Variational Inference with NoFAS: Normalizing Flow with Adaptive Surrogate for Computationally Expensive Models

A Preprint

Yu Wang, Fang Liu, and Daniele E. Schiavazzi

Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN, USA

April 29, 2022

Abstract

Fast inference of numerical model parameters from data is an important prerequisite to generate predictive models for a wide range of applications. Use of sampling-based approaches such as Markov chain Monte Carlo may become intractable when each likelihood evaluation is computationally expensive. New approaches combining variational inference with normalizing flow are characterized by a computational cost that grows only linearly with the dimensionality of the latent variable space, and rely on gradient-based optimization instead of sampling, providing a more efficient approach for Bayesian inference about the model parameters. Moreover, the cost of frequently evaluating an expensive likelihood can be mitigated by replacing the true model with an offline trained surrogate model, such as neural networks. However, this approach might generate significant bias when the surrogate is insufficiently accurate around the posterior modes. To reduce the computational cost without sacrificing inferential accuracy, we propose Normalizing Flow with Adaptive Surrogate (NoFAS), an optimization strategy that alternatively updates the normalizing flow parameters and surrogate model parameters. We also propose an efficient sample weighting scheme for surrogate model training that preserves global accuracy while effectively capturing high posterior density regions. We demonstrate the inferential and computational superiority of NoFAS against various benchmarks, including cases where the underlying model lacks identifiability. The source code and numerical experiments used for this study are available at `https://github.com/cedricwangyu/NoFAS`.

1 Introduction

Numerical models are increasingly used to improve our understanding of physical processes in engineering and science. They can achieve a remarkable realism, even in the simulation of complex multi-physics phenomena, but their computational cost typically increases with their complexity, and with the level of detail they are designed to provide. These models might, in addition, contain a large number of parameters that need to be carefully tuned to reproduce observed data, before *predicting* the response of the system of interest for unobserved conditions.

This two-step process of first inferring the model parameters from data and then using an optimally trained model to make predictions is essential to construct *digital twins*, i.e., predictive digital replicas of physical systems. Recent applications combining digital twins with Bayesian network-based probabilistic reasoning are discussed, for example, in [1] for damage assessment in unmanned aerial vehicles, and [2] for multi-physics simulations for hypersonic systems. A recent digital twin for computational physiology is also discussed in [3] for predicting group II pulmonary hypertension in adults.

The most expensive computational task in the development of predictive models is the solution of an inverse problem. Since the exact posterior distribution of the model parameters given a set of observations is not available in closed form, posterior sampling is often employed to approximate the posterior distribution numerically, using approaches such as Approximate Bayesian Computation (ABC) [4] and, in the presence of a tractable likelihood, Markov chain Monte Carlo sampling (MCMC) [5]. For complicated posteriors with multiple modes or ridges, advanced MCMC with adaptive proposal distribution were proposed in [6, 7], with more recent approaches being the Hamiltonian Monte Carlo [8], the no U-turn sampler [9] and the differential evolution adaptive Metropolis (DREAM) algorithm [10, 11] to cite a few.

As an alternative to sampling-based approaches, variational inference (VI) [12, 13, 14] leverages numerical optimization to determine the member of a parametric family of distributions that is the closest to a desired posterior in some sense (e.g., the Kullback-Leibler divergence). VI can be combined with stochastic optimization to improve its efficiency for large datasets [15], but still required significant model-specific analysis. To overcome this problem, Black-Box VI (BBVI [16]) was developed to support a larger class of models, leveraging Monte Carlo estimates of the evidence lower bound (ELBO) and reparameterized gradients with lower variance [17, 18, 19, 20]. However, a common approximation made by BBVI is the mean field assumption, which enforces an a-priori independence among groups of parameters, and may introduce significant bias in the distributional approximation, when the independence assumption does not hold. To go beyond the mean field assumption, many approaches have been proposed using, e.g., Gaussian mixture models [14], hierarchical variational models [21], CRVI [22], copula VI [23] and others (see, e.g., [24]), but they often rely on strong parametric assumption when introducing a dependency structure.

Normalizing flows was proposed in [25]. It can be used for VI via a composition of invertible transformations with easily computable Jacobian determinants. Simple flows were initially proposed such as planar and radial flows [25], followed by Inverse Autoregressive Flow (IAF [26]), real-valued non volume preserving transformations (RealNVP [27]), masked autoregressive flow (MAF [28]), generative flow (GLOW [29]), among others. Readers may refer to [30] for an extensive overview on NF.

For both sampling-based posterior inference or approximate inference via VI through NF, the likelihood function given the assumed model and data need to be repeatedly evaluated for new posterior samples, which becomes quickly infeasible, in practice, for computationally expensive models. One of the solutions is to approximate the models using a surrogate $\hat{f}$ which is generally obtained through a three-phase process, i.e., input dimensionality reduction, design of experiments, and formulation of a surrogate from an appropriate family (see, e.g., [31]). Reduction of the inputs dimensionality can be achieved through, e.g., principal component analysis [32], variable screening [33, 34, 35], sensitivity analysis [36, 37, 38], and Bayesian updating [39, 40]. Sampling locations can be selected through factorial design [41], central composite design [42] and orthogonal arrays [43], among others. Finally, possible surrogate formulations include response surface analysis [44], Kriging [45], radial basis functions [46], boosting trees and random forests [47], adaptive and active learning [48, 49], among others. Popular non-intrusive approaches used in computational mechanics include the generalized polynomial chaos expansion (gPC) [50, 51], stochastic collocation on tensor isotropic and anisotropic quadrature grids [52, 53], multi-element adaptive gPC [54], hierarchical sparse grids [55], simplex stochastic collocation [56], sparsity-promoting gPC [57], generalized multi-resolution expansion [58, 59] and, more recently, deep neural networks [60]. Finally, other approaches in the literature have proposed the combination of an inference task and an adaptive surrogate model. For example, a local approximant is combined with MCMC in [61, 62, 63] and a Gaussian process surrogate coupled with Hamiltonian Monte Carlo sampling is presented in [64] with applications to one-dimensional hemodynamics.

In this work, we combine NF and surrogate modelling and propose a new method – Normalizing Flow with Adaptive Surrogate (NoFAS) – for variational inference with computationally expensive models. NoFAS is a general approach for Bayesian inference and uncertainty quantification, designed to sample from complex or high-dimensional posterior distributions with significantly reduced computational cost. This is an alternated optimization algorithm, where a surrogate is adaptively refined using posterior samples which are, in turn, computed by optimizing the NF parameters with respect to a loss function that depends on the surrogate. Any expressive NFs can be used, and we find the autoregressive NFs to be effective in approximating even complicated posterior distributions. Our contributions are listed below.

- We combine variational inference, normalizing flow and surrogate modelling in a data efficient and computationally affordable framework to obtain inference on true model parameters.
- We propose an efficient sample weighting scheme for the loss function that remembers samples in the pre-grid (providing global surrogate accuracy) as well as more recent samples acquired during the NF iterations. Older samples acquired during the early stages of VI are instead progressively forgotten.
- We demonstrate the application of NoFAS in multiple experiments with different types of models, and showcase its advantages over the fixed-surrogate model approach, MCMC and BBVI with the mean-field assumption, for indentifiable and non-identifiable models.

The paper is organized as follows. In Section 2, we review normalizing flow, surrogate modeling and present our NoFAS approach. In Section 3, we apply NoFAS for the solution of inverse problems in four numerical experiments, and compare its performance in posterior inference to an approach where the surrogate is kept fixed, MH, and BBVI. We provide some discussions and final remarks in Section 4.

Table 1: List of mathematical symbols.

Symbol	Description	Symbol	Description
b	Batch size for NF.	M	Total size of the adaptive samples stored.
$b_1, \dots, b_{L+1}$	Bias terms in MADE.	n	Total number of observations.
β_0	Pre-grid weight factor.	p	Target density function.
β_1	Memory decay factor. samples in loss.	$\pi(\cdot)$	Prior distribution.
c	Calibration frequency.	$q_k(\boldsymbol{z}_k)$	Density function of $\boldsymbol{z}_k$.
d	Latent space dimensionality.	φ_F, φ_S	Optimizers for NF and surrogate update.
$D(\cdot\|\cdot)$	KL-Divergence.	$\sigma_j,\ j = 1, \cdots, m$	Prescribed standard deviations in log-likelihood.
f_{μ_i}, f_{α_i}	MADE networks in MAF.	σ	Soft-max activation.
f	True model.	S_0	Pre-grid size.
$\hat{f}$	Surrogate model.	T_F	Total number of NF iterations.
F_k, F	NF bijections and their composition.	T_S	Total number of surrogate update iterations.
$h_1, \dots, h_{L+1}$	Activation functions in MAF.	V, M^V, W, M^W	MAF: Mask matrices used in MAF.
G	Queue for sample storage for calibration.	$\boldsymbol{x}, X$	Observations and and their space.
k, K	Current and total number of NF layers.	$\boldsymbol{z}, Z$	Latent variables and their space.
		$\boldsymbol{y}$	Variables in hidden MADE layer.
$\ell(\boldsymbol{z}; f, \boldsymbol{x})$	Likelihood function.	Z_G	Adaptively selected training set for $\hat{f}$
$\lambda, \lambda_k, \Lambda, \Lambda_k$	NF parameters and their space.	Z_P	Pre-grid.
$L_j(\cdot)$	Loss function.	$\{z_K^{(i)}\}_{i=1}^b$	Batch samples.
$m^i(k)$	functions returning an integer from 1 to d.	$\boldsymbol{z}_k$	Output from the k-th NF layer.
m	Output space dimensionality, $\boldsymbol{x} \in \mathbb{R}^m$.	ω^*	Optimal surrogate model parameters.

2 Method

In this section, we first review the framework of NF for VI, focusing on MAF, and RealNVP in Section 2.1. We then discuss a surrogate model formulation for computationally expensive models in Section 2.2. We introduce NoFAS in Section 2.3, along with its algorithm. The notation introduced in this section is summarized in Table 1.

