

Variational Inference with Vine Copulas: An efficient Approach for Bayesian Computer Model Calibration

Vojtech Kejzlar¹ and Tapabrata Maiti²

¹Department of Mathematics and Statistics, Skidmore College

²Department of Statistics and Probability, Michigan State University

Abstract

With the advancements of computer architectures, the use of computational models proliferates to solve complex problems in many scientific applications such as nuclear physics and climate research. However, the potential of such models is often hindered because they tend to be computationally expensive and consequently ill-fitting for uncertainty quantification. Furthermore, they are usually not calibrated with real-time observations. We develop a computationally efficient algorithm based on variational Bayes inference (VBI) for calibration of computer models with Gaussian processes. Unfortunately, the speed and scalability of VBI diminishes when applied to the calibration framework with dependent data. To preserve the efficiency of VBI, we adopt a pairwise decomposition of the data likelihood using vine copulas that separate the information on dependence structure in data from their marginal distributions. We provide both theoretical and empirical evidence for the computational scalability of our methodology and describe all the necessary details for an efficient implementation of the proposed algorithm. We also demonstrate the opportunities given by our method for practitioners on a real data example through calibration of the Liquid Drop Model of nuclear binding energies.

Keywords— Bayesian inference; Computer experiments; Gaussian process; Nuclear physics; Vine copula; Uncertainty quantification; Prediction

1 Introduction

The ever-growing access to high performance computing in scientific communities has enabled development of complex computer models in fields such as nuclear physics, climatology, and engineering that produce massive amounts of data. These models need real-time calibration with quantified uncertainties. Bayesian methodology combined with Gaussian process modeling has been heavily utilized for calibration of computer models due to its natural way to account for various sources of uncertainty; see [Higdon et al. \(2015\)](#), and [King et al. \(2019\)](#) for examples in nuclear physics, [Sexton et al. \(2012\)](#) and [Pollard et al. \(2016\)](#) for examples in climatology, and [Williams et al. \(2006\)](#), [Plumlee et al. \(2016\)](#) and [Zhang et al. \(2019\)](#) for applications in engineering.

The framework for Bayesian calibration of computer models was developed by [Kennedy and O’Hagan \(2001\)](#) with extensions provided by [Higdon et al. \(2005, 2008\)](#); [Bayarri et al. \(2007\)](#); [Plumlee \(2017, 2019\)](#), and [Gu and Wang \(2018\)](#), to name a few. Despite its popularity, however, Bayesian calibration becomes infeasible in big-data scenarios with complex and many-parameter models because it relies on Markov chain Monte Carlo (MCMC) algorithms to approximate posterior densities.

This text presents a scalable and statistically principled approach to Bayesian calibration of computer models. We offer an alternative approximation to posterior densities using variational Bayesian inference (VBI), which originated as a machine learning algorithm that approximates a target density through optimization. Statisticians and computer scientists (starting with [Peterson and Anderson \(1987\)](#); [Jordan et al. \(1999\)](#)) have been widely using variational techniques because they tend to be faster and easier to scale to massive datasets. Moreover, the recently published frequentist consistency of variational Bayes by [Wang and Blei \(2018\)](#) established VBI as a theoretically valid procedure. The scalability of VBI in modern applications hinges on efficiency of stochastic optimization in scenarios with independent data points. This efficiency, however, diminishes in the case of Bayesian calibration of computer models due to dependence structure in data

(Robbins and Monro, 1951; Hoffman et al., 2013). To maintain the speed and scalability of VBI, we adopt a pairwise decomposition of data likelihood using vine copulas that separate the information on dependence structure in data from their marginal distributions (Cooke and Kurowicka, 2006). Our specific contributions are as follows:

1. We propose a novel version of the black-box variational inference (Ranganath et al., 2014) for Bayesian calibration of computer models that preserves the efficiency of stochastic optimization in scenario with dependent data. Python code with our algorithm is available at https://github.com/kejzlarv/VBI_Calibration.
2. We incorporate Rao-Blackwellization, control variates, and importance sampling to reduce the variance of noisy gradient estimates involved in our algorithm.
3. We provide both theoretical and empirical evidence for scalability of our methodology and establish its superiority over the Metropolis-Hastings algorithm and the No-U-Turn sampler both in terms of time efficiency and memory requirements.
4. Finally, we demonstrate the opportunities in uncertainty quantification given by the proposed algorithm on a real-world example in the field of nuclear physics.

1.1 Outline of this paper

In Section 2, we describe the framework for Bayesian calibration of computer models and give an overview of VBI. In Section 3, we derive our proposed VBI approach to perform inexpensive and scalable calibration. We establish statistical validity of the method and provide theoretical justification for its scalability. Subsequently, in Section 4, we discuss the implementation details with focus on strategies to reduce the variance of the gradient. Section 5 presents a simulation study comparing our approach with the state-of-the-art methods to approximate posterior distribution and illustrates our method on a real-data application.

2 Background and Theoretical Framework

Formally, let $\mathbf{y} = (y_1, \dots, y_n)$ be observations of a physical process $\zeta(\mathbf{t}_i)$ depending on a known set of inputs $\mathbf{t}_i \in \Omega \subset \mathbb{R}^p$. Assume that y_i follows

$$y_i = \zeta(\mathbf{t}_i) + \sigma\epsilon_i, \tag{1}$$

where σ represent the scale of observation error $\epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, 1)$. As a mathematical description of ζ , we consider a computer model f_m defined as the mapping $(\mathbf{t}, \boldsymbol{\theta}) \mapsto f_m(\mathbf{t}, \boldsymbol{\theta})$ which depends on an additional set of inputs $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^{p'}$ that we call calibration parameters. These are fixed but unknown quantities representing fundamental properties of the physical process that cannot be directly measured or controlled in an experiment. Model calibration corresponds to determining the unknown and hypothetical *true* value of the parameter $\boldsymbol{\theta}$, at which the physical process $\zeta(\mathbf{t})$ would satisfy $\zeta(\mathbf{t}) = f(\mathbf{t}, \boldsymbol{\theta}) + \delta(\mathbf{t})$; $\delta(\mathbf{t})$ is the systematic discrepancy of the model whose form is generally unknown. We assume a single value of calibration parameter $\boldsymbol{\theta}$ to be common among all the observations y_i and all the future instances of the physical process.

Overall, we can write the complete statistical model as

$$y_i = f(\mathbf{t}_i, \boldsymbol{\theta}) + \delta(\mathbf{t}_i) + \epsilon_i. \tag{2}$$

It is often the case that the evaluation of computer model f_m is too expensive in terms of both time and space (memory). Common practice is to reduce the number of necessary computer model evaluations by considering a Gaussian process (GP) prior model:

$$f_m(\mathbf{t}, \boldsymbol{\theta}) \sim \mathcal{GP}(m_f(\mathbf{t}, \boldsymbol{\theta}), k_f((\mathbf{t}, \boldsymbol{\theta}), (\mathbf{t}', \boldsymbol{\theta}'))).$$

In this setup, the data also include set of model evaluations $\mathbf{z} = (z_1, \dots, z_s)$ over a grid $\{(\tilde{\mathbf{t}}_1, \tilde{\boldsymbol{\theta}}_1), \dots, (\tilde{\mathbf{t}}_s, \tilde{\boldsymbol{\theta}}_s)\}$. These are usually selected sequentially using some space-filling design such us uniform or Latin hypercube design (Morris and Mitchell, 1995), which is a design that has a good coverage of the space with evenly distributed points in each one-dimensional projection. The discrepancy function $\delta(\mathbf{t})$, while intrinsically deterministic, is also modeled by a GP. The complete dataset \mathbf{d} consists of n

observations y_i from the physical process ζ and s evaluations z_j of the computer model f_m , i.e. $\mathbf{d} = (d_1, \dots, d_{n+s}) := (\mathbf{y}, \mathbf{z})$, and follows the multivariate normal distribution

$$\mathbf{d}|\phi \sim \mathcal{N}(M(\phi), K(\phi)), \quad (3)$$

where $\phi = (\boldsymbol{\theta}, \boldsymbol{\gamma}, \sigma)$ is the set of all unknown parameters with $\boldsymbol{\gamma}$ denoting the set of hyperparameters of the GPs' mean and covariance functions. $M(\phi)$ is the mean vector and $K(\phi)$ is the covariance matrix given by the GPs' specifications.

Under this framework, the Bayesian predictions of new values \mathbf{y}^* of a physical process ζ are given by the posterior predictive distribution $p(\mathbf{y}^*|\mathbf{d})$, namely

$$p(\mathbf{y}^*|\mathbf{d}) = \int p(\mathbf{y}^*|\mathbf{d}, \phi)p(\phi|\mathbf{d}) \mathrm{d}\phi. \quad (4)$$

The conditional density $p(\mathbf{y}^*|\mathbf{d}, \phi)$ is a multivariate normal density given by the statistical model (2) and the specification of GPs. The posterior distribution of the unknown parameters $p(\phi|\mathbf{d})$ is given by the Bayes' theorem. The term ‘‘calibration’’ in the Bayesian paradigm includes both an estimation of ϕ and a full evaluation of uncertainty for every parameter under a prior uncertainty expressed by $p(\phi)$. It is also worth noting that the posterior predictive density is rarely computed directly from (4). Instead, we first generate samples $\phi^{(1)}, \dots, \phi^{(M)}$ from $p(\phi|\mathbf{d})$ and then obtain samples $\mathbf{y}^{*(1)}, \dots, \mathbf{y}^{*(M)}$ so that $\mathbf{y}^{*(i)} \sim p(\mathbf{y}^*|\mathbf{d}, \phi^{(i)})$, $i = 1, \dots, M$. The posterior predictive density is approximated using the empirical density of $\mathbf{y}^{*(1)}, \dots, \mathbf{y}^{*(M)}$.

As a consequence of this simple two-step algorithm, we are interested in effective sampling (approximation) from the posterior distribution $p(\phi|\mathbf{d})$. *This becomes quickly infeasible with increasing size of datasets, number of parameters, and model complexity.* Traditional MCMC methods that approximate $p(\phi|\mathbf{d})$ —such as the Metropolis-Hastings (MH) algorithm (Chib and Greenberg, 1995) or more advanced ones including Hamiltonian Monte Carlo or the No-U-Turn Sampler (NUTS) (Homan and Gelman, 2014)—typically fail because of the computational costs associated with the evaluation of $p(\mathbf{d}|\phi)$. The standard approaches to scalable Bayesian inference are in general not applicable here because of the highly correlated structure of $K(\phi)$ or the nature of calibration itself. Indeed, parallelization of MCMC (Neiswanger et al., 2014) works in the case of and indepen-

dent latest \mathbf{d} , and GP approximation methods are developed in the context of regression problems (Quiñero-Candela and Rasmussen, 2005; Titsias, 2009; Bauer et al., 2016). We emphasize that our context is much more complex and that our approach is not focused on developing parallel computing, but rather exploiting probabilistic theory of approximation to reduce the computational cost.

2.1 Variational Bayes Inference (VBI)

VBI is an optimization based method that approximates $p(\phi|\mathbf{d})$ by a family of distributions $q(\phi|\lambda)$ over latent variables with its own variational parameter λ . Many commonly used families exist with the simplest mean-field family assuming independence of all the components in ϕ ; see Wainwright and Jordan (2008); Hoffman and Blei (2015); Ranganath et al. (2016); Tran et al. (2015, 2017) for examples of more sophisticated families. The approximate distribution q^* is chosen to satisfy

$$q^* = \arg \min_{q(\phi|\lambda)} KL(q(\phi|\lambda)||p(\phi|\mathbf{d})). \quad (5)$$

Here, KL denotes the Kullback-Leibler divergence of $q(\phi|\lambda)$ from $p(\phi|\mathbf{d})$. Finding q^* is done in practice by maximizing the *evidence lower bound (ELBO)*

$$\mathcal{L}(\lambda) = \mathbb{E}_q \left[\log p(\mathbf{d}|\phi) \right] - KL(q(\phi|\lambda)||p(\phi)), \quad (6)$$

which is a sum of the expected data log-likelihood and the negative KL divergence between the combined prior distribution $p(\phi)$ of calibration parameters, the error scale σ , and GP hyperparameters and the variational distribution $q(\phi|\lambda)$. Note that we set $\mathcal{L}(\lambda) := \mathcal{L}(q(\phi|\lambda))$ for the ease of notation. Minimizing the ELBO is equivalent to minimizing the original objective function.

The ELBO can be optimized via the standard coordinate or gradient ascent methods. These techniques are inefficient for large datasets, because we must optimize the variational parameters globally for the whole dataset. Instead, it has become common practice to use a stochastic gradient ascent (SGA) algorithm, which Hoffman et al. (2013) named “stochastic variational inference”

(SVI). Similarly to the traditional gradient ascent, SGA updates $\boldsymbol{\lambda}$ at the t^{th} iteration with

$$\boldsymbol{\lambda}_{t+1} \leftarrow \boldsymbol{\lambda}_t + \rho_t \tilde{\ell}(\boldsymbol{\lambda}_t). \quad (7)$$

Here, $\tilde{\ell}(\boldsymbol{\lambda})$ is a realization of the random variable $\tilde{\mathcal{L}}(\boldsymbol{\lambda})$, so that $\mathbb{E}(\tilde{\mathcal{L}}(\boldsymbol{\lambda})) = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$, and [Ranganath et al. \(2014\)](#) showed that the gradient of ELBO with respect to the variational parameter $\boldsymbol{\lambda}$ can be written as

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) (\log p(\mathbf{d}|\boldsymbol{\phi}) - \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})}) \right], \quad (8)$$

where $\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda})$ is the gradient of the variational log-likelihood with respect to $\boldsymbol{\lambda}$.

SGA converges to a local maximum of $\mathcal{L}(\boldsymbol{\lambda})$ (global for $\mathcal{L}(\boldsymbol{\lambda})$ concave ([Bottou et al., 1997](#))) when the learning rate ρ_t follows the Robbins-Monro conditions ([Robbins and Monro, 1951](#))

$$\sum_{t=1}^{\infty} \rho_t = \infty, \quad \sum_{t=1}^{\infty} \rho_t^2 < \infty. \quad (9)$$

The bottleneck in the computation of the ELBO gradient $\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$ is the evaluation of the log-likelihood $\log p(\mathbf{d}|\boldsymbol{\phi})$, which makes the traditional gradient methods as hard to scale as MCMC methods. SGA algorithms address this challenge. If we consider N independent observations $d_i \sim p(d_i|\boldsymbol{\phi})$, then we can define a noisy estimate of the gradient $\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$ as

$$\begin{aligned} \tilde{\mathcal{L}}(\boldsymbol{\lambda}) &:= N \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) (\log p(d_I|\boldsymbol{\phi})) \right] \\ &\quad - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right], \end{aligned} \quad (10)$$

where $I \sim U(1, \dots, N)$ with $\mathbb{E}(\tilde{\mathcal{L}}(\boldsymbol{\lambda})) = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$. Each update of $\boldsymbol{\lambda}$ computes the likelihood only for one observation d_i at a time and makes the SVI scalable for large datasets. One can easily see that, under the framework for Bayesian calibration, $\mathbb{E}(\tilde{\mathcal{L}}(\boldsymbol{\lambda})) \neq \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$ and that the corresponding **noisy estimates are biased**.

3 Variational Calibration of Computer Models

In this section, we derive the algorithm for scalable variational inference approach to Bayesian computer model calibration. The first step is finding a convenient decomposition of the likelihood $p(\mathbf{d}|\phi)$ that allows for an unbiased stochastic estimate of the gradient $\nabla_{\lambda}\mathcal{L}(\lambda)$ that depends only on a small subset of data. Multivariate copulas, and specifically their pairwise construction which we shall introduce below, provide such a decomposition. We are not the first ones to use copulas in the context of VBI. For instance, [Tran et al. \(2015\)](#) and [Smith et al. \(2020\)](#) proposed a multivariate copula as a possible variational family. However, we are the first ones using copulas in the context of computer model calibration implementing via VBI.

