sum_{n=0}^{r} \binom{r}{n} M_n(\tau, q)^2$.

Skellam distribution and polynomial kernel Consider a Skellam distribution whose probability mass function $P(X = x) = e^{-(\lambda_1+\lambda_2)} (\frac{\lambda_1}{\lambda_2})^{x/2} I_{|x|}(2\sqrt{\lambda_1 \lambda_2})$ for any $x \in \mathbb{Z}$, and a polynomial kernel $k(x, y) = (xy + 1)^r$. The kernel mean embedding can be computed as: for any $y \in \mathbb{Z}$,

$$\mu_{\mathbb{P}}(y) = \sum_{n=0}^{r} \binom{r}{n} y^n \mathbb{E}[X^n] = \sum_{n=0}^{r} \binom{r}{n} y^n B_n(\lambda_1 - \lambda_2, \ldots, \lambda_1 + (-1)^n \lambda_2),$$

where B_n are the complete exponential Bell polynomials (Bell, 1934). The initial error can be computed as $\mathbb{E}_{X,X' \sim \mathbb{P}}[k(X, X')] = \sum_{n=0}^{r} \binom{r}{n} B_n(\lambda_1 - \lambda_2, \ldots, \lambda_1 + (-1)^n \lambda_2)^2$.

Uniform distribution over $\{-1,+1\}^d$ and exponential Hamming distance kernel Consider a uniform distribution over all possible states $\{-1,+1\}^d$ of an Ising model, and an exponential Hamming distance kernel $k(x,y)=\exp(-\lambda d_H(x,y))$ with a lengthscale $\lambda>0$. In this particular domain of $\{-1,+1\}^d$, the exponential Hamming distance kernel can be equivalently written as $k(x,y)=\exp(-\frac{\lambda}{2}(d-\sum_{i=1}^d x_iy_i))$. As a result, the kernel mean embedding can be computed as follows:

$$\begin{aligned}
\mu_{\mathbb{P}}(y) &= \frac{1}{2^d}\sum_{x\in\{-1,+1\}^d}\left[\exp\left(-\frac{\lambda d}{2}\right)\cdot\exp\left(\frac{\lambda}{2}\sum_{i=1}^d x_iy_i\right)\right]\\
&= \frac{1}{2^d}\cdot\exp\left(-\frac{\lambda d}{2}\right)\sum_{x\in\{-1,+1\}^d}\prod_{j=1}^d\exp\left(\frac{\lambda}{2}x_jy_j\right)\\
&= \frac{1}{2^d}\cdot\exp\left(-\frac{\lambda d}{2}\right)\prod_{j=1}^d\sum_{x_j\in\{-1,1\}}\exp\left(\frac{\lambda}{2}x_jy_j\right)\\
&= \exp\left(-\frac{\lambda d}{2}\right)\prod_{j=1}^d\cosh\left(\frac{\lambda y_j}{2}\right)\\
&= \exp\left(-\frac{\lambda d}{2}\right)\left[\cosh\left(\frac{\lambda}{2}\right)\right]^d.
\end{aligned}$$

Since the kernel mean embedding is a constant function, the initial error is the same as the kernel mean embedding, i.e. $\mathbb{E}_{X,X'\sim\mathbb{P}}[k(X,X')]=\exp(-\frac{\lambda d}{2})[\cosh(\frac{\lambda}{2})]^d$.

Uniform distribution over $\{0,+1\}^d$ and Tanimoto kernel Consider a uniform distribution over all possible states $\{0,+1\}^d$ of an Ising model. We use the Tanimoto kernel $k(x,y)=\frac{x^\top y}{\|x\|^2+\|y\|^2-x^\top y}$ whenever $(x,y)\neq(0,0)$ and adopt the convention $k(0,0)=1$. The exceptional pair contributes $2^{-d}1\{y=0\}$ to the kernel mean embedding. Hence,

$$\mu_{\mathbb{P}}(y)=\frac{1}{2^d}1\{y=0\}+\frac{1}{2^d}\sum_{s=0}^d\sum_{\substack{a=\max(0,s-d+\|y\|^2),\dots,\min(s,\|y\|^2)\\ s+\|y\|^2-a>0}}\frac{a}{s+\|y\|^2-a}\binom{\|y\|^2}{a}\binom{d-\|y\|^2}{s-a}.$$

