



Scalable Stochastic Parametric Verification with Stochastic Variational Smoothed Model Checking

This is the peer reviewed (post-print) version of the following article:

Original

Scalable Stochastic Parametric Verification with Stochastic Variational Smoothed Model Checking / Bortolussi, L., Cairoli, F., Carbone, G., Pulcini, P.. - 14245 LNCS:(2023), pp. 45-65. (23rd International Conference on Runtime Verification Thessaloniki, Greece October 3 - 6, 2023) [10.1007/978-3-031-44267-4_3].

Availability:

This version is available at: 11368/3065900 since: 2023-12-04T14:39:48Z

Publisher:

Springer, Cham

Published

DOI:10.1007/978-3-031-44267-4_3

Terms of use:

Some rights reserved. More details in the PostPrint rights section below.

License:

Copyright autore

Publisher copyright

--

(Article begins on next page)

Scalable Stochastic Parametric Verification with Stochastic Variational Smoothed Model Checking

Luca Bortolussi¹, Francesca Cairoli¹, Ginevra Carbone¹, and Paolo Pulcini¹

Department of Mathematics and Geoscience, University of Trieste, Italy

Abstract. Parametric verification of linear temporal properties for stochastic models requires to compute the satisfaction probability of a certain property as a function of the parameters of the model. Smoothed model checking (smMC) [8] infers the satisfaction function over the entire parameter space from a limited set of observations obtained via simulation. As observations are costly and noisy, smMC leverages the power of Bayesian learning based on Gaussian Processes (GP), providing accurate reconstructions with statistically sound quantification of the uncertainty. In this paper we propose Stochastic Variational Smoothed Model Checking (SV-smMC), which exploits stochastic variational inference (SVI) to approximate the posterior distribution of the smMC problem. The strength and flexibility of SVI, a stochastic gradient-based optimization making inference easily parallelizable and enabling GPU acceleration, make SV-smMC applicable both to Gaussian Processes (GP) and Bayesian Neural Networks (BNN). SV-smMC extends the smMC framework by greatly improving scalability to higher dimensionality of parameter spaces and larger training datasets, thus overcoming the well-known limits of GP.

1 Introduction

Parametric verification of logical properties aims at providing meaningful insights into the behaviour of a system, checking whether its evolution satisfies or not a certain requirement while varying some parameters of the system’s model. The requirement is typically expressed as a temporal logic formula. Stochastic systems, however, require the use of probabilistic model checking (PMC) techniques as the satisfaction of a property is itself a stochastic quantity, facing significant scalability issues. To ameliorate such problems, statistical model checking (SMC) uses statistical tools to estimate the satisfaction probability of logical properties from trajectories sampled from the stochastic model. These estimates are enriched with probabilistic bounds of the estimation errors. If the number of sampled trajectories is sufficiently large, the satisfaction probability, estimated as the average of satisfaction on individual runs, will converge to the true probability. However, if the parameters of the stochastic model vary, the dynamics of the system will also vary. Therefore, SMC has to be performed from scratch for each set of parameter values, making SMC computationally unfeasible for stochastic parametric verification at runtime. This includes, for instance, estimating a satisfaction probability from the current state of the system, considering the current

state as the varying parameter, or estimating the effect on future behaviour of tuning some control parameter. In other words, we can easily express runtime verification (RV) of a stochastic system, whose goal is to monitor how the future satisfaction of a given requirement behaves w.r.t. the current state of the system, as a parametric verification problem. The current state of the system, which changes over time, can be represented as a varying parameter. Therefore, RV inherits the computational unfeasibility of stochastic parametric verification.

Population continuous-time Markov chains (CTMC) offer a very expressive formalism to describe the dynamical behaviour of stochastic processes. Moreover, the satisfaction probability of a signal temporal logic (STL) requirement over parametric stochastic models, in particular CTMCs, has been proved to be a smooth function of the parameters of the model [8]. This result enables the use of machine learning techniques to infer an approximation of this function from a limited pool of observations. Observations, computed via SMC for a small number of parameter values, are noisy and may be computationally demanding to obtain. This calls for Bayesian approaches, where predictions are efficiently computed and enriched with a probabilistic quantification of the predictive uncertainty. In this regard, in [8] the authors present smoothed model checking (smMC), a fully Bayesian solution based on Gaussian Processes (GP). Since the observation process is non-Gaussian, outputs are in fact realizations of a Bernoulli distribution, exact GP inference is unfeasible. The authors thus resort to the Expectation Propagation (EP) algorithm to approximate the posterior inference. Unfortunately, the cost of EP is cubic in the number of observations used to train the GP, making smMC applicable only to models with a low dimensional parameter space as they require a limited number of training observations. In [26], this scalability problem is tackled by using a sparse variational approach to GP inference that builds on [30]. Variational inference (VI) is used to perform approximate inference of a GP classification (GPC) problem, whereas sparsification is used to reduce the computational complexity, by performing inference on a limited set of observations, called inducing points. However, the objective function used in [26] presents limitations that severely affect the overall performance in terms of both accuracy and efficiency. More precisely, it leaves no room for an optimal selection of the inducing variables and it does not allow for mini-batch optimization, a popular technique used to improve the efficiency of gradient-descent methods over large-scale problems. Moreover, in [26] the observation process is modeled by a Bernoulli likelihood instead of a binomial as in [8]. The latter can condense the satisfaction of the M simulations into a single observation, whereas the former must consider them as M different observations. This has a strong effect on the dimension of the training set which is of paramount importance for the sake of scalability. While smMC in [8] and in [26] renders parametric verification feasible at runtime, the number of parameters that are allowed to vary is still limited (up to a 4-dimensional parameter space with a reasonable computational budget). Finding an effective solution that makes smMC scale to large datasets, thus to large parameter spaces, remains an open issue.

Main contributions. The main contribution is to tackle the scalability issues of smMC by leveraging stochastic variational inference (SVI), instead of EP or VI,

to solve the smMC Bayesian inference problem. More precisely, we propose a novel approach for scalable stochastic parametric verification, called Stochastic Variational Smoothed Model Checking (SV-smMC), that leverages Stochastic Variational Inference (SVI) to make the smMC Bayesian inference scale to large datasets. The variational rationale is to transform the inference problem into an optimization one. The approach is stochastic in the sense that stochastic gradient descent (SGD) is used in the gradient-based optimization of a suitable variational objective function. SVI is extremely flexible and it is thus applied both to Gaussian Processes (GP) and to Bayesian Neural Networks (BNN). The main advantage of SVI, compared to the VI used for example in [26], is the use of mini-batches that makes inference easily parallelizable, enabling also GPU acceleration. Moreover, in SVI the inducing variables are optimally selected during inference so that sparsification causes a less pronounced drop in the reconstruction accuracy w.r.t. [26]. As a result, SV-smMC can face extremely large datasets. This result has significant implications for RV as it enables us to consider variability both in the rate parameters and in the initial state of the stochastic system. One can also take into account the variability solely in the state space, thereby accommodating larger state spaces that are commonly encountered in complex systems.

Overview of the paper. This paper is structured as follows. We start by presenting the background theory in Sect. 2, comprised of the formal definition of the smMC Bayesian inference problem (Sect. 2.3). In Sect. 3 the theoretical details of SV-smMC are presented both for the GP version (Sect. 3.1) and for the BNN version (Sect. 3.2). Sect. 4 compares the performances of SV-smMC against those of smMC on three stochastic models with increasing parametric complexity. Two of these case studies are taken from [8] to make a fair comparison of the performances. Moreover, in order to have a better quantification of the scalability of the proposed solution, we test SV-smMC on a pool of randomly generated stochastic processes over parameter spaces of increasing dimension for multiple randomly generated temporal properties. Finally, Sect. 5.1 briefly discuss how to obtain local statistical guarantees over the prediction error.