3.1 Multivariate Copulas and Likelihood Decomposition

Fundamentally, a copula separates the information on the dependence structure of $N > 1$ random variables D_1, \dots, D_N from their marginal distributions. Let us assume, for simplicity, that the marginal cumulative distribution functions (CDFs) F_1, \dots, F_N are continuous and possess the inverse functions $F_1^{-1}, \dots, F_N^{-1}$. It follows from the probability integral transform that $U_i := F_i(D_i) \sim U(0, 1)$ and conversely that $D_i = F_i^{-1}(U_i)$. With this in mind, we have

$$\begin{aligned} P(D_1 \leq F_1^{-1}(d_1), \dots, D_N \leq F_N^{-1}(d_N)) \\ = P(U_1 \leq d_1, \dots, U_N \leq d_N) := C(d_1, \dots, d_N). \end{aligned}$$

The function C is a distribution with support on $[0, 1]^N$, uniform marginals, and is called a copula. Under the above assumptions, a one-to-one correspondence exists between copula C and the joint distribution of $\mathbf{D} = (D_1, \dots, D_N)^T$, as stated in the following theorem due to [Sklar \(1959\)](#). To keep the notation consistency and readability, we re-state the theorem here.

Theorem 1 ([Sklar \(1959\)](#)) *Given random variables D_1, \dots, D_N with continuous marginals F_1, \dots, F_N and joint distribution functions F , there exists a unique copula C such that for all $\mathbf{d} = (d_1, \dots, d_N)^T \in \mathbb{R}^N$ implies that $F(d_1, \dots, d_N) = C(F_1(d_1), \dots, F_n(d_N))$. Conversely, given F_1, \dots, F_N and copula C , the joint distribution F defined through $C(F_1(x_1), \dots, F_n(x_N))$ is an N -variate distribution functions with marginals F_1, \dots, F_N .*

Consequently, one can write the joint probability density function (pdf) f of \mathbf{D} as

$$f(d_1, \dots, d_N) = c(F_1(d_1), \dots, F_N(d_N)) \prod_{i=1}^N f_i(d_i), \quad (11)$$

where c represents the copula density and f_i is the marginal pdf of D_i .

The key reason for considering copulas is that one can decompose the N -dimensional copula density c into a product of bivariate copulas. The starting point for this construction is a recursive decomposition of the density f into a product of conditional densities

$$f(d_1, \dots, d_N) = \prod_{i=2}^N f(d_i | d_1, \dots, d_{i-1}) f(d_1). \quad (12)$$

For $N = 2$, the Sklar's theorem implies that

$$f(d_1, d_2) = c_{12}(F_1(d_1), F_2(d_2)) f_1(d_1) f_2(d_2), \quad (13)$$

and

$$f(d_1 | d_2) = c_{12}(F_1(d_1), F_2(d_2)) f_1(d_1), \quad (14)$$

where

$$c_{12} := c_{12}(F_1(d_1), F_2(d_2)) \quad (15)$$

is a density of $C(F_1(d_1), F_2(d_2)) = F(d_1, d_2)$. Using [\(14\)](#) for the decomposition of (D_1, D_p) given D_2, \dots, D_{p-1} , we obtain

$$f(d_p | d_1, \dots, d_{p-1}) = \left(\prod_{r=1}^{p-2} c_{r,p;r+1,\dots,p-1} \right) c_{(p-1),p} \cdot f_p(d_p), \quad (16)$$

where

$$c_{i,j;i_1,\dots,i_m} := c_{i,j;i_1,\dots,i_m}(F(d_i | d_{i_1}, \dots, d_{i_m}), F(d_j | d_{i_1}, \dots, d_{i_m})) \quad (17)$$

and

$$F(d_i, d_j | d_{i_1}, \dots, d_{i_m}) := C_{i,j;i_1,\dots,i_m}(F(d_i | d_{i_1}, \dots, d_{i_m}), F(d_j | d_{i_1}, \dots, d_{i_m})). \quad (18)$$

Using (12) and (16) with the specific index choices $p = i, r = i + j$, we have that

$$f(d_1, \dots, d_N) = \left[\prod_{j=1}^{N-1} \prod_{i=1}^{N-j} c_{i,(i+j);(i+1),\dots,(i+j-1)} \right] \prod_{k=1}^N f_k(d_k). \quad (19)$$

Note that $c_{i,j;i_1,\dots,i_m}$ is a two-dimensional copula evaluated at the conditional CDFs $F(d_i | d_{i_1}, \dots, d_{i_m})$ and $F(d_j | d_{i_1}, \dots, d_{i_m})$. This decomposition is called a *D-vine*. A similar class of decompositions is possible when one applies (14) on (D_{p-1}, D_p) given D_1, \dots, D_{p-2} and sets $j = p - r, j + i = p$ to get a *canonical vine (C-vine)* (Cooke and Kurowicka, 2006):

$$f(d_1, \dots, d_N) = f_1(d_1) \left[\prod_{p=2}^N \prod_{k=1}^{p-1} c_{p-k,t;1,\dots,(p-k-1)} \cdot f_p(d_p) \right] = \left[\prod_{j=1}^{N-1} \prod_{i=1}^{N-j} c_{j,(j+i);1,\dots,(j-1)} \right] \prod_{k=1}^N f_k(d_k). \quad (20)$$

One can easily imagine that many such pair-copula decompositions exist. Bedford and Cooke (2002) observed that these can be represented graphically as a sequence of nested trees with undirected edges, which are referred to as *vine trees* and their decompositions as regular vines. Here, we focus exclusively on the D-vine and C-vine decompositions because they represent the most-studied instances of regular vines and provide an especially efficient notation. We note, however, that the following results can be extended to any regular vines.

Properties of vine copulas (Cooke and Kurowicka, 2006): The vine copula construction is particularly attractive for two reasons. First, each pair of variables occurs only once as a conditioning set. Second, the bivariate copulas involved in the decompositions have convenient form in the case of Gaussian likelihood f . In particular, let $\mathbf{D} = (D_1, \dots, D_N)^T$ follows a multivariate normal distribution with $F_j = \Phi, j = 1, \dots, N$, where Φ is the standard normal CDF. The bivariate

copula density is

$$c_{i,j;i_1,\dots,i_m}(u_i, u_j) = \frac{1}{\sqrt{1-\kappa^2}} \exp\left\{-\frac{\kappa^2(w_i^2 + w_j^2) - 2\kappa w_i w_j}{2(1-\kappa^2)}\right\}. \quad (21)$$

Here, $u_i = F(d_i|d_{i_1}, \dots, d_{i_m})$, $u_j = F(d_j|d_{i_1}, \dots, d_{i_m})$, $w_i = \Phi^{-1}(u_i)$, $w_j = \Phi^{-1}(u_j)$, and $\kappa = \rho_{i,j;i_1,\dots,i_m}$ is the partial correlation of variables i, j given i_1, \dots, i_m . The D-vine and C-vine decompositions also involve conditional CDFs, for which we need further expressions. Let v be an index in a set Ξ and $\Xi_{-v} := \Xi \setminus v$ so that Ξ contains more than one element, $F(d_j|\mathbf{d}_\Xi)$ is typically computed recursively as

$$F(d_j|\mathbf{d}_\Xi) = h(F(d_j|\mathbf{d}_{\Xi_{-v}}), F(d_v|\mathbf{d}_{\Xi_{-v}})|\rho_{j,v;\Xi_{-v}}) \quad (22)$$

and the function h is for the Gaussian case given by

$$h(u_i, u_j|\rho_{i,j;i_1,\dots,i_m}) = \Phi\left(\frac{\Phi^{-1}(u_i) - \rho_{i,j;i_1,\dots,i_m}\Phi^{-1}(u_j)}{\sqrt{1 - \rho_{i,j;i_1,\dots,i_m}^2}}\right). \quad (23)$$

Lastly, the partial correlation can be also computed recursively as

$$\rho_{i,j;\Xi} = \frac{\rho_{i,j;\Xi_{-v}} - \rho_{i,v;\Xi_{-v}}\rho_{v,j;\Xi_{-v}}}{\sqrt{1 - \rho_{i,v;\Xi_{-v}}^2}\sqrt{1 - \rho_{v,j;\Xi_{-v}}^2}}. \quad (24)$$

3.2 Scalable Algorithm with Truncated Vine Copulas

We now consider the data likelihood $p(\mathbf{d}|\phi)$ according to (3) and make use of vines to construct a noisy estimate of the gradient $\nabla_{\lambda}\mathcal{L}(\lambda)$. We additionally assume that $N = n + s$, where n is the number of observations y_i from the physical process, and s is the number of computer model runs z_j . The log-likelihood $\log p(\mathbf{d}|\phi)$ can be rewritten according to the D-vine decomposition as

$$\log p(\mathbf{d}|\phi) = \sum_{j=1}^{N-1} \sum_{i=1}^{N-j} p_{i,i+j}^{\Delta}(\phi), \quad (25)$$

where

$$p_{i,i+j}^\Delta(\boldsymbol{\phi}) = \log c_{i,(i+j);(i+1),\dots,(i+j-1)} + \frac{1}{n-1} (\log p_i(d_i|\boldsymbol{\phi}) + \log p_{i+j}(d_{i+j}|\boldsymbol{\phi})). \quad (26)$$

This can be conveniently used in the expression of the ELBO gradient. For a D-vine, we have that

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}) &= \sum_{j=1}^{N-1} \sum_{i=1}^{N-j} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda})(p_{i,i+j}^\Delta(\boldsymbol{\phi})) \right] \\ &\quad - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right]. \end{aligned} \quad (27)$$

The following proposition gives a noisy unbiased estimate $\tilde{\mathcal{L}}_\Delta(\boldsymbol{\lambda})$ of the gradient (27). Similarly, we can derive a noisy estimate $\tilde{\mathcal{L}}_C(\boldsymbol{\lambda})$ of the gradient using a C-vine. We leave the details of the C-vine case together with the proof of proposition 1 to the Appendix.

Proposition 1 *Let $\tilde{\mathcal{L}}_\Delta(\boldsymbol{\lambda})$ be an estimate of the ELBO gradient $\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$ defined as*

$$\begin{aligned} \tilde{\mathcal{L}}_\delta(\boldsymbol{\lambda}) &= \frac{N(N-1)}{2} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda})(p_{I_\Delta(K)}^\Delta(\boldsymbol{\phi})) \right] \\ &\quad - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right], \end{aligned}$$

where $K \sim U(1, \dots, \frac{N(N-1)}{2})$, and I_Δ is the bijection

$$I_\Delta : \{1, \dots, \frac{N(N-1)}{2}\} \rightarrow \{(i, i+j) : i \in \{1, \dots, N-j\} \text{ for } j \in \{1, \dots, N-1\}\},$$

then $\tilde{\mathcal{L}}_\Delta(\boldsymbol{\lambda})$ is unbiased i.e., $\mathbb{E}(\tilde{\mathcal{L}}_\Delta(\boldsymbol{\lambda})) = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$.

As in the case of SVI for independent data, these noisy estimates allow to update the variational parameter $\boldsymbol{\lambda}$ without the need to evaluate the whole likelihood $p(\mathbf{d}|\boldsymbol{\phi})$. We need to consider only the data consisting of a copula's conditioning and conditioned sets. Unfortunately, both $\tilde{\mathcal{L}}_\Delta(\boldsymbol{\lambda})$ and $\tilde{\mathcal{L}}_C(\boldsymbol{\lambda})$ can be relatively costly to compute for large datasets because of the recursive nature of calculations involved in the copula densities' evaluation. According to Brechmann et al. (2012); Dissmann et al. (2013), and Brechmann and Joe (2015), the most important and strongest

dependencies among variables can be typically captured best by the pair copulas of the first trees. This notion motivates the use of *truncated vine copulas*, where the copulas associated with the higher-order trees are set to the independence copulas. From the definition of a regular vine, one can show that the joint density f can be decomposed as

$$f(d_1, \dots, d_N) = \left[\prod_{j=1}^{N-1} \prod_{e \in E_i} c_{j(e), k(e); \Xi(e)} \right] \prod_{k=1}^N f_k(d_k),$$

where $e = j(e), k(e); \Xi(e) \in E_i$ is an edge in the i^{th} tree of the vine specification. We define the truncated regular vine copula as follows.

Definition 1 ([Brechmann et al. \(2012\)](#)) *Let $\mathbf{U} = \{U_1, \dots, U_N\}$ be a random vector with uniform marginals, and let $l \in \{1, \dots, N-1\}$ be the truncation level. Let Π denote the bivariate independence copula. Then, \mathbf{U} is said to be distributed according to an N -dimensional l -truncated R -vine copula if C is an N -dimensional R -vine copula with*

$$C_{j(e), k(e); \Xi(e)} = \Pi \quad \forall e \in E_i \quad i = l+1, \dots, N-1.$$

For the case of an **l-truncated D-vine**, we have

$$f(d_1, \dots, d_N) = \left[\prod_{j=1}^l \prod_{i=1}^{N-j} c_{i, (i+j); (i+1), \dots, (i+j-1)} \right] \prod_{k=1}^N f_k(d_k), \quad (28)$$

and analogically to the case of D-vine with no truncation, the log-likelihood $p(\mathbf{d}|\boldsymbol{\phi})$ can be written as a sum of unique elements given in Proposition [2](#).

Proposition 2 *If the copula of $p(\mathbf{d}|\boldsymbol{\phi})$ is distributed according to an l -truncated D-vine, we can rewrite*

$$\log p(\mathbf{d}|\boldsymbol{\phi}) = \sum_{j=1}^l \sum_{i=1}^{N-j} p_{i, i+j}^{\Delta_l}(\boldsymbol{\phi}), \quad (29)$$

where

$$p_{i, i+j}^{\Delta_l}(\boldsymbol{\phi}) = \log c_{i, (i+j); (i+1), \dots, (i+j-1)} + \frac{1}{a_i} \log p_i(d_i|\boldsymbol{\phi}) + \frac{1}{b_{i+j}} \log p_{i+j}(d_{i+j}|\boldsymbol{\phi}), \quad (30)$$

and

$$\begin{aligned} a_i &= 2l - \left[(l+1-i)\mathbb{1}_{i \leq l} + (l-N+i)\mathbb{1}_{i > N-l} \right], \\ b_{i+j} &= 2l - \left[(l+1-j-i)\mathbb{1}_{i+j \leq l} + (l-N+j+i)\mathbb{1}_{i+j > N-l} \right]. \end{aligned}$$

The main idea for the scalable variational calibration (VC) of computer models is replacing the full log-likelihood $\log(\mathbf{d}|\phi)$ in the definition of ELBO with the likelihood based on a truncated vine copula. This yields the *l-truncated ELBO* for the l-truncated D-vine

$$\mathcal{L}_{\Delta_l}(\boldsymbol{\lambda}) = \mathbb{E}_q \left[\sum_{j=1}^l \sum_{i=1}^{N-j} p_{i,i+j}^{\Delta_l}(\phi) \right] - KL(q(\phi|\boldsymbol{\lambda})||p(\phi)) \quad (31)$$

with its gradient

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\Delta_l}(\boldsymbol{\lambda}) = \sum_{j=1}^l \sum_{i=1}^{N-j} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\phi|\boldsymbol{\lambda})(p_{i,i+j}^{\Delta_l}(\phi)) \right] - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\phi|\boldsymbol{\lambda}) \log \frac{q(\phi|\boldsymbol{\lambda})}{p(\phi)} \right].$$

The following proposition gives a noisy unbiased estimate $\tilde{\mathcal{L}}_{\Delta_l}(\boldsymbol{\lambda})$ of the gradient $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\Delta_l}(\boldsymbol{\lambda})$. We can analogously derive an unbiased estimate $\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda})$ of the gradient using C-vine (see the Appendix for details and the proof).

