14th International Conference on Structural Safety and Reliability - ICOSSAR'25
1-6 June, 2025 - Los Angeles California, USA
R. Ghanem, E. Johnson, S. Masri and A. Olivier (Eds)

VARIATIONAL BAYESIAN MODEL UPDATING USING NORMALIZING FLOWS

FELIX METT[1], JAN GRASHORN[2] AND MICHAEL BEER[1,3,4]

[1] Institute for Risk and Reliability, Leibniz University Hannover
Callinstr. 34, 30167 Hannover, Germany
{mett, beer}@irz.uni-hannover.de

[2] Chair of Engineering Materials and Building Preservation, Helmut-Schmidt-University
Friedrich-Ebert-Damm 245, 22159 Hamburg, Germany
grashorn@hsu-hh.de

[3] Institute for Risk and Uncertainty, University of Liverpool
Liverpool L69 7ZL, 610101 Liverpool, United Kingdom

[4] International Joint Research Center for Engineering Reliability and Stochastic Mechanics
Tongji University, Siping road 1239, 200092 Shanghai, China

Key words: Bayesian updating, Variational inference, Normalizing flows, Gaussian processes

Abstract. We investigate the use of normalizing flows to approximate transport maps from tractable reference densities to complex Bayesian posterior distributions for Bayesian model updating. A Gaussian process (GP) surrogate with active sampling is used to provide a differentiable target density for optimizing the transport map. While results show normalizing flows can capture multimodal behavior in a simple example, further work is needed to refine the active sampling strategy and enable mode identification in the GP surrogate for robust multimodal density approximation.

1 INTRODUCTION

Incorporating recorded data to update model parameters enhances the robustness and reliability of mathematical models in engineering. This process is typically framed in a Bayesian context to address model uncertainty, structural inaccuracies, and observational noise.

A central challenge in Bayesian model updating is the intractability of the posterior distribution, which often involves an inaccessible normalization constant and costly likelihood evaluations. These challenges make exact inference impractical for many real-world applications, requiring approximation methods.

Approximate inference methods fall into two categories: sampling-based methods, such as Markov Chain Monte Carlo (MCMC) [1], which aim to draw samples from the posterior, and variational inference methods, which find an approximate density that is close to the posterior. For example, [2] models the posterior density using a Gaussian mixture model, parameterized with the help of a Gaussian process (GP) surrogate (see e.g. [3]), trained via active sampling. Later, [4] introduced a cyclical annealing schedule to improve the performance of this approach for multimodal posteriors.

A related method based on transport maps [5] constructs an approximate, deterministic, and invertible mapping from a reference distribution to the target posterior but depends heavily on the chosen approximation framework. The polynomial-based approach in [5], introduced in [6], struggles with multimodal posteriors. Normalizing flows [7] provide a more flexible alternative by using neural networks to construct invertible, differentiable transformations.

Training normalizing flows requires differentiable likelihood models and many gradient descent iterations. [8] proposed a method where a neural network surrogate is trained adaptively alongside the normalizing flow, enabling efficient, end-to-end differentiable inference.

This work proposes a hybrid approach that combines the GP-based surrogate posterior from [2] with the multimodal-aware active sampling strategy from [4] to train a normalizing flow-based transport map. To our knowledge, this is the first integration of active-sampling-based GP posterior surrogates with normalizing flows for Bayesian inference. By using the GP predictive mean as a differentiable approximation of the posterior, we enable efficient training of the normalizing flow. This method leverages the sample efficiency of GPs to minimize the costly and repetitive evaluation of the underlying computational model, while benefiting from the expressiveness of normalizing flows. The resulting invertible map facilitates posterior sampling and expectation computation.

Our results show that this approach effectively addresses challenges in approximating multimodal posteriors, as encountered in [5], and we highlight current limitations and suggest future improvements.

The paper is structured as follows: Section 2 introduces Bayesian model updating and transport map-based variational posterior approximation. Section 3 presents a GP-based surrogate model that utilizes active sampling and cyclical annealing for efficient posterior approximation. Section 4 discusses normalizing flows for transport map construction. Section 5 demonstrates the developed approach through two numerical examples: one with a complex synthetic posterior and another with a small physical system and multimodal posterior. Finally, Section 6 concludes the paper.

2 BAYESIAN MODEL UPDATING AND VARIATIONAL APPROXIMATION

In the following, we first introduce the general framework for Bayesian model updating problems. Subsequently, we outline a variational approximation method based on transport maps, which forms the basis for the posterior inference approach developed in this work.

2.1 Problem formulation

Let $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ denote m measured observations, and let $\mathcal{M}(\boldsymbol{\theta}) \colon \mathbb{R}^d \to \mathbb{R}^n$ represent a computational model parameterized by a d-dimensional random variable $\boldsymbol{\theta}$ that simulates the physical process underlying the data. The goal is to infer the posterior distribution of $\boldsymbol{\theta}$ given $\boldsymbol{D}$. According to Bayes' theorem, the posterior is [9]:

$$p(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{p(\boldsymbol{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{D})}, \tag{1}$$

where $p(\boldsymbol{D}|\boldsymbol{\theta})$ is the likelihood, $p(\boldsymbol{\theta})$ is the prior, and $p(\boldsymbol{D})$ is the evidence. Since $p(\boldsymbol{D})$ is often intractable, we work with the unnormalized form:

$$p(\boldsymbol{\theta}|\boldsymbol{D}) \propto p(\boldsymbol{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \tilde{p}(\boldsymbol{\theta}|\boldsymbol{D}). \tag{2}$$

Direct computation of the posterior is often impractical due to the complexity of the likelihood function, which may involve expensive simulations or high-dimensional spaces. Approximation

methods are necessary, typically categorized as sampling-based (e.g., MCMC [1]) or variational (e.g., [5, 2, 4]). In this work, we adopt a transport map-based variational approach to approximate the posterior.