2 Background

2.1 Population Continuous Time Markov Chain

A population of interacting agents can be modeled as a stochastic system evolving continuously in time over a finite or countable state space $\mathcal{X}$. Assuming the system is Markovian, meaning the memory-less property holds, we can rely on the formalism of population Continuous Time Markov Chains (CTMC) $\mathcal{M}$. A population is specified by n different species $\{S_1, \dots, S_n\}$ subject to a dynamics described by r different rules (reactions) $\{R_1, \dots, R_r\}$. The respective CTMC is described by:

- a state vector, $X(t) = (X_1(t), \dots, X_n(t))$, taking values in $\mathcal{X} \subseteq \mathbb{N}^n$ and counting the number of agents in each species at time t ;

- a finite set of reactions $R = (R_1, \dots, R_r)$ describing how the state of the population changes in time. A general reaction R_i is identified by the tuple (τ_i, ν_i) , where:
 - $\rho_i : \mathcal{X} \times \Theta_i \rightarrow \mathbb{R}_{\geq 0}$ is the *rate function* of reaction R_i that associates with each reaction the rate of an exponential distribution, as a function of the global state of the model and of parameters θ_i , and
 - ν_i is the *update vector*, giving the net change of agents due to the reaction, so that the firing of reaction R_i results in a transition of the system from state $X(t)$ to state $X(t) + \nu_i$.

Reaction rules are easily visualised in the chemical reaction style, as

$$R_i: \sum_{j \in \{1, \dots, n\}} \alpha_{ij} S_j \xrightarrow{\rho_i(X, \theta_i)} \sum_{j \in \{1, \dots, n\}} \beta_{ij} S_j.$$

The stoichiometric coefficients $\alpha_i = [\alpha_{i1}, \dots, \alpha_{in}]$, $\beta_i = [\beta_{i1}, \dots, \beta_{in}]$ can be arranged so that they form the update vector $\nu_i = \beta_i - \alpha_i$ for reaction R_i .

The parameters $\theta = (\theta_1, \dots, \theta_r)$ have a crucial effect on the dynamics of the system: changes in θ can lead to qualitatively different dynamics. We stress such crucial dependency by using the notation $\mathcal{M}_\theta$. The trajectories of such a CTMC can be seen as samples of random variables $X(t)$ indexed by time t over a state space $\mathcal{X}$. A parametric CTMC (pCTMC) is a family $\mathcal{M}_\theta$ of CTMCs where the parameters θ vary in a domain Θ . For simplicity, but without loss of generality, for the rest of the paper we fix the initial state of the pCTMC and consider only the rate parameters as varying conditions. We stress that all the results remain valid if the initial state of the system varies, e.g. when it represents the current state in an RV framework.

Running example. A well-known example of pCTMC is the SIR model describing the spread of an epidemic in a population of fixed size N . Let S , I and R denote respectively the species of susceptible, infected and recovered individuals. The stochastic dynamics is described by an infection reaction $R_1 : S + I \rightarrow 2I$ happening with rate function $\beta X_S X_I / N$, and a recovery reaction $R_2 : I \rightarrow R$ happening with rate function γX_I . The rate parameters, $\theta = (\beta, \gamma)$, and the initial state of the population, $X(0) = (X_S(0), X_I(0), N - X_S(0) - X_I(0))$, play a crucial role in shaping the future dynamics of the system, ultimately determining whether and when the disease will be eradicated.

2.2 Signal Temporal Logic

Properties of CTMC trajectories can be expressed via Signal Temporal Logic (STL) [21] formulas. STL allows the specification of properties of dense-time, real-valued signals, and the automatic generation of monitors for testing properties on individual trajectories. The rationale of STL is to transform real-valued signals into Boolean ones, using formulae built on the following *STL syntax*:

$$\varphi := \text{true} \mid \mu \mid \neg \varphi \mid \varphi \wedge \varphi \mid \varphi \mathcal{U}_I \varphi, \quad (1)$$

where $\mathcal{U}_I$ is the until operator, $I \subseteq \mathbb{T}$ is a temporal interval, either bounded, $I = [a, b]$, or unbounded, $I = [a, +\infty)$, for any $0 \leq a < b$. Atomic propositions μ are (non-linear) inequalities on population variables. From this essential syntax it is easy to define other operators: $false := \neg true$, $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, eventually $F_I := true \mathcal{U}_I \varphi$ and globally $G_I := \neg F_I \neg\varphi$. Monitoring the satisfaction of a formula is done recursively on the parsing tree structure of the STL formula. See [21] for the details on STL Boolean semantics and on Boolean STL monitors.

Running example (continued). In the context of the previously described SIR model, we can define an STL property to monitor the termination of the epidemic in a time between 100 and 120 time units from the epidemic onset. Formally, $\varphi = (X_I > 0) \mathcal{U}_{[100, 120]} (X_I = 0)$. Because of the stochastic nature of the system, even when both the rate parameters and the initial state are fixed, the satisfaction of this property is a random variable over Boolean truth values.

2.3 Smoothed Model Checking

Probabilistic Model Checking (PMC). Verification of temporal properties is of paramount importance, especially for safety-critical processes. When the system evolves stochastically, probabilistic model checking [2] comes into play. For linear time properties, like those of STL, the goal is to compute the probability $Pr(\varphi|\mathcal{M})$ that a stochastic system $\mathcal{M}$ satisfies a given STL formula φ . Exact computation of $Pr(\varphi|\mathcal{M})$ suffers from very limited scalability and furthermore requires the full knowledge of the stochastic model.

Statistical Model Checking (SMC). Statistical model checking [33] fights the two aforementioned issues: instead of analyzing the model, it evaluates the satisfaction of the given formula on a number of observed runs of the system, and derives a statistical estimate of $Pr(\varphi|\mathcal{M})$, valid only with some confidence. Given a CTMC $\mathcal{M}_\theta$ with fixed parameters θ , time-bounded CTMC trajectories are sampled by standard simulation algorithms, such as SSA [13], and monitoring algorithms for STL [21] are used to assess if the formula φ is satisfied for each sampled trajectory. This process produces samples from a Bernoulli random variable equal to 1 if and only if φ is true. SMC [34, 37] then uses standard statistical tools, either frequentist [34] or Bayesian [37], to estimate from these samples the satisfaction probability $Pr(\varphi|\mathcal{M}_\theta)$ or to test if $Pr(\varphi|\mathcal{M}_\theta) > q$ with a prescribed confidence level.

Satisfaction Function for pCTMCs. Building on [8], our interest is in parametric verification [16], i.e. to quantify how the satisfaction of STL formulae depend on the unknown parameters of the pCTMC. We define the *satisfaction function* $f_\varphi: \Theta \rightarrow [0, 1]$ associated to φ as

$$f_\varphi(\theta) = Pr(\varphi = true | \mathcal{M}_\theta). \quad (2)$$

In practice, for every $\theta \in \Theta$, $f_\varphi(\theta)$ quantifies the probability that a realization of $\mathcal{M}_\theta$ satisfies φ . An accurate estimation of the satisfaction function f_φ over the

entire parameter space Θ by means of SMC would require a prohibitively large number of evaluations. In particular, SMC must be run from scratch for every parameter $\theta \in \Theta$. In [8] – Theorem 1 – it has been shown that $f_\varphi(\theta)$ is a smooth function of the model parameters and thus machine learning techniques can be used to infer this function from a limited set of observations and this is exactly the goal of smoothed model checking (smMC).