Proposition 3 *Let $\tilde{\mathcal{L}}_{\Delta_l}(\boldsymbol{\lambda})$ be an estimate of the ELBO gradient $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\Delta_l}(\boldsymbol{\lambda})$ defined as*

$$\tilde{\mathcal{L}}_{\Delta_l}(\boldsymbol{\lambda}) = \frac{l(2N-(l+1))}{2} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\phi|\boldsymbol{\lambda})(p_{I_{\Delta_l}(K)}^{\Delta_l}(\phi)) \right] - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\phi|\boldsymbol{\lambda}) \log \frac{q(\phi|\boldsymbol{\lambda})}{p(\phi)} \right],$$

where $K \sim U(1, \dots, \frac{l(2N-(l+1))}{2})$, and I_{Δ_l} is the bijection

$$I_{\Delta_l} : \left\{ 1, \dots, \frac{l(2N-(l+1))}{2} \right\} \rightarrow \{(i, i+j) : i \in \{1, \dots, N-j\} \text{ for } j \in \{1, \dots, l\}\},$$

then $\tilde{\mathcal{L}}_{\Delta_l}(\boldsymbol{\lambda})$ is unbiased i.e., $\mathbb{E}(\tilde{\mathcal{L}}_{\Delta_l}(\boldsymbol{\lambda})) = \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\Delta_l}(\boldsymbol{\lambda})$.

Considering the l-truncated ELBO (31), our proposed algorithm for the VC of computer models with truncated vine copulas is stated in the Algorithm 1. Note that $\tilde{\mathcal{L}}_{\Delta_l}(\boldsymbol{\lambda})$ does not have closed

form expression in general due to expectations involved in the computation. Therefore, we resort to a Monte Carlo (MC) approximation of $\tilde{\mathcal{L}}_{\Delta_l}(\lambda)$ using samples from the variational distribution.

Algorithm 1: Variational calibration with truncated D-vine copulas.

Input: Data \mathbf{d} , mean and covariance functions for GPs, variational family $q(\phi|\boldsymbol{\lambda})$,
truncation level l

```

1  $\lambda \leftarrow$  random initial value
2  $t \leftarrow 1$ 
3 repeat
4   for  $s = 1$  to  $S$  do
5      $\phi[s] \sim q(\phi|\boldsymbol{\lambda})$ 
6      $K \leftarrow U(1, \dots, \frac{l(2N-(l+1))}{2})$ 
7      $\rho \leftarrow t^{\text{th}}$  value of a Robbins-Monro sequence
8      $\boldsymbol{\lambda} \leftarrow$ 
        $\boldsymbol{\lambda} + \rho \frac{1}{S} \sum_{s=1}^S \left[ \frac{l(2N-(l+1))}{2} \nabla_{\boldsymbol{\lambda}} \log q(\phi[s]|\boldsymbol{\lambda}) \times \left( p_{I_{\Delta_l}(K)}^{\Delta_l}(\phi[s]) - \frac{2}{l(2N-(l+1))} \log \frac{q(\phi[s]|\boldsymbol{\lambda})}{p(\phi[s])} \right) \right]$ 
9      $t \leftarrow t + 1$ 
10 until change of  $\boldsymbol{\lambda}$  is less than  $\epsilon$ 

```

Scalability Discussion: The complexity of bivariate copula evaluation depends on the size of conditioning dataset due to the recursive nature of the calculations (Cooke and Kurowicka, 2006). From the vine tree construction, the cardinality of the conditioning set for D-vine and C-vine is in the worst case $N - 2$. Nevertheless, on average, we can do better. Indeed, let X be the cardinality of the conditioning set in $p_{I_{\Delta}(K)}^{\Delta}$ (or $p_{I_C(K)}^C$), then

$$P(X = i) = \frac{N - (i + 1)}{\binom{n}{2}} \quad \text{for } i \in \{0, \dots, N - 2\} \quad (32)$$

and $\mathbb{E}(X) = \frac{N-2}{3}$. The cardinality of conditioning set is on average roughly $N/3$. On the other hand, the cardinality of conditioning set is for the case of Algorithm 1 at most $l - 1$. Now, let X_l be the cardinality of the conditioning set in the updating step of the variational parameter λ in the Algorithm 1, then

$$P(X_l = i) = \frac{N - (i + 1)}{\frac{l(2N-(l+1))}{2}} \quad \text{for } i \in \{0, \dots, l - 1\}, \quad (33)$$

and $\mathbb{E}(X_l) = [(l-1)(3N-2l-2)]/[3(2N-l-1)]$. $\mathbb{E}(X_l) \approx 2$ for $N = 10^5$ and truncation level $l = 5$, which is a significant improvement to the average case $p_{I_{\Delta}(K)}^{\Delta}$ and $p_{I_C(K)}^C$ (≈ 33333 for $N = 10^5$).

This provides a heuristic yet compelling argument for the scalability.

4 Implementation Details

4.1 Selection of Truncation Level

Selection of the truncation level l is an important element in effective approximation of the posterior distribution $p(\phi|\mathbf{d})$ under Algorithm 1. [Dissmann et al. \(2013\)](#) propose a sequential approach for selection of l in the case of vine estimation. One sequentially fits models with an increasing truncation level until the quality of fit stays stable or computational resources are depleted. We adopt similar idea for the case of VC of computer models with vine copulas. Let $\lambda(l)$ represents the value of variational parameter estimated with Algorithm 1 for a fixed truncation level l . One can then sequentially increase l until $\|\lambda(l+1) - \lambda(l)\| < \epsilon$ for some norm $\|\cdot\|$ and a desired tolerance ϵ .

4.2 Variance Reduction of Monte Carlo Approximations

The computational convenience of MC approximations of the gradient estimators based on the l -truncated D-vine and C-vine copulas $\tilde{\mathcal{L}}_{\Delta_l}(\lambda)$ and $\tilde{\mathcal{L}}_{C_l}(\lambda)$ (see Section 3.2) is typically accompanied by their large variance. The consequence in practice is the need for small step size ρ_t in the SGA portion of Algorithm 1 which results in a slower convergence. In order to reduce the variance of MC approximations, we adopt similar approach as [Ruiz et al. \(2016\)](#) and use Rao-Blackwellization ([Casella and Robert, 1996](#)), control variates (CV) ([Ross, 2006](#)), and importance sampling. The remainder of this section focuses on the case of D-vine decomposition, see Appendix for the derivations for C-vines.

4.2.1 Rao-Blackwellization

The idea of Rao-Blackwellization is to replace the noisy estimate of gradient with its conditional expectation with respect to a subset of ϕ . For simplicity, let us consider a situation with $\phi = (\phi_1, \phi_2) \in \mathbb{R}^2$ and variational family $q(\phi|\lambda)$ that factorizes into $q(\phi_1|\lambda_1)q(\phi_2|\lambda_2)$. Addi-

tionally, let $\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)$ be the MC approximation of the gradient $\nabla_{\lambda}\mathcal{L}(\lambda)$. Now, the conditional expectation $\mathbb{E}[\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)|\phi_1]$ is also an unbiased estimate of $\nabla_{\lambda}\mathcal{L}(\lambda)$ since $\mathbb{E}_q(\mathbb{E}[\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)|\phi_1]) = \mathbb{E}_q(\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2))$ and

$$\text{Var}_q(\mathbb{E}[\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)|\phi_1]) = \text{Var}_q(\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)) - \mathbb{E}[(\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2) - \mathbb{E}[\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)|\phi_1])^2]$$

shows that $\text{Var}_q(\mathbb{E}[\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)|\phi_1]) \leq \text{Var}_q(\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2))$. The factorization of the variational family also makes the conditional expectation straightforward to compute as

$$\mathbb{E}[\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)|\phi_1] = \int_{\phi_2} \mathbb{E}[\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)] \frac{q(\phi_1|\lambda_1)q(\phi_2|\lambda_2)}{q(\phi_1|\lambda_1)} d\phi_2 = \mathbb{E}_{q(\phi_2|\lambda_2)}(\hat{\mathcal{L}}_{\lambda}(\phi_1, \phi_2)),$$

i.e., we just need to integrate out some variables. Let us consider the MC approximation of the gradient estimator $\tilde{\mathcal{L}}_{\Delta_l}(\lambda)$. The j^{th} entry of the Rao-Blackwellized estimator is

$$\frac{1}{S} \sum_{s=1}^S \left[\frac{l(2N - (l + 1))}{2} \nabla_{\lambda_j} \log q(\phi_j[s]|\lambda_j) (\tilde{p}_{(j)}(\phi[s]) - \frac{2}{l(2N - (l + 1))} \log \frac{q(\phi_j[s]|\lambda_j)}{p(\phi_j[s])}) \right],$$

where $\tilde{p}_{(j)}(\phi)$ are the components of $p_{I_{\Delta_l}(K)}^{\Delta_l}(\phi)$ that include ϕ_j .

4.2.2 Control Variates

To further reduce the variance of the MC approximations we will replace the Rao-Blackwellized estimate above with a function that has the same expectation but again smaller variance. For illustration, let us first consider a target function $\xi(\phi)$ whose variance we want to reduce, and a function $\psi(\phi)$ with finite expectation. Define

$$\hat{\xi}(\phi) = \xi(\phi) - a(\psi(\phi) - \mathbb{E}_q[\psi(\phi)]), \quad (34)$$

where a is a scalar and $\mathbb{E}_q(\hat{\xi}(\phi)) = \mathbb{E}_q[\xi(\phi)]$. The variance of $\hat{\xi}(\phi)$ is

$$\text{Var}_q(\hat{\xi}(\phi)) = \text{Var}_q(\xi(\phi)) + a^2 \text{Var}_q(\psi(\phi)) - 2a \text{Cov}_q(\xi(\phi), \psi(\phi)). \quad (35)$$

This shows that a good choice for function $\psi(\phi)$ is one that has high covariance with $\xi(\phi)$. Moreover, the value of a that minimizes (35) is

$$a^* = \frac{\text{Cov}_q(\xi(\phi), \psi(\phi))}{\text{Var}_q(\psi(\phi))}. \quad (36)$$

Let us place the CV back into the context of calibration. Meeting the above described criteria, Ranganath et al. (2014) propose $\psi(\phi)$ to be $\nabla_{\lambda} \log q(\phi|\lambda)$, because it depends only on the variational distribution and has expectation zero. We can now set the target function $\xi(\phi)$ to be

$$\frac{l(2N - (l + 1))}{2} \nabla_{\lambda_j} \log q(\phi_j|\lambda_j) \left(\tilde{p}_{(j)}(\phi) - \frac{2}{l(2N - (l + 1))} \log \frac{q(\phi_j|\lambda_j)}{p(\phi_j)} \right),$$

which gives the following j^{th} entry of the MC approximation of the gradient estimator $\tilde{\mathcal{L}}_{\Delta_l}(\lambda)$ with CV

$$\tilde{\mathcal{L}}_{\Delta_l}^{\text{CV}(j)}(\lambda) = \frac{1}{S} \sum_{s=1}^S \left[\frac{l(2N - (l + 1))}{2} \nabla_{\lambda_j} \log q(\phi_j[s]|\lambda_j) \left(\tilde{p}_{(j)}(\phi[s]) - \frac{2(\log \frac{q(\phi_j[s]|\lambda_j)}{p(\phi_j[s])} + \hat{a}_j^{\Delta})}{l(2N - (l + 1))} \right) \right],$$

where \hat{a}_j^{Δ} is the estimate of a^* based on additional independent draws from the variational approximation (otherwise the estimator would be biased).

4.2.3 Importance sampling

The ultimate variance reduction technique used is the importance sampling. We refer to Ruiz et al. (2016) for full description of the method and illustration of its efficiency in the VBI framework. Fundamentally, instead of taking samples from the variational family $q(\phi|\lambda)$ to carry out the MC approximation of the ELBO gradient estimate, we will take samples from an overdispersed distribution $r(\phi|\lambda, \tau)$ in the same family that depends on an additional dispersion parameter $\tau > 1$. Namely, we can write the estimate $\tilde{\mathcal{L}}_{\Delta_l}(\lambda)$ as

$$\mathbb{E}_{r(\phi|\lambda, \tau)} \left[\frac{l(2N - (l + 1))}{2} \nabla_{\lambda} \log q(\phi|\lambda) (p_{I_{\Delta_l}(K)}^{\Delta_l}(\phi) - \frac{2}{l(2N - (l + 1))} \log \frac{q(\phi|\lambda)}{p(\phi)}) w(\phi) \right],$$

where $w(\phi) = q(\phi|\boldsymbol{\lambda})/r(\phi|\boldsymbol{\lambda}, \tau)$ is the importance weight which guarantees the estimator to be unbiased. The reason to formulate the $\tilde{\mathcal{L}}_{\Delta_l}(\lambda)$ this way comes from the fact the optimal proposal (Robert and Casella, 2005) distribution to form the MC estimate is not $q(\phi|\boldsymbol{\lambda})$, but rather

$$r^*(\phi) \propto q(\phi|\boldsymbol{\lambda})|\xi(\phi)|, \quad (37)$$

where

$$\xi(\phi) = \frac{l(2N - (l + 1))}{2} \nabla_{\boldsymbol{\lambda}} \log q(\phi|\boldsymbol{\lambda}) \left(p_{I_{\Delta_l}(K)}^{\Delta_l}(\phi) - \frac{2}{l(2N - (l + 1))} \log \frac{q(\phi|\boldsymbol{\lambda})}{p(\phi)} \right). \quad (38)$$

However, the normalizing constant for the optimal $r^*(\phi)$ is intractable, and so Ruiz et al. (2016) propose that an overdispersed version of the variational family that assigns higher probability to the tails of $q(\phi|\boldsymbol{\lambda})$ is closer to the optimum than $q(\phi|\boldsymbol{\lambda})$ itself. For example, if the value of $\boldsymbol{\lambda}$ makes the variational family a poor fit, then the samples $\phi[s] \sim q(\phi|\boldsymbol{\lambda})$ have a high value for the variational distribution but low for the true posterior. On the other hand, $r^*(\phi)$ proposes values of $\phi[s]$ for which $\xi(\phi)$ is large that are in the tails of $p(\phi|\boldsymbol{\lambda})$.

To see how the importance sampling leads to the reduction of variance of the MC estimates, let us consider the following estimator

$$\hat{\mathcal{L}}_{MC} = \frac{1}{S} \sum_{s=1}^S \xi(\phi[s]), \quad \phi[s] \sim p(\phi|\boldsymbol{\lambda}), \quad (39)$$

then

$$\text{Var}[\hat{\mathcal{L}}_{MC}] = \frac{1}{S} \mathbb{E}_q[\xi^2(\phi)] - \frac{1}{S} [\tilde{\mathcal{L}}_{\Delta_l}(\lambda)]^2. \quad (40)$$

Similarly, we can derived the variance of the MC estimator with the importance weights

$$\hat{\mathcal{L}}_{MC}^O = \frac{1}{S} \sum_{s=1}^S \xi(\phi[s]) \frac{q(\phi[s]|\boldsymbol{\lambda})}{r(\phi[s]|\boldsymbol{\lambda}, \tau)}, \quad \phi[s] \sim r(\phi|\boldsymbol{\lambda}, \tau), \quad (41)$$

as

$$\text{Var}[\hat{\mathcal{L}}_{MC}^O] = \frac{1}{S} \mathbb{E}_q[\xi^2(\phi) \frac{q(\phi|\boldsymbol{\lambda})}{r(\phi|\boldsymbol{\lambda}, \tau)}] - \frac{1}{S} [\tilde{\mathcal{L}}_{\Delta_l}(\lambda)]^2. \quad (42)$$

The extension of the Algorithm [1](#) with the variance reductions of the MC approximations due to Rao-Blackwellization, CV, and importance sampling is summarized in the Algorithm [2](#).

Algorithm 2: Variational calibration with truncated D-vine copulas II.