2.2 Transport map based Bayesian posterior estimation

As detailed in [10], a transport map is a differentiable, invertible transformation T between two probability densities. In this work, we focus on transformations from a reference density $\rho(\boldsymbol{z})$ to the target density $\pi(\boldsymbol{\theta})$, defined by:

$$T(\boldsymbol{z}) = \boldsymbol{\theta}, \quad \boldsymbol{z} \sim \rho(\boldsymbol{z}), \quad T^{-1}(\boldsymbol{\theta}) = \boldsymbol{z}, \quad \boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}). \tag{3}$$

The densities are related by the change of variables formula [11]:

$$\rho(\boldsymbol{z}) = \pi\left(T(\boldsymbol{z})\right) \left|\det J_T(\boldsymbol{z})\right|, \quad \pi(\boldsymbol{\theta}) = \rho\left(T^{-1}(\boldsymbol{\theta})\right) \left|\det J_{T^{-1}}(\boldsymbol{\theta})\right| = \rho(\boldsymbol{z}) \left|\det J_T(\boldsymbol{z})\right|^{-1}. \tag{4}$$

Finding an exact transport map is impractical, but we can parameterize it using function approximators, such as polynomials [12, 6] or normalizing flows [7].

Following [5], transport maps can be used to approximate Bayesian posteriors by optimizing a parametric map $T(\boldsymbol{z}; \boldsymbol{\phi})$ to minimize the discrepancy with the true posterior $p(\boldsymbol{\theta}|\boldsymbol{D})$. This discrepancy is measured by the Kullback-Leibler (KL) divergence:

$$\mathcal{D}_{\mathrm{KL}}\left[\pi(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta}|\boldsymbol{D})\right] = \mathbb{E}_{\pi(\boldsymbol{\theta})}\left[\log \pi(\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}|\boldsymbol{D})\right]. \tag{5}$$

The KL divergence can be rewritten in terms of the reference density using Eq. (4):

$$\mathbb{E}_{\pi(\boldsymbol{\theta})}\left[\log \pi(\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}|\boldsymbol{D})\right] = \mathbb{E}_{\rho(\boldsymbol{z})}\left[\log \rho(\boldsymbol{z}) - \log\left|\det J_T(\boldsymbol{z}; \boldsymbol{\phi})\right| - \log p\left(T(\boldsymbol{z}; \boldsymbol{\phi})|\boldsymbol{D}\right)\right]. \tag{6}$$

Furthermore, by using the unnormalized posterior $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{D})$ (Eq. (2)), the optimization problem becomes:

$$\underset{\boldsymbol{\phi}}{\operatorname{argmin}}\ \mathbb{E}_{\rho(\boldsymbol{z})}\left[\log \rho(\boldsymbol{z}) - \log\left|\det J_T(\boldsymbol{z}; \boldsymbol{\phi})\right| - \log \tilde{p}\left(T(\boldsymbol{z}; \boldsymbol{\phi})|\boldsymbol{D}\right)\right]. \tag{7}$$

This objective simplifies to an expectation over the known reference density and only involves optimizing the map parameters $\boldsymbol{\phi}$. To efficiently compute gradients and minimize costly model evaluations, we use a surrogate model of the posterior for optimization, as discussed in the next section.

3 ACTIVE LEARNING GAUSSIAN PROCESSES FOR POSTERIOR APPROXIMATION

To approximate the posterior, we rely on GP regression to create a surrogate for the unnormalized Bayesian log-posterior $\log \tilde{p}\left(\boldsymbol{\theta}|\boldsymbol{D}\right)$ [2, 4]. This GP surrogate provides both the posterior estimate and the gradients for optimization. By leveraging the uncertainty in the GP, we use active sampling with an acquisition function based on [2] to minimize costly likelihood evaluations. Additionally, a cyclical annealing schedule helps explore multimodal distributions [4].

3.1 Gaussian process surrogate for log-posterior approximation

A GP is a collection of random variables with a mean function $m(\boldsymbol{\theta})$ and covariance function $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ [3]. We model the log-posterior $f(\boldsymbol{\theta}) = \log \tilde{p}(\boldsymbol{\theta}|\boldsymbol{D})$ as a GP:

$$f(\boldsymbol{\theta}) \sim \mathcal{GP}\left(m(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right). \tag{8}$$

Following [2], we use a negative quadratic mean function:

$$m(\boldsymbol{\theta}) = m_0 - \frac{1}{2}\sum_{i=1}^{d}\frac{(\theta_i - \theta_i^*)^2}{r_i}, \tag{9}$$

where m_0 is the maximum mean, $\boldsymbol{\theta}^*$ its location, and d the dimensionality of $\boldsymbol{\theta}$. The θ_i and θ_i^* are the components of $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$, respectively, while $\boldsymbol{r} \in \mathbb{R}_+^d$ represents the length scale parameters. The covariance kernel is a scaled squared exponential:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2\sqrt{(2\pi)^d}\left(\prod_{i=1}^{d}\sqrt{l_i}\right)\cdot\exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}')^T\Sigma_l^{-1}(\boldsymbol{\theta}-\boldsymbol{\theta}')\right), \tag{10}$$

with output scale σ_f^2 and diagonal length-scale matrix $\Sigma_l \in \mathbb{R}_+^{d\times d}$. We add small Gaussian noise to the training data to improve stability [13]:

$$\hat{f}_i = f(\hat{\boldsymbol{\theta}}_i) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2). \tag{11}$$