The same convention contributes 2^{-2d} to the initial error. Therefore,

$$\mathbb{E}_{X,X'\sim\mathbb{P}}[k(X,X')]=\frac{1}{2^{2d}}+\frac{1}{2^{2d}}\sum_{t=0}^d\sum_{s=0}^d\sum_{\substack{a=\max(0,s-d+t),\dots,\min(s,t)\\ s+t-a>0}}\frac{a}{s+t-a}\binom{d}{t}\binom{t}{a}\binom{d-t}{s-a}.$$

The displayed sums contain $\mathcal{O}(d^2)$ terms for the kernel mean embedding and $\mathcal{O}(d^3)$ terms for the initial error. These are substantially cheaper than direct summation over $\mathcal{O}(2^d)$ states for the kernel mean embedding and $\mathcal{O}(2^{2d})$ pairs of states for the initial error.

Uniform distribution over $\{0,1,2\}^d$ and exponential Hamming distance kernel Consider a uniform distribution over all possible states $\{0,1,2\}^d$ of a Potts model, and an exponential Hamming distance kernel $k(x,y)=\exp(-\lambda d_H(x,y))$ with a lengthscale $\lambda>0$. The kernel mean embedding can be computed as follows:

$$\mu_{\mathbb{P}}(y)=\frac{1}{3^d}\sum_x\exp(-\lambda d_H(x,y))=\frac{1}{3^d}\sum_{s=0}^d\binom{d}{s}\cdot 2^s\cdot e^{-\lambda s}=\frac{1}{3^d}\sum_{s=0}^d\binom{d}{s}(2e^{-\lambda})^s=\left(\frac{1+2e^{-\lambda}}{3}\right)^d.$$

Since the kernel mean embedding is a constant over y, the initial error is the same as the kernel mean embedding, i.e. $\mathbb{E}_{X,X'\sim\mathbb{P}}[k(X,X')]=(\frac{1+2e^{-\lambda}}{3})^d$. The same derivations can be extended to uniform distribution over $\{0,1,\dots,K\}^d$ for any positive integer K with the corresponding kernel mean embedding $\mu_{\mathbb{P}}(y)=(\frac{1+Ke^{-\lambda}}{K+1})^d$ and initial error $\mathbb{E}_{X,X'\sim\mathbb{P}}[k(X,X')]=(\frac{1+Ke^{-\lambda}}{K+1})^d$.

C An Example on the Tightness of the Upper Bound in Theorem 1

The worst-case optimality results cited in the proof of Theorem 1 state that

$$\hat{\sigma}=\sup_{\|f\|_{\mathcal{H}_k}\leq 1}\left|\sum_{x\in\mathcal{X}}f(x)p(x)-\sum_{i=1}^N w_if(x_i)\right|, \tag{C.1}$$