Input: Data \mathbf{d} , mean and covariance functions for GPs, variational family $q(\phi|\boldsymbol{\lambda})$, dispersion parameter τ , **truncation level l**

```

1  $\lambda \leftarrow$  random initial value
2  $t \leftarrow 1$ 
3 repeat
4   for  $s = 1$  to  $S$  do
5      $\phi[s] \sim r(\phi|\boldsymbol{\lambda}, \tau)$ 
6      $K \leftarrow U(1, \dots, \frac{l(2N-(l+1))}{2})$ 
7      $\rho \leftarrow t^{\text{th}}$  value of a Robbins-Monro sequence
8      $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \boldsymbol{\rho} \sum_{s=1}^S \left[ \frac{l(2N-(l+1))}{2S} \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j[s]|\boldsymbol{\lambda}_j) \times (\tilde{p}_{(j)}(\phi[s]) - \right.$ 
        $\left. \frac{2(\log \frac{q(\phi_j[s]|\boldsymbol{\lambda}_j)}{p(\phi_j[s])} + \tilde{a}_j^\Delta)}{l(2N-(l+1))}) w(\phi_j[s]) \right]$ 
9      $t \leftarrow t + 1$ 
10 until change of  $\boldsymbol{\lambda}$  is less than  $\epsilon$ 

```

4.3 Choice of the learning rate

Even though the SGA is straightforward in its general definition, the choice of learning rate ρ_t can be challenging in practice. Ideally, one would want the rate to be small in the situations where the noisy estimates of the gradient have large variance and vice-versa. The elements of variational parameter λ can also differ in scale, and one needs to set the learning rate so that the SGA can accommodate even the smallest scales. The rapidly increasing usage of machine learning techniques in recent years produced various algorithms for element-wise adaptive-scale learning rates. We use the adaptive gradient (AdaGrad) algorithm (Duchi et al., 2011) which has been considered in similar problems before, e.g., Ranganath et al. (2014), however, there are other popular algorithms such as the ADADELTA (Zeiler, 2012) or the RMSProp (Tieleman and Hinton, 2012). Let \mathbf{g}_T be the gradient used in the T^{th} step of the SGA algorithm, and \mathbf{G}_t be the matrix consisting of the sum of the outer products of these gradients across the first t iterations, namely

$$\mathbf{G}_t = \sum_{T=1}^t \mathbf{g}_T \mathbf{g}_T^T. \quad (45)$$

The AdaGrad defines the element-wise adaptive scale learning rate as $\rho_t = \eta \cdot \text{diag}(\mathbf{G}_t)^{-1/2}$, where η is the initial learning rate. It is a common practice, however, to add a small constant value to $\text{diag}(\mathbf{G}_t)$ (typically of order 10^{-6}) to avoid division by zero.

4.4 Parametrizations

Variational families. We use a Gaussian distribution for real valued components of ϕ and a gamma distribution for positive variables. Both families are parametrized in terms of their mean and standard deviation. Moreover, in order to avoid constrained optimization, we transform all the positive variational parameters λ to $\tilde{\lambda} = \log(e^\lambda - 1)$ and optimize with respect to $\tilde{\lambda}$.

Overdispersed families. Given a fixed dispersion coefficient τ , the overdispersed Gaussian distribution with mean μ and standard deviation σ is a Gaussian distribution with mean μ and standard deviation $\sigma\sqrt{\tau}$. The overdispersed gamma distribution with mean μ and standard deviation σ is a gamma distribution with mean $\mu + (\tau - 1)\frac{\sigma^2}{\mu}$ and standard deviation $\frac{\sigma\sqrt{\tau\mu^2 + \tau\sigma^2(\tau - 1)}}{\mu}$.

(Ruiz et al., 2016).

5 Applications

5.1 Simulation study

In this section, we study Algorithm 2 in a simulated scenario, where we first demonstrate the method’s fidelity in approximating the posterior distribution of calibration parameters $p(\boldsymbol{\theta}|\mathbf{d})$ and substantiate the indispensability of the variance reduction techniques described in Section 4 in order to achieve convergence. Second, we show the scalability of our method in comparison to the popular MH algorithm and the NUTS.

	GP mean	GP covariance function
f_m	$\theta_1 \cos(t_1) + \theta_2 \sin(t_2)$	$\eta_f \cdot \exp(-\frac{\ \mathbf{t}-\mathbf{t}'\ ^2}{2l_f^2} - \frac{\ \boldsymbol{\theta}-\boldsymbol{\theta}'\ ^2}{2l_\theta^2})$
δ	β_δ	$\eta_\delta \cdot \exp(-\frac{\ \mathbf{t}-\mathbf{t}'\ ^2}{2l_\delta^2})$

Table 1: The specification of GPs for the simulation study.

Let us consider a simple scenario following the model (2) with a two-dimensional calibration parameter $\boldsymbol{\theta} = (0.39, 0.60)$ that was obtained as a sample from its prior distribution $p(\boldsymbol{\theta})$ and a two-dimensional input variable $\mathbf{t} = (t_1, t_2)$. We model $f_m(\mathbf{t}, \boldsymbol{\theta})$ and $\delta(\mathbf{t})$ with GPs according to the specifications in Table 1 with the particular choices of $\eta_f = \frac{1}{30}$, $l_t = 1$, $l_\theta = 1$, $\eta_\delta = \frac{1}{30}$, $l_\delta = \frac{1}{2}$, and $\beta_\delta = 0.15$.

We choose the variational family to be the mean-field family with Gaussian distributions for real valued parameters and gamma distributions for positive variables following the parametrization discussed in Section 4.4. The variational parameters are initialized to match the prior distributions, and we use the AdaGrad for the learning rate updates.

5.1.1 Calibration

For the purpose of model calibration, we sampled the data \mathbf{d} jointly from the prior with the experimental noise following $\mathcal{N}(0, \frac{1}{100})$. The calibration parameter values for the model runs \mathbf{z} were selected on a uniform grid over $[0, 1]^2$ and the inputs \mathbf{t} over $[0, 3]^2$. For the first set of experiments, the size of the dataset was $N = 225$ with $n = 144$ and $s = 81$. We used 50 samples from the variational family to approximate the expectations in Algorithm 2 and 10 samples to implement the control variates.

Figure 1 demonstrates the quality of the variational approximation (Algorithm 2) in comparison to the MH algorithm and the NUTS. We can see that our method was able to accurately match both MCMC-based approximations with a minor deviation in θ_1 . It is important to note, however, that the variance reduction through the combination of the Rao-Blackwellization, control variates, and importance sampling was necessary to achieve meaningful convergence.

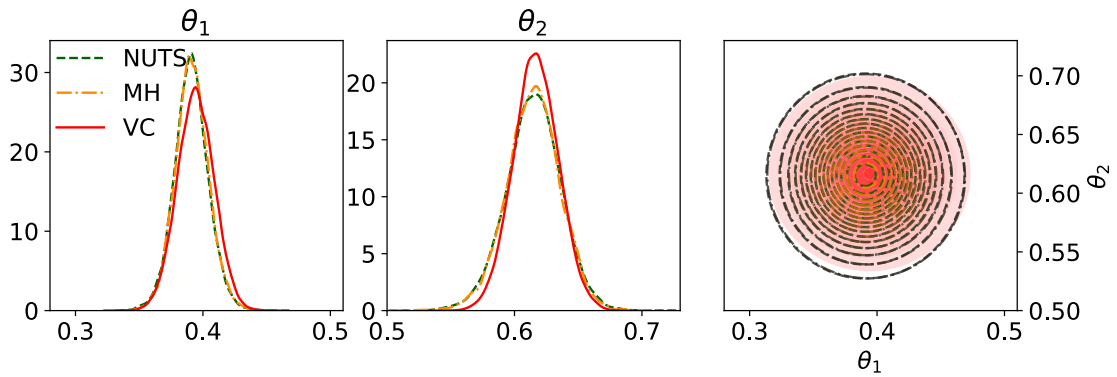


Figure 1: The approximate posterior distributions for the target calibration parameters. The VC (Algorithm 2) was carried out using $l = 3$ truncated D-vine and compared with the results from the NUTS and the MH algorithm.

In particular, Figure 2 shows the mean squared error (MSE) of the posterior predictive means, evaluated on an independently generated set of 50 data points, based on the VC with cumulatively implemented variance reduction techniques. Algorithm 2 which employs the importance sampling clearly outperforms the calibration with only the Rao-Blackwellization

and the calibration with control variates. In fact, each additional attempt to reduce the variance tends to decrease the MSE by one order of magnitude. There is naturally a time and space (memory) cost associated with each variance reduction technique. Figure 2 shows that the control variates and the importance sampling practically double the time per iteration of the algorithm. This additional complexity is, however, outweighed by the gain in the MSE reduction. The increase in memory consumption is less significant and is due to the storage of dispersion coefficients used for importance sampling and samples needed to compute control variates. Note that the memory consumed by the algorithms rises over time, because we chose to store the values of variational parameters during each step; the memory demands can be dramatically reduced if we drop these intermediate results.

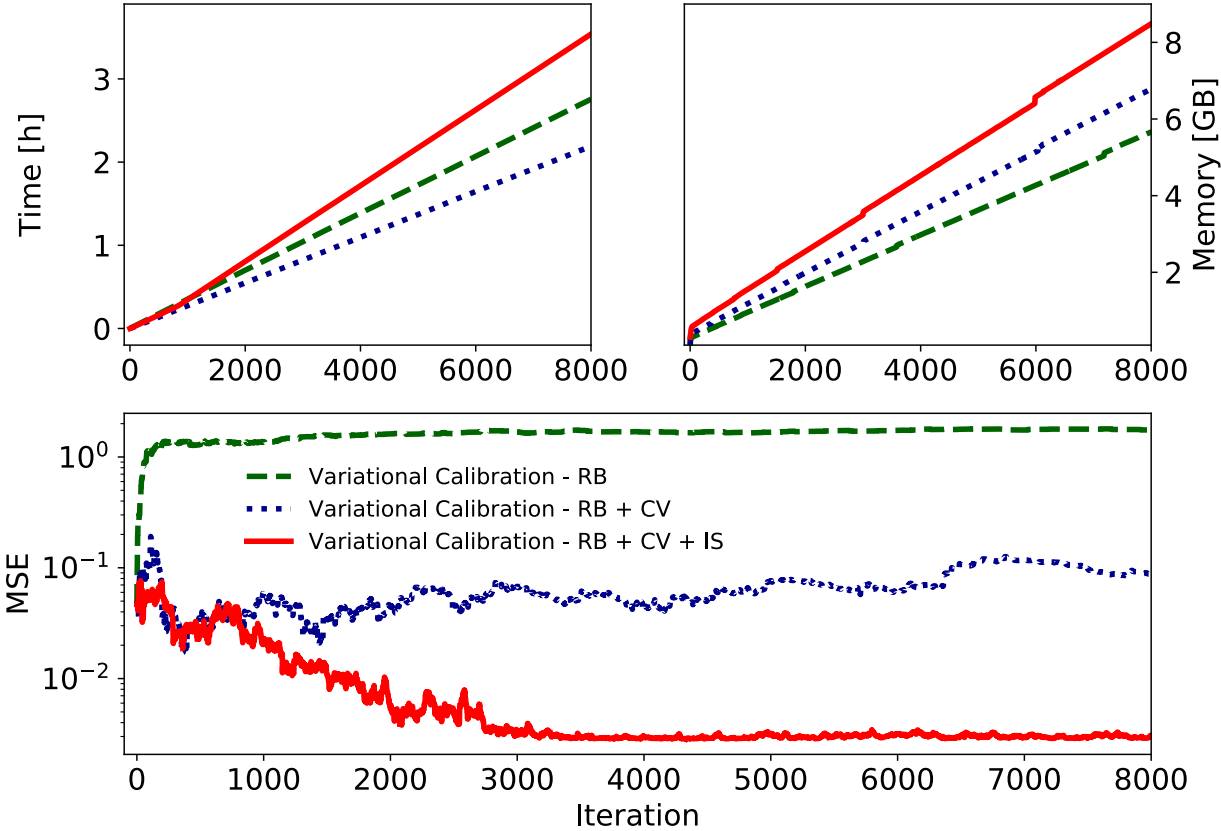


Figure 2: The evolution of MSE of the posterior predictive means based on the VC with cumulatively implemented variance reduction techniques described in Section 4.2. The figure is based on an independently generated set of 50 testing points. Time and memory demands for each of the implementations are also plotted the VC (Algorithm 2) was carried out using $l = 3$ truncated D-vine.

For completeness, in Table 2 we also compare the MSE of MCMC approximations and the VC at the point of convergence of the algorithms. The resulting errors in the predictions were, for all the practical purposes, equivalent.

Algorithm	<i>MSE</i>
VC with RB + CV + IS	2.9×10^{-3}
Metropolis-Hastings	3.0×10^{-3}
No-U-Turn	3.0×10^{-3}

Table 2: Comparison of the MSE for the simple scenario using the MH, the NUTS, and the VC algorithms.

5.1.2 Scalability

We now significantly increase the size of the dataset from $N = 225$ to 0.5×10^4 and eventually to 2×10^4 with the simulated experimental measurements and the model runs split equally ($n = s$). For better numerical stability, we expand the space of the input variables to $\mathbf{t} \in [0, 10]^2$ and select those using the Latin hypercube design. We also enlarge the testing dataset to 200 points. All the remaining simulation parameters are unchanged. The conventional MCMC methods are already impractical for the purpose of Bayesian calibration with these moderately large amounts of data. We were able to obtain only around 600 posterior samples in the case of $N = 1 \times 10^4$ and about 120 for $N = 2 \times 10^4$ in 25 hours of sampling using the MH algorithm (significantly less with the NUTS).

Algorithm 2 (D-vine with truncation $l = 5$) converges to the predictive MSE of about 0.003 under 4 hours for $N = 2 \times 10^4$ and 2 hours for $N = 0.5 \times 10^4$. It took similar time for the MH to achieve this MSE value for $N = 0.5 \times 10^4$ but almost 25 hours for the NUTS. Once we increased the data size to 2×10^4 , neither the NUTS nor the MH were able to achieve a similar predictive MSE as the VC within the 25 hour window allotted for sampling. In fact, they were by an order of magnitude larger. It is important to mention that both MCMC-based algorithms have also substantially larger memory demands than the VC. See Appendix C for more details.

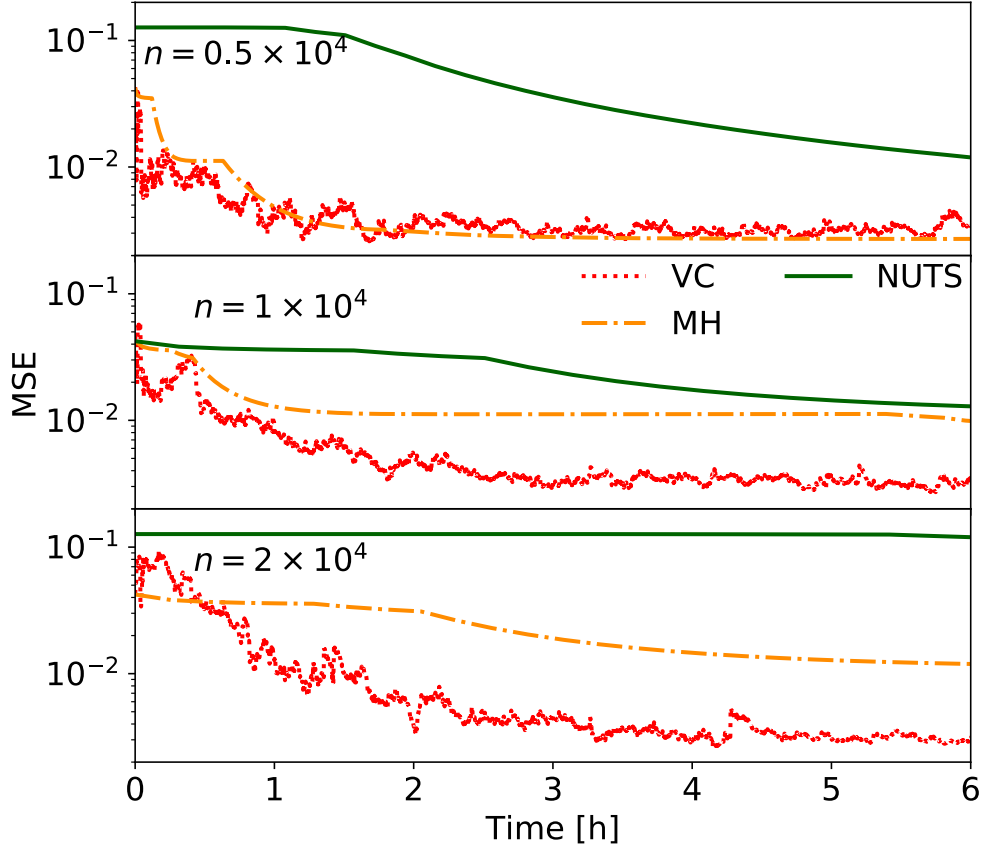


Figure 3: The evolution of the MSE of the posterior predictive means based on the VC (Algorithm 2), the MH algorithm, and the NUTS. The figure is based on an independently generated set of 200 testing points. The VC (Algorithm 2) was carried out using $l = 5$ truncated D-vine.