This noise is incorporated into the covariance kernel as [3]:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2\sqrt{(2\pi)^d}\left(\prod_{i=1}^{d}\sqrt{l_i}\right)\cdot\exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}')^T\Sigma_l^{-1}(\boldsymbol{\theta}-\boldsymbol{\theta}')\right) + \sigma_\varepsilon^2. \tag{12}$$

The GP is conditioned on noisy training data $\boldsymbol{\Xi} = \{(\hat{\boldsymbol{\theta}}_i, \hat{f}_i) | i = 1, \ldots, N\}$. Its hyperparameters are $\boldsymbol{\psi} = \left[m_0, \boldsymbol{\theta}^*, \boldsymbol{r}, \sigma_f^2, \boldsymbol{l}, \sigma_\varepsilon^2\right]$. The predictive mean $\bar{f}_{\boldsymbol{\Xi}}$ and variance $s_{\boldsymbol{\Xi}}^2 := \mathbb{V}\left[f_{\boldsymbol{\Xi}}\right]$ are given by [3]:

$$\bar{f}_{\boldsymbol{\Xi}}(\boldsymbol{\theta};\boldsymbol{\psi}) = m(\boldsymbol{\theta};\boldsymbol{\psi}) + k(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}};\boldsymbol{\psi})\left[k(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}};\boldsymbol{\psi}) + \sigma_\varepsilon^2\boldsymbol{I}\right]^{-1}\left(\hat{\boldsymbol{f}} - m(\hat{\boldsymbol{\theta}};\boldsymbol{\psi})\right), \tag{13}$$

$$s_{\boldsymbol{\Xi}}^2(\boldsymbol{\theta})(\boldsymbol{\theta};\boldsymbol{\psi}) = k(\boldsymbol{\theta}, \boldsymbol{\theta};\boldsymbol{\psi}) - k(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}};\boldsymbol{\psi})^T\left[k(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}};\boldsymbol{\psi}) + \sigma_\varepsilon^2\boldsymbol{I}\right]^{-1}k(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}};\boldsymbol{\psi}), \tag{14}$$

where $\boldsymbol{I}$ is the identity matrix. Note that the kernel and mean are represented as matrices and vectors when dictated by the input dimensionality. The optimal hyperparameters $\boldsymbol{\psi}$ are estimated by maximizing the log-marginal likelihood of the training data $\boldsymbol{\Xi}$. For details, we refer the reader to [3]. The parameters m_0 and $\boldsymbol{\theta}^*$ are excluded from the optimization procedure and set to the maximum observed unnormalized log posterior. Throughout this work, we transform the model inputs $\boldsymbol{\theta}$ into standard normal space, which allows us to use fixed initial values for the length-scale parameters. The initial values for the hyperparameters are summarized in Table 3.1.

Hyperparameter	$\boldsymbol{r}$	σ_f^2	$\boldsymbol{l}$	σ_ε^2
Initial value	1	1	1	10^{-4}

Table 1: Initial hyperparameter values for the GP-based surrogate model. For vector-valued hyperparameters, the given initial value is applied to each component.

To maximize the efficiency of building the surrogate model, selecting appropriate training data is essential. Next, we describe the active sampling approach used for this purpose.

3.2 Initial training set and active sampling for efficient refinement

We use a modified active sampling strategy inspired by [2]. The process begins with an initial training set generated using Latin Hypercube Sampling (LHS) [14], ensuring broad initial coverage of the input space. New training points $\hat{\boldsymbol{\theta}}$ are then selected by maximizing the acquisition function:

$$a(\boldsymbol{\theta}) = s^2_{\boldsymbol{\Xi}}(\boldsymbol{\theta}) \exp\left(\bar{f}_{\boldsymbol{\Xi}}(\boldsymbol{\theta})\right), \tag{15}$$

where $s^2_{\boldsymbol{\Xi}}(\boldsymbol{\theta})$ and $\bar{f}_{\boldsymbol{\Xi}}(\boldsymbol{\theta})$ denote the GP's predictive variance and mean, respectively. The exponential weighting of the mean term biases the acquisition function toward predicted high posterior density regions, while the variance term encourages exploration of uncertain regions.

To approximate the maximizer of the acquisition function, we evaluate $a(\boldsymbol{\theta})$ over a large set of randomly drawn candidates and select the top $n_a = 5$ points with the highest acquisition values for inclusion in the training set.

Although effective for unimodal posteriors, this approach may struggle with multimodal distributions, where the acquisition function can converge to a local mode. To address this, we incorporate a cyclical annealing schedule inspired by [4], which encourages broader exploration. Further details are provided in the next section.