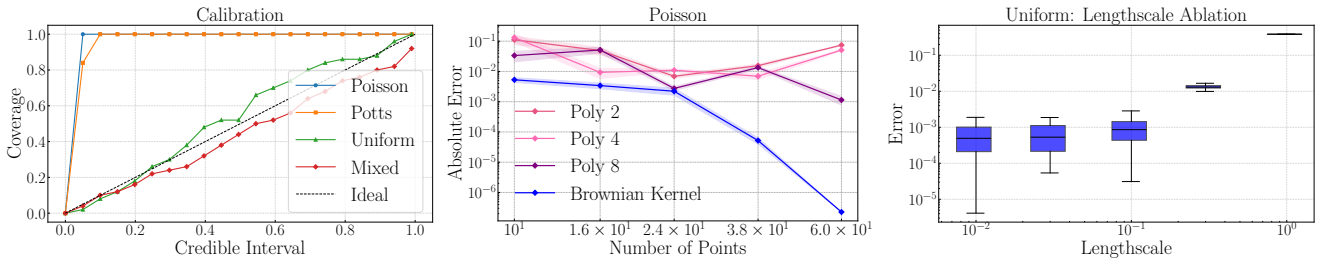

Figure 5: **Left**: Calibration of the posterior variance of BayesSum in synthetic settings (a)—(d). **Middle**: Ablation study on different kernels in synthetic setting (a): polynomial kernels of degree 2, 5, and 8 and Brownian motion kernel. **Right**: Ablation study of kernel lengthscales of the exponential Hamming kernel in synthetic setting (b).

where w_i are the BayesSum weights (see Remark 1). Suppose that $\mathcal{X} = \mathbb{N}$ and k is any Matérn kernel. The RKHS of k on $\mathbb{R}$ is a Sobolev space Kanagawa et al. (2018). Being compactly supported and infinitely differentiable, the bump function f given by $f(x) = \exp(-1/(1-x^2))$ for $|x| < 1$ and $f(x) = 0$ otherwise is in every Sobolev space. Because Matérn kernels are stationary, the norm of the translation $f(\cdot - a)$ does not depend on a. Since $f(\cdot - a)|_{\mathbb{N}} \in \mathcal{H}_k$ for every a, by selecting $a = N + 1$ and using (C.1) we obtain

$$\hat{\sigma} \geq \frac{1}{\|f|_{\mathbb{N}}\|_{\mathcal{H}_k}} f(0) p(N+1) = \frac{\exp(-1)}{\|f|_{\mathbb{N}}\|_{\mathcal{H}_k}} p(N+1),$$

where we used the fact that $f(i - (N+1)) = 0$ if $i \neq N+1$. In combination with Theorem 1 we thus have

$$C_1 p(N+1) \leq \hat{\sigma} \leq C_2 \sum_{i>N} p(i)$$

for all $N \in \mathbb{N}$ and some constants C_1 and C_2. If $p(i) = \Theta(i^{-\alpha})$ for $\alpha > 1$, the lower bound is of order $N^{-\alpha}$ and the upper bound of order $N^{-\alpha+1}$. If $p(i) = \Theta(e^{-ci})$ for $c > 0$, both the lower and upper bound are of order e^{-cN}.

D Additional experiments and details

More details on the synthetic settings: Here, we provide details on the baseline methods: Monte Carlo (MC), Russian Roulette (RR), importance sampling (IS), and stratified sampling (SS).

(a) The ground truth is computed by summation over the first 200 terms $n = 0, ..., 200$. For IS we use a negative binomial distribution $\mathrm{NB}(\tau, p)$ as the proposal distribution $\mathbb{Q}$. We pick $\tau = 5$ and the success probability $p = \eta/(\tau+\eta)$ such that the mean of the proposal distribution matches the mean of the distribution $\mathbb{P} = \mathrm{Poisson}(\eta)$. For RR, we draw a random subset $\mathcal{X}' \subset \mathcal{X}$ according to a geometric distribution with parameter $\rho_\star = 0.95$. Hence, the inclusion probability for the element at position $x' \in \mathcal{X}$ is given by ρ_*^j. For SS, the set $\mathcal{X} = \mathbb{N}$ is partitioned into two disjoint regions, $\mathcal{R}_1 = \{n \in \mathbb{N} : n < c\}$ and $\mathcal{R}_2 = \{n \in \mathbb{N} : n \geq c\}$. The cutoff parameter $c = 100$. An equal number of $N/2$ samples are drawn independently via inverse cumulative distribution function of each region. For Active BayesSum, a candidate subset $\mathcal{X}_{\text{sub}} \subset \mathcal{X}$ is chosen as $\mathcal{X}_{\text{sub}} = \{0, 1, ..., 200\}$ over which the utility function is evaluated and maximized.