2.1 Autoregressive Normalizing Flow

NF uses the map $F : \mathbb{R}^d \times \Lambda \to \mathbb{R}^d$ with parameters $\lambda \in \Lambda$ to transform realizations from an easy-to-sample distribution, such as $\boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{0}, I_d)$, to samples from a desired *target* distribution.

Specifically, F is obtained as a composition of K bijections $F_k : \mathbb{R}^d \times \Lambda_k \to \mathbb{R}^d$, each parameterized by $\lambda_k \in \Lambda_k$: $F(\boldsymbol{z}_0; \lambda) = [F_K(\,\cdot\,; \lambda_K) \circ F_{K-1}(\,\cdot\,; \lambda_{K-1}) \circ \cdots \circ F_1(\,\cdot\,; \lambda_1)]\,(\boldsymbol{z}_0)$, where $\boldsymbol{z}_k = F_k(\boldsymbol{z}_{k-1}; \lambda_k)$ for $k = 1, \dots, K$. Since $F_k(\,\cdot\,; \lambda_k)$ is a bijection from $\boldsymbol{z}_{k-1}$ to $\boldsymbol{z}_k$, $q_k(\boldsymbol{z}_k)$, the distribution of $\boldsymbol{z}_k$, can be obtained by the change of variable

$$q_k(\boldsymbol{z}_k) = q_{k-1}(\boldsymbol{z}_{k-1}) \left|\det \frac{\partial F_k^{-1}}{\partial \boldsymbol{z}_{k-1}}\right| = q_{k-1}(\boldsymbol{z}_{k-1}) \left|\det \frac{\partial F_k}{\partial \boldsymbol{z}_{k-1}}\right|^{-1}. \tag{1}$$

Taking the logarithm and summing over k, Eqn. (1) becomes

$$\log q_K(\boldsymbol{z}_K) = \log q_0(\boldsymbol{z}_0) - \sum_{k=1}^{K} \log \left| \det \frac{\partial F_k}{\partial \boldsymbol{z}_{k-1}} \right| . \tag{2}$$

The goal is to determine an *optimal* set of parameters $\lambda^* = (\lambda_1^*, \lambda_2^*, \cdots, \lambda_K^*) \in \Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_K = \Lambda$ so the density q_K can approximate a target density p. A commonly used objective (loss) function to achieve this goal is the *flow-based free energy bound*, expressed as

$$\begin{aligned}\mathcal{F}(\boldsymbol{x}) &= \mathbb{E}_{q_K(\boldsymbol{z})}\left[\log q_K(\boldsymbol{z}) - \log p(\boldsymbol{x}, \boldsymbol{z})\right] = \mathbb{E}_{q_0(\boldsymbol{z}_0)}\left[\log q_K(\boldsymbol{z}_K) - \log p(\boldsymbol{x}, \boldsymbol{z}_K)\right] \\ &= \mathbb{E}_{q_0(\boldsymbol{z}_0)}[\log q_0(\boldsymbol{z}_0)] - \mathbb{E}_{q_0(\boldsymbol{z}_0)}[\log p(\boldsymbol{x}, \boldsymbol{z}_K)] - \mathbb{E}_{q_0(\boldsymbol{z}_0)}\left[\sum_{k=1}^{K} \log \left| \det \frac{\partial F_k}{\partial \boldsymbol{z}_{k-1}} \right| \right] .\end{aligned} \tag{3}$$

The expectations in Eqn. (3) are approximated by their Monte-Carlo (MC) estimates using samples $\boldsymbol{z}_0$ from the basic distribution q_0. However, the computation of the Jacobian determinants in Eqn. (3) may be computationally intensive, especially when using a large number of bijections. To efficiently compute the determinants, the coupling layer-based NF with block-triangular Jacobian matrices (RealNVP [27] and GLOW [29]), and autoregressive transformation-based NF with lower-triangular Jacobian matrices (MAF [28] and IAF [26]) have been proposed. In this study, we focus on autoregressive transformations in NF.

According to the chain rule, the joint distribution $p(\boldsymbol{z})$ can be written as $p(\boldsymbol{z}) = p_1(z_1) \prod_{i=2}^{d} p_i(z_i | z_1, \ldots, z_{i-1})$. If the components of $\boldsymbol{z}$ are not independent, the autoregressive flow can be applied to capture their dependency. For example, MAF [28] uses $p(z_i | z_1, \ldots, z_{i-1}) = \phi((z_i - \mu_i)/e^{\alpha_i})$, where ϕ is the density function of the standard normal distribution, $\mu_i = f_{\mu_i}(z_1, \ldots, z_{i-1})$, $\alpha_i = f_{\alpha_i}(z_1, \ldots, z_{i-1})$, and f_{μ_i} and f_{α_i} are masked autoencoder neural networks (MADE [65]). Let $\boldsymbol{z}$ be the input and $\hat{\boldsymbol{z}}$ be the output of a MADE network having L hidden layers with d_l nodes per layer, for $l = 1, \cdots, L$. The mappings between the input and the first hidden layer, among hidden layers, and from the last hidden layer to the output are, respectively

$$\begin{cases} \boldsymbol{y}_1 &= h_1(b_1 + (\boldsymbol{W}^1 \odot \boldsymbol{M}^1)\boldsymbol{z}), & \text{where } \boldsymbol{M}^1 \text{ is } d_1 \times d \text{ and } \boldsymbol{M}^1_{u,v} = \mathbb{1}_{m^1(u) \geq v} \\ \boldsymbol{y}_l &= h_l(b_l + (\boldsymbol{W}^l \odot \boldsymbol{M}^l)\boldsymbol{y}_{l-1}), & \text{where } \boldsymbol{M}^l \text{ is } d_l \times d_{l-1} \text{ and } \boldsymbol{M}^l_{u,v} = \mathbb{1}_{m^l(u) \geq m^{l-1}(v)} \\ \hat{\boldsymbol{z}} &= h_{L+1}(b_{L+1} + (\boldsymbol{W}^{L+1} \odot \boldsymbol{M}^{L+1})\boldsymbol{y}_L) & \text{where } \boldsymbol{M}^{L+1} \text{ is } d \times d_L \text{ and } \boldsymbol{M}^{L+1}_{u,v} = \mathbb{1}_{u > m^L(v)}, \end{cases}$$

where h_l is the activation function between layer $l-1$ and l for $l = 1, \cdots, L+1$, $m^l(k)$ is a pre-set or random integer from 1 to $d-1$, b_l and $\boldsymbol{W}^l$ are the bias and weight parameters for $l = 1, \ldots, L+1$.

The matrix $\prod_{l=L+1}^{1} \boldsymbol{M}^l$, encoding the dependence between the components of the input $\boldsymbol{z}$ and output $\hat{\boldsymbol{z}}$, is strictly lower diagonal, thus satisfying the sought autoregressive property [65]. The masks $\boldsymbol{M}^1, \ldots, \boldsymbol{M}^{L+1}$ enable the computation of all μ_i and α_i in a single forward pass [28]; additionally, the Jacobian of each MAF layer is lower-triangular, with determinant equal to $|\det \partial f / \partial \boldsymbol{z}|^{-1} = \exp(\sum_{i=1}^{d} \alpha_i)$.

Between consecutive autoregressive layers and after the basic distribution, batch normalization [66] can be used to normalize the outputs from the previous layer so they have approximately zero mean and unit variance [28]. Batch normalization accelerates training by reducing the oscillations in the magnitudes of the flow parameters between consecutive layers, without complicating the computation of the Jacobian determinant in Eqn. (3). A batch normalization layer with input $\boldsymbol{z}$ and output $\hat{\boldsymbol{z}}$, parameterized through $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, is

$$\hat{\boldsymbol{z}} = F^B(\boldsymbol{z}) = \boldsymbol{\beta} + (\boldsymbol{z} - \boldsymbol{m}) \odot (\boldsymbol{v} + \epsilon)^{-1/2} \odot e^{\boldsymbol{\gamma}} \text{ and } \left| \det \frac{\partial F^B}{\partial \boldsymbol{z}} \right| = \exp \left[\sum_i \left(\gamma_i - \frac{1}{2} \log(v_i + \epsilon) \right) \right],$$

where $\boldsymbol{m}$ and $\boldsymbol{v}$ refer to the sample mean and variance of $\boldsymbol{z}$, respectively, and the tolerance ϵ (e.g. 10^{-5}) ensures numerical stability if $\boldsymbol{v}$ has near zero components.

Besides MAF, RealNVP is another commonly used auto-regressive flow of the form

$$\begin{cases} \hat{\boldsymbol{z}}_i = \boldsymbol{z}_i & \text{for } i \leq d' \\ \hat{\boldsymbol{z}}_i = \boldsymbol{z}_i \odot e^{\boldsymbol{\alpha}} + \boldsymbol{\mu} & \text{for } d' < i \leq d \end{cases},$$

where $\boldsymbol{\mu} = f_\mu(z_1, \ldots, z_{d'})$ and $\boldsymbol{\alpha} = f_\alpha(z_{d'+1}, \ldots, z_d)$. The output $\hat{\boldsymbol{z}}$ consists of identical copies of the input $\boldsymbol{z}$ for the first $d' < d$ elements, while the remaining $d - d'$ components are transformed by the MADE autoencoders f_μ and f_α. MAF could be seen as a generalization of RealNVP by setting $\mu_i = \alpha_i = 0$ for $i \leq d'$ [28]. We employ both RealNVP and MAF in our experiments in Section 3.