3.3 Cyclical annealing to improve exploration of multimodal posteriors

In this section, we describe the cyclical annealing schedule introduced in [4] in more detail. Adapted from [15], this method introduces an annealing parameter $\beta_n \in [0, 1]$ defined as:

$$\beta_n = \begin{cases} 1 - \frac{\tau}{R}, & \tau \leq R \\ 0, & \tau > R, \end{cases} \tag{16}$$

where τ controls the cyclical behavior of β_n and is defined as:

$$\tau = \frac{\mathrm{mod}\ (n-1, N/M)}{N/M}. \tag{17}$$

In these expressions, $n \in 1, 2, \ldots, N$ is the iteration index, N is the total number of iterations, M is the number of annealing cycles, and $R \in [0, 1]$ determines the portion of each cycle during which β_n decreases (as per the first case of Eq. (16)). By adjusting R, we can control the balance between exploration (higher R) and exploitation (lower R) during active sampling.

To integrate cyclical annealing into the active sampling process, we map β_n to a temperature $\mathcal{T}_a(\beta_n) \in [1, 50]$, as in [4]. This temperature decreases linearly from 50 to 1 over M cycles within the total iteration count N. The unnormalized log-posterior is then modified as:

$$\log \tilde{p}_{\mathcal{T}_a}(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{1}{\mathcal{T}_a} \log \tilde{p}(\boldsymbol{\theta}|\boldsymbol{D}). \tag{18}$$

We build a GP surrogate for this annealed log-posterior and proceed with active sampling as before. The cyclical annealing schedule dynamically adjusts the exploration-exploitation balance. During high-temperature phases, the inverse of temperature $\mathcal{T}_a$ reduces exploitation of already identified modes. As the temperature reaches its minimum, the process shifts back to exploitation, focusing more on the areas already identified as promising.

This combined approach of GP surrogate modeling, active sampling, and cyclical annealing is used to approximate the unnormalized posterior $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{D})$ throughout this work. In the next section, we describe how transport maps are approximated using normalizing flows for posterior approximation.

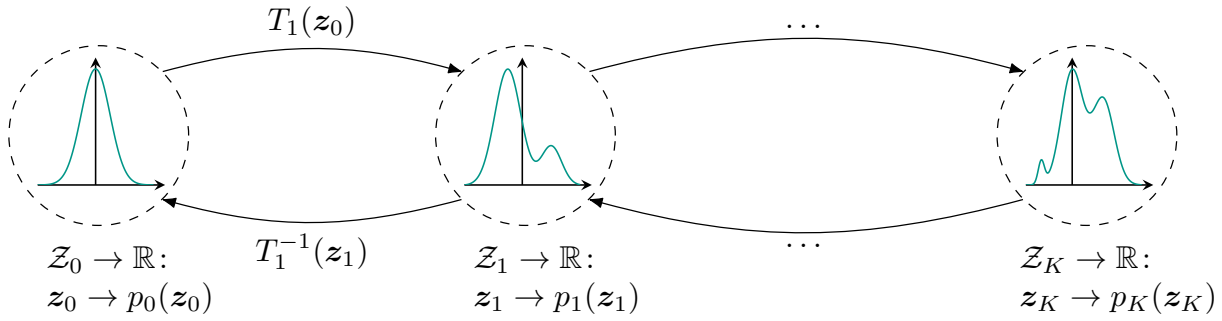

Figure 1: Illustration of a normalizing flow, where a simple reference density is progressively transformed into a complex target distribution through a series of invertible transformations. Note that $p_0(\boldsymbol{z}_0) = \rho(\boldsymbol{z})$ and $p_K(\boldsymbol{z}_K) = \pi(\boldsymbol{\theta})$.

4 NORMALIZING FLOWS FOR DENSITY APPROXIMATION

This section introduces the use of parameterized transport maps to approximate the posterior $p\left(\boldsymbol{\theta}|\boldsymbol{D}\right)$ by transforming a reference density $\rho(\boldsymbol{z})$. These maps are represented as normalizing flows, which are trained with a GP surrogate for the log-posterior. We detail the flow-based framework, the transformations employed, and the training procedure.

4.1 Normalizing flows

Normalizing flows approximate transport maps by composing differentiable, invertible transformations. These building blocks form flexible, high-dimensional mappings in a tractable way [7].

Let T be the overall transformation, composed of K transformations $T_1, \dots, T_K$:

$$T(\boldsymbol{z}_0) = T_K(\boldsymbol{z}_{k-1}) \circ \cdots \circ T_1(\boldsymbol{z}_0) = \boldsymbol{z}_K, \tag{19}$$

where each T_k maps $\boldsymbol{z}_{k-1}$ into $\boldsymbol{z}_k$, with $\boldsymbol{z}_0 = \boldsymbol{z}$ and $\boldsymbol{z}_K = \boldsymbol{\theta}$. The general idea of such a composed transformation is illustrated in Figure 1, where a simple reference density is transformed into a complex target.

To apply this transformation in the transport map framework, we need expressions for the inverse and Jacobian determinant (see Eq. (4)). The inverse of the composed transformation is:

$$\left(T_K \circ \cdots \circ T_1\right)^{-1} = T_1^{-1} \circ \cdots \circ T_K^{-1}, \tag{20}$$

and the Jacobian determinant follows from the chain rule:

$$\det J_{T_K \circ \dots \circ T_1}(\boldsymbol{z}) = \prod_{k=1}^{K} \det J_{T_k}(\boldsymbol{z}_{k-1}). \tag{21}$$

For a comprehensive review of normalizing flows, including the transformations used here, we refer the reader to [7], which also covers a detailed discussion of various architectures. In this work, we focus on two types of transformations, outlined next.

4.2 Invertible transformations used in this work

We use two types of invertible transformations: affine coupling layers [16] and activation normalization (ActNorm) layers [17]. Both transformations are affine, easily invertible, and allow efficient computation of the Jacobian determinant, making them well-suited for normalizing flows.