(b) The ground truth is computed by exact summation over the full index set. For IS, we uses a fully factorized proposal distribution $\mathbb{Q}$ with $q(x) = \prod_{i=1}^d q_i(x_i)$, where each $q_i(x_i) \propto \exp(-\beta\, h_i^\top x_i)$ is a categorical distribution. For RR, we first denote $\mathcal{I} = \{0, 1, \ldots, 3^L\}$ as the index set of all possible sequences in $\mathcal{X} = \{0, 1, 2\}^L$. Hence, each index $j \in \mathcal{I}$ is in bijective correspondence with the sequence in $\mathcal{X}$. Next, we draw the random index subset $\mathcal{I}' \subset \mathcal{I}$ according to a geometric distribution with parameter $\rho_* = 0.95$. Therefore, for each index $j \in \mathcal{I}$, the inclusion probability is given explicitly by ρ_*^j. For SS, we partition the index set $\mathcal{I}$ into four index subsets $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$ of equal sizes. Then we draw $\lfloor N/4 \rfloor$ number of samples uniformly and independently from each subset, which gives N samples in total. For Active BayesSum, a candidate subset $\mathcal{X}_{\text{sub}} \subset \mathcal{X}$ with 2000 points are chosen randomly out of $\mathcal{X}$, over which the utility function is evaluated and maximized.

(c) The ground truth is computed by exact enumeration over all configurations. Samples from the un-normalized Potts model are drawn using either Gibbs sampling or the Metropolis–Hastings algorithm (Gelman et al., 1995), with a burn-in of 10 iterations and thinning by a factor of 5. Both MC and BayesSum are computed with the same set of samples in order for consistent comparison.

(d) The ground truth admits a closed-form expression:

$$I = \frac{1}{17^2} \sum_{h_1 \in H} \sum_{h_2 \in H} \left[-20\, \frac{1 - e^{-0.2\,\alpha L}}{0.2\,\alpha L} - \left(I_0(1) + \frac{2}{\beta L} \sum_{n=1}^{\infty} \frac{I_n(1)}{n} \sin(n\beta L) \right) + 20 + e \right],$$

$$\alpha = 1 + 0.1\,(h_1 + h_2), \qquad \beta = 2\pi\big(1 + 0.05\,(h_1^2 + h_2^2)\big)\,, \qquad H = \{\,-1 + 0.125\,j \;:\; j = 0, 1, \ldots, 16\,\}$$

and $I_n(x)$ denotes the modified Bessel function of the first kind of order.

Moreover, in this setting, we use the following kernel

$$k\big((x,y),(x',y')\big) = k_{\mathcal{X}}(x,x') + k_{\mathcal{Y}}(y,y') + k_{\mathcal{X}}(x,x')\,k_{\mathcal{Y}}(y,y'),$$

where $k_{\mathcal{X}}$ depends only on the continuous coordinates and $k_{\mathcal{Y}}$ depends only on the categorical coordinates. In particular, $k_{\mathcal{X}}(x,x')$ is the Gaussian kernel with lengthscale $\ell = 0.8$; and $k_{\mathcal{Y}}(y,y')$ is the exponential–Hamming kernel with lengthscale $\lambda = 1.0$. Under the joint distribution $\mathbb{P} = \mathbb{P}_X \times \mathbb{P}_Y$, The kernel mean embedding at (x',y') equals $\mu_{\mathbb{P}}(x',y') = \mu_{\mathbb{P}_X}(x') + \mu_{\mathbb{P}_Y}(y') + \mu_{\mathbb{P}_X}(x') \cdot \mu_{\mathbb{P}_Y}(y')$. Here, both kernel mean embeddings $\mu_{\mathbb{P}_X}$ and $\mu_{\mathbb{P}_Y}$ admit closed-form expressions from Briol et al. (2025) and our Table 1.

Additional experimental results: In Figure 5 (**Left**), we report the calibration of the BayesSum posterior variance. Here, coverage refers to the proportion of times that a confidence interval contains the true value across repeated experiments. The black diagonal line denotes perfect calibration, while curves above or below this line indicate underconfidence or overconfidence, respectively. We observe that the posterior variances are well-calibrated for the uniform distribution over $\{0,1,2\}^d$ in (b) and the distribution over mixed domain in (d), whereas the posterior variance is underconfident for the Poisson distribution in (a) and the Potts distribution in (c). This behavior arises because the shifted Brownian motion kernel $k_c(x,y) = A(c + \min(x,y))$ in (a) and the Stein kernel in (c) are both unbounded, which leads to inflated posterior variances even when the kernel amplitude A is carefully selected.