5.2 Calibration of Liquid Drop Model

Over the past decade or so, the statistical tools of uncertainty quantification have experienced a robust rump-up in use in the field of nuclear physics (Ireland and Nazarewicz, 2015). Bayesian calibration has been especially popular because it enhances the understanding of nuclear model’s structure through parameter estimation and potentially advances the quality of nuclear modeling by accounting for systematic errors. In this context, we use our variational Algorithm 2 to calibrate the 4-parameter Liquid Drop Model (LDM) (Myers and Swiatecki, 1966; Kirson, 2008; Benzaid et al., 2020) which is a global (across the whole nuclear chart) model of nuclear binding energies; the minimum energy needed to disassemble

the nucleus of an atom into free protons and neutrons. Nuclear binding energy is equivalent (energy-mass equivalence explained by $E = mc^2$) to the mass defect that corresponds to the difference between the mass number of a nucleus and its actual measured mass. In principle, the LDM treats the nucleus like molecules in a drop of incompressible fluid of very high density. Despite this simplification, the LDM accounts for the spherical shape of most nuclei and makes reasonable estimates of average properties of nuclei. The LDM is formulated through the semi-empirical mass formula as:

$$E_B(N, Z) = \theta_{\text{vol}}A - \theta_{\text{surf}}A^{2/3} - \theta_{\text{sym}}\frac{(N - Z)^2}{A} - \theta_C\frac{Z(Z - 1)}{A^{1/3}}. \quad (46)$$

where Z is the proton number, N is the neutron number, and $A = Z + N$ is the mass number of an atom. The calibration parameters are $\boldsymbol{\theta} = (\theta_{\text{vol}}, \theta_{\text{surf}}, \theta_{\text{sym}}, \theta_C)$ representing the volume, surface, symmetry and Coulomb energy, respectively. These parameters have specific physical meaning, where θ_{vol} is proportional to the volume of the nucleus for instance. See [Krane \(1987\)](#) for more details. Here we note that this is by no means the first case when Bayesian methodology is applied to study the LDM. In fact, the LDM is a popular model for statistical application ([Bertsch et al., 2005](#); [Yuan, 2016](#); [Bertsch and Bingham, 2017](#); [Kejzlar et al., 2021](#)) which is why we choose the model to illustrate our methodology as well. The LDM also generally performs better on heavy nuclei as compared to the light nuclei which alludes to the existence of a significant systematic discrepancy between the model and the experimental binding energies ([Reinhard et al., 2006](#); [Kejzlar et al., 2020](#)). Namely, we consider the following statistical model

$$y = E_B(N, Z) + \delta(N, Z) + \sigma\epsilon, \quad (47)$$

where $\delta(N, Z)$ represents the unknown systematic discrepancy between the semi-empirical mass formula and the experimental binding energies y . The parameter σ is as usual the scale of observation error $\epsilon \sim \mathcal{N}(0, 1)$. The nuclear physics community often ([Dobaczewski et al.,](#)

[2014] considers the least squares (LS) estimator of θ defined as

$$\hat{\theta}_{L_2} = \arg \min_{\theta} \sum_{i=1}^n (y_i - E_B(N_i, Z_i))^2 \quad (48)$$

which is also the maximum likelihood estimate of θ in the case of $\delta = 0$. The benefit of this estimator is that it is fast, easy-to-compute, and allows for analysis under the standard linear regression theory. It, however, neglects some sources of uncertainty that are accounted for in the Bayesian calibration framework.

To this end, we shall consider a GP prior with the mean zero and the squared exponential covariance function for the systematic discrepancy $\delta(Z, N)$. Since the main purpose of the example is to provide a canonical illustration of the methodology in a real data scenario, we also set a GP prior for the LDM and treat $E_B(Z, N)$ as an unknown function. We use 2000 experimental binding energies randomly selected from the AME2003 dataset [Audi et al., 2003] (publicly available at <http://amdc.impcas.ac.cn/web/masstab.html>) for calibration, see Figure 4, and an additional set of 10^4 model evaluations. The calibration

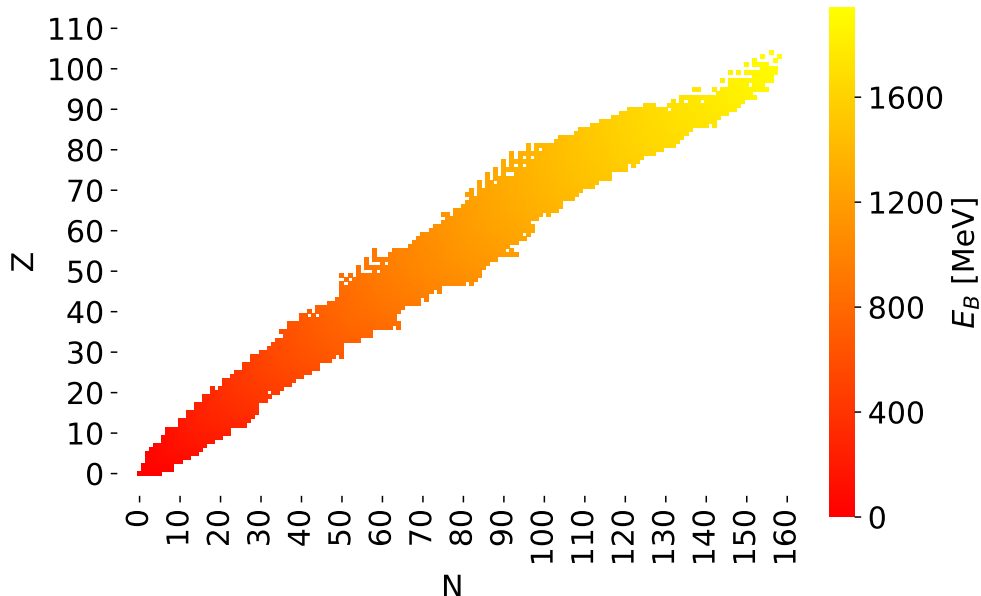


Figure 4: Experimental binding energies of nuclei in AME2003 dataset (2225 observations).

inputs were generated with the Latin hypercube design so that all the reasonable values of

$(\theta_{\text{vol}}, \theta_{\text{surf}}, \theta_{\text{sym}}, \theta_{\text{C}})$ given by the literature are covered (Weizsäcker, 1935; Bethe and Bacher, 1936; Myers and Swiatecki, 1966; Kirson, 2008; Benzaid et al., 2020). The model inputs (Z, N) were selected from the set of 2000 experimental binding energies, duplicated five-fold, and randomly permuted among the generated calibration inputs to span only the set of relevant nuclei. This relatively large number of model runs was chosen so that the combined 6 dimensional space of calibration parameters and model inputs is sufficiently covered considering the existence of a non-trivial systematic discrepancy. In fact, the uniform experimental design would amount only to 4-5 points per dimension.

Independent Gaussian distributions centered at the LS estimates $\hat{\theta}_{L_2}$ (in Table 3) with standard deviations large enough to cover the space of inputs used for generating the model runs were selected to represent the prior knowledge about the calibration parameters. Independent gamma distributions were used as the prior models for the hyperparameters of the GP’s covariance functions. We choose the variational family to be fully-factorized with the Gaussian distributions for real valued parameters and the gamma distributions for positive variables. The means of variational families were initialized as random samples from their respective prior distributions and the variances were set to match those of the prior distributions. We used the AdaGrad for stochastic optimization. See Appendix D for further discussion on the prior distributions and experimental design.

5.2.1 Results

Including the generated model runs, the overall size of training dataset is 1.2×10^4 which already makes the MCMC based Bayesian calibration impractical, as illustrated by the simulation study in Section 5.1. We therefore asses the quality of variational approximation only against the LDM with the standard LS estimation and do not consider the MCMC methods. In particular, we consider the testing dataset of the remaining 225 experimental binding energies in AME2003 that were excluded from the training data. The predictions $\hat{\mathbf{y}}^*$ of these testing binding energies \mathbf{y}^* were calculated, under the variational approximation,

as the posterior means of \mathbf{y}^* conditioned on the 1.2×10^4 binding energies from the training data set, i.e., the posterior means of the predictive distribution $p(\mathbf{y}^*|\mathbf{d})$. The predictions under the LS estimates $\hat{\boldsymbol{\theta}}_{L_2}$ were given by the semi-empirical mass formula (46).

Table 3 gives the root MSE for both methods under consideration. The VC (Algorithm 2) results are based on a 24 hour window dedicated to running the algorithm with 50 samples used to approximate the expectations, 10 samples used to implement the control variates, and the truncation level selected to be $l = 3$. By using GPs to account for the systematic discrepancies of the semi-empirical mass formula and the uncertainty of the LDM itself, we were able to significantly reduce the root MSE approx. 57% compared to the LS benchmark. Table 3 additionally shows the calibration parameter estimates and their standard errors. The estimates under the VC are given by the means of their variational families. Both the methods calibrate the LDM around the same values with notably low standard errors of the LS estimates. This is, however, expected since $\hat{\boldsymbol{\theta}}_{L_2}$ are ordinary LS estimates that in the presence of heteroscedasticity (see Figure 5) become inefficient and tend to significantly underestimate the true variance (Goldberger, 1966; Johnston, 1976).

Method	Parameter estimate and standard errors				Testing error
	a_{vol}	a_{surf}	a_{sym}	a_C	\sqrt{MSE} (MeV)
LS	15.42 (0.027)	16.91 (0.086)	22.47 (0.070)	0.69 (0.002)	3.54
VC	15.78 (0.198)	15.99 (0.681)	21.94 (0.510)	0.68 (0.018)	1.52

Table 3: The root MSE of the VC (Algorithm 2) after 24 hours dedicated to running the algorithm compared with the root MSE based on the LS estimates. The parameter estimates (and their standard errors) are also displayed.

The residual plot in Figure 5, showing the difference between \mathbf{y}^* and $\hat{\mathbf{y}}^*$ as a function of the nuclear mass number A , clearly demonstrates a better fit of the testing data with our methodology than is achieved by the simple LS fit. The majority of the residuals appear to be randomly spread around 0 which strongly supports the efficiency of GPs in accounting for the systematic discrepancy between the model and the physical process.

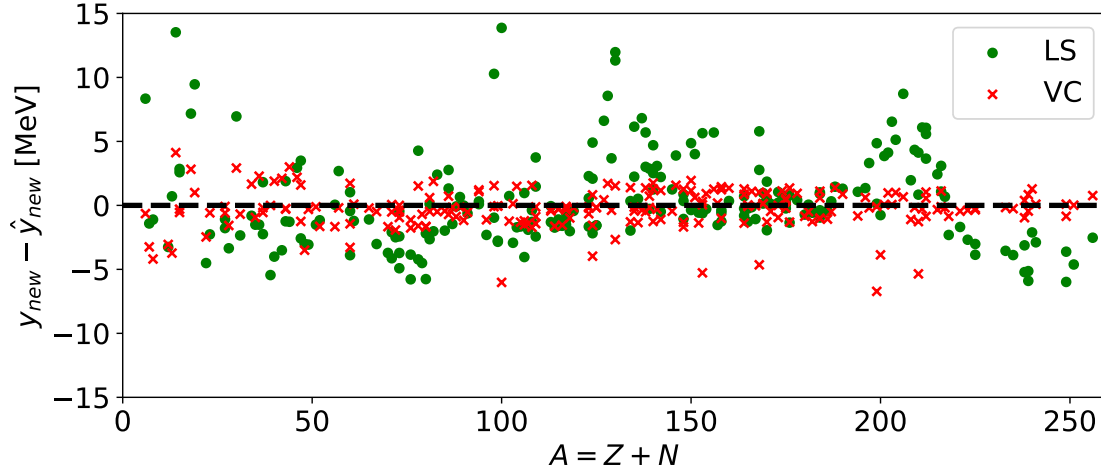


Figure 5: The residual plot for 225 experimental binding energies in the testing dataset.

6 Discussion

We developed and studied a VBI based approach to Bayesian calibration of computer models under the celebrated framework of [Kennedy and O’Hagan \(2001\)](#) which has been heavily utilized by practitioners for almost two decades. Our method consists of scalable and statistically principled tools for UQ of computationally complex and many-parameter computer models. We exploit the probabilistic theory of approximation coupled with pairwise construction of multivariate copulas using truncated regular vines to establish these tools. The theoretical justification for scalability was also discussed. Here we also note that while our work mainly focused on the model calibration framework, we can use the VBI algorithm developed in this paper for any models with complex Gaussian likelihoods with minimal modifications needed. Additionally, we dedicated a significant portion of this text to the description of implementation details that are often neglected in the literature. We discussed the choice of learning rate for stochastic optimization and outlined techniques to reduce the variance of noisy gradient estimates which include the Rao-Blackwellization, control variates, and importance sampling.

We are aware of efficient modeling of covariance modeling in spatial statistics that facilitates interpolation and prediction in the spatial domain. Our objective here is not to

model spatial covariance function instead build variational Bayes inference for calibrated computer models. The issues in calibrated computer models are fundamentally different. Therefore, the standard practice of spatial statistics is not directly importable for inference. We anticipate this research will have an impact on Bayesian spatial statistics in the near future.

In our examples, we first carried out an extensive simulation study that provided empirical evidence for accuracy and scalability of our method in scenarios where traditional MCMC based approaches become impractical. We established the superiority of the VC over the MH algorithm and the NUTS in terms of time efficiency and memory requirements. We also demonstrated the opportunities given by our method for practitioners on a real data example through calibration of the Liquid Drop Model of nuclear binding energies.

There are a few natural directions to enhance the methodology provided in this work from both computational and theoretical perspectives. First, an *a priori* method to select a sufficient truncation level for vine copulas would be beneficial to avoid the current sequential approach. For example, Brechmann and Joe (2015) discuss the use of fit indices for finding sufficient truncation. Secondly, the theoretical justification for our method would greatly benefit from establishing the link between the ELBO and the l-truncated ELBO which is the ultimate driving force behind the computational efficiency of the VC. Additionally, there are other alternatives to the traditional MCMC than VBI that have shown to be effective in handling massive datasets. The stochastic gradient MCMC (Ma et al., 2015) algorithm, for instance, utilizes similar data subsampling trick as VBI (see Section 2.1 for details) which has been successfully applied in deep learning (Deng et al., 2019) or state space models (Aicher et al., 2019). A similar copula likelihood decomposition to the one proposed in this paper could be used for computer model calibration via stochastic gradient MCMC, however, it would require a non-trivial algorithmic development that is beyond the scope of this work.