Smoothed model checking. Given a pCTMC $\mathcal{M}_\theta$ and an STL formula φ , the goal of smMC is to find a statistical estimate of the satisfaction function of (2) from a set of noisy observations of f_φ obtained at few parameter values $\theta_1, \theta_2, \dots$. The task is to construct a statistical model that, for any value $\theta^* \in \Theta$, computes efficiently an estimate of $f_\varphi(\theta^*)$ together with a credible interval associated with the prediction. More precisely, given an input point θ , our observations are obtained by evaluating a property φ on few trajectories sampled from the stochastic model $\mathcal{M}_\theta$ via SSA. Thus, given a set of N_t parameter values, $\Theta_t = \{\theta_1, \dots, \theta_{N_t}\}$, we simulate, for each parameter θ_i , M_t trajectories, obtaining M_t Boolean values $\ell_i^j \in \{0, 1\}$ for $j = 1, \dots, M_t$. We condense these Boolean values in a vector $L_i = [\ell_i^1, \dots, \ell_i^{M_t}]$. The noisy observations form the *training set*

$$D_t = \{(\theta_i, L_i) \mid i = 1, \dots, N_t\}. \quad (3)$$

In practice, smMC can be framed as a *Bayesian inference problem* that aims at inferring an accurate probabilistic estimate of the unknown satisfaction function $f_\varphi : \Theta \rightarrow [0, 1]$. In the following, let $f : \Theta \rightarrow [0, 1]$. The main ingredients of a Bayesian approach are the following:

1. Choose a *prior* distribution, $p(f)$, over a suitable function space, encapsulating the beliefs about function f prior to any observations being taken.
2. Determine the functional form of the observation process by defining a suitable *likelihood* function that effectively models how the observations depend on the uncertain parameter θ . Our observation process can be modeled by a binomial over M_t trials with parameter $f_\varphi(\theta)$. Given the nature of our training set, defined in (3), we define the probabilistic likelihood as

$$p(D_t | f) = \prod_{i=1}^{N_t} \text{Binomial}(L_i \mid M_t, f(\theta_i)).$$

3. Leverage Bayes' theorem to define the *posterior* distribution over functions given the observations

$$p(f | D_t) = \frac{p(D_t | f)p(f)}{p(D_t)}.$$

Computing $p(D_t) = \int p(D_t | f)p(f)df$ is almost always computationally intractable as we have non-conjugate prior-likelihood distributions. Therefore, we need algorithms to accurately approximate such posterior distribution.

4. Evaluate such posterior at points θ_* , resulting in a predictive distribution $p(f_*|\theta_*, D_t)$, whose statistics are used to obtain the desired estimate of the satisfaction probability together with the respective credible interval.

Two main ingredients are essential to define the smMC solution strategy:

- (i) the probabilistic model chosen to describe the distribution over functions f ,
- (ii) the approximate inference strategy.

Running example (continued). For visualization purposes, consider a one-dimensional configuration, where $\beta = 0.12$ and γ varies in the interval $[0.005, 0.2]$. The exact satisfaction function $f_\varphi(\gamma)$ is unknown, but we can obtain noisy observations via SMC. We randomly sample N_t values of γ and, for each value of γ , we simulate M_t trajectories and evaluate the STL satisfaction of each realization. The purple bars in Fig. 1 represent the SMC 95% confidence intervals around the empirical mean of the satisfaction probability. SmMC learns a probabilistic model of the satisfaction function over γ , i.e. infers the posterior distribution (point 3). Estimating the satisfaction probability at a new point γ_* reduces to evaluating the posterior predictive distribution in γ_* (point 4). Fig. 1 shows the 95% credible intervals of the posterior predictive distribution over a grid of test parameters.

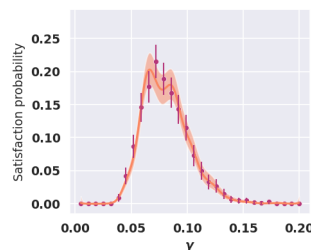


Fig. 1. Bayesian inference of the satisfaction function $f_\varphi(\gamma)$.

Previous works and limitations. The smMC technique presented in [8] uses Gaussian Processes (GP) as probabilistic model – ingredient (i) – and the Expectation Propagation (EP) algorithm to approximate the posterior distribution – ingredient (ii). EP [22,27] is an iterative algorithm that aims at matching the product of non-Gaussian likelihoods with a product of univariate Gaussian distributions. If the prior is Gaussian, this results in a closed-form analytic expression of the approximate posterior. However, the resulting analytic expression involves the inversion of the covariance matrix estimated at all pairs of parameter values. Therefore, this solution scales as $\mathcal{O}(N_t^3)$, it is thus unfeasible for large datasets. On the other hand, the rationale of VI, used in [26], is to directly approximate the non-Gaussian posterior with a parametric distribution (typically a Gaussian) building an optimization problem to search for the variational parameters that best fit the data. However, computing the optimization loss requires the derivation of the analytical form of a Gaussian with the inversion of a covariance matrix. Therefore, the cost is still cubic in the number of parameter values. Moreover, if the observation process is modeled by a Bernoulli, as in [26], meaning if Boolean values ℓ_i^j are considered as individual observations instead of condensing them in a vector L_i , inference scales as $\mathcal{O}((N_t M_t)^3)$. In order to mitigate such scalability issues, the authors of [26] propose a sparsification technique. Sparsification reduces the computational complexity to $\mathcal{O}((m M_t)^3)$, where m is the number of

sparse observations, known as inducing variables, considered. Nonetheless, sparsification strongly reduces the reconstruction accuracy, especially if the m sparse observations are randomly selected. The objective function used in [26] does not explicitly depend on the inducing variables, forcing them to be fixed a priori, leaving no room for an optimal selection of such points. Moreover, the variational optimization problem of [26] cannot be framed in terms of stochastic gradient descent optimization. These issues strongly limit the scalability capabilities of the proposed solution.

Our contribution. The main contribution of this paper, explained in the next section, is to introduce stochastic variational inference (SVI) as an efficient and scalable alternative for ingredient (ii). The rationale is to frame Bayesian inference as an optimization problem, typical workaround of variational approaches, and define an objective function that enables optimization in terms of stochastic gradient descent (SGD) so that smMC becomes efficient and scalable even on extremely large datasets. Furthermore, our objective function explicitly depends on the inducing variables that can be considered as additional variational parameters to be optimized with SGD. Therefore, our sparsification strategy allows for an optimal selection of sparse observations. SVI is then applied on two different probabilistic models, i.e. on two alternatives for ingredient (i): Gaussian Processes (GP) and Bayesian Neural Networks (BNN).

3 Stochastic Variational Smoothed Model Checking

The goal of Stochastic Variational Smoothed Model Checking (SV-smMC) is to make smMC scale to high-dimensional models. SV-smMC proposes stochastic variational inference as ingredient (ii) to efficiently compute the approximate posterior distribution $p(f|D_t)$ so that inference scales well to large datasets D_t .