Affine coupling layer. Affine coupling layers split the input $\boldsymbol{x} \in \mathbb{R}^L$ into two parts: the first ℓ components remain unchanged, and the remaining $L - \ell$ components undergo an affine transformation:

$$\boldsymbol{z}_{1:\ell} = \boldsymbol{x}_{1:\ell} \tag{22}$$

$$\boldsymbol{z}_{\ell+1:L} = \boldsymbol{x}_{\ell+1:L} \odot \exp\left(\boldsymbol{s}(\boldsymbol{x}_{1:\ell})\right) + \boldsymbol{b}(\boldsymbol{x}_{1:\ell}), \tag{23}$$

where $\boldsymbol{s}$ and $\boldsymbol{b}$ are neural networks with trainable parameters $\boldsymbol{\phi}$ producing scale and translation vectors. The operator $\odot$ denotes the element-wise product. The transformation is trivially invertible, and the Jacobian determinant simplifies to:

$$\det J_T = \exp\left(\sum \boldsymbol{s}(\boldsymbol{x}_{1:\ell})\right). \tag{24}$$

To ensure all input dimensions are eventually transformed, we apply fixed random permutations between layers.

Activation Normalization. Activation Normalization (ActNorm) is a per-dimension affine transformation:

$$\boldsymbol{z} = \boldsymbol{x} \odot \exp(\boldsymbol{s}) + \boldsymbol{b}, \tag{25}$$

with learnable scale $\boldsymbol{s}$ and translation $\boldsymbol{b}$. These parameters are initialized from the first batch to standardize its mean and variance. Like affine coupling, ActNorm is efficiently invertible and contributes a simple term to the Jacobian:

$$\det J_T = \exp\left(\sum \boldsymbol{s}\right). \tag{26}$$

Together, these layers form the building blocks of our normalizing flow, offering both expressiveness and computational tractability.

Network architecture. In our implementation, the scale and translation functions $\boldsymbol{s}$ and $\boldsymbol{b}$ within each affine coupling layer are represented by a single two-layer multilayer perceptron (MLP). The MLP takes an input of size ℓ, processes it through a hidden layer with h units and ReLU activation, and outputs a vector of size $2(L - \ell)$. The overall architecture is $\ell \to h \to 2(L - \ell)$. The specific value of h is task-dependent and provided in the respective example sections.

4.3 Selection of the reference density

In normalizing flows, the reference density $\rho(\boldsymbol{z})$ is the initial distribution from which the transformation starts. It must be tractable and computationally manageable. While the reference density does not need to be trainable, we can introduce additional parameters $\boldsymbol{\vartheta}$ to allow for training.

A significant challenge arises when approximating multimodal target distributions. The Jacobian determinant in the change of variables formula (see Eq. (4)) represents the probability mass transported through the transformation, ensuring that strictly positive probability mass is maintained across the entire space. This requirement prevents assigning zero probability to regions between the modes of multimodal target distributions [7].

To address this, we use a multimodal reference density with the same number of modes as the target distribution for multimodal inference tasks. Since the number of modes is typically unknown a priori, part of our ongoing work involves leveraging the GP approximation of the posterior density to guide this process.

4.4 Training a normalizing flow on the approximate posterior

Training a normalizing flow on the approximate posterior involves using the mean of the trained GP (see Section 3), $\bar{f}_{\boldsymbol{\Xi}}(\boldsymbol{\theta})$, to express the log-posterior. Substituting this expression into Eq. (7) results in the following optimization problem:

$$\underset{\boldsymbol{\phi},\boldsymbol{\vartheta}}{\operatorname{argmin}} \, \mathbb{E}_{\rho(\boldsymbol{z};\boldsymbol{\vartheta})} \left[\log \rho(\boldsymbol{z};\boldsymbol{\vartheta}) - \log \left| \det J_T(\boldsymbol{z};\boldsymbol{\phi}) \right| - \bar{f}_{\boldsymbol{\Xi}} \left(T(\boldsymbol{z};\boldsymbol{\phi}) \right) \right], \tag{27}$$

where $\boldsymbol{\phi}$ are the learnable parameters of the normalizing flow transformations and $\boldsymbol{\vartheta}$ are the distribution parameters of the reference density. This optimization problem can be solved using stochastic gradient descent methods.

5 NUMERICAL EXAMPLES

We now demonstrate the proposed framework on two numerical examples to assess the quality of posterior approximations and highlight current limitations and future directions.

5.1 Warped Gaussian posterior

The first example approximates a two-dimensional warped Gaussian posterior with log-density:

$$\pi(\boldsymbol{\theta}) = -\frac{1}{2} \left(\left(\frac{\theta_1}{\sigma_1} \right)^2 + \left(\theta_2 \sigma_2 + \theta_1^2 \frac{b(\sigma_2 - 1)}{\sigma_2} \right)^2 \right), \tag{28}$$

where $\theta_1, \theta_2 \sim \mathcal{N}(0,1)$ are independent standard normal random variables. The involved parameters are set as $\sigma_1 = 1.1$, $\sigma_2 = 3.0$, and $b = -4.0$. Note that this example does not rely on the Bayesian theorem for formulating the posterior density, and is included solely to demonstrate the variational approximation performance of the proposed framework.