Appendix A Scalable Algorithm with Truncate C-Vine Copulas

Here we present the details of the C-vine based versions of Algorithm [1](#) and Algorithm [2](#). First, we can decompose the log-likelihood $\log p(\mathbf{d}|\boldsymbol{\phi})$ using a C-vine as

$$\log p(\mathbf{d}|\boldsymbol{\phi}) = \sum_{j=1}^{N-1} \sum_{i=1}^{N-j} p_{j,j+i}^C(\boldsymbol{\phi}), \quad (49)$$

where

$$p_{j,j+i}^C(\boldsymbol{\phi}) = \log c_{j,(j+i);1,\dots,(j-1)} + \frac{1}{N-1} (\log p_j(d_j|\boldsymbol{\phi}) + \log p_{j+i}(d_{j+i}|\boldsymbol{\phi})). \quad (50)$$

This now yields the following expression for the ELBO gradient:

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}) = \sum_{j=1}^{N-1} \sum_{i=1}^{N-j} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) (p_{j,j+i}^C(\boldsymbol{\phi})) \right] - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right]. \quad (51)$$

Equivalently to Proposition [1](#), we have the following proposition that establishes the noisy unbiased estimate of the gradient [\(51\)](#) using the C-vine copula decomposition.

Proposition 4 *Let $\tilde{\mathcal{L}}_C(\boldsymbol{\lambda})$ be an estimate of the ELBO gradient $\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$ defined as*

$$\tilde{\mathcal{L}}_C(\boldsymbol{\lambda}) = \frac{N(N-1)}{2} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) (p_{I_C(K)}^C(\boldsymbol{\phi})) \right] - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right],$$

where $K \sim U(1, \dots, \frac{N(N-1)}{2})$, and I_C is the bijection

$$I_C : \left\{ 1, \dots, \frac{N(N-1)}{2} \right\} \rightarrow \left\{ (j, j+i) : i \in \{1, \dots, N-j\} \text{ for } j \in \{1, \dots, N-1\} \right\},$$

then $\tilde{\mathcal{L}}_C(\boldsymbol{\lambda})$ is unbiased i.e., $\mathbb{E}(\tilde{\mathcal{L}}_C(\boldsymbol{\lambda})) = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$.

Again, $\tilde{\mathcal{L}}_C(\boldsymbol{\lambda})$ can be relatively costly to compute for large datasets due to the recursive

nature of the copula density computations. We now carry out exactly the same development an using l-truncated C-vine as in the case of Proposition 2 and Proposition 3.

Proposition 5 *If the copula of $p(\mathbf{d}|\boldsymbol{\phi})$ is distributed according to an l-truncated C-vine, we can rewrite*

$$\log p(\mathbf{d}|\boldsymbol{\phi}) = \sum_{j=1}^l \sum_{i=1}^{N-j} p_{i,i+j}^{C_l}(\boldsymbol{\phi}), \quad (52)$$

where

$$p_{i,i+j}^{C_l}(\boldsymbol{\phi}) = \log c_{j,(j+i);1,\dots,(j-1)} + \frac{1}{a_j} \log p_j(d_j|\boldsymbol{\phi}) + \frac{1}{b_{j+i}} \log p_{j+i}(d_{j+i}|\boldsymbol{\phi}), \quad (53)$$

and

$$\begin{aligned} a_j &= N - 1, \\ b_{j+i} &= (N - 1 - l) \mathbb{1}_{j+i \leq l} + l. \end{aligned}$$

Let us now replace the full log-likelihood $\log(\mathbf{d}|\boldsymbol{\phi})$ in the definition of ELBO with the likelihood based on a truncated vine copula. This yields the l-truncated ELBO for the l-truncated C-vine

$$\mathcal{L}_{C_l}(\boldsymbol{\lambda}) = \mathbb{E}_q \left[\sum_{j=1}^l \sum_{i=1}^{N-j} p_{j,j+i}^{C_l}(\boldsymbol{\phi}) \right] - KL(q(\boldsymbol{\phi}|\boldsymbol{\lambda})||p(\boldsymbol{\phi})) \quad (54)$$

with its gradient

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{C_l}(\boldsymbol{\lambda}) = \sum_{j=1}^l \sum_{i=1}^{N-j} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda})(p_{j,j+i}^{C_l}(\boldsymbol{\phi})) \right] - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right].$$

Consequently, we get the following proposition that establishes the noisy unbiased estimate of $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{C_l}(\boldsymbol{\lambda})$.

Proposition 6 Let $\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda})$ be an estimate of the ELBO gradient $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{C_l}(\boldsymbol{\lambda})$ defined as

$$\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda}) = \frac{l(2N - (l + 1))}{2} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda})(p_{I_{C_l}(K)}^{C_l}(\boldsymbol{\phi})) \right] - \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right],$$

where $K \sim U(1, \dots, \frac{l(2N - (l + 1))}{2})$, and I_{C_l} is the bijection

$$I_{C_l} : \left\{1, \dots, \frac{l(2N - (l + 1))}{2}\right\} \rightarrow \{(j, j + i) : i \in \{1, \dots, N - j\} \text{ for } j \in \{1, \dots, l\}\},$$

then $\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda})$ is unbiased i.e., $\mathbb{E}(\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda})) = \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{C_l}(\boldsymbol{\lambda})$.

Algorithm 3 postulates the version of Algorithm 1 based on the truncated C-vine decomposition.

Algorithm 3: Variational calibration with truncated C-vine copulas.

Input: Data \mathbf{d} , mean and covariance functions for GPs, variational family $q(\boldsymbol{\phi}|\boldsymbol{\lambda})$,
truncation level l

1 $\lambda \leftarrow$ random initial value

2 $t \leftarrow 1$

3 **repeat**

4 **for** $s = 1$ to S **do**

5 $\boldsymbol{\phi}[s] \sim q(\boldsymbol{\phi}|\boldsymbol{\lambda})$

6 $K \leftarrow U(1, \dots, \frac{l(2N - (l + 1))}{2})$

7 $\rho \leftarrow t^{\text{th}}$ value of a Robbins-Monro sequence

8 $\boldsymbol{\lambda} \leftarrow$

$$\boldsymbol{\lambda} + \rho \frac{1}{S} \sum_{s=1}^S \left[\frac{l(2N - (l + 1))}{2} \nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}[s]|\boldsymbol{\lambda}) \times \left(p_{I_{C_l}(K)}^{C_l}(\boldsymbol{\phi}[s]) - \frac{2}{l(2N - (l + 1))} \log \frac{q(\boldsymbol{\phi}[s]|\boldsymbol{\lambda})}{p(\boldsymbol{\phi}[s])} \right) \right]$$

9 $t \leftarrow t + 1$

10 **until** change of $\boldsymbol{\lambda}$ is less than ϵ

A.1 Variance Reduction

Let us now consider the MC approximation of the gradient estimator $\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda})$. The j^{th} entry of the estimator with Rao-Blackwellization is

$$\frac{1}{S} \sum_{s=1}^S \left[\frac{l(2N - (l + 1))}{2} \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j[s] | \boldsymbol{\lambda}_j) (\tilde{p}_{(j)}(\phi[s]) - \frac{2}{l(2N - (l + 1))} \log \frac{q(\phi_j[s] | \boldsymbol{\lambda}_j)}{p(\phi_j[s])}) \right],$$

where $\tilde{p}_{(j)}(\boldsymbol{\phi})$ are here the components of $p_{I_{C_l}(K)}^{C_l}(\boldsymbol{\phi})$ that include ϕ_j .

We can again use the control variates to reduce the variance of MC approximation of the gradient estimator $\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda})$. In particular, we consider the following j^{th} entry of the Rao-Blackwellized MC approximation of the gradient estimator $\tilde{\mathcal{L}}_{C_l}(\boldsymbol{\lambda})$ with control variates

$$\tilde{\mathcal{L}}_{C_l}^{CV(j)}(\boldsymbol{\lambda}) = \sum_{s=1}^S \left[\frac{l(2N - (l + 1))}{2S} \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j[s] | \boldsymbol{\lambda}_j) (\tilde{p}_{(j)}(\phi[s]) - \frac{2(\log \frac{q(\phi_j[s] | \boldsymbol{\lambda}_j)}{p(\phi_j[s])} + \hat{a}_j^C)}{l(2N - (l + 1))}) \right],$$

where \hat{a}_j^C is the estimate of the optimal control variate scalar a^* based on S (or fewer) independent draws from the variational distribution. Namely,

$$\hat{a}_j^C = \frac{\widehat{\text{Cov}}_q(\xi^C(\boldsymbol{\phi}), \psi^C(\boldsymbol{\phi}))}{\widehat{\text{Var}}_q(\psi^C(\boldsymbol{\phi}))},$$

where

$$\xi^C(\boldsymbol{\phi}) = \frac{l(2N - (l + 1))}{2} \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j | \boldsymbol{\lambda}_j) \left(\tilde{p}_{(j)}(\boldsymbol{\phi}) - \frac{2 \log q(\phi_j | \boldsymbol{\lambda}_j)}{l(2N - (l + 1)) \log p(\phi_j)} \right)$$

and $\psi^C(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j | \boldsymbol{\lambda}_j)$.

As in the case of the D-vine, we now derive the ultimate Algorithm [4](#). Again, instead of taking the samples from $q(\boldsymbol{\phi} | \boldsymbol{\lambda})$ to approximate the gradient estimates, we will take samples from an overdispersed distribution $r(\boldsymbol{\phi} | \boldsymbol{\lambda}, \tau)$. Combining the Rao-Blackwellization, control variates, and importance sampling, we have the following j^{th} entry of the MC approximation

of the gradient estimator $\tilde{\mathcal{L}}_{C_l}(\lambda)$

$$\tilde{\mathcal{L}}_{C_l}^{OCV(j)}(\boldsymbol{\lambda}) = \sum_{s=1}^S \left[\frac{l(2N - (l + 1))}{2S} \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j[s] | \boldsymbol{\lambda}_j) (\tilde{p}_{(j)}(\phi[s]) - \frac{2(\log \frac{q(\phi_j[s] | \boldsymbol{\lambda}_j)}{p(\phi_j[s])} + \tilde{a}_j^C)}{l(2N - (l + 1))}) w(\phi_j[s]) \right],$$

where $\phi[s] \sim r(\phi | \boldsymbol{\lambda}, \tau)$ and $w(\phi[s]) = q(\phi[s] | \boldsymbol{\lambda}) / r(\phi[s] | \boldsymbol{\lambda}, \tau)$ with

$$\tilde{a}_j^C = \frac{\widehat{\text{Cov}}_q(\tilde{\xi}^C(\boldsymbol{\phi}), \tilde{\psi}^C(\boldsymbol{\phi}))}{\widehat{\text{Var}}_q(\tilde{\psi}^C(\boldsymbol{\phi}))},$$

where

$$\tilde{\xi}^C(\boldsymbol{\phi}) = \frac{l(2N - (l + 1))w(\phi_j)}{2} \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j | \boldsymbol{\lambda}_j) \left(\tilde{p}_{(j)}(\boldsymbol{\phi}) - \frac{2 \log \frac{q(\phi_j | \boldsymbol{\lambda}_j)}{p(\phi_j)}}{l(2N - (l + 1))} \right)$$

and $\tilde{\psi}^C(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j | \boldsymbol{\lambda}_j) w(\phi_j)$.

Algorithm 4: Variational calibration with truncated C-vine copulas II.

Input: Data \mathbf{d} , mean and covariance functions for GPs, variational family $q(\boldsymbol{\phi} | \boldsymbol{\lambda})$, dispersion parameter τ , and **truncation level** l

1 $\lambda \leftarrow$ random initial value

2 $t \leftarrow 1$

3 **repeat**

4 **for** $s = 1$ to S **do**

5 $\phi[s] \sim r(\phi | \boldsymbol{\lambda}, \tau)$ // Random sample from r

6 $K \leftarrow U(1, \dots, \frac{l(2N - (l + 1))}{2})$

7 $\rho \leftarrow t^{\text{th}}$ value of a Robbins-Monro sequence

8 $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \rho \sum_{s=1}^S \left[\frac{l(2N - (l + 1))}{2S} \nabla_{\boldsymbol{\lambda}_j} \log q(\phi_j[s] | \boldsymbol{\lambda}_j) \times \left(\tilde{p}_{(j)}(\phi[s]) - \frac{2(\log \frac{q(\phi_j[s] | \boldsymbol{\lambda}_j)}{p(\phi_j[s])} + \tilde{a}_j^C)}{l(2N - (l + 1))}) w(\phi_j[s]) \right] \right]$

9 $t \leftarrow t + 1$

10 **until** change of $\boldsymbol{\lambda}$ is less than ϵ

Appendix B Proofs

Proof of Proposition 1. Since $P(K = k) = \frac{2}{N(N-1)}$, we have directly from the definition of expectation

$$\begin{aligned} \mathbb{E}(\tilde{\mathcal{L}}_D(\boldsymbol{\lambda})) &= \frac{N(N-1)}{2} \sum_{k=1}^{\frac{N(N-1)}{2}} \frac{2}{N(N-1)} \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda})(p_{I_D(k)}^D(\boldsymbol{\phi})) \right] \\ &- \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\phi}|\boldsymbol{\lambda}) \log \frac{q(\boldsymbol{\phi}|\boldsymbol{\lambda})}{p(\boldsymbol{\phi})} \right] = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}). \end{aligned}$$

The final equality is the consequence of the uniqueness of the pairs of variables in the conditioned sets of the copula density $c_{i,(i+j);(i+1),\dots,(i+j-1)}$, and that $\frac{N(N-1)}{2}$ is the number of unordered pairs of N variables.

Proof of Proposition 2. It is sufficient to show that for $l \in \{1, \dots, N-1\}$ the following equality holds:

$$\sum_{j=1}^l \sum_{i=1}^{N-j} \left[\frac{1}{a_i} \log p_i(d_i|\boldsymbol{\phi}) + \frac{1}{b_{i+j}} \log p_{i+j}(d_{i+j}|\boldsymbol{\phi}) \right] = \sum_{k=1}^N \log p(\mathbf{d}_k|\boldsymbol{\phi}), \quad (55)$$

where

$$\begin{aligned} a_i &= 2l - \left[(l+1-i)\mathbb{1}_{i \leq l} + (l-N+i)\mathbb{1}_{i > N-l} \right], \\ b_{i+j} &= 2l - \left[(l+1-j-i)\mathbb{1}_{i+j \leq l} + (l-N+j+i)\mathbb{1}_{i+j > N-l} \right]. \end{aligned}$$

To show this, let us consider the summation

$$\begin{aligned} &\sum_{j=1}^l \sum_{i=1}^{N-j} \left[\log p_i(d_i|\boldsymbol{\phi}) + \log p_{i+j}(d_{i+j}|\boldsymbol{\phi}) \right] \\ &= \sum_{j=1}^l \left[(\log p_1(d_1|\boldsymbol{\phi}) + \log p_{1+j}(d_{1+j}|\boldsymbol{\phi})) + \dots + (\log p_{N-j}(d_{N-j}|\boldsymbol{\phi}) + \log p_N(d_N|\boldsymbol{\phi})) \right]. \end{aligned}$$

For $l = 1$, we get

$$\begin{aligned} & \sum_{j=1}^l \sum_{i=1}^{N-j} \left[\log p_i(d_i|\boldsymbol{\phi}) + \log p_{i+j}(d_{i+j}|\boldsymbol{\phi}) \right] \\ &= (\log p_1(d_1|\boldsymbol{\phi}) + \log p_2(d_2|\boldsymbol{\phi})) + \cdots + (\log p_{N-1}(d_{N-1}|\boldsymbol{\phi}) + \log p_N(d_N|\boldsymbol{\phi})), \end{aligned}$$

and for $l \geq 2$

$$\begin{aligned} & \sum_{j=1}^l \sum_{i=1}^{N-j} \left[\log p_i(d_i|\boldsymbol{\phi}) + \log p_{i+j}(d_{i+j}|\boldsymbol{\phi}) \right] \\ &= \left[(\log p_1(d_1|\boldsymbol{\phi}) + \log p_2(d_2|\boldsymbol{\phi})) + \cdots + (\log p_{N-1}(d_{N-1}|\boldsymbol{\phi}) + \log p_N(d_N|\boldsymbol{\phi})) \right] + \cdots \\ & \quad + \left[(\log p_1(d_1|\boldsymbol{\phi}) + \log p_{1+l}(d_{1+l}|\boldsymbol{\phi})) + \cdots + (\log p_{N-l}(d_{N-l}|\boldsymbol{\phi}) + \log p_N(d_N|\boldsymbol{\phi})) \right]. \end{aligned}$$