2.2 Surrogate Likelihood for Computationally Expensive Models

Consider a black-box model as a generic map $f : Z \to X$ between the random inputs $\boldsymbol{z} = (z_1, z_2, \cdots, z_d)^T \in Z$ and the outputs $(x_1, x_2, \cdots, x_m)^T \in X$, and assume n observations $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^n \subset X$ to be available. Without loss of generality, we assume $\boldsymbol{x}$ to come from a Gaussian distribution with mutually independent components (other distributional assumptions can be made, depending on the specific problem). With the Gaussian distribution assumptions, the log-likelihood of $\boldsymbol{z}$ given $\boldsymbol{x}$ is

$$\ell(\boldsymbol{z}; f, \boldsymbol{x}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{f(\boldsymbol{z})_j - x_{ij}}{\sigma_j}\right)^2 - \frac{n \cdot m}{2}\log(2\pi) - n\sum_{j=1}^{m}\log(\sigma_j).$$

Our goal is to infer $\boldsymbol{z}$ and to quantify its uncertainty given $\mathbf{x}$. We employ a variational Bayesian paradigm and sample from the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x}) \propto \ell(\boldsymbol{z}; f, \boldsymbol{x})\pi(\boldsymbol{z})$, with prior $\pi(\boldsymbol{z})$ via NF. VI-NF requires the evaluation of the gradient of the cost function in Eqn. (3) with respect to the NF parameters λ, replacing $p(\boldsymbol{x}, \boldsymbol{z}_K)$ with $p(\boldsymbol{x}|\boldsymbol{z}_K)\pi(\boldsymbol{z}_k) = \ell(\boldsymbol{z}_K; f, \boldsymbol{x})\pi(\boldsymbol{z}_k)$, and approximating the expectations with their MC estimates. However, the likelihood function needs to be evaluated at every MC realization, which can be costly if the model $f(\boldsymbol{z})$ is computationally expensive. In addition, automatic differentiation through a legacy (e.g. physics-based) solver may be an impractical, time-consuming, or require the development of an adjoint solver.

One solution is to replace the model f with a computationally inexpensive surrogate $\hat{f} : Z \times \Omega \to X$ parameterized by $\omega \in \Omega$, whose derivatives can be obtained at a relatively low computational cost, but intrinsic bias in the selected surrogate formulation (i.e., $f(\boldsymbol{z}) \notin \{\hat{f}(\boldsymbol{z}; \omega) | \forall \omega \in \Omega\}$), a limited number of training examples, and locally optimal ω can compromise the accuracy of $\hat{f}$.

To resolve these issues, we propose to update the surrogate model adaptively by smartly weighting the samples of $\mathbf{z}$ from NF. Once a newly updated surrogate is obtained, the likelihood function is updated, leading to a new posterior distribution that will be approximated by VI-NF, producing, in turn, new samples for the next surrogate model update, and so on. In the next section, we will introduce this new approach in detail and provide its algorithm.

2.3 Variational Inference via Normalizing Flow with an Adaptive Surrogate (NoFAS)

The computational cost of training a surrogate model with an uniform accuracy over the entire parameter space Z grows exponentially with the problem dimensionality (see, e.g., [42]). In such a case, many training samples would correspond to model outputs that are relatively far from the available observations $\mathbf{x}$ and contribute minimally to learning the posterior distribution of $\mathbf{z}$ given $\mathbf{x}$, leading to a massive waste of computational resources.

Motivated by this observation, we propose a strategy to alternate gradient-based updates for the surrogate model parameters ω and NF parameters λ. Once every c normalizing flow iterations (*calibration frequency*), a sub-sample from the *batch* $\{\boldsymbol{z}_K^{(s)}\}_{s=1}^b$ (used in Eqn. (3) to evaluate the MC expectations) is used to provide additional training samples to *adaptively* improve the surrogate $\hat{f}$. This approach, referred to as VI using Normalizing Flow with an Adaptive Surrogate (NoFAS for short), is illustrated in Figure 1 and its algorithmic steps are presented in Algorithm 1.

NoFAS starts with an initial surrogate model trained from an a-priori selected *pre-grid*, $Z_P = \{\boldsymbol{z}_P^{(s)}\}_{s=1}^{S_P}$ of realizations from Z (typically tensor product grids or low-discrepancy sequences [67]). At every $(j \cdot c)$, $j \in \mathbb{N}$ flow parameter update, $S_G < b$ realizations $Z_{G,j} = \{\boldsymbol{z}_{G,j}^{(t)}\}_{t=1}^{S_G}$ are sub-sampled at random from the batch $\{\boldsymbol{z}_K^{(s)}\}_{s=1}^b$. The solutions from the true model $f(Z_{G,j})$ are computed and used to update the surrogate model $\hat{f}$ on the training set

$$Z_T = \left(\cup_{\alpha=\max(1,j-M+1)}^{j} Z_{G,\alpha}\right) \cup Z_P. \tag{4}$$

Additionally, the sequentially collected samples carry different weights in the loss function for the surrogate model update; more recently collected parameter samples Z_G receive a larger weight than those collected earlier so to achieve better accuracy around regions of high posterior density progressively discovered by NF. Larger weights are also assigned to the pre-grid realizations Z_P, since they are responsible to ensure some *generalizability* of the trained surrogate over the parameter space Z, rather than just capturing local features. Taken together, if a l_2 norm is used

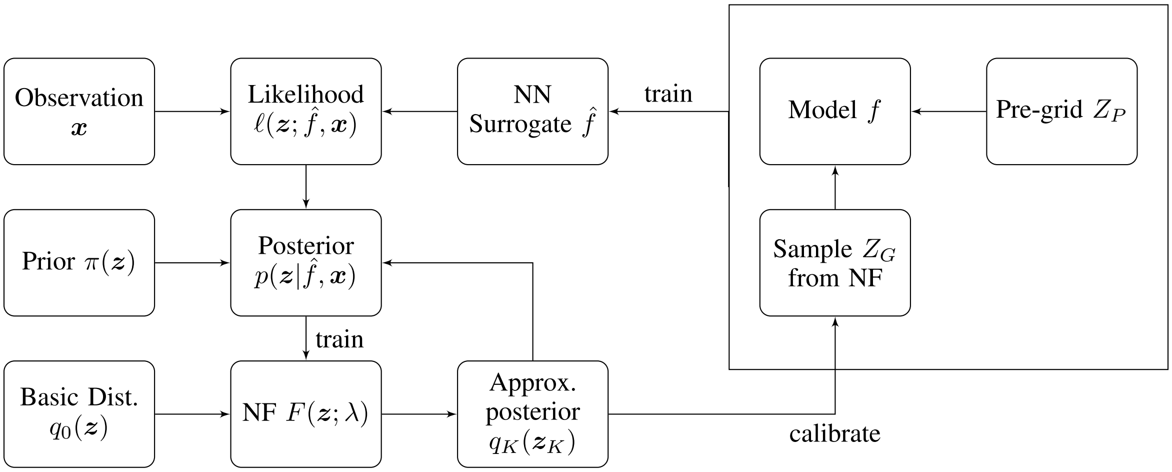

Observation $\boldsymbol{x}$
Likelihood $\ell(\boldsymbol{z};\hat{f},\boldsymbol{x})$
NN Surrogate $\hat{f}$
train
Model f
Pre-grid Z_P
Prior $\pi(\boldsymbol{z})$
Posterior $p(\boldsymbol{z}|\hat{f},\boldsymbol{x})$
Sample Z_G from NF
train
Basic Dist. $q_0(\boldsymbol{z})$
NF $F(\boldsymbol{z};\lambda)$
Approx. posterior $q_K(\boldsymbol{z}_K)$
calibrate

Figure 1: Illustrative Diagram for the NoFAS algorithm.

(but other types of loss can be chosen), the surrogate model optimization loss would take the form

$$\begin{aligned} L_j(\beta_0,\beta_1,\omega,Z_P,Z_G) &= \beta_0 \sum_{s=1}^{S_P} \|\hat{f}(\boldsymbol{z}_P^{(s)};\omega) - f(\boldsymbol{z}_P^{(s)})\|_2^2/S_P + \\ +(1-\beta_0) &\sum_{\alpha=\max(1,j-M+1)}^{j} \sum_{s=1}^{S_G} \sigma(\exp(-\beta_1(j-\alpha))\,\|\hat{f}(\boldsymbol{z}_{G,\alpha}^{(s)};\omega) - f(\boldsymbol{z}_{G,\alpha}^{(s)})\|_2^2/S_G, \end{aligned} \tag{5}$$

where M denotes the *memory* of the proposed adaptive scheme, i.e., the number of the more recent realizations in Z_G included in the loss, β_0 represents the weight assigned to the pre-grid realizations, $\beta_1 > 0$ defines the rate of exponentially decaying weights for $Z_{G,\alpha}$, $\alpha = \max(1, j-M+1),\ldots,j$ (with realizations ordered from the most recent j to the least recent $\max(1, j-M+1)$) and $\sigma(\cdot)$ is the softmax function.

In our numerical experiments, the behaviors of batch $\{\boldsymbol{z}_K^{(s)}\}_{s=1}^{b}$ could be observed to evolve in three phases. In the first phase, the samples are aggregated in clusters occupying a small region of the parameter space. In the second phase, the aggregated clusters move together to a region of high posterior density, followed by a third stage where they are scattered to better cover the posterior distribution. Batch sub-samples extracted in the first two phases can be similar in some cases and lack diversity to provide a good characterization of the local response of the true model f, negatively affecting convergence. Additionally, evaluating the true model at almost identical inputs (parameter samples) is a waste of computational resources. A possible remedy is to perturb the parameter realizations by injecting Gaussian noise, before storing them in $Z_{G,j}$. For the experiments presented in Section 3, we added Gaussian noise $\mathcal{N}(0,\varepsilon^2)$ if the batch has a standard deviation $< \varepsilon$ and we used $\varepsilon = 0.1$. We conjecture that several factors affect the choice of ε, such as the prior knowledge about the parameters and the local curvature of the true model response;

future work is needed to better understand this phenomenon and to develop a systematic approach for choosing ε.