The core idea behind variational approaches is to translate the posterior inference into an optimization problem, where a parametric distribution is proposed as a candidate approximator of the unknown posterior distribution. The optimization problem aims at minimizing the difference, measured by the Kullback-Leibler (KL) divergence, between these two distributions. However, as the posterior is unknown, various model-specific strategies can be developed to derive from the KL formula a lower bound of the marginal log-likelihood of our data. This lower bound, known as evidence lower bound (ELBO), is used as new objective function as it does not depend explicitly on the unknown posterior. This bound is then optimized with respect to the parameters of the proposed variational distribution. The latter is typically chosen to have nice statistical properties, making predictions feasible and efficient. Owing to the work of [15], in order to scale VI over very large datasets, the (black-box VI) optimization task can be phrased as a stochastic optimization problem [36] by estimating the gradient of the ELBO with Monte Carlo methods. Moreover, the dataset can be divided into mini-batches. As for ingredient (i), SV-smMC can use two alternative probabilistic models to define distributions over function f . The first one is based on Gaussian Processes (GP), whereas the second one is based on Bayesian Neural Networks (BNN). SVI

is applied to both probabilistic models with the proper model-specific adjustments to the variational formulation. Below, we provide an intuitive presentation of the approximate inference processes, whereas more formal details are provided in Appendix A in [6].

3.1 Gaussian Processes over non-Gaussian Likelihoods

Gaussian Processes (GP) are a well-known formalism to define distributions over real-valued functions of the form $g: \Theta \rightarrow \mathbb{R}$. A GP distribution is uniquely identified by its mean $\mu(\theta) = \mathbb{E}[g(\theta)]$ and its covariance function $k_\gamma(\theta, \theta')$ and characterized by the fact that the distribution of g over any finite set of points $\hat{\theta}$ is Gaussian with mean $\mu(\hat{\theta})$ and variance $k_\gamma(\hat{\theta}, \hat{\theta})$. In the following, we let g_t , μ_t and $K_{N_t N_t}$ denote respectively the latent, the mean and the covariance functions evaluated on the training inputs Θ_t .

The GP prior over latent functions g evaluated at training inputs Θ_t – step 1 – is defined as $p(g|\Theta_t) = \mathcal{N}(g|\mu_t, K_{N_t N_t})$. The posterior over latent variables $p(g_t|D_t)$ – step 3 – is not available in closed form since it is the convolution of a Gaussian and a binomial distribution. Hence, we have to rely on SVI for posterior approximation (details later). Once we obtain a tractable posterior approximation, in order to make predictions over a test input θ_* , with latent variable g_* , we have to compute an empirical approximation of the predictive distribution

$$p(f_*|\theta_*, D_t) = \int \Phi(g_*) p(g_*|\theta_*, D_t) dg_*, \quad (4)$$

in which the outputs of the latent function $g: \Theta \rightarrow \mathbb{R}$ are mapped into the $[0, 1]$ interval by means of a so-called link function Φ , typically the inverse logit or the inverse probit function [5], so that $f: \Theta \rightarrow [0, 1]$ is obtained as $f = g \circ \Phi$.

Stochastic Variational Inference. Here we outline an intuitive explanation of the SVI steps to approximate the GP posterior when the likelihood is non-Gaussian. For a more detailed mathematical description see Appendix A in [6]. The main issue with GP inference is the inversion of the $N_t \times N_t$ covariance matrix $K_{N_t N_t}$. This is the reason why variational approaches to GP start with sparsification, i.e. with the selection of $m \ll N_t$ inducing points that live in the same space of Θ_t and, from them, define a set of inducing variables u_t . The covariance matrix over inducing points, K_{mm} , is less expensive to invert and thus it acts as a low-rank approximation of $K_{N_t N_t}$. We introduce a Gaussian variational distribution $q(u_t)$ over inducing variables whose goal is to be as similar as possible to the posterior $p(u_t|D_t)$. A classical VI result is to transform the expression of the KL divergence between the variational distribution $q(u_t)$ and the posterior $p(u_t|D_t)$ into a lower bound over the marginal log-likelihood $\log p(D_t)$. As our likelihood – step 2 – factors as $p(D_t|g_t) = \prod_{i=1}^{N_t} p(L_i|g_i)$ and because of the Jensen inequality we obtain the following ELBO (see Appendix A.1 in [6] for the mathematical details):

$$\log p(D_t) \geq \sum_{i=1}^{N_t} \mathbb{E}_{q(g_i)} [\log p(L_i|g_i)] - KL[q(u_t)||p(u_t)] := \mathcal{L}_{GP}(\nu, \gamma), \quad (5)$$

where L_i denotes the set of observed Boolean tuples corresponding to points in θ_i in D_t , $p(u_t)$ denotes the prior distribution over inducing variables and ν denotes the hyper-parameters introduced to describe the sparsification and the variational distribution. The distribution $q(g_t)$ is Gaussian with an exact analytic derivation from $q(u_t)$ that requires $\mathcal{O}(m^2)$ computations (no matrix inversion involved here). The SVI algorithm then consists of maximizing $\mathcal{L}_{GP}$ with respect to its parameters using gradient-based stochastic optimization. We stress that, at this step, the selection of inducing variable is optimized, resulting in a more effective sparsification. Computing the KL divergence in (5) requires only $\mathcal{O}(m^3)$ computations. Most of the work will thus be in computing the expected likelihood terms. Given the ease of parallelizing the simple sum over N_t , we can optimize $\mathcal{L}_{GP}$ in a stochastic fashion by selecting mini-batches of the data at random.

Predictive distribution. The predictive posterior $p(g_*|\theta_*, D_t)$ is now approximated by a variational distribution $q(g_*)$, which is Gaussian and whose mean and variance can be analytically computed with cost $\mathcal{O}(m^2)$. From the mean and the variance of $q(g_*)$, we obtain the respective credible interval and we can use the link function Φ to map it to a subset of the interval $[0, 1]$, in order to obtain the mean and the credible interval of the posterior predictive distribution $p(f_*|\theta_*, D_t)$ of equation (4).

3.2 Bayesian Neural Networks

The core idea of Bayesian neural networks (BNNs) [18, 31, 14, 5], is to place a probability distribution over the weights $\mathbf{w}$ of a neural network $f_{\mathbf{w}}: \Theta \rightarrow [0, 1]$, transforming the latter into a probabilistic model.

The Bayesian learning process starts by defining a prior distribution $p(\mathbf{w})$ over $\mathbf{w}$ – step 1 – that expresses our initial belief about the values of the weights. As we observe data D_t , we update this prior to a posterior distribution $p(\mathbf{w}|D_t)$ – step 3 – using Bayes’ rule. Because of the non-linearity introduced by the neural network function $f_{\mathbf{w}}(\theta)$ and since the likelihood $p(D_t|\mathbf{w})$ – step 2 – is binomial, the posterior $p(\mathbf{w}|D_t)$ is non-Gaussian and it cannot be computed analytically. In order to predict the satisfaction function over an unobserved input θ_* , we marginalize the predictions with respect to the posterior distribution of the parameters, obtaining

$$p(f_*|\theta_*, D_t) = \int f_{\mathbf{w}}(\theta_*) p(\mathbf{w}|D_t) d\mathbf{w}. \quad (6)$$

The latter is called *posterior predictive* distribution and it can be used to retrieve information about the uncertainty of a specific prediction f_* . Unfortunately, the integration is analytically intractable due to the non-linearity of the neural network function [5, 20] so we empirically estimate such quantity.

Stochastic Variational Inference. The rationale of SVI for BNNs is to choose a parametric variational distribution $q_{\psi}(\mathbf{w})$ that approximates the unknown posterior distribution $p(\mathbf{w}|D_t)$ by minimizing the KL divergence $KL[q_{\psi}(\mathbf{w})||p(\mathbf{w}|D_t)]$ between these two distributions. Since the posterior distribution is not known, the

classic variational approach is to transform the minimization of the KL divergence into the maximization of the Evidence Lower Bound (ELBO) [17], defined as

$$\mathcal{L}_{BNN}(\psi) := \mathbb{E}_{q_\psi(\mathbf{w})}[\log p(D_t|\mathbf{w})] - KL[q_\psi(\mathbf{w})||p(\mathbf{w})] \leq \log p(D_t), \quad (7)$$

see Appendix A.2 in [6] for the mathematical details. The first term is the expected log-likelihood of our data with respect to values of $f_{\mathbf{w}}$ sampled from $q_\psi(\mathbf{w}|D_t)$, whereas the second term is the KL divergence between the proposal distribution and the prior. The distribution q_ψ should be a distribution easy to sample from and such that the KL divergence is easy to compute. A common choice for q_ψ is the Gaussian distribution (where ψ denotes its mean and variance). KL divergence among two Gaussian distributions has an exact analytical form, hence the ELBO of (7) can be computed and it can be used as the objective function of a maximization problem over ψ .