Note that in the case of $l = N - 1$, the last summation consists of only one element $\log p_1(d_1|\boldsymbol{\phi}) + \log p_{1+l}(d_{1+l}|\boldsymbol{\phi})$. By careful examination of the two cases above, we get the following results. For $2l \leq N$:

$$\begin{aligned} & \sum_{j=1}^l \sum_{i=1}^{N-j} \left[\log p_i(d_i|\boldsymbol{\phi}) + \log p_{i+j}(d_{i+j}|\boldsymbol{\phi}) \right] \\ &= \sum_{k=1}^l (l+k-1) \log p_k(d_k|\boldsymbol{\phi}) + \sum_{k=l+1}^{N-l} 2l \log p_k(d_k|\boldsymbol{\phi}) + \sum_{k=N-l+1}^N (N-i+l) \log p_k(d_k|\boldsymbol{\phi}), \end{aligned}$$

where the middle term disappears in the case $2l = N$, and for $2l > N$:

$$\begin{aligned} & \sum_{j=1}^l \sum_{i=1}^{N-j} \left[\log p_i(d_i|\boldsymbol{\phi}) + \log p_{i+j}(d_{i+j}|\boldsymbol{\phi}) \right] \\ &= \sum_{k=1}^{N-l} (l+k-1) \log p_k(d_k|\boldsymbol{\phi}) + \sum_{k=N-l+1}^l (N-1) \log p_k(d_k|\boldsymbol{\phi}) + \sum_{k=l+1}^N (N-i+l) \log p_k(d_k|\boldsymbol{\phi}). \end{aligned}$$

If we now check that a_i equals to the factors in front of the log-likelihoods in the two cases above, the proof of Proposition [2](#) is complete. Note that once we check the equality for a_i ,

the same directly translates to b_{i+j} since b_{i+j} is a_i with indices set to $i + j$ instead of i .
Indeed, for $2l \leq N$

$$a_i = \begin{cases} l + i - 1 & i \leq l \\ 2l & l < i \leq N - l, \\ N - i + l & N - l < i \end{cases}$$

and for $2l > N$

$$a_i = \begin{cases} l + i - 1 & i \leq N - l \\ N - 1 & N - l < i \leq l. \\ N - i + l & l < i \end{cases}$$

Proof of Proposition 3. By the construction of R-vine (see [Cooke and Kurowicka \(2006\)](#)), each tree \mathcal{T}_i , for $i = 1, \dots, N - 1$ has exactly $N - i$ edges (these are the unique conditioned variable pairs). For any R-vine truncated at level $l \in \{1, \dots, N - 1\}$, we get the number of edges to be

$$\sum_{i=1}^l (N - i) = lN - \frac{l(l+1)}{2} = \frac{l(2N - (l+1))}{2}$$

The rest of the proof is identical with that of Proposition 1 due to the uniqueness of the conditioned variable pairs in the copula density $c_{i,(i+j);(i+1),\dots,(i+j-1)}$, but in this case $P(K = k) = \frac{2}{l(2N-(l+1))}$.

Proof of Proposition 4. The proof is identical with that of Proposition 1 since each conditioned pair in the copula density $c_{j,(j+i);1,\dots,(j-1)}$ is unique as well.

Proof of Proposition 5. It is sufficient to show that for $l \in \{1, \dots, N - 1\}$ the following equality holds:

$$\sum_{j=1}^l \sum_{i=1}^{N-j} \left[\frac{1}{a_j} \log p_j(d_j | \phi) + \frac{1}{b_{j+i}} \log p_{j+i}(d_{j+i} | \phi) \right] = \sum_{k=1}^N \log p(\mathbf{d}_k | \phi), \quad (56)$$

where

$$a_j = N - 1,$$

$$b_{j+i} = (N - 1 - l)\mathbb{1}_{j+i \leq l} + l.$$

To show this, let us consider the following summation

$$\begin{aligned} & \sum_{j=1}^l \sum_{i=1}^{N-j} \left[\log p_j(d_j | \boldsymbol{\phi}) + \log p_{j+i}(d_{j+i} | \boldsymbol{\phi}) \right] = \\ & \sum_{j=1}^l \left[(N - j) \log p_j(d_j | \boldsymbol{\phi}) + \sum_{i=1}^{N-j} \log p_{j+i}(d_{j+i} | \boldsymbol{\phi}) \right] = \\ & \sum_{j=1}^l (N - j) \log p_j(d_j | \boldsymbol{\phi}) + \sum_{j=1}^l \left[\log p_{j+1}(d_{j+1} | \boldsymbol{\phi}) + \dots + \log p_N(d_N | \boldsymbol{\phi}) \right]. \end{aligned}$$

Now, for $l = 1$, we have

$$\sum_{j=1}^1 \left[\log p_{j+1}(d_{j+1} | \boldsymbol{\phi}) + \dots + \log p_N(d_N | \boldsymbol{\phi}) \right] = \log p_2(d_2 | \boldsymbol{\phi}) + \dots + \log p_N(d_N | \boldsymbol{\phi}).$$

For $l \geq 2$, we have

$$\begin{aligned} & \sum_{j=1}^l \left[\log p_{j+1}(d_{j+1} | \boldsymbol{\phi}) + \dots + \log p_N(d_N | \boldsymbol{\phi}) \right] = \\ & \left[\log p_2(d_2 | \boldsymbol{\phi}) + \dots + \log p_N(d_N | \boldsymbol{\phi}) \right] + \dots + \left[\log p_{l+1}(d_{l+1} | \boldsymbol{\phi}) + \dots + \log p_N(d_N | \boldsymbol{\phi}) \right]. \end{aligned}$$

Therefore we can rewrite

$$\sum_{j=1}^l \left[\log p_{j+1}(d_{j+1} | \boldsymbol{\phi}) + \dots + \log p_N(d_N | \boldsymbol{\phi}) \right] = \sum_{j=1}^l (j - 1) \log p_j(d_j | \boldsymbol{\phi}) + \sum_{j=l+1}^N l \log p_j(d_j | \boldsymbol{\phi})$$

Overall,

$$\begin{aligned}
& \sum_{j=1}^l \sum_{i=1}^{N-j} \left[\log p_j(d_j|\phi) + \log p_{j+i}(d_{j+i}|\phi) \right] = \\
& \sum_{j=1}^l (N-j) \log p_j(d_j|\phi) + \sum_{j=1}^l (j-1) \log p_j(d_j|\phi) + \sum_{j=l+1}^N l \log p_j(d_j|\phi) = \\
& \sum_{k=1}^l (N-1) \log p_k(d_k|\phi) + \sum_{k=l+1}^N l \log p_k(d_k|\phi).
\end{aligned}$$

Since $j \in \{1, \dots, l\}$ and

$$b_{j+i} = \begin{cases} N-1 & j+i \leq l \\ l & j+i > l \end{cases},$$

the equality [56](#) holds.

Proof of Proposition [6](#). The proof is identical with that of Proposition [3](#) since each conditioned pair in the copula density $c_{j,(j+i);1,\dots,(j-1)}$ is unique, and a C-vine is a special case of R-vine.

Appendix C Simulation: Memory Profile

Here we present the memory profiles for the MH, the NUTS, and the Algorithm [2](#) under the simulation scenario studied in Chapter 5. These were recorded during a one hour period of running the algorithms. The MH algorithm and the NUTS were implemented in Python 3.0 using the PyMC3 module version 3.5. The memory profiles were measured using the memory-profiler module version 0.55.0 in Python 3.0. The VC was also implemented in Python 3.0. The code was run on the high performance computing cluster at the Institute for Cyber-Enabled Research at Michigan State University.

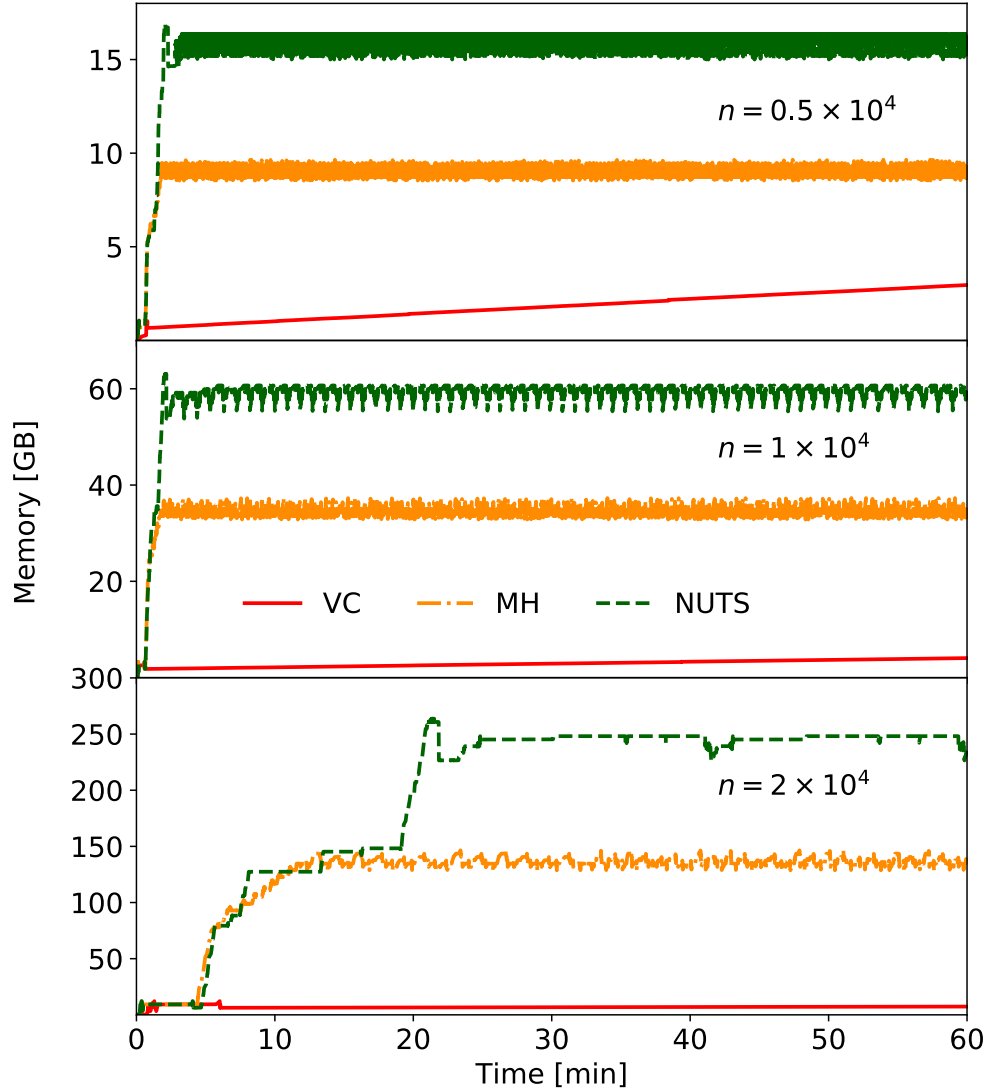


Figure 6: Recorded memory profiles of the Algorithm [2](#), the MH algorithm, and the NUTS for the duration of 1h under the simulation scenario with $n = 0.5 \times 10^4$, $n = 1 \times 10^4$, and $n = 2 \times 10^4$.

Appendix D Application: Liquid Drop Model (LDM)

D.1 GP specifications

In the case of the LDM $E_B(Z, N)$, we consider the GP prior with mean zero and covariance function

$$\eta_f \times \exp\left(-\frac{\|Z - Z'\|^2}{2\nu_Z^2} - \frac{\|N - N'\|^2}{2\nu_N^2} - \frac{\|\theta_{\text{vol}} - \theta'_{\text{vol}}\|^2}{2\nu_1^2} - \frac{\|\theta_{\text{surf}} - \theta'_{\text{surf}}\|^2}{2\nu_2^2} - \frac{\|\theta_{\text{sym}} - \theta'_{\text{sym}}\|^2}{2\nu_3^2} - \frac{\|\theta_C - \theta'_C\|^2}{2\nu_4^2}\right).$$

Similarly, we consider the GP process prior for the systematic discrepancy $\delta(Z, N)$ with mean zero and covariance function

$$\eta_\delta \times \exp\left(-\frac{\|Z - Z'\|^2}{2l_Z^2} - \frac{\|N - N'\|^2}{2l_N^2}\right).$$

D.2 Experimental design

Kennedy and O'Hagan (2001) recommend to select the calibration inputs for the model runs so that any plausible value θ of the true calibration parameter is covered. In this context, we consider the space of calibration parameters to be centered at the values of least squares estimates $\hat{\theta}_{L_2}$ and broad enough to contain the majority of values provided by the nuclear physics literature (Weizsäcker, 1935; Bethe and Bacher, 1936; Myers and Swiatecki, 1966; Kirson, 2008; Benzaid et al., 2020). Table 4 gives the lower and upper bounds for the parameter space so that Lower bound = $\hat{\theta}_{L_2} - 15 \times SE(\hat{\theta}_{L_2})$ and Upper bound = $\hat{\theta}_{L_2} + 15 \times SE(\hat{\theta}_{L_2})$. Here $SE(\hat{\theta}_{L_2})$ is given by the standard linear regression theory.

Parameter	Lower bound	Upper bound
θ_{vol}	15.008	15.829
θ_{surf}	15.628	18.193
θ_{sym}	21.435	23.505
θ_C	0.665	0.72

Table 4: The space of calibration parameters used for generating the outputs of semi-empirical mass formula (46).