In Figure 5 (**Middle**), we run an ablation study on the choice of kernels in case (a) of the synthetic setting using non-repetitive samples. The Brownian motion kernel achieves lower error compared to polynomial kernels since the integrand is not an polynomial function. Meanwhile, the performance of polynomial kernels improves as the degree of the polynomial increases from 2 to 8. This is expected since the higher-degree polynomial kernels can better approximate the integrand. In Figure 5 (**Right**), we run an ablation study on the choice of lengthscales in the exponential Hamming distance in case (b) of the synthetic setting. Performance is similar for lengthscales $0.01, 0.03, 0.1$ whereas larger lengthscales $0.3, 1.0$ achieve worse performances. This highlights the importance of lengthscale selection in BayesSum. In the experiments in the main text, we optimize the kernel lengthscales via maximizing the log marginal likelihood, which gives a lengthscale of 0.01, falling within the optimal range as shown in the ablation study as desired.

In Figure 6, we compare the absolute integration error of BayesSum under two sampling schemes: (i) points drawn i.i.d. from a distribution with replacement; and (ii) points drawn i.i.d. from the same distribution without replacement. This distinction is unique to discrete domains, since repeated samples occur with probability zero in continuous spaces. As illustrated in Figure 6(**Left**) and (**Right**), using unique (non-duplicated) points consistently yields lower error. This is expected: repeated evaluation points provide no additional information to the BayesSum estimator and therefore do not improve its accuracy. In Figure 6(**Middle**), the two curves are nearly identical. This is because the state space has cardinality 3^{15} ; when drawing samples i.i.d. from the uniform distribution over $\mathcal{X} = \{0,1,2\}^{15}$, duplicates almost never occur in practice. Consequently, the i.i.d. and unique-sample schemes achieve indistinguishable performance.

More details on the unnormalized models: Here, we provide additional details for the two unnormalized models: (a) the Conway–Maxwell–Poisson (CMP) model for count data, and (b) the Potts model for sequence data.

(a) The initial parameters are set to $(\theta_1, \theta_2) = (0.5, 1.2)$. BayesSum employs the shifted Brownian motion kernel $k_c(x,y) = c + \min(x,y)$ with $c = 1$ and amplitude 1. In Figure 3, the optimal parameter location is obtained by computing the log-likelihood over the entire grid $[0.0, 3.0] \times (0.0, 0.8]$ and selecting the point that achieves the maximum value.

(b) The initial parameters $\boldsymbol{\theta} = (\mathbf{h}, \mathbf{J})$ are drawn independently from a normal distribution $\mathcal{N}(0,\, 0.01^2)$ at each coordinate. BayesSum employs an exponential Hamming kernel $k(x,y) = \exp\big(-\lambda\, H(x,y)\big)$, with lengthscale $\lambda = 0.1104$ and amplitude 1. The discrete KSD values in Figure 4 are based on the discrete difference Stein operator of Yang et al. (2018): the cyclic permutation on the single-site state space $\{0,1,2\}$ is defined as $s \mapsto (s+1) \bmod 3$ (i.e. $0 \to 1 \to 2 \to 0$), and is applied to the one-hot-encoded vectors; its inverse is implemented by a cyclic shift in the opposite direction. The discrete KSD values are computed between the trained Potts model $p(x;\boldsymbol{\theta})$ and a fixed set of 2000 samples drawn from the true data distribution. For KSD computation, we use the same exponential Hamming kernel as in BayesSum. In terms of computational costs, reported wall-times were calculated excluding hyperparameter-selection cost for both BayesSum and MC. Lastly, experiments with higher dimensions, $L = 20$ and $L = 30$, were implemented. Figure 7 shows that overall, BayesSum continues to outperform MC in regimes with larger L. However, marginal advantages slowly decrease as L increases, which is expected with the curse of dimensionality that adversely affects the kernel-based methods.