We first construct a GP approximation of the log-posterior using the procedure from Section 3. The GP training set $\boldsymbol{\Xi}$ is initialized with 10 LHS samples $\hat{\boldsymbol{\theta}}_{init}$. The total budget of further function calls is set to 160. We perform $M = 2$ annealing cycles with control parameter of $R = 0.5$, where each iteration adds $n_a = 5$ samples to the GP training set. This results in $N = 32$ total iterations to build the surrogate.

A normalizing flow is then trained to approximate the surrogate posterior density, using a fixed Gaussian reference density $\rho(\boldsymbol{z})$ with zero mean and unit covariance. The flow consists of three blocks, each containing an affine coupling layer and an ActNorm layer. The scale and translation functions in the coupling layers are modeled by a two-layer MLP with $h = 10$ hidden units, where $\ell = 1$ and $2(L - \ell) = 2$, with $L = d = 2$ (see Section 4.2). Fixed random permutations are applied between blocks to ensure all dimensions are transformed. The model, with 142 trainable parameters, is trained using the Adam optimizer [18] for 2000 iterations with a learning rate of 0.01 and batch size of 256. The first and second moment decay rates in Adam were set to $\beta_1 = 0.9$ and $\beta_2 = 0.99$, respectively.

Figure 2 shows (from left to right) the true posterior, the GP approximation with training sample locations, and the transport map output. The GP approximation identifies the region of maximum posterior density and captures the overall shape well, though the tails in low-density regions are not fully resolved. The scattered training points outside the main mode result from high-temperature phases during cyclical annealing and the initial LHS. The normalizing flow closely matches the GP approximation, indicating effective modeling, though future work should refine the active sampling strategy to better capture the tails.

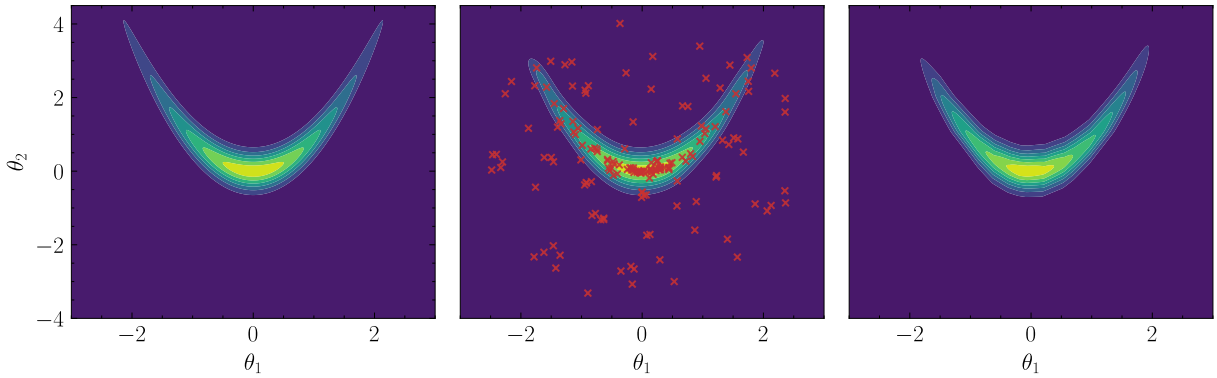

Figure 2: From left to right: True posterior, surrogate GP posterior with training sample locations (indicated by red markers), and the normalizing flow approximation of the surrogate GP posterior.

5.2 Mass-spring system

The second example considers a three-degree-of-freedom mass-spring system, previously studied in [5], which yields a multimodal posterior. The system, shown in Figure 3, comprises three masses and three springs.

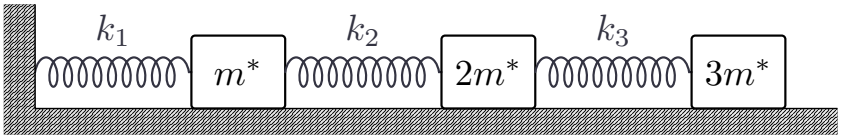

Figure 3: Three-degree-of-freedom mass-spring system. Masses are sliding freely on a frictionless surface.

The goal is to infer the spring stiffnesses k_1, k_2 and k_3 based on noisy observations of the system's natural frequencies $\boldsymbol{\omega}$, obtained by solving:

$$\det\left(\boldsymbol{K}\boldsymbol{M}^{-1} - \omega\boldsymbol{I}\right) = 0, \tag{29}$$

with

$$\boldsymbol{K} = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}, \quad \boldsymbol{M} = \begin{bmatrix} m^* & 0 & 0 \\ 0 & 2m^* & 0 \\ 0 & 0 & 3m^* \end{bmatrix}, \tag{30}$$

where $m^* = 100$. Synthetic data $\boldsymbol{D}$ were generated using $m = 50$ Monte Carlo samples from the distributions $k_1 \sim \mathcal{N}(1500, 150)$, $k_2 \sim \mathcal{N}(750, 32.5)$ and $k_3 \sim \mathcal{N}(1200, 24)$. Units are omitted for simplicity, as the focus lies on evaluating the inference method.