Algorithm 1: NoFAS algorithm.

```
input  : Model $f$, observations $\boldsymbol{x}$, batch size $b$, calibration frequency $c$ and size $S_G$
Generate initial surrogate model training set $Z_P = \{\boldsymbol{z}_P^{(s)}\}_{s=1}^{S_0}$ from prior $\pi(\boldsymbol{z})$;
Let $\omega_0 = \arg\min_{\omega\in\Omega} L(\omega; Z_P)$ and initialize surrogate $\hat{f}(\cdot\,;\omega_0)$;
Initialize† flow parameters $\lambda_0$ and set $t = 0$;
Initialize hyper-parameters (e.g. learning rate $\eta_0$, learning scheduler) for optimizer $\psi_F(\lambda, \mathcal{F}(\boldsymbol{x}))$ over $\lambda$ that
  minimizes the free energy bound in Eq. (3);
while $t \leq T_F$ or $|\mathcal{F}_t(\boldsymbol{x}) - \mathcal{F}_{t-1}(\boldsymbol{x})| > \epsilon$ do
    Obtain a batch of samples $\{\boldsymbol{z}_0^{(s)}\}_{s=1}^{b}$ from a basic distribution (e.g., $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$);
    Compute $\boldsymbol{z}_K^{(s)} = F(\boldsymbol{z}_0^{(s)}; \lambda_t),\ s = 1, \dots, b$;
    if ($t$ mod $c$ == $0$) then
        Randomly draw a subset of $\{\boldsymbol{z}_K^{(s)}\}_{s=1}^{b}$ and push it into $Z_{G,t}$;
        Reset the parameters of the scheduler‡ for optimizer $\psi_S(\omega, L_t(\beta_0, \beta_1, \omega, Z_P, Z_G))$ over $\omega$, minimizing
          the surrogate loss in Eq. (5);
        for $\tau = 0, 1, \cdots, T_S$ do
            $\omega_{\tau+1} \leftarrow \psi_S^{(t)}(\omega_\tau, L_t(\beta_0, \beta_1, \omega, Z_P, Z_G))$;
    Update the likelihood $\ell(\boldsymbol{z}_K^{(s)}; \hat{f}(\cdot\,;\omega_{T_S}), \boldsymbol{x})$ for $s = 1, \cdots, b$;
    $\lambda_{t+1} \leftarrow \psi_F(\lambda_t, \mathcal{F}(\boldsymbol{x}))$;
    $t \leftarrow t + 1$;
```

† We used the uniform Glorot initialization [68] in the experiments in Section 3. However, other initialization schemes, such as Kaiming Uniform [69] can also be used.

‡The learning rate scheduler parameter reduces the learning rate at every iteration. Our algorithm *resets* the learning rate back to η_0 at every surrogate model update.

3 Experiments

In this section, we run four numerical experiments to demonstrate the application and performance of NoFAS in parameter estimation, for models formulated as algebraic or differential equations. The first two experiments have identifiable model parameters, the model in the third experiment has highly correlated parameters, and we purposely design an over-parameterized model in the fourth experiment to examine the robustness of the NoFAS procedure for variational inference. In each of the four experiments, we employed a fully connected neural network with two hidden layers with 64 and 32 nodes, respectively, as the surrogate model, and set $\beta_0 = 0.5$ and $\beta_1 = 0.1$. We investigated numerically the sensitivity of NoFAS to different choices of β_0 and β_1 and the results are presented in Appendix D.1. The results suggest that NoFAS performs well for $\beta_0 \in [0.5, 0.7]$ and $\beta_1 \in [0.01, 1.0]$. We also compare NoFAS with MH, BBVI, and NF with a fixed surrogate model, evaluating their performance based on the accuracy in the recovered posterior and predictive posterior distributions, and the computational cost savings as measured by the number of true model evaluations.

3.1 Experiment 1: Model with Closed-form Solution

The output from model $f : \mathbb{R}^2 \to \mathbb{R}^2$ in this experiment has the closed-form expression

$$f(\boldsymbol{z}) = f(z_1, z_2) = (z_1^3/10 + \exp(z_2/3), z_1^3/10 - \exp(z_2/3))^T. \tag{6}$$

Observations $\widetilde{\boldsymbol{x}}$ are generated as

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x}^* + 0.05\,|\boldsymbol{x}^*| \odot \boldsymbol{x}_0, \tag{7}$$

where $\boldsymbol{x}_0 \sim \mathcal{N}(0, \boldsymbol{I}_2)$. We set the *true* model parameters at $\boldsymbol{z}^* = (3, 5)^T$, with output $x^* = f(\boldsymbol{z}^*) = (7.99, -2.59)$, and simulated 50 sets of observations from Eqn. (7). The likelihood of $\boldsymbol{z}$ given the observed data $\widetilde{\boldsymbol{x}}$ is Gaussian and we adopt a noninformative uniform prior $\pi(\boldsymbol{z})$.

For surrogate model estimation, we set the maximum number of true model evaluations – referred to as the *budget* – at 64, and examine two scenarios for allocating the budget. In the first scenario, the entire budget is assigned to a 2-dimensional $8 \times 8 = 64$ pre-grid Z_p. The surrogate $\hat{f}$ is learned from Z_P only, and never updated. In the second scenario, we allocate a budget of $4 \times 4 = 16$ model solutions to Z_P and use the rest to calibrate $\hat{f}$ with $S_G = 2$ samples every $c = 1000$ NF iterations. We refer to these two scenarios as the *fixed* and *adaptive* surrogate, respectively, where the latter coincides with NoFAS. A RealNVP architecture is employed, alternating 5 batch normalization layers and 5 linear masked coupling layers having 1 hidden layer of 100 neurons. We run the RMSprop optimizer with a learning rate of 0.03 and an exponential decay coefficient of 0.9995 to update λ.

The results are presented Figure 2. The adaptive surrogate is less biased and leads to an improved quantification of uncertainty compared to the fixed surrogate. By better capturing f locally around $\boldsymbol{z}^*$ (third column in Figure 2), the adaptive surrogate also generates a posterior predictive distribution that agrees with the observations (second column in Figure 2).

We also run the MH algorithm to obtain the posterior samples on $\boldsymbol{z}$ and present the results in Figure 3. MH used 4×10^6 iterations, resulting in 3600 effective samples using a burn-in and thinning rate of 10% and 1/1000, respectively. The accuracy of MH in capturing the posterior and posterior predictive distributions is similar to NoFAS, but MH evaluated the true model 4×10^6 times compared to only 64 times for NoFAS.

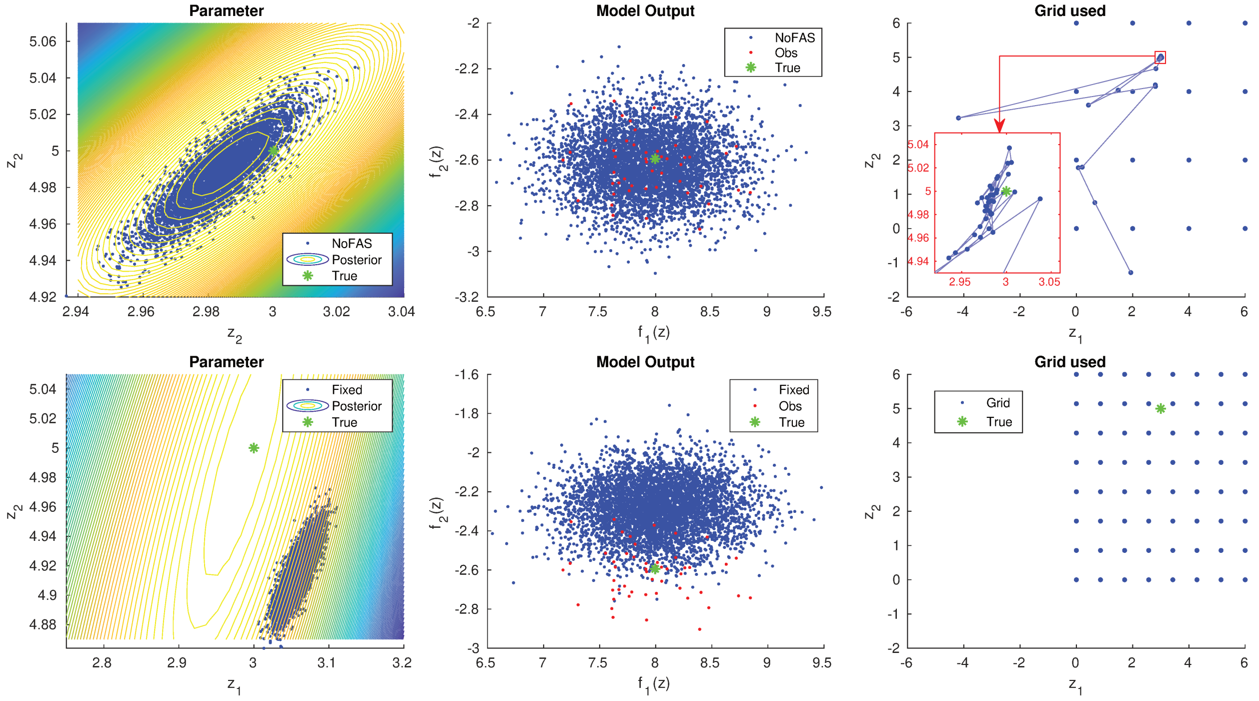


Figure 2: Parameter samples from the posterior distribution and the corresponding model solutions using NoFAS (first row) and a fixed surrogate (second row) in Experiment 1. The plots in the first column show the posterior parameter realizations (blue points), the true parameter values $\boldsymbol{z}^*$ (green star), and the contour for the true posterior distribution of $\boldsymbol{z}$. The plots in the second column display the observations $\widetilde{\boldsymbol{x}}$ (red dots), the posterior predictive samples, and $\boldsymbol{x}^*$ (green star). The pre-grid and the adaptive realizations in Z_G are illustrated in the third column.