Predictive distribution. The predictive distribution (6) is a non-linear combination of Gaussian distributions, and thus it is not Gaussian. However, samples can be easily extracted from $q_\psi(\mathbf{w})$, which allows us to obtain an empirical approximation of the predictive distribution. Let $[w_1, \dots, w_C]$ denote a vector of C realizations of the random variable $\mathbf{w} \sim q_\psi(\mathbf{w})$. Each realization w_i induces a deterministic function f_{w_i} that can be evaluated at θ_* , the unobserved input, providing an empirical approximation of $p(f_*|\theta_*, D_t)$.

Running example (end). The SIR model is rather simple with a two-dimensional parameter space and a two-dimensional state space. However, if one is interested in estimating the satisfaction function w.r.t. both states and rate parameters (a typical RV framework) we notice how EP inference already reaches its computational bottleneck, VI fails, whereas, SV-smMC easily succeeds (see Section 4.3 below).

4 Experiments

4.1 Case studies

We briefly introduce the case studies used to investigate the scalability and the accuracy of SV-smMC. In order to make a fair comparison, we start by reproducing the case studies presented in [8], Network Epidemics and Prokaryotic Gene Expression, and then add a third biological model, Three-layer Phosphorelay. Finally, to have a better and unbiased quantification of the performances and of the scalability of SV-smMC, we test it over randomly generated pCTMC with parameter spaces of increasing dimensions.

- **Network Epidemics (SIR)**: see running example.
- **Prokaryotic Gene Expression (PGE)**: model of LacZ, X_{LacZ} , protein synthesis in E. coli. The dynamics is governed by 11 parameters $k_1, k_2, \dots, k_{11}$. We choose an STL property for monitoring bursts of gene expression, rapid increases in LacZ counts followed by long periods of lack of protein production: $\varphi = F_{[1600, 2100]}(\Delta X_{LacZ} > 0 \wedge G_{[10, 200]}(\Delta X_{LacZ} \leq 0))$, where $\Delta X_{LacZ}(t) = X_{LacZ}(t) - X_{LacZ}(t-1)$.

- **Three-layer Phosphorelay (PR)**: network of three proteins $L1$, $L2$, $L3$ involved in a cascade of phosphorylation reactions (changing the state of the protein), in which protein Lj , in its phosphorylated form Ljp , acts as a catalyser of phosphorylation of protein $L(j + 1)$. There is a ligand B triggering the first phosphorylation in the chain. The dynamics depends on 6 parameters $k_p, k_1, k_2, k_3, k_4, k_d$. The chosen STL property models a switch in the most expressed protein between $L1p$ and $L3p$ after time 300: $\varphi = G_{[0,300]}(L1p - L3p \geq 0) \wedge F_{[300,600]}(L3p - L1p \geq 0)$.
- **Random pCTMC**: we randomly generate pCTMC over parameter spaces of increasing dimension. The number of interacting species n and the type and the number of reactions are randomly determined. The STL properties considered captures different behaviours: $\varphi_1 = G_{[0,T]}(S_i \leq S_j)$, $\varphi_2 = F_{[0,T]}(S_i \leq S_j)$ and $\varphi_3 = F_{[0,T]}(G(S_i < \tau))$, where τ and T are two fixed hyper-parameters and the species to be monitored, S_i and S_j , are randomly sampled for each property.

Details about the dynamics, i.e. about the reactions, the selected initial states and the chosen parametric ranges, are provided in Appendix B in [6].

4.2 Experimental Details

Dataset generation. The training set D_t is built as per (3). The test set, used to validate our results, can be summarized as $D_v = \{(\theta_j, (\ell_j^1, \dots, \ell_j^{M_v})) \mid j = 1, \dots, N_v\}$, where M_v is chosen very large, $M_v \gg M_t$, so that we have a good estimate of the true satisfaction probability over each test input. Input data, i.e. the parameter values, are scaled to the interval $[-1, 1]$ to enhance the performances of the inferred models and to avoid sensitivity to different scales in the parameter space. In the first three case studies, the biology-inspired ones, we choose different subsets of varying parameters and train a separate model on each of these choices. In other words, we fix some of the parameters and let only the remaining ones vary. In particular, in SIR we consider the following configurations: (a) fix γ and let β vary, (b) fix β and let γ vary, (c) vary both β and γ ; in PGE we consider the following configurations: (d) k_2 is the only parameter allowed to vary, (e) we let k_2 and k_7 vary; in PR we consider the following configurations: (f) only k_1 varies, (g) only k_p, k_d vary, (h) only k_1, k_2, k_3 vary, (i) only k_1, k_2, k_3, k_4 vary, (l) all six parameters are allowed to vary.

On the other hand, to better analyze the scalability of SV-smMC we randomly generate pCTMC over parameter spaces Θ of dimension 2, 3, 4, 8, 12, 16 and 20. For each dimension, we generate three different models and each model is tested over the three random properties $\varphi_1, \varphi_2, \varphi_3$ defined above. Table 2 in Appendix B in [6] shows the chosen dimensions for each generated dataset. In general, the number of observed parameters N_t and the number of observations per point M_t increase proportionally to the dimensionality of Θ .

Experimental settings. The CTMC dynamics is simulated via StochPy SSA¹ simulator for biology inspired models, whereas GillesPy2² is used to generate and

¹ <https://github.com/SystemsBioinformatics/stochpy>

² <https://github.com/StochSS/GillesPy2>

simulate random pCTMC. The Boolean semantics of pcheck library³ is used to check the satisfaction of a certain formula for a specific trajectory. GPyTorch [12] library is used to train the VI-GP and SVI-GP models, whereas Pyro [4] library is used to train the SVI-BNN models, both built upon PyTorch [25] library. Instead, EP-GP is implemented in NumPy⁴. The experiments were conducted on a shared virtual machine with a 32-Core Processor, 64GB of RAM and an NVidia A100 GPU with 20GB, and 8 VCPU. Code and data are available at: <https://github.com/ailab-units/SV-smMC.git>.

Training and evaluation. We apply Stochastic Variational Inference on both Gaussian Processes (SVI-GPs) and Bayesian Neural Networks (SVI-BNNs) and compare them to the baseline smMC approaches, where Gaussian Processes were inferred using either Expectation Propagation (EP-GPs) [8] or Variational Inference (VI-GPs) [26]. All models (EP-GP, VI-GP, SVI-GP and SVI-BNN) are Bayesian and trained over the training set D_t . Once the training phase is over, for each pair $(\theta_j, (\ell_j^1, \dots, \ell_j^{M_v})) \in D_v$ in the test set, we obtain a probabilistic estimate of the satisfaction probability $f(\theta_j)$ (defined in Sect. 3). We compare such distribution to the satisfaction probability $f_\varphi(\theta_j)$ estimated as the mean $\bar{L}_i$ over the Bernoulli trials $(\ell_j^1, \dots, \ell_j^{M_v})$ and we call the latter SMC satisfaction probability. We stress that SMC estimates provably converge to the true satisfaction probabilities, meaning that the width of confidence intervals converges to zero in the limit of infinite samples, while Bayesian inference quantifies the predictive uncertainty. Consequently, regardless of the number of samples, SMC and Bayesian estimates have different statistical meanings.