A uniform prior $\tilde{k}_i \sim \mathcal{U}(500, 2500)$ is assumed for each stiffness. To facilitate GP modeling, we transform parameters to standard normal space:

$$\tilde{k}_i(\theta_i) = 500 + \Phi_{std}(\theta_i) \cdot (2500 - 500), \tag{31}$$

where Φ_{std} denotes the standard normal CDF and $\boldsymbol{\theta} \in \mathbb{R}^3$ is the random variable to be inferred. Following [5], we define the likelihood as:

$$p(\boldsymbol{D}|\boldsymbol{\theta}) = \prod_{i=1}^{m} \frac{1}{\sqrt{(2\pi)^{|\boldsymbol{D}|}\det\boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{D}_i - \mathcal{M}(\tilde{\boldsymbol{k}}(\boldsymbol{\theta}))\right)^T \cdot \boldsymbol{\Sigma}^{-1} \cdot \left(\boldsymbol{D}_i - \mathcal{M}(\tilde{\boldsymbol{k}}(\boldsymbol{\theta}))\right)\right), \tag{32}$$

where $\boldsymbol{\Sigma}$ is the covariance of $\boldsymbol{D}$ and $\mathcal{M}(\bullet)$ denotes the model, specified in Eq. (29). The target log-posterior is formulated using Eq. (2).

To approximate the posterior, we again use the GP framework from Section 3, initializing the training set $\boldsymbol{\Xi}$ with 10 LHS samples $\hat{\boldsymbol{\theta}}_{init}$. The total budget of further model evaluations is 300. We performed $M = 5$ annealing cycles with a control parameter of $R = 0.5$, adding $n_a = 5$ samples per iteration through active sampling, resulting in $N = 60$ total iterations to build the surrogate.

A normalizing flow is then trained using a fixed reference density $\rho(\boldsymbol{z})$, defined as a mixture of two three-dimensional Gaussians:

$$\rho(\boldsymbol{z}) \sim 0.5 \cdot \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{C}_1) + 0.5 \cdot \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{C}_2), \tag{33}$$

with $\boldsymbol{\mu}_1 = [1, 1, 1]$, $\boldsymbol{\mu}_2 = [-1, -1, -1]$, and diagonal covariances $\boldsymbol{C}_1 = \boldsymbol{C}_2 = \text{diag}([0.3, 0.3, 0.3])$. This mixture distribution is chosen to facilitate the approximation of the multimodal posterior distribution, as discussed in Section 4.3.

The flow architecture mirrors that of the previous example, consisting of three blocks with affine coupling and ActNorm layers, interleaved with fixed random permutations. In this case, the scale and translation functions are modeled by a two-layer MLP with input size $\ell = 2$, hidden size $h = 8$, and output size $2(L-\ell) = 2$, where $L = d = 3$ (see Section 4.2). The model has 144 trainable parameters and is trained for 1000 iterations using Adam with the same hyperparameters as before.

Figure 4 shows the marginal densities of the true posterior and the transport map approximation. The GP successfully captures the bimodal structure, which is reflected in the flow-based approximation. However, similar to the previous example, the posterior remains overly concentrated in high-density regions, highlighting limitations in the current active sampling strategy. Additionally, selecting the reference density is nontrivial, particularly in multimodal settings where the number of modes is unknown a priori. Future work could explore using the GP surrogate to identify the number of modes in the posterior, enabling the adaptive construction of the reference density, such as a mixture model with the corresponding number of components.

6 CONCLUSION

We presented a variational framework for approximating Bayesian posterior distributions by combining active sampling GP regression with normalizing flows. The proposed method enables efficient sampling from complex posteriors using a surrogate-based transport map trained on adaptively acquired data. Numerical examples demonstrate the method's capability to approximate both warped and multimodal densities with a relatively small number of forward model evaluations.

While the results are promising, current limitations include insufficient exploration of low-density regions and a reliance on manual selection of the reference distribution in flow training. Future work will focus on improving the active sampling strategy to enhance coverage of low-density tails and on leveraging the GP surrogate to guide the adaptive design of mixture-based reference densities, especially in the presence of unknown multimodal structures.