3.2 Experiments 2 and 3: ODE Hemodynamic Models

In this section, NoFAS is applied to estimate the parameters of a two-element and a three-element Windkessel models. These are lumped parameter (or circuit) models of the interaction between flow, pressure, resistance and compliance in a vascular system. These models are also frequently used to mimic physiological boundary conditions in numerical hemodynamics, and the solutions of inverse problems for Windkessel models is often a prerequisite to determine boundary conditions parameters so that simulation outputs can match patient-specific responses. These systems also offer a concise representation (i.e., with a small number of parameters) of circulatory sub-systems in humans, and therefore constitute an ideal test bed for the application of advanced inference algorithms to physiological models with many more parameters.

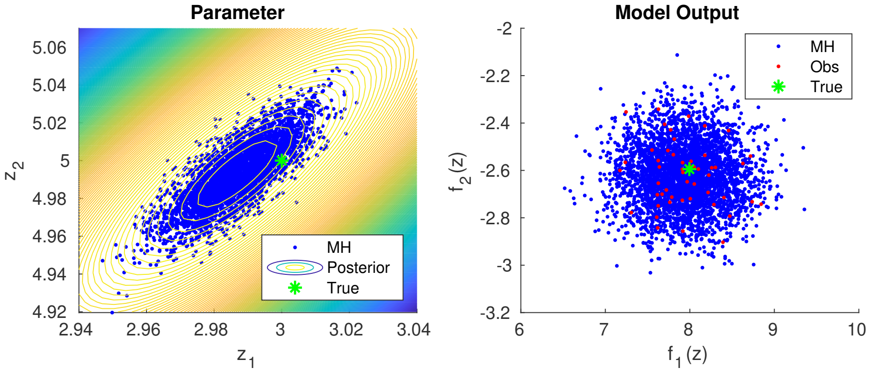


Figure 3: Parameters and model solutions from Experiment 1 using a MH sampler. The blue points represent the parameter samples of $\boldsymbol{z}$ from MH (left plot), the corresponding model solutions $\boldsymbol{x}$ (right plot). A green star represents the true parameters $\boldsymbol{z}^*$ (left) and model solutions $\boldsymbol{x}^*$ (right). Red dots indicate the observations $\widetilde{\boldsymbol{x}}$. A contour plot of the true posterior distribution is also superimposed to the posterior samples on the left plot.

3.2.1 Experiments 2: Two-element Windkessel Model

The two-element Windkessel model was developed by Otto Frank [70] and, when applied to the flow-pressure relation in the aorta, is successful in explaining the exponential decay of the aortic pressure following the closure of the aortic valve. This model requires two parameters, i.e., a systemic resistance $R \in [100, 1500]$ Barye· s/ml and a systemic capacitance $C \in [1 \times 10^{-5}, 1 \times 10^{-2}]$ ml/Barye, which are responsible for its alternative name as *RC model*. We provide a periodic time history of the aortic flow in Figure 4 and use the RC model to predict the time history of the proximal pressure $P_p(t)$, specifically its maximum (max), minimum (min) and average (ave) values over a typical heart cycle, while assuming the distal resistance $P_d(t)$ as a constant in time, equal to 55 mmHg. In our experiment, we set the true resistance and capacitance as $z_1^* = R^* = 1000$ Barye· s/ml and $z_2^* = C^* = 5 \times 10^{-5}$ ml/Barye and determine $P_p(t)$ from a RK4 numerical solution of the following algebraic-differential system of two equations

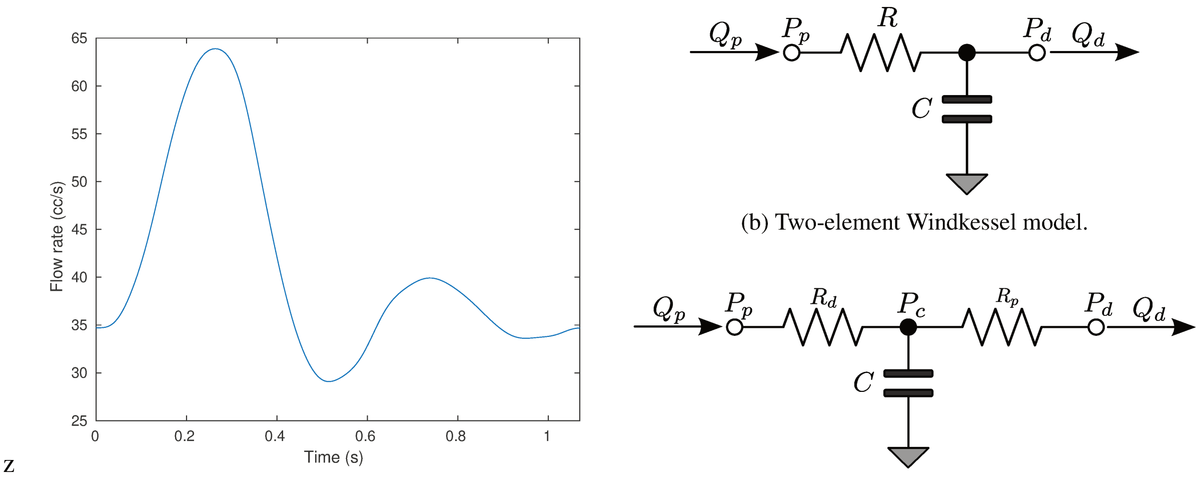
$$Q_d = \frac{P_p - P_d}{R}, \quad \frac{\mathrm{d}P_p}{\mathrm{d}t} = \frac{Q_p - Q_d}{C}, \tag{8}$$

where Q_p is the flow entering the RC system and Q_d is the distal flow (see Figure 4). Synthetic observations are generated by adding Gaussian noise to the true model solution $\boldsymbol{x}^* = (P_{p,\text{min}}, P_{p,\text{max}}, P_{p,\text{ave}}) = (78.28, 101.12, 85.75)$, i.e., $\widetilde{\boldsymbol{x}}$ follows a multivariate Gaussian distribution with mean $\boldsymbol{x}^*$ and a diagonal covariance matrix with entries $0.05\, x_i^*$, where $i = 1, 2, 3$ corresponds to the maximum, minimum, and average pressures, respectively. The aim is to quantify the uncertainty in the RC model parameters given 50 repeated pressure measurements. We imposed a non-informative prior on R and C. Also note that the two-element Windkessel model is *identifiable*. The resistance R can be estimated based on the mean flow and pressure, while the capacitance C directly affects the pulse pressure (i.e. the difference between systolic and diastolic pressure).

Similar to Experiment 1, we consider a total budget of 64 output evaluations via the true model f and compare NoFAS with the fixed surrogate approach, a MH sampler, and BBVI. For NoFAS, $M = 20$ batches of $S_G = 2$ samples per batch are collected at a calibration frequency of $c = 1000$ NF parameter updates. The NF architecture is MAF composed of five alternated batch normalization and 5 MADE layers, each consisting of a MADE autoencoder with 1 hidden layer of 100 nodes and ReLU activation.

The results from NoFAS and from NF with a fixed surrogate are presented in Figure 5. The approximate posterior distribution of the RC model parameters with a fixed surrogate is clearly biased, and so is the corresponding predictive posterior distribution. In contrast, NoFAS provides significantly more accurate results.

The results from BBVI and MH are presented in Figure 6. BBVI uses a normal variational distribution. For the RC model, where no posterior correlation between parameters R and C is expected, BBVI with the mean field assumption captures the posterior and predictive posterior distributions accurately. MH operates on a fixed surrogate trained using a 30 × 30 pre-grid and requires 2×10^6 true model evaluations. A total of 1800 effective posterior samples were generated using a burn-in and thinning rate of 10% and 1/1000, respectively. Even though this fixed surrogate is trained from a large number of examples, it still introduces bias in the estimated parameters. The posterior samples from MH deviate instead significantly from the true model parameters, and so do the posterior predictive samples, though less pronounced.



(a) Time history for proximal flow rate Q_p.

(b) Two-element Windkessel model.

(c) Three-element Windkessel model.

Figure 4: RC and RCR hemodynamic models.

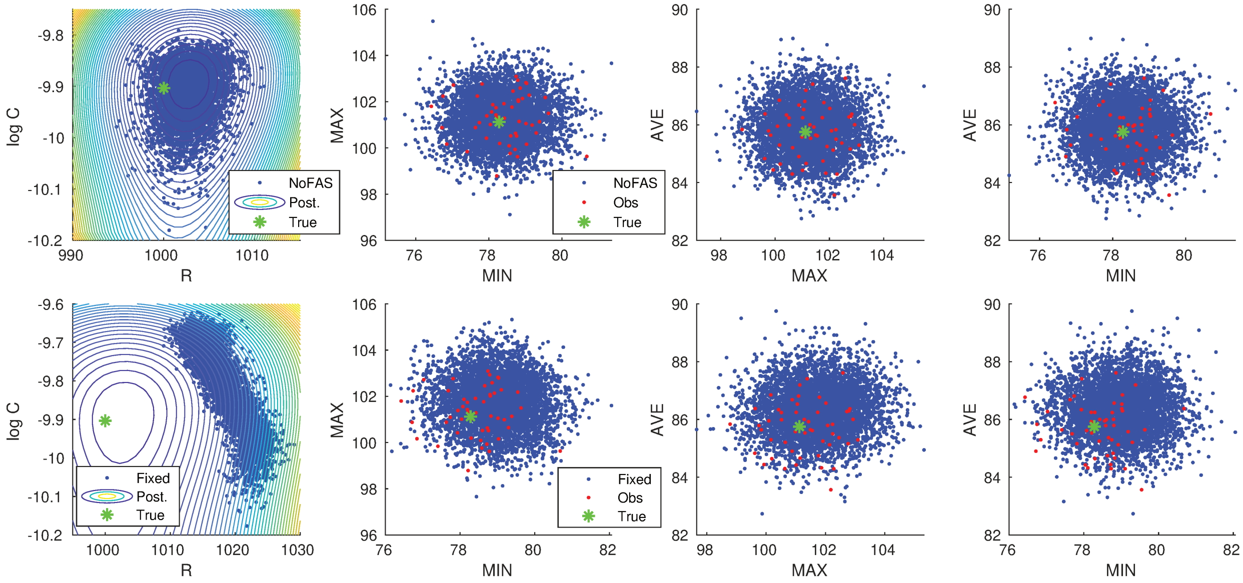


Figure 5: Parameters and model solutions from inference using NoFAS (first row) and NF with a fixed surrogate (second row) in Experiment 2. The first column shows the posterior samples $\boldsymbol{z}_K$ while the other columns illustrate the corresponding model outputs. Symbols and colors are consistent with those in Figure 2.