Evaluation metrics. To define meaningful measures of performance, let's clarify the notation. For each point in the test set, $j \in \{1, \dots, N_v\}$, let $\bar{L}_j$ and σ_j denote respectively the average and the standard deviation over the M_v Bernoulli trials $(\ell_j^1, \dots, \ell_j^{M_v})$. The inferred models, on the other hand, provide a posterior predictive distribution $p(f_j | \theta_j, D_t)$, let q_ϵ^j denote the ϵ -th quantile of such distribution. The metrics used to quantify the overall performances of the models over each case study and each configuration are the following:

- (i) the *root mean squared error* (RMSE) between SMC and the expected satisfaction probabilities, i.e. the root of the average of the squared residuals

$$\text{RMSE} = \frac{1}{N_v} \sum_{j=1}^{N_v} (\bar{L}_j - \mathbb{E}_{p(f|\theta_j, D_t)}[f(\theta_j)])^2.$$

This measure evaluates the quality of reconstruction provided by the mean of the posterior predictive distribution;

³ <https://github.com/simonesilvetti/pcheck>

⁴ Our implementation builds on <https://github.com/simonesilvetti/pyCheck/blob/master/smoothed/smoothedMC.py>

- (ii) the *accuracy* over the test set, i.e. the fraction of non empty intersections between SMC confidence intervals and estimated $(1-\epsilon)$ credible intervals:

$$\text{Acc} = \frac{1}{N_v} \cdot \left| \left\{ j \in \{1, \dots, N_v\} : \left[-\frac{z\sigma_j}{\sqrt{M_v}}, \frac{z\sigma_j}{\sqrt{M_v}} \right] \cap \left[q_{\epsilon/2}^j, q_{1-\epsilon/2}^j \right] \neq \emptyset \right\} \right|.$$

In particular, we set $z=1.96$ and $\epsilon=0.05$ in order to have the 95% confidence intervals and the 95% credible intervals respectively;

- (iii) the *average width* of the estimated credible intervals

$$\text{Unc} = \frac{1}{N_v} \sum_{j=1}^{N_v} (q_{1-\epsilon/2}^j - q_{\epsilon/2}^j),$$

which quantifies how informative the predictive uncertainty is and allows us to detect over-conservative predictors.

A good predictor should be balanced in terms of low RMSE, high test accuracy, i.e. high values for Acc, and narrow credible intervals, i.e. low values for Unc.

Implementation details. Both SVI-GP and SVI-BNN models are trained for $2k$ epochs with mini-batches of size 100 and a learning rate of 0.001. In SVI-GP the prior is computed on a maximum of $1k$ inducing points selected from the training set. SVI-BNNs have a fully connected architecture with 3 layers and Leaky ReLU nonlinear activations. To evaluate SVI-BNNs we take $1k$ samples from the posterior distribution evaluated over test inputs.

Prior tuning. Choosing an adequate prior is of paramount importance. In this paper we leverage model-specific strategies to pick reasonable ones, however, no guarantees can be provided about the adequacy of such priors. For GP, the prior is strongly related to the chosen kernel and adapted to data by type II maximum likelihood maximization. In EP-GP the optimal kernel hyperparameters are searched beforehand (as in [8]), whereas in SVI-GP the kernel hyperparameters are optimized on the fly, i.e. the variational loss is maximized also w.r.t. the kernel hyperparameters. In SVI-BNN, a deterministic NN (with the same architecture of the BNN) is trained and the learned weights are used to center the prior distribution which is Gaussian with standard deviation equal to $1/m$ (m is the layer-width).

4.3 Experimental Results

Computational costs. The cost of EP-GP inference is dominated by the cost of matrix inversion, which is cubic in the number of points in the training set. The cost of VI-GP is cubic in the number of randomly selected inducing points and linear in the total number of observations, i.e. the number of training points times the number of observations per point. The cost of SVI-GP inference is cubic in the number of optimally selected inducing points, which is chosen to be sufficiently

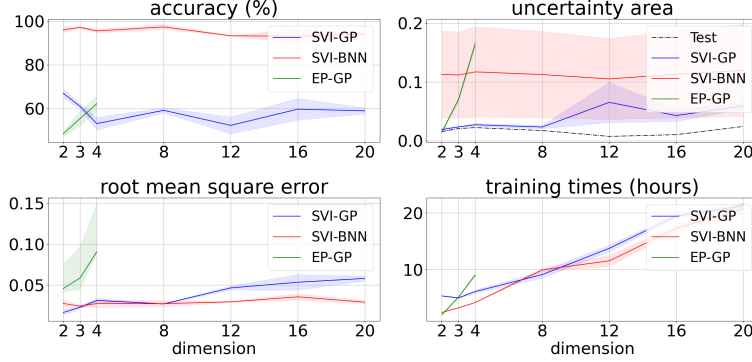


Fig. 2. Quantitative analysis of the scalability performances over randomly generated pCTMC with parameter spaces of increasing dimension (x-axis). For each dimension, we plot the 95% confidence interval of Acc, RMSE and Unc over each test set corresponding to that dimension.

small, and linear in the number of training instances. The cost of SVI-BNN is linear in the number of training points but it also depends on the architectural complexity of the chosen neural network. Stochastic variational models are trained by means of SGD, which is a stochastic inference approach. Thus, at least on simple configurations, it is likely to take longer than EP in reaching convergence. The computational advantage becomes significant as the complexity of the case study increases, i.e., when the training set is sufficiently large. EP faces memory limits, becoming unfeasible on configurations with parameter space of dimension higher than four. As a collateral advantage, SVI-GP optimizes the kernel hyperparameters on the fly during the training phase, whereas in EP-GP the hyperparameters search is performed beforehand and it is rather expensive. Compared to our SV-smMC approaches, VI-GPs do not allow for GPU acceleration, as a consequence the training times over one-dimensional case studies go from the 40 minutes of our SVI-based approaches to 3 hours.

For the randomly generated pCTMC we let the dimension of the training set grow linearly with the dimension of the parameter space ($N_t = 5000 \cdot r$) and so does the training time of SVI-GP and SVI-BNN (see Fig. 2). On the other hand, Fig. 2 shows how the training time of EP-GP grows much faster with respect to r and becomes soon unfeasible ($r > 4$). SV-smMC is trained leveraging GPU acceleration. Its convergence times are comparable to EP’s on simple configurations and they outperform EP on more complex ones. Evaluation time for EP-GPs and SVI-GPs is negligible as it is computed from the analytic posterior. The evaluation time for SVI-BNNs with $1k$ posterior samples is in turn negligible.

Performance evaluation. The evaluation metrics are the root mean square error (RMSE), the accuracy (Acc) and the width of the uncertainty quantification area (Unc). Results over the randomly generated pCTMC are summarized in Fig. 2, whereas results over the biological case studies are summarized in Table 1. For completeness, we compare our SV-smMC with the VI-GPs approach presented

in [26]. However, using a Bernoulli likelihood instead of a Binomial creates an extremely inefficient computational bottleneck that makes VI-GP feasible only over the one-dimensional scenario, where it is outperformed by all methods (see Table 1). Fig. 2 shows, for each dimension, the 95% confidence interval of Acc, RMSE and Unc over each test set corresponding to that dimension. More precisely, each dimension has nine associated datasets, three models with three properties each, so that nine different smMC models have been trained. Notice how the width of the inferred credible intervals is compared against the width of SMC confidence intervals.