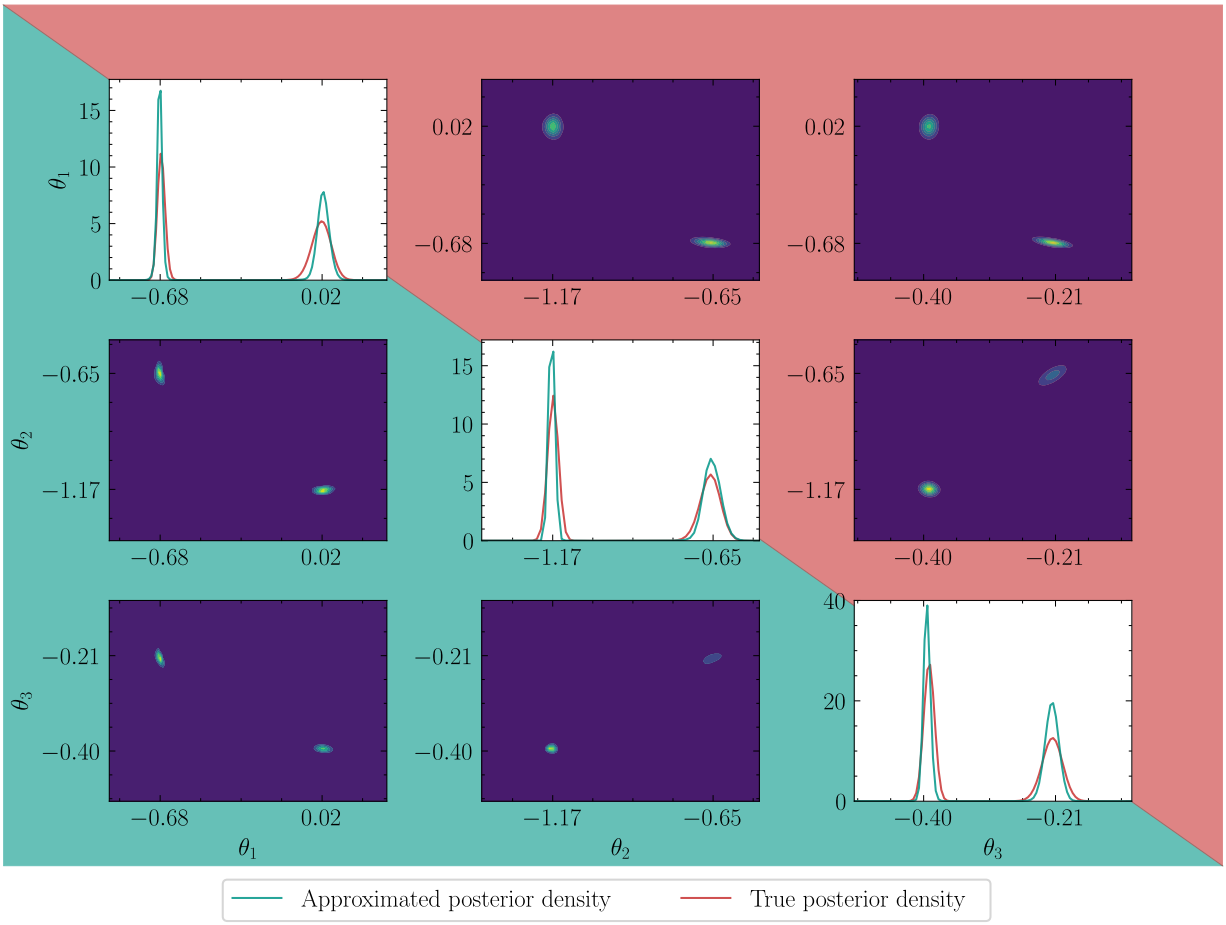

Figure 4: Comparison of target and approximated posterior densities: one-dimensional marginals are shown along the diagonal; two-dimensional marginals appear off-diagonal. Please note that plots above the diagonal are transposed versions of those below, and non-square subplot shapes may cause apparent axis rescaling, which can affect visual interpretation.

ACKNOWLEDGEMENT

Felix Mett acknowledges support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 496491159. Jan Grashorn acknowledges support from dtec.bw (Digitalization and Technology Research Center of the Bundeswehr), funded by the European Union through NextGenerationEU.

REFERENCES

[1] Beck, J.L., Au, S.-K. Bayesian Updating of Structural Models and Reliability using Markov Chain Monte Carlo Simulation. *J. Eng. Mech.* (2002) **128**(4):380–391.

[2] Acerbi, L. Variational Bayesian Monte Carlo. *Adv. Neural Inf. Process. Syst.* (2018) **31**:8222–8232.

[3] Rasmussen, C.E. and Williams, C.K.I. Gaussian Processes for Machine Learning. *The MIT Press* (2006).

[4] Igea, F., Cicirello, A. Cyclical Variational Bayes Monte Carlo for efficient multi-modal posterior distributions evaluation. *Mech. Syst. Signal Process.* (2023) **186**(4):109868.

[5] Grashorn, J., Broggi, M., Chamoin, L. and Beer, M. Efficiency comparison of MCMC and Transport Map Bayesian posterior estimation for structural health monitoring. *Mech. Syst. Signal Process.* (2024) **216**:111440.

[6] Baptista, R., Marzouk, Y. and Zahm, O. On the representation and learning of monotone triangular transport maps. *Found. Comput. Math.* (2024) **24**:2063–2108.

[7] Papamakarios, G., Nalisnick, N., Rezende, D.J., et al. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.* (2021) **22**(1):2617–2680.

[8] Wang, Y., Liu, F. and Schiavazzi, D.E. Variational inference with NoFAS: Normalizing flow with adaptive surrogate for computationally expensive models. *J. Comput. Phys.* (2022) **467**:111454.

[9] Beck, J.L., Katafygiotis, L.S. Updating models and their uncertainties. I: Bayesian statistical framework. *J. Eng. Mech.* (1998) **124**(4):455–461.

[10] Spantini , A., Bigoni , D. and Marzouk, Y. Inference via Low-Dimensional Couplings. *J. Mach. Learn. Res.* (2018) **19**:1–71.

[11] Grigoriu, M. Stochastic Calculus: Applications in Science and Engineering. *Birkhäuser Boston, MA* (2002).

[12] El Moselhy, T.A. and Marzouk, Y. Bayesian inference with optimal maps. *J. Comput. Phys.* (2012) **23**:7815–7850.

[13] Gramacy, R.B. and Lee, H.K. Cases for the nugget in modeling computer experiments. *Stat. Comput.* (2012) **22**:713–722.

[14] McKay, M. D., Beckman, R. J. and Conover, W. J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* (1979) **21**:239–245.

[15] Fu, H., Li, C., Liu, X. et al. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. *Proc. NAACL-HLT 2019* (2019) **1**:240–250.

[16] Dinh, L., Sohl-Dickstein, J. and Bengio, S. Density estimation using Real NVP. *Int. Conf. Learn. Represent.* (2017).

[17] Kingma, D.P. and Dhariwal, P. Glow: Generative flow with invertible 1×1 convolutions. *Adv. Neural Inf. Process. Syst.* (2018) **31**:10215—10224.

[18] Kingma, D.P. and Ba, J. Adam: A Method for Stochastic Optimization. *Int. Conf. Learn. Represent.* (2015)