3.2.2 Experiments 3: Three-element Windkessel Model

The three-parameter Windkessel or *RCR* model is characterized by proximal and distal resistance parameters $R_p, R_d \in [100, 1500]$ Barye· s/ml and one capacitance parameter $C \in [1 \times 10^{-5}, 1 \times 10^{-2}]$ ml/Barye. Even if it consists of a relatively simple lumped parameter formulation, the RCR circuit model is not identifiable. The average distal pressure is only affected by the total system resistance, i.e. the sum $R_p + R_d$, leading to a negative correlation between these two parameters. Thus, an increment in the proximal resistance is compensated by a reduction in the distal resistance (so the average distal pressure remains the same) which, in turn, reduces the friction encountered by the flow exiting the capacitor. An increase in the value of C is finally needed to restore the average, minimum and maximum pressure. This leads to a positive correlation between C and R_d.

Similar to the RC model, the output consists of the proximal pressure $P_p(t)$, specifically its maximum, minimum and average values $(P_{p,\text{min}}, P_{p,\text{max}}, P_{p,\text{ave}})$ over one heart cycle. The true parameters are $z_1^* = R_p^* = 1000$ Barye·s/ml, $z_2^* = R_d^* = 1000$ Barye·s/ml and $C^* = 5 \times 10^{-5}$ ml/Barye and the proximal pressure is computed from the solution

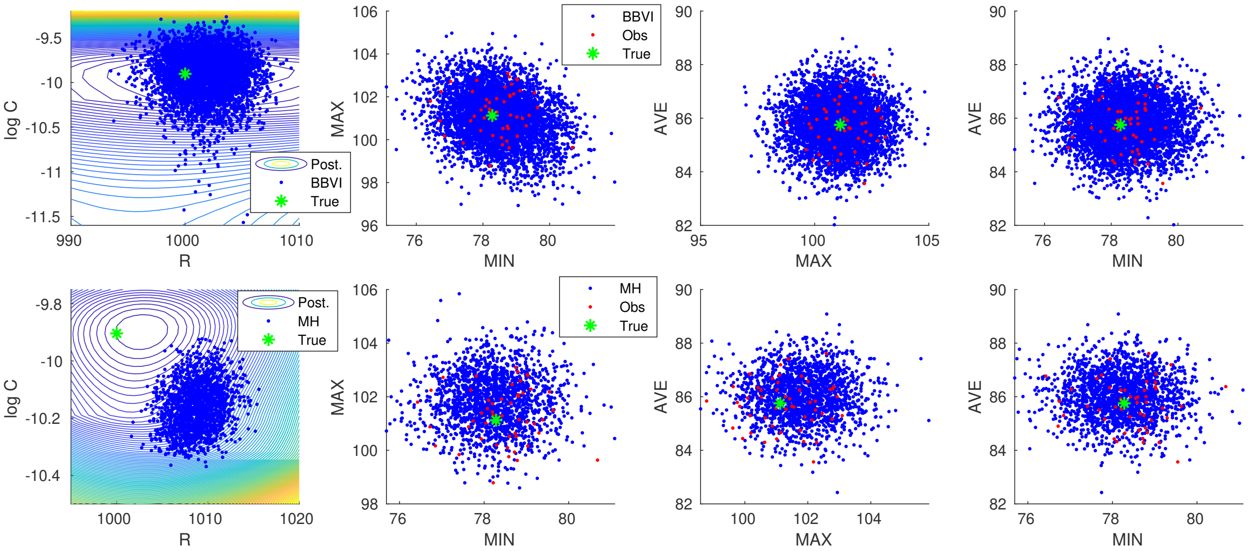


Figure 6: Parameters and model solutions from BBVI (first row) and MH (second row) for Experiment 2. Symbols and colors are consistent with those in Figure 2.

of the algebraic-differential system

$$Q_p = \frac{P_p - P_c}{R_p}, \quad Q_d = \frac{P_c - P_d}{R_d}, \quad \frac{\mathrm{d}P_c}{\mathrm{d}t} = \frac{Q_p - Q_d}{C}, \tag{9}$$

where the distal pressure is set to $P_d = 55$ mmHg. Synthetic observations are generated from $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\mu = (f_1(\boldsymbol{z}^*), f_2(\boldsymbol{z}^*), f_3(\boldsymbol{z}^*))^T = (P_{p,\text{min}}, P_{p,\text{max}}, P_{p,\text{ave}})^T = (100.96, 148.02, 116.50)^T$ and $\boldsymbol{\Sigma}$ is a diagonal matrix with entries $(5.05, 7.40, 5.83)^T$. The budgeted number of true model solutions is 216; the fixed surrogate model is evaluated on a $6 \times 6 \times 6 = 216$ pre-grid while the adaptive surrogate is evaluated with a pre-grid of size $4 \times 4 \times 4 = 64$ and the other 152 evaluations are adaptively selected. The NF architecture and hyper-parameter specifications are the same as for the RC model, except a more frequent surrogate update of $c = 300$ and a larger batch size $b = 500$.

The results are presented in Figure 7. The posterior samples obtained through NoFAS capture well the non-linear correlation among the parameters and generate a fairly accurate posterior predictive distribution that overlaps with the observations but has a slightly larger dispersion, as expected. In contrast, NF with a fixed surrogate fails to capture the parameter correlations and the complex shape of the posterior distribution; in addition, the posterior predictive samples deviate significantly from the observed data.

The results from the BBVI and MH are shown in Figure 8. BBVI used a Gaussian variational distribution; 3600 posterior samples were obtained from 4×10^6 MCMC iterations with a burn-in and thinning rate of 10% and $1/1000$ respectively. Since the RCR model parameters are highly correlated, it is not surprising that the mean field assumption for BBVI leads to biased posterior distributions. MH produces better results than BBVI, but still has some bias in the posterior predictive distribution, particularly in the tails. Similar to the RC model, a fixed surrogate model is used for MH, trained with a large pre-grid consisting of $20^3 = 8000$ samples.

This experiment showcases the inferential and computational superiority of NoFAS over BBVI with the mean field assumption and MH with a fixed surrogate trained from a large training dataset, for cases where there is a strong posterior correlation among parameters.

3.3 Experiment 4: Non Isomorphic Sobol Function

In this experiment, we consider a mapping from a five-dimensional parameter vector $\boldsymbol{z} \in \mathbb{R}^5$ onto a four-dimensional output

$$f(\boldsymbol{z}) = \boldsymbol{A}\, \boldsymbol{g}(e^{\boldsymbol{z}}). \tag{10}$$

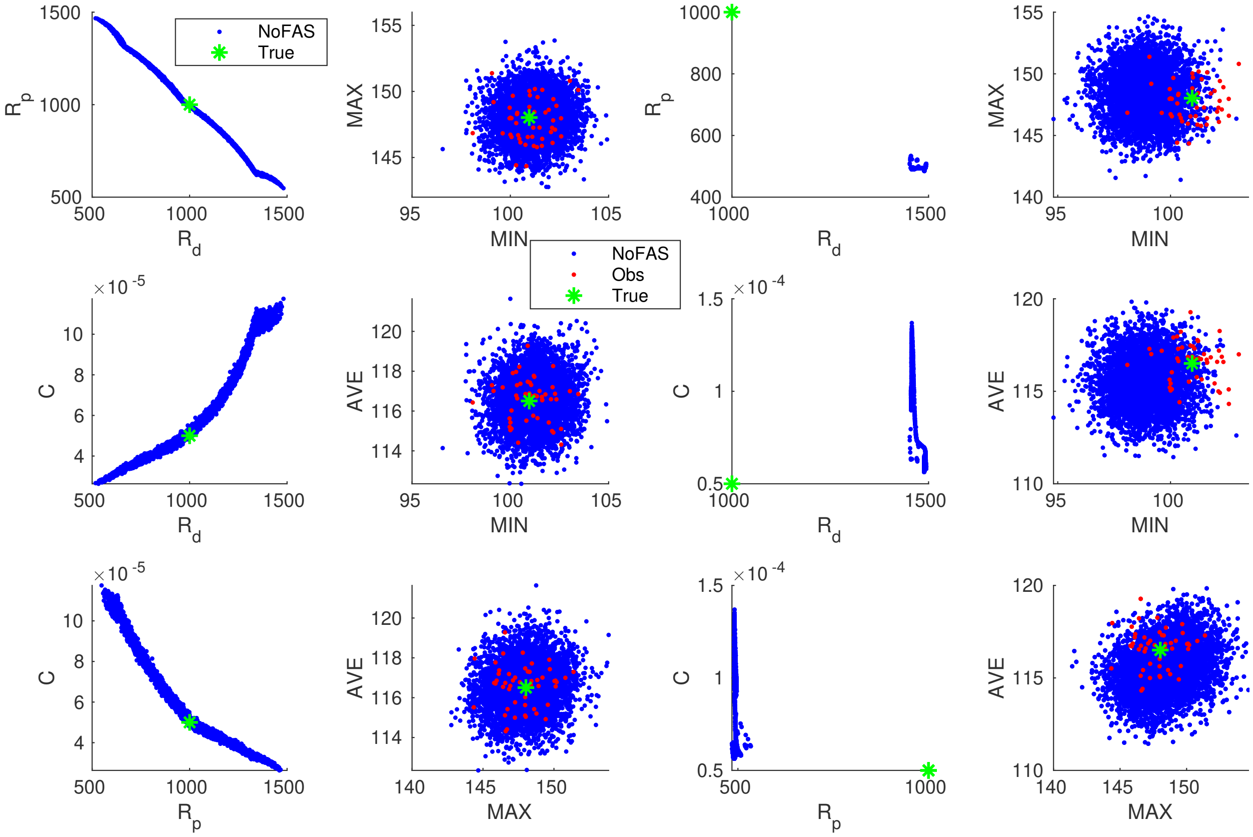


Figure 7: Parameter posterior samples and corresponding model solutions for the RCR model. The two columns on the left show the results of NoFAS, while the two columns on the right contain the results for NF with a fixed surrogate. The first and third columns show the posterior samples, and the second and fourth columns show the posterior predictive samples (blue) and observations (red). The green stars represent ether the true parameters R_p^*, R_d^*, C^* (1st and 3rd columns) or the corresponding model solutions (2nd and 4th columns).