In addition, Fig. 7-Fig. 10 in Appendix D of [6] show the results over one-dimensional configurations - (a), (b), (d) and (f) respectively - whose results over the test set are easy to visualise. In particular, we show the mean and the 95% credible intervals of the estimated satisfaction probability $f(\theta_j)$ for EP-GPs, SVI-GPs and SVI-BNNs. Fig. 11-Fig. 13 in Appendix D of [6] compare the results of EP-GPs, SVI-GPs and SVI-BNNs over two-dimensional configurations - (c), (e) and (g). In particular, we compare the SMC estimate of the satisfaction probability $f_\varphi(\theta_j)$ to the average satisfaction probability $\mathbb{E}[f(\theta_j)]$ estimated by EP-GPs, SVI-GPs and SVI-BNNs over each input θ_j of the test set.

We now compare the performances obtained by the variational approaches of SV-smMC to those of the smMC baseline based on EP-GP. Table 1 and Fig. 2 show how the RMSE of SV-smMC solutions is almost always lower than that of smMC. In addition, the baseline solution presents an RMSE that grows proportionally to the complexity and the dimensionality of the underlying configuration. On the contrary, SV-smMC solutions do not reflect such behaviour, as the value of the RMSE is almost constant across all the different configurations. About the informativeness of uncertainty estimations, we notice how SVI-BNN tends to produce credible intervals that are always larger than the one of SVI-GP, which, in turn, tends to underestimate the underlying uncertainty. This phenomenon appears in all the different configurations and it is easily observable in Fig. 2. We argue that SVI-GP tends to provide overconfident predictions due to the sparsification strategies used during inference. Such behaviour is well-known and discussed in [10,28]. SVI-BNN does not present such behaviour as it does not undergo any

Configuration	RMSE				Accuracy				Uncertainty				
	EP-GP	VI-GP	SVI-GP	SVI-BNN	EP-GP	VI-GP	SVI-GP	SVI-BNN	Test	EP-GP	VI-GP	SVI-GP	SVI-BNN
(a) SIR β	1.41	2.11	1.38	1.43	100.00	96.3	98.80	100.00	0.044	0.044	0.060	0.032	0.097
(b) SIR γ	1.02	2.12	0.84	0.96	77.80	59.8	77.30	92.60	0.018	0.018	0.041	0.012	0.058
(c) SIR β, γ	1.25	-	1.20	0.99	72.75	-	85.75	92.25	0.019	0.042	-	0.019	0.047
(d) PGE k_2	5.34	6.06	5.83	5.51	93.75	95.75	94.00	97.50	0.039	0.030	0.057	0.037	0.080
(e) PGE k_2, k_7	6.48	-	3.44	2.06	78.25	-	89.25	95.00	0.043	0.021	-	0.056	0.093
(f) PR k_1	2.25	1.76	1.99	1.78	99.70	99.80	99.20	100.00	0.058	0.059	0.066	0.046	0.100
(g) PR k_p, k_d	2.95	-	2.31	1.89	99.25	-	96.25	99.75	0.055	0.108	-	0.037	0.093
(h) PR k_1, k_2, k_3	6.97	-	2.30	2.01	99.80	-	93.70	100.0	0.050	0.340	-	0.030	0.121
(i) PR $k_1, \dots, k_4$	10.63	-	2.44	2.67	99.02	-	93.65	99.92	0.050	0.682	-	0.030	0.150
(l) PR $k_p, k_1, \dots, k_4, k_d$	-	-	1.87	1.56	-	-	97.02	99.80	0.049	-	-	0.028	0.087

Table 1. Root Mean Square Error ($\times 10^{-2}$), test accuracy (%) and average uncertainty width for EP-GP, VI-GP, SVI-GP and SVI-BNN. SVI-BNNs are evaluated on $1k$ posterior samples. For each case study, we highlight the minimum MSE, the highest accuracy values and the **lowest** and **highest** uncertainty values. Uncertainty is compapurple to the average uncertainty width of the test set.

sparsification. On the other hand, the baseline smMC tends to have tight uncertainty estimates on low-dimensional configurations, but it becomes excessively over-conservative in high-dimensional configurations, making the predicted credible intervals almost uninformative. All models reach extremely high accuracies over the test set. SVI-BNN reaches the best performances over all the configurations: the average accuracy is around 95% and it is always higher than 71%. This result is not surprising given that SVI-BNN shows low RMSEs (overall average around 0.02) and slightly over-conservative credible intervals (overall average uncertainty width around 0.1). The SVI-GP accuracy, on the other hand, fluctuates around 60% as it tends to provide over-confident credible intervals with an overall average RMSE comparable to that of SVI-BNN but with overall average uncertainty width of around 0.03.

Related work. A number of recent works tackle the problem of parametric verification of a stochastic process, we here present the most relevant. In [1] the parameter space of a pCTMC is explicitly explored, while [11] assumes a probability distribution over the parameters and proposes a sampling-based approach. In [19] conformal predictions are used over the expected value of the stochastic process rather than its distribution and [7] presents a frequentist, rather than Bayesian, approach based on quantile regression. The conformal recalibration is constant over the entire state space, whereas our Bayesian quantification of uncertainty is point-specific.

Discussion. To summarize, we can see how, in general, SV-smMC solutions scale a lot better to high-dimensional problems compared to smMC, both in terms of feasibility and in terms of quality of the results. SVI-BNN reaches the highest accuracy and provides rather conservative predictions. SVI-GP, on the other, reaches low RMSEs and tends to provide overconfident predictions. Therefore BNNs should be preferred in the verification of safety-critical systems, whereas GPs can be a good solution when overconfidence is more tolerable. Finally, we see how EP-GP is competitive only on extremely simple configurations. As the dimensionality increases, so does the error: the RMSE increases and the credible intervals become excessively broad. Moreover, we soon reach the memory-bound wall that makes EP-GP solution unfeasible on configurations with more than four parameters.

We conclude by suggesting how smMC, either the original EP-based version or the novel SVI-based version, may be enriched with *probabilistically approximately correct statistical guarantees over the generalization error* holding point-wise at any prediction point without affecting the scalability at runtime. See Section 5.1 for an intuition of the solution based on Conformal Predictions [32]. A detailed presentation of such an extension is left for future work.

5 Conclusions

This paper presents SV-smMC, an extension of Smoothed Model Checking, based on stochastic variational inference, that scales well to high dimensional parameter spaces and that enables GPU acceleration. In addition, this paper offers a

comparison of the performances of stochastic variational inference over two different Bayesian approaches - namely Gaussian processes (SVI-GP) and Bayesian neural networks (SVI-BNN) - against those of the baseline smMC, based on the expectation propagation technique. In particular, our experiments show that the posterior predictive distribution provided by SVI-BNN provides the best overall results in terms of the estimated satisfaction probabilities. On the other hand, thanks to GPU acceleration, SVI-GP is able to achieve competitive performances with a significant speed-up in computational time. Furthermore, we show how variational approaches are able to overcome the computational limitations of the expectation propagation algorithm over large datasets.

SV-smMC can be naturally extended with active learning ideas, following the line of [9,3,29], solving efficiently parameter synthesis and design tasks.