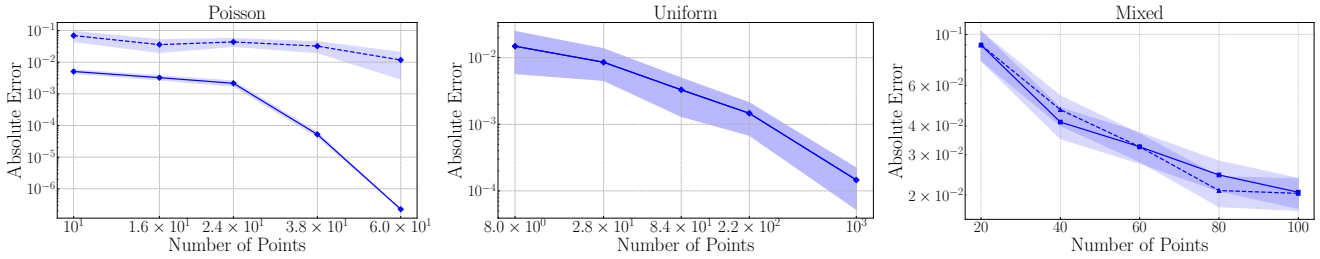

Figure 6: Comparison of BayesSum with i.i.d. samples with replacement (solid line) and without replacement (dashed line) in the setting of the Poisson distribution (**Left**), Uniform distribution (**Middle**) and the mixed distributions over continuous and discrete domain (**Right**).

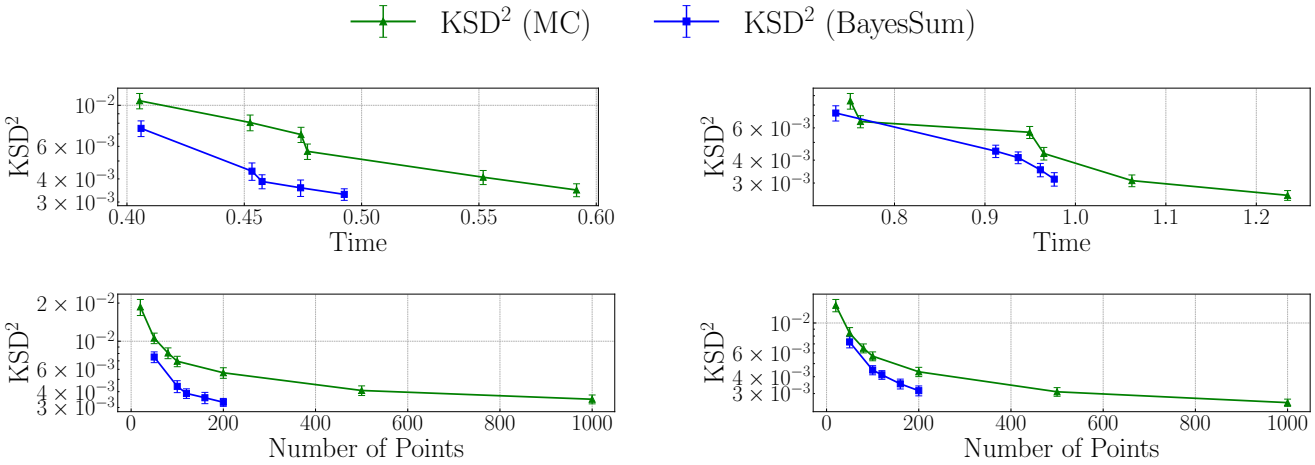

Figure 7: Comparison of a large Potts model trained via maximum log likelihood, with the normalization constant estimated by BayesSum and Monte Carlo, under the same computational time (top) and sample size N (bottom). Error bars represent the standard error repeated over 30 random seeds. **Left:** $L = 20$. **Right:** $L = 30$.