where $g_i(\boldsymbol{r}) = (2 \cdot |2\,a_i - 1| + r_i)/(1 + r_i)$ with $r_i > 0$ for $i = 1, \dots, 5$ is the *Sobol* function [71] and $\boldsymbol{A}$ is a 4×5 matrix. We also set

$$\boldsymbol{a} = (0.084, 0.229, 0.913, 0.152, 0.826)^T \text{ and } \boldsymbol{A} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The true parameter vector is set at $\boldsymbol{z}^* = (2.75, -1.5, 0.25, -2.5, 1.75)^T$. The Sobol function is bijective and analytic but $f : \mathbb{R}^5 \to \mathbb{R}^4$ leads to an over-parameterized model and non-identifiability. This is also confirmed by the fact that the curve segment $\gamma(t) = g^{-1}(g(\boldsymbol{z}^*) + \boldsymbol{v}\,t) \in Z$ gives the same model solution as $\boldsymbol{x}^* = f(\boldsymbol{z}^*) = f(\gamma(t)) \approx (1.4910, 1.6650, 1.8715, 1.7011)^T$ for $t \in (-0.0153, 0.0686]$, where $\boldsymbol{v} = (1, -1, 1, -1, 1)^T$. This is consistent with the one-dimensional null-space of the matrix $\boldsymbol{A}$. Since the output $g_i(\boldsymbol{r})$ of the Sobol function is $(1, 2|2a_i - 1|]$, hence $t \in \bigcap_{i=1}^5 ((1 - g_i(\boldsymbol{z}^*))/v_i, (2|2a_i - 1| - g_i(\boldsymbol{z}^*))/v_i] = (-0.0153, 0.0686]$.

Similar to the other 3 experiments, we generated the model output observations from a Gaussian distribution as

$$\boldsymbol{x} = \boldsymbol{x}^* + 0.01 \cdot |\boldsymbol{x}^*| \odot \boldsymbol{x}_0, \text{ where } \boldsymbol{x}_0 \sim \mathcal{N}(0, \boldsymbol{I}_5). \tag{11}$$

The aim is to obtain posterior samples on the latent variables $\boldsymbol{z}$ and quantify the uncertainty given the observed data. The likelihood function is Gaussian and we impose a uniform prior on $\boldsymbol{z}$. The fixed surrogate model was estimated on a grid of $4^5 = 1024$ points; for NoFAS, the adaptive surrogate model was initially trained on a $3^5 = 243$ pre-grid, and then successively updated using $S_G = 12$ samples from the $b = 250$ batches computed every $c = 200$ NF parameter updates. We used a ReLU-activated RealNVP NF with 15 layers, where each linear masked coupling layer contains one hidden layer of 100 nodes, and 15 batch normalization layers are added before each RealNVP layer. A RMSProp optimizer was used, with the learning rate and its exponential decay factor at 0.0005 and 0.9999, respectively.

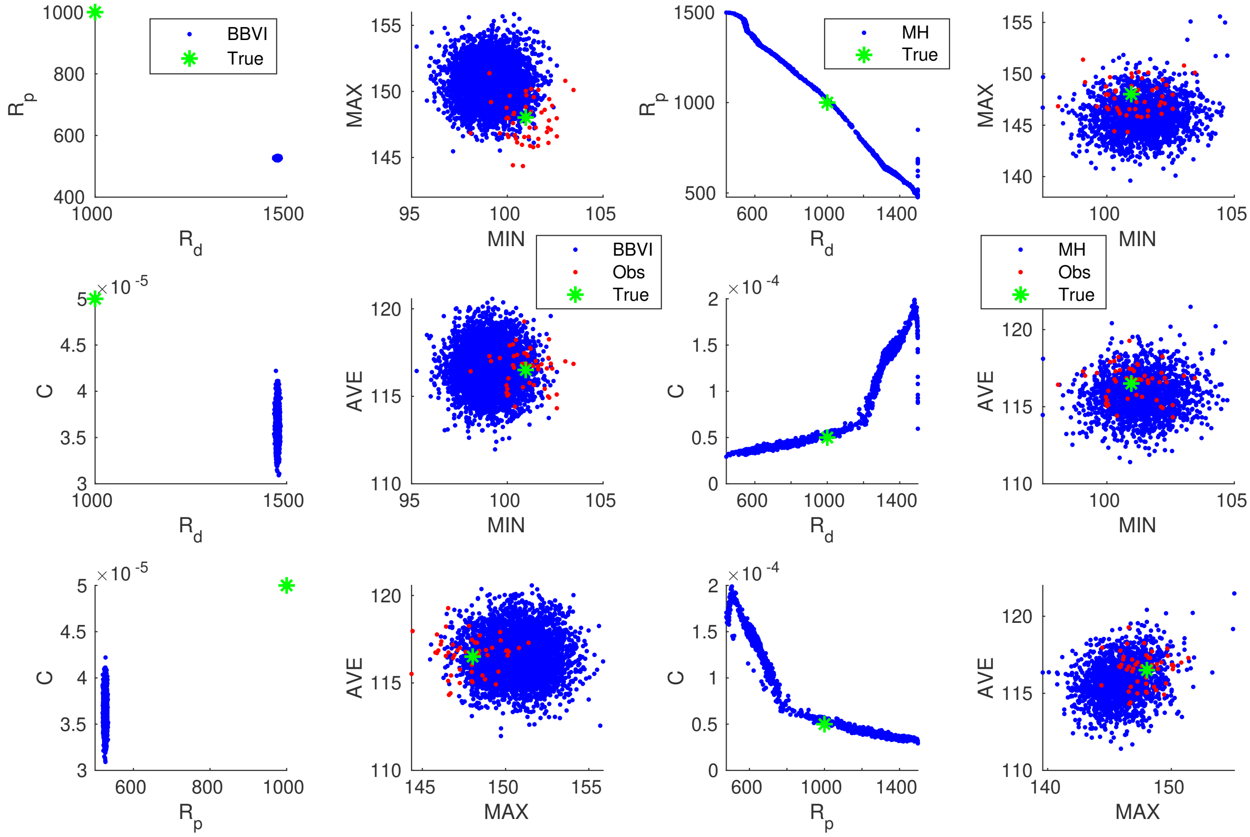


Figure 8: RCR model parameters and corresponding outputs from BBVI (left two columns) and MH (right two columns) in Experiment 3. Symbols and colors are consistent with those in Figure 7.

The marginal posterior histograms for the 5 model parameters and their pairwise 2-dimensional scatter plots are presented in Figure 9. Additionally, since the model is non-identifiable, the 5-dimensional joint posterior distribution has a *ridge* $\gamma(t)$ for $t \in (-0.0153, 0.0686]$, highlighted in red in the pairwise plots. In the plots of z_2 vs. the other latent variables, the ridge is also characterized by a vertical or horizontal red segment, suggesting z_2 becomes unimportant sufficiently away from its true value $z_2 \approx 0.223$ and making the inference of this parameter particularly challenging. The samples from the posterior predictive distributions are presented in Figure 10. The model outputs generated from the posterior predictive distributions overlap well with the actual observations.

The results from the NF with a fixed surrogate are presented in Figure 11. Consistent with the findings in the other experiments, the posterior samples are biased and deviate from the true parameter values. The samples from the posterior predictive distributions are presented in Figure 12, which also shows some deviation of the model outputs generated from the posterior predictive distributions from the actual observations. We also generated posterior samples using a MH sampler. Due to the non-identifiability of the model, MH had trouble converging on parameter z_2 and the Markov chains for z_2 moved freely in the region where this parameter is unimportant, despite a satisfactory convergence for the other 4 parameters. To mitigate this problem, we constrained the prior by forcing $\boldsymbol{z} \in [-4, 4]^5$, which leads to convergence on all parameters, as measured by the Gelman-Rubin metric [72]. However, the Markov chains still suffered from poor mixing and we used a burn-in of 10% and a thinning interval of 1×10^6 to reduce the sample auto-correlation, generating 5400 posterior samples. The results are presented in Figure 13 and Figure 14. The results suggest that MH can provide parameter estimates compatible with those produced by NoFAS but might have to leverage stronger prior knowledge and requires substantially more samples for models with non-identifiable parameters.