D.3 Prior distributions

First, we consider the independent Gaussian distributions centered at the LS estimates $\hat{\boldsymbol{\theta}}_{L_2}$ (in Table 3) with standard deviations $7.5 \times SE(\hat{\boldsymbol{\theta}}_{L_2})$ so that the calibration parameters used for generating the model runs are covered roughly within two standard deviations of the priors. Namely,

$$\theta_{\text{vol}} \sim \mathcal{N}(15.42, 0.203),$$

$$\theta_{\text{surf}} \sim \mathcal{N}(16.91, 0.645),$$

$$\theta_{\text{sym}} \sim \mathcal{N}(22.47, 0.525),$$

$$\theta_{\text{C}} \sim \mathcal{N}(0.69, 0.015).$$

The prior distributions for hyperparameters of the GPs were selected as Gamma(α, β) with the shape parameter α and scale parameter β , so that they represent a vague knowledge about the scale of these parameters given by the literature on nuclear mass models (Weizsäcker, 1935; Bethe and Bacher, 1936; Myers and Swiatecki, 1966; Fayans, 1998; Kirson, 2008; McDonnell et al., 2015; Kortelainen et al., 2010, 2012, 2014; Benzaid et al., 2020; Kejzlar et al., 2020). In particular, the error scale σ is in the majority of nuclear applications within units of MeV, therefore we set

$$\sigma \sim \text{Gamma}(2, 1),$$

with the scale of the systematic error being

$$\eta_{\delta} \sim \text{Gamma}(10, 1),$$

to allow for this quantity to range between the units and tens of MeV. It is also reasonable to assume that the mass of a given nucleus is correlated mostly with its neighbours on the nuclear chart. We express this notion through these reasonably wide prior distributions

$$\begin{aligned}
l_Z &\sim \text{Gamma}(10, 1), \\
l_N &\sim \text{Gamma}(10, 1), \\
\nu_Z &\sim \text{Gamma}(10, 1), \\
\nu_N &\sim \text{Gamma}(10, 1), \\
\nu_i &\sim \text{Gamma}(10, 1), \quad i = 1, 2, 3, 4.
\end{aligned}$$

Finally, the majority of the masses in the training dataset of 2000 experimental binding energies fall into the range of [1000, 2000] MeV (1165 of masses precisely). We consider the following prior distribution for the parameter η_f to reflect on the scale of the experimental binding energies:

$$\eta_f \sim \text{Gamma}(110, 10).$$

References

- Aicher C, Ma YA, Foti NJ, Fox EB (2019) Stochastic gradient mcmc for state space models. SIAM Journal on Mathematics of Data Science 1(3):555–587, DOI 10.1137/18M1214780, URL <https://doi.org/10.1137/18M1214780>, <https://doi.org/10.1137/18M1214780>
- Audi G, Wapstra A, Thibault C (2003) The AME2003 atomic mass evaluation: (ii). tables, graphs and references. Nuclear Physics A 729:337–676, URL <http://www.sciencedirect.com/science/article/pii/S0375947403018098>
- Bauer M, van der Wilk M, Rasmussen CE (2016) Understanding probabilistic sparse gaussian process approximations. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, NeurIPS’16, pp 1533–1541, URL <http://dl.acm.org/citation.cfm?id=3157096.3157268>

- Bayarri MJ, Berger JO, Paulo R, Sacks J, Cafeo JA, Cavendish J, Lin CH, Tu J (2007) A framework for validation of computer models. *Technometrics* 49:138–154, URL <https://doi.org/10.1198/004017007000000092>
- Bedford T, Cooke RM (2002) Vines—a new graphical model for dependent random variables. *The Annals of Statistics* 30(4):1031–1068
- Benzaid D, Bentriddi S, Kerraci A, Amrani N (2020) Bethe–Weizsäcker semiempirical mass formula coefficients 2019 update based on AME2016. *Nucl Sci Tech* 31:9, DOI 10.1007/s41365-019-0718-8, URL <https://doi.org/10.1007/s41365-019-0718-8>
- Bertsch GF, Bingham D (2017) Estimating parameter uncertainty in binding-energy models by the frequency-domain bootstrap. *Physical Review Letters* 119:252501, URL <https://link.aps.org/doi/10.1103/PhysRevLett.119.252501>
- Bertsch GF, Sabbey B, Uusnäkki M (2005) Fitting theories of nuclear binding energies. *Phys Rev C* 71:054311, DOI 10.1103/PhysRevC.71.054311, URL <https://link.aps.org/doi/10.1103/PhysRevC.71.054311>
- Bethe HA, Bacher RF (1936) Nuclear physics a. stationary states of nuclei. *Rev Mod Phys* 8:82–229, DOI 10.1103/RevModPhys.8.82, URL <https://link.aps.org/doi/10.1103/RevModPhys.8.82>
- Bottou L, Le Cun Y, Bengio Y (1997) Global training of document processing systems using graph transformer networks. In: *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp 489–493, URL <http://leon.bottou.org/papers/bottou-97>
- Brechmann EC, Joe H (2015) Truncation of vine copulas using fit indices. *Journal of Multivariate Analysis* 138:19–33
- Brechmann EC, Czado C, Aas K (2012) Truncated regular vines in high dimensions with application to financial data. *The Canadian Journal of Statistics* 40(1):68–85

- Casella G, Robert CP (1996) Rao-blackwellisation of sampling schemes. *Biometrika* 83(1):81–94, URL <http://www.jstor.org/stable/2337434>
- Chib S, Greenberg E (1995) Understanding the Metropolis-Hastings algorithm. *The American Statistician* 49:327–335, URL <http://www.jstor.org/stable/2684568>
- Cooke R, Kurowicka D (2006) *Uncertainty Analysis With High Dimensional Dependence Modelling*. Wiley
- Deng W, Zhang X, Liang F, Lin G (2019) An adaptive empirical bayesian method for sparse deep learning. In: *Advances in Neural Information Processing Systems* 32, Curran Associates, Inc., pp 5563–5573, URL <http://papers.nips.cc/paper/8794-an-adaptive-empirical-bayesian-method-for-sparse-deep-learning.pdf>
- Dissmann J, Brechmann E, Czado C, Kurowicka D (2013) Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis* 59:52–69, DOI <https://doi.org/10.1016/j.csda.2012.08.010>, URL <http://www.sciencedirect.com/science/article/pii/S0167947312003131>
- Dobaczewski J, Nazarewicz W, Reinhard PG (2014) Error estimates of theoretical models: a guide. *Journal of Physics G: Nuclear and Particle Physics* 41(7):074001, DOI <https://doi.org/10.1088/0954-3899/41/7/074001>, URL <https://doi.org/10.1088/0954-3899/41/7/074001>
- Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research* 12:2121–2159, URL <http://dl.acm.org/citation.cfm?id=1953048.2021068>
- Fayans SA (1998) Towards a universal nuclear density functional. *Journal of Experimental and Theoretical Physics Letters* 68(3):169–174, DOI [10.1134/1.567841](https://doi.org/10.1134/1.567841)

- Goldberger A (1966) *Econometric theory*. Wiley publications in statistics, J. Wiley, URL <https://books.google.com/books?id=QN6xAAAAIAAJ>
- Gu M, Wang L (2018) Scaled Gaussian stochastic process for computer model calibration and prediction. *SIAM/ASA Journal on Uncertainty Quantification* 6(4):1555–1583, DOI 10.1137/17M1159890
- Higdon D, Kennedy M, Cavendish JC, Cafo JA, Ryne RD (2005) Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing* 26:448–466, URL <https://doi.org/10.1137/S1064827503426693>
- Higdon D, Gattiker J, Williams B, Rightley M (2008) Computer model calibration using high-dimensional output. *Journal of the American Statistical Association* 103:570–583, URL <http://www.jstor.org/stable/27640080>
- Higdon D, McDonnell JD, Schunck N, Sarich J, Wild SM (2015) A Bayesian approach for parameter estimation and prediction using a computationally intensive model. *Journal of Physics G: Nuclear and Particle Physics* 42(3):034009, DOI 10.1088/0954-3899/42/3/034009
- Hoffman M, Blei D (2015) Stochastic structured variational inference. In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, PMLR, San Diego, CA, vol 38, pp 361–369, URL <http://proceedings.mlr.press/v38/hoffman15.html>
- Hoffman MD, Blei DM, Wang C, Paisley J (2013) Stochastic variational inference. *Journal of Machine Learning Research* 14:1303–1347, URL <http://jmlr.org/papers/v14/hoffman13a.html>
- Homan MD, Gelman A (2014) The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* 15:1351–1381, URL <http://dl.acm.org/citation.cfm?id=2627435.2638586>

- Ireland DG, Nazarewicz W (2015) Enhancing the interaction between nuclear experiment and theory through information and statistics. *Journal of Physics G: Nuclear and Particle Physics* 42(3):030301, DOI 10.1088/0954-3899/42/3/030301
- Johnston J (1976) *Econometric Methods*. McGraw-Hill, URL <https://books.google.com/books?id=TLt0tAEACAAJ>
- Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. *Machine Learning* 37:183–233
- Kejzlar V, Neufcourt L, Nazarewicz W, Reinhard PG (2020) Statistical aspects of nuclear mass models. *Journal of Physics G: Nuclear and Particle Physics* 47(9):094001, DOI 10.1088/1361-6471/ab907c, URL <https://doi.org/10.1088%2F1361-6471%2Fab907c>
- Kejzlar V, Son M, Bhattacharya S, Maiti T (2021) Statistical aspects of nuclear mass models. *Journal of Physics G: Nuclear and Particle Physics* 47(9):094001, DOI 10.1007/s11222-021-10024-8, URL <https://doi.org/10.1088%2F1361-6471%2Fab907c>
- Kennedy MC, O’Hagan A (2001) Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63:425–464, DOI 10.1111/1467-9868.00294, URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00294>
- King GB, Lovell AE, Neufcourt L, Nunes FM (2019) Direct comparison between Bayesian and frequentist uncertainty quantification for nuclear reactions. *Physical Review Letters* 122:232502
- Kirson MW (2008) Mutual influence of terms in a semi-empirical mass formula. *Nucl Phys A* 798(1):29 – 60, DOI 10.1016/j.nuclphysa.2007.10.011, URL <http://www.sciencedirect.com/science/article/pii/S0375947407007531>

- Kortelainen M, Lesinski T, Moré JJ, Nazarewicz W, Sarich J, Schunck N, Stoitsov MV, Wild SM (2010) Nuclear energy density optimization. *Physical Review C* 82(2):024313, DOI 10.1103/PhysRevC.82.024313
- Kortelainen M, McDonnell J, Nazarewicz W, Reinhard PG, Sarich J, Schunck N, Stoitsov MV, Wild SM (2012) Nuclear energy density optimization: large deformations. *Physical Review C* 85:024304, DOI 10.1103/PhysRevC.85.024304
- Kortelainen M, McDonnell J, Nazarewicz W, Olsen E, Reinhard PG, Sarich J, Schunck N, Wild SM, Davesne D, Erler J, Pastore A (2014) Nuclear energy density optimization: Shell structure. *Phys Rev C* 89:054314, DOI 10.1103/PhysRevC.89.054314
- Krane K (1987) *Introductory Nuclear Physics*. Wiley, URL <https://books.google.com/books?id=ConwAAAAAAAJ>
- Ma YA, Chen T, Fox E (2015) A complete recipe for stochastic gradient mcmc. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) *Advances in Neural Information Processing Systems* 28, Curran Associates, Inc., pp 2917–2925, URL <http://papers.nips.cc/paper/5891-a-complete-recipe-for-stochastic-gradient-mcmc.pdf>
- McDonnell JD, Schunck N, Higdon D, Sarich J, Wild SM, Nazarewicz W (2015) Uncertainty quantification for nuclear density functional theory and information content of new measurements. *Physical Review Letters* 114(12):122501, DOI 10.1103/PhysRevLett.114.122501
- Morris MD, Mitchell TJ (1995) Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* 43(3):381 – 402, DOI [https://doi.org/10.1016/0378-3758\(94\)00035-T](https://doi.org/10.1016/0378-3758(94)00035-T), URL <http://www.sciencedirect.com/science/article/pii/S037837589400035T>
- Myers WD, Swiatecki WJ (1966) Nuclear masses and deformations. *Nucl Phys* 81(2):1 – 60,

- DOI 10.1016/S0029-5582(66)80001-9, URL <http://www.sciencedirect.com/science/article/pii/S0029558266800019>
- Neiswanger W, Wang C, Xing EP (2014) Asymptotically exact, embarrassingly parallel mcmc. In: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, AUAI Press, Arlington, VA, UAI'14, pp 623–632, URL <http://dl.acm.org/citation.cfm?id=3020751.3020816>
- Peterson C, Anderson JR (1987) A mean field theory learning algorithm for neural networks. *Complex Systems* 1:995–1019
- Plumlee M (2017) Bayesian calibration of inexact computer models. *Journal of the American Statistical Association* 112:1274–1285, DOI 10.1080/01621459.2016.1211016
- Plumlee M (2019) Computer model calibration with confidence and consistency. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 81(3):519–545, DOI 10.1111/rssb.12314, URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12314>, <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssb.12314>
- Plumlee M, Joseph VR, Yang H (2016) Calibrating functional parameters in the ion channel models of cardiac cells. *Journal of the American Statistical Association* 111:500–509
- Pollard D, Chang W, Haran M, Applegate P, DeConto R (2016) Large ensemble modeling of the last deglacial retreat of the West Antarctic Ice Sheet: comparison of simple and advanced statistical techniques. *Geoscientific Model Development* 9(5):1697–1723
- Quiñonero-Candela J, Rasmussen CE (2005) A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* pp 1939–1959, URL <http://dl.acm.org/citation.cfm?id=1046920.1194909>
- Ranganath R, Gerrish S, Blei D (2014) Black box variational inference. In: Proceedings of

- the Seventeenth International Conference on Artificial Intelligence and Statistics, PMLR, Proceedings of Machine Learning Research, vol 33, pp 814–822
- Ranganath R, Tran D, Blei DM (2016) Hierarchical variational models. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning – Volume 48, JMLR, ICML'16, pp 2568–2577
- Reinhard PG, Bender M, Nazarewicz W, Vertse T (2006) From finite nuclei to the nuclear liquid drop: Leptodermous expansion based on self-consistent mean-field theory. Phys Rev C 73:014309, DOI 10.1103/PhysRevC.73.014309, URL <https://link.aps.org/doi/10.1103/PhysRevC.73.014309>
- Robbins H, Monro S (1951) A stochastic approximation method. Annals of Mathematical Statistics 22(3):400–407
- Robert C, Casella G (2005) Monte Carlo Statistical Methods. Springer Texts in Statistics, Springer New York, URL <https://books.google.com/books?id=HfhGAXn5GugC>
- Ross SM (2006) Simulation, 4th edn. Academic Press, Inc., Orlando, FL
- Ruiz FJR, Titsias MK, Blei DM (2016) Overdispersed black-box variational inference. In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, AUAI Press, Arlington, Virginia, USA, UAI'16, p 647–656
- Sexton DMH, Murphy JM, Collins M, Webb MJ (2012) Multivariate probabilistic projections using imperfect climate models Part i: outline of methodology. Climate Dynamics 38(11):2513–2542
- Sklar A (1959) Fonctions de répartition à n dimensions et leurs marges. Publications de l'Institut de Statistique de l'Université de Paris 8:229–231
- Smith MS, Loaiza-Maya R, Nott DJ (2020) High-dimensional copula variational approximation through transformation. Journal of Computational and Graphical Statistics 0(ja):1–

- 35, DOI 10.1080/10618600.2020.1740097, URL <https://doi.org/10.1080/10618600.2020.1740097>, <https://doi.org/10.1080/10618600.2020.1740097>
- Tieleman T, Hinton G (2012) Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning
- Titsias M (2009) Variational learning of inducing variables in sparse Gaussian processes. In: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, PMLR, vol 5, pp 567–574
- Tran D, Blei DM, Airoldi EM (2015) Copula variational inference. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, MIT Press, Cambridge, MA, NeurIPS'15, pp 3564–3572, URL <http://dl.acm.org/citation.cfm?id=2969442.2969637>
- Tran D, Ranganath R, Blei DM (2017) Hierarchical implicit models and likelihood-free variational inference. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NeurIPS'17, pp 5529–5539, URL <http://dl.acm.org/citation.cfm?id=3295222.3295304>
- Wainwright MJ, Jordan MI (2008) Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning 1(1–2):1–305, DOI 10.1561/2200000001, URL <http://dx.doi.org/10.1561/2200000001>
- Wang Y, Blei DM (2018) Frequentist consistency of variational Bayes. Journal of the American Statistical Association 0(0):1–15
- Weizsäcker CFv (1935) Zur theorie der kernmassen. Z Phys 96(7):431–458, DOI 10.1007/BF01337700, URL <https://doi.org/10.1007/BF01337700>
- Williams B, Higdon D, Gattiker J, Moore L, McKay M, Keller-McNulty S (2006) Com-

binning experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis* 1(4):765–792

Yuan C (2016) Uncertainty decomposition method and its application to the liquid drop model. *Phys Rev C* 93:034310, DOI 10.1103/PhysRevC.93.034310, URL <https://link.aps.org/doi/10.1103/PhysRevC.93.034310>

Zeiler MD (2012) Adadelta: An adaptive learning rate method. ArXiv 1212.5701

Zhang L, Jiang Z, Choi J, Lim CY, Maiti T, Baek S (2019) Patient-specific prediction of abdominal aortic aneurysm expansion using Bayesian calibration. *IEE Journal of Biomedical and Health Informatics* URL [DOI:10.1109/JBHI.2019.2896034](https://doi.org/10.1109/JBHI.2019.2896034)