5.1 Statistical Guarantees

The Bayesian quantification of uncertainty, despite being based on statistically sound operations, offers no guarantees per se as it strongly depends on the chosen prior. Additionally, in smMC, either EP- or SVI-based, we further add the error of approximate inference on top of this. Here we briefly discuss how smMC, in all its versions, can be enriched with *probabilistically approximately correct statistical guarantees over the generalization error* holding point-wise at any prediction point. This result is obtained by combining Bayesian estimates of uncertainty with Inductive Conformal Prediction [32,35], a framework that can be applied on top of any deterministic regressor to enrich its predictions with statistically valid quantification of the predictive uncertainty. The rationale is to transform point-wise predictions made by the expectation over the posterior $q(f)$ into statistically valid prediction intervals. ICP is non-informative to check how the uncertainty distributes over Θ as the interval width is constant over the entire parameter space Θ . Normalized Inductive Conformal Predictions (NICP) [23,24] consider locally weighted residuals to overcome this limitation, so that prediction intervals are tighter for parameters that are deemed easy to predict and vice-versa. Retrieving such guarantees does not affect the scalability at runtime. Our intuition is to exploit the Bayesian quantification of uncertainty of smMC as the normalizing function of a NICP approach, that in turn will provide us with point-specific statistical guarantees over the error coverage. More details and some preliminary results can be found in Appendix C in [6].

Acknowledgments. This work has been partially supported by the PRIN project “SEDUCE” n. 2017TWRCNB and by the PNRR project iNEST (Interconnected North-East Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS_00000043).

References

1. Badings, T.S., Jansen, N., Junges, S., Stoelinga, M., Volk, M.: Sampling-based verification of ctmcS with uncertain rates. In: Computer Aided Verification: 34th Inter-

- national Conference, CAV 2022, Haifa, Israel, August 7–10, 2022, Proceedings, Part II. pp. 26–47. Springer (2022)
2. Baier, C., Katoen, J.P.: Principles of model checking. MIT press (2008)
 3. Bartocci, E., Bortolussi, L., Nenzi, L., Sanguinetti, G.: System design of stochastic models using robustness of temporal properties. *Theor. Comput. Sci.* **587**, 3–25 (2015). <https://doi.org/10.1016/j.tcs.2015.02.046>, <https://doi.org/10.1016/j.tcs.2015.02.046>
 4. Bingham, E., Chen, J.P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletos, T., Singh, R., Szerlip, P.A., Horsfall, P., Goodman, N.D.: Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.* **20**, 28:1–28:6 (2019), <http://jmlr.org/papers/v20/18-403.html>
 5. Bishop, C.M.: Pattern recognition and machine learning. Springer (2006)
 6. Bortolussi, L., Cairoli, F., Carbone, G., Pulcini, P.: Stochastic variational smoothed model checking. arXiv preprint arXiv:2205.05398 (2022)
 7. Bortolussi, L., Cairoli, F., Paoletti, N.: Conformal quantitative predictive monitoring of stl requirements for stochastic processes. In: 26th ACM International Conference on Hybrid Systems: Computation and Control (2023)
 8. Bortolussi, L., Milios, D., Sanguinetti, G.: Smoothed model checking for uncertain continuous-time markov chains. *Information and Computation* **247**, 235–253 (2016)
 9. Bortolussi, L., Silveti, S.: Bayesian statistical parameter synthesis for linear temporal properties of stochastic models. In: Beyer, D., Huisman, M. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference, TACAS 2018. Lecture Notes in Computer Science, vol. 10806, pp. 396–413. Springer (2018). https://doi.org/10.1007/978-3-319-89963-3_23, https://doi.org/10.1007/978-3-319-89963-3_23
 10. Candela, J.Q., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.* **6**, 1939–1959 (2005)
 11. Češka, M., Dannenberg, F., Paoletti, N., Kwiatkowska, M., Brim, L.: Precise parameter synthesis for stochastic biochemical systems. *Acta Informatica* **54**, 589–623 (2017)
 12. Gardner, J.R., Pleiss, G., Bindel, D., Weinberger, K.Q., Wilson, A.G.: Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 7587–7597. NIPS’18, Curran Associates Inc., Red Hook, NY, USA (2018)
 13. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry* **81**(25), 2340–2361 (1977)
 14. Goan, E., Fookes, C.: Bayesian neural networks: An introduction and survey. In: Case Studies in Applied Bayesian Data Science, pp. 45–87. Springer (2020)
 15. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. *Journal of Machine Learning Research* (2013)
 16. Jansen, N., Corzilius, F., Volk, M., Wimmer, R., Ábrahám, E., Katoen, J.P., Becker, B.: Accelerating parametric probabilistic verification. In: International Conference on Quantitative Evaluation of Systems. pp. 404–420. Springer (2014)
 17. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. *Machine learning* **37**(2), 183–233 (1999)
 18. Lampinen, J., Vehtari, A.: Bayesian approach for neural networks—review and case studies. *Neural networks* **14**(3), 257–274 (2001)
 19. Lindemann, L., Qin, X., Deshmukh, J.V., Pappas, G.J.: Conformal prediction for stl runtime verification. arXiv preprint arXiv:2211.01539 (2022)
 20. MacKay, D.J.: A practical bayesian framework for backpropagation networks. *Neural computation* **4**(3), 448–472 (1992)

21. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, pp. 152–166. Springer (2004)
22. Minka, T.P.: Expectation propagation for approximate bayesian inference. arXiv preprint arXiv:1301.2294 (2013)
23. Papadopoulos, H., Haralambous, H.: Reliable prediction intervals with regression neural networks. *Neural networks : the official journal of the International Neural Network Society* **24** 8, 842–51 (2011)
24. Papadopoulos, H., Vovk, V., Gammerman, A.: Regression conformal prediction with nearest neighbours. *J. Artif. Intell. Res.* **40**, 815–840 (2014)
25. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8024–8035 (2019)
26. Piho, P., Hillston, J.: Active and sparse methods in smoothed model checking. In: International Conference on Quantitative Evaluation of Systems. pp. 217–234. Springer (2021)
27. Rasmussen, C.E.: Gaussian processes in machine learning. In: Summer school on machine learning, pp. 63–71. Springer (2003)
28. Rasmussen, C.E., Candela, J.Q.: Healing the relevance vector machine through augmentation. *Proceedings of the 22nd international conference on Machine learning* (2005)
29. Silvetti, S., Policriti, A., Bortolussi, L.: An active learning approach to the falsification of black box cyber-physical systems. In: Polikarpova, N., Schneider, S.A. (eds.) Integrated Formal Methods - 13th International Conference, IFM 2017, Turin, Italy, September 20–22, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10510, pp. 3–17. Springer (2017). https://doi.org/10.1007/978-3-319-66845-1_1, https://doi.org/10.1007/978-3-319-66845-1_1
30. Titsias, M.: Variational learning of inducing variables in sparse gaussian processes. In: Artificial intelligence and statistics. pp. 567–574. PMLR (2009)
31. Titterton, D.: Bayesian methods for neural networks and related models. *Statistical science* pp. 128–139 (2004)
32. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world, vol. 29. Springer (2005)
33. Younes, H.L., Simmons, R.G.: Probabilistic verification of discrete event systems using acceptance sampling. In: International Conference on Computer Aided Verification. pp. 223–235. Springer (2002)
34. Younes, H.L., Simmons, R.G.: Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation* **204**(9), 1368–1409 (2006)
35. Zeni, G., Fontana, M., Vantini, S.: Conformal prediction: a unified review of theory and new challenges. *ArXiv abs/2005.07972* (2020)
36. Zinkevich, M., Weimer, M., Li, L., Smola, A.: Parallelized stochastic gradient descent. *Advances in neural information processing systems* **23** (2010)
37. Zuliani, P., Platzer, A., Clarke, E.M.: Bayesian statistical model checking with application to simulink/stateflow verification. In: Proceedings of the 13th ACM international conference on Hybrid systems: computation and control. pp. 243–252 (2010)