4 Discussion

We propose NoFAS, an approach to efficiently solve inverse problems by combining optimization-based inference with the adaptive construction of surrogate models using samples from NF. The number of forward model evaluations

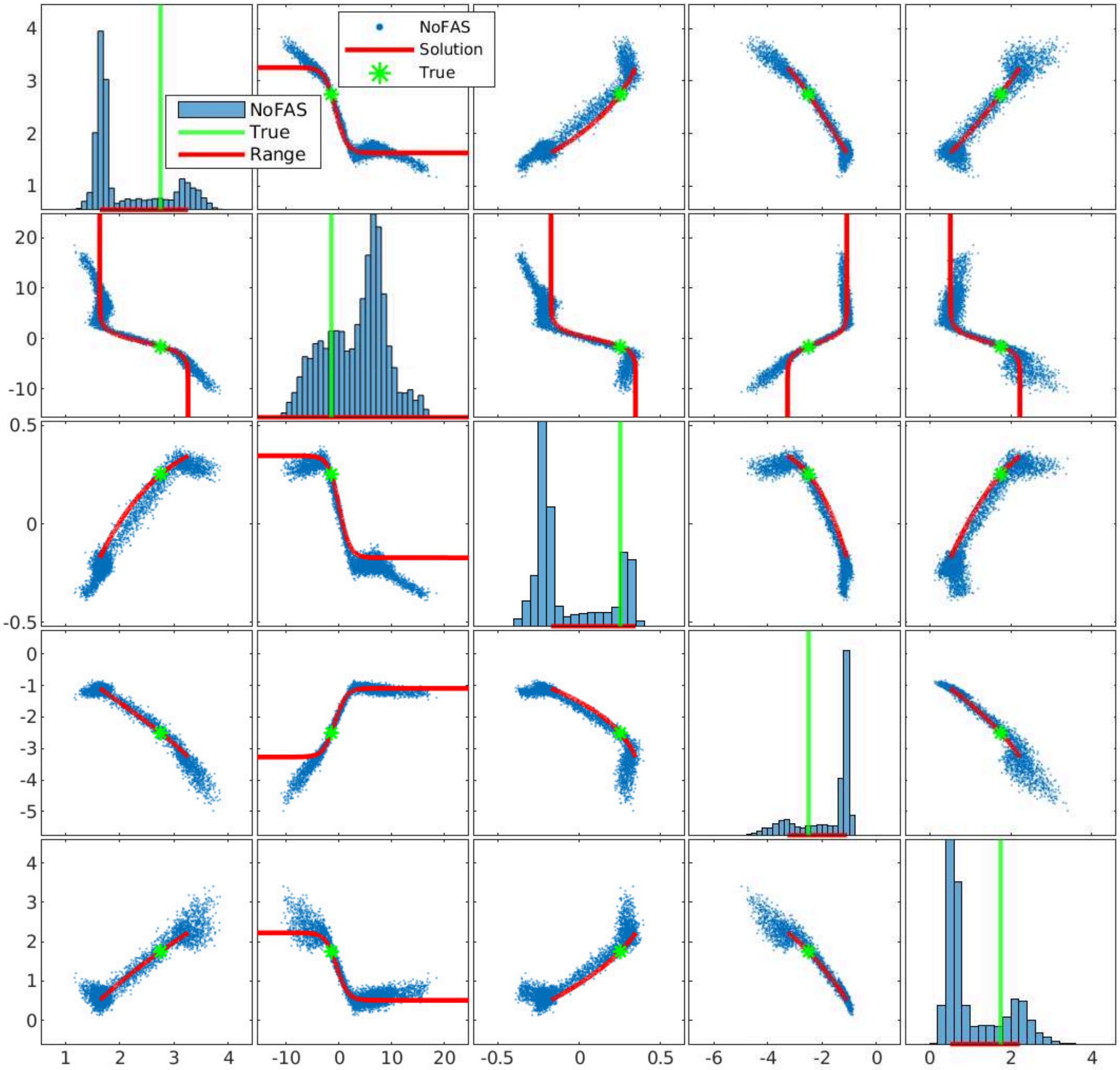


Figure 9: Marginal histograms and pairwise scatter plots of the posterior samples from NoFAS in Experiment 4. The vertical green lines and the green stars represent the true parameters $\boldsymbol{z}^*$; the horizontal red lines indicate admissible parameter ranges compatible with the output $f(\boldsymbol{z}^*)$, and the blue points represent the posterior samples.

can be greatly reduced by adaptively training and improving the surrogate with samples from high-density posterior regions that are progressively discovered by NF. We also propose a flexible sampling weighting mechanism, where a number of parameter realizations from a pre-selected grid and the most recent training samples are assigned larger weights in the loss function, which significantly improves the inference results compared to NF with uniform weights. The trade-off between global and local surrogate accuracy can be tuned by deciding the relative budget to assign either to the pre-grid or to sample locations adaptively selected through the NF iterations. Based on our empirical studies, the assignment of the 20% to 30% of the solution budget to the pre-grid produces generally accurate posterior and posterior predictive distributions, even if characterized by complex features, such as the presence of multiple modes or ridges. Assigning an excessive budget to the pre-grid can lead to poor approximation for the posterior distribution, whereas an insufficient number of pre-grid samples can cause slow convergence or even divergence.

NoFAS requires several hyperparameters to be specified or tuned, including the batch size b, the calibration interval c, and the calibration size S_G. Specifically, b needs to exceed a certain threshold for NoFAS to converge. We tested different batch sizes on the RCR model and observed that the minimal loss stops decreasing for $b > 50$, but b needs to be at least 400 to achieve accurate uncertainty quantification. We also tested various c and found that small c tends

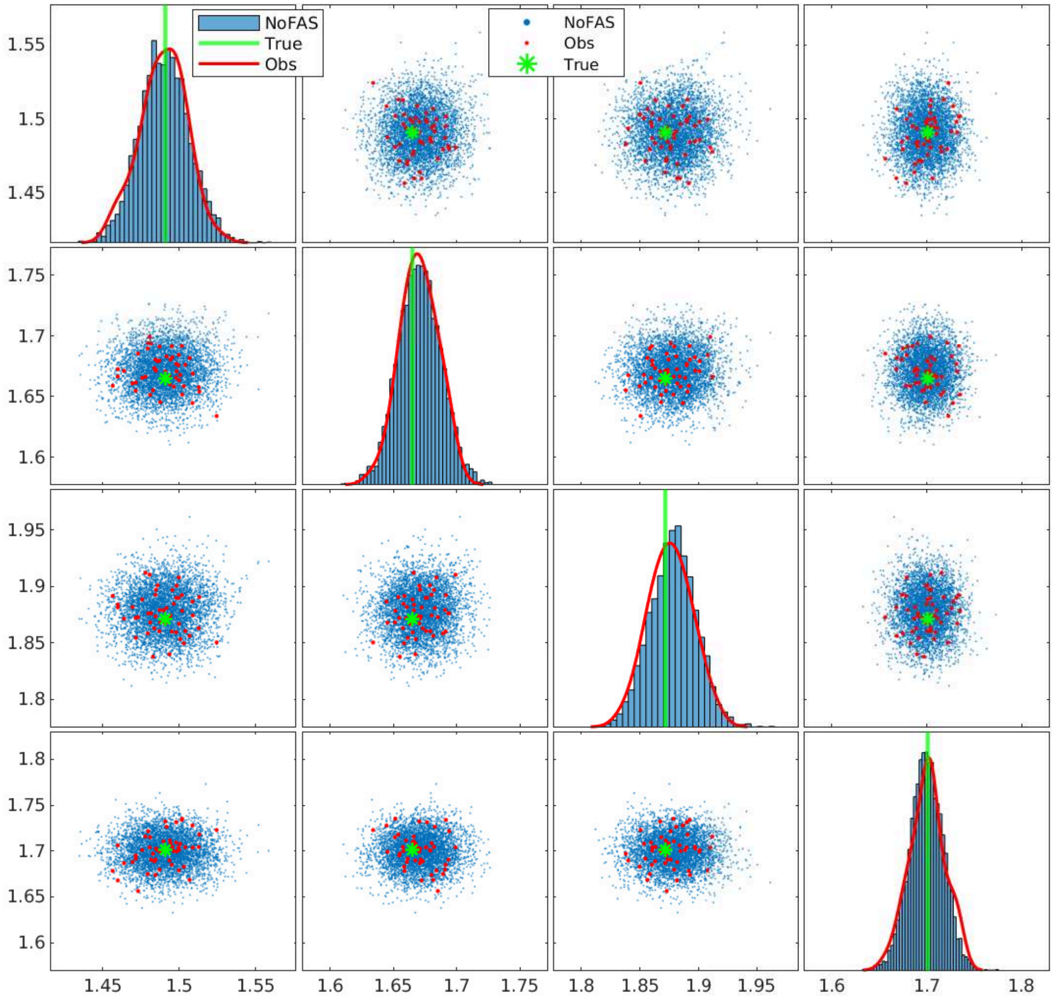


Figure 10: Marginal histograms and pairwise scatter plots for the predictive posterior distribution from NoFAS in Experiment 4. The histograms on the diagonal show the marginal distributions of the posterior predictive samples for the model solutions, the green vertical lines indicate the true model solution, and the red curves are the kernel density estimation. For the non-diagonal scatter plots, the blue, red, and green dots represent the posterior predictive samples, the observations and the true model solutions, respectively.

to produce too much surrogate adaptation in the early stage of NoFAS, consuming the budget too quickly or even ran out before convergence. Conversely, large c could lead to inaccurate gradients, delaying the exploration of high posterior density regions. Note that approximation of multi-modal posterior distributions typically requires a higher S_G than for uni-modal distributions. In our preliminary experiments on bimodal posterior distributions (not shown), using $S_G = 2$ was sufficient to locate both modes. A large S_G leads to a fast consumption of the budget in the early stage of NoFAS, while a small S_G could lead to an inaccurate surrogate. We recommend c and S_G be jointly selected by taking the parameter space dimensionality d into account. Generally speaking, problems characterized by a higher dimensionality d would require a larger b/S_G ratio. As for the pre-grid weight factor β_0 and memory decay factor β_1, β_0 represents the proportion of loss computed from the pre-grid in Eqn. (5), while β_1 controls the decay rate for the proportion of the loss function computed from adaptively collected samples from most to least recent. The larger β_0, the more impact the pre-grid samples will have on the results from NoFAS; the larger β_1, the shorter the memory on adaptively collected samples. We recommend values of $\beta_0 \in [0.5, 0.7]$ and $\beta_1 \in [0.01, 1.0]$ based on our experiment results.

NoFAS is designed to be agnostic with respect to the NF formulation, provided the selected flow is sufficiently expressive. In our experiments, we used RealNVP and MAF. The latter requires fewer parameters than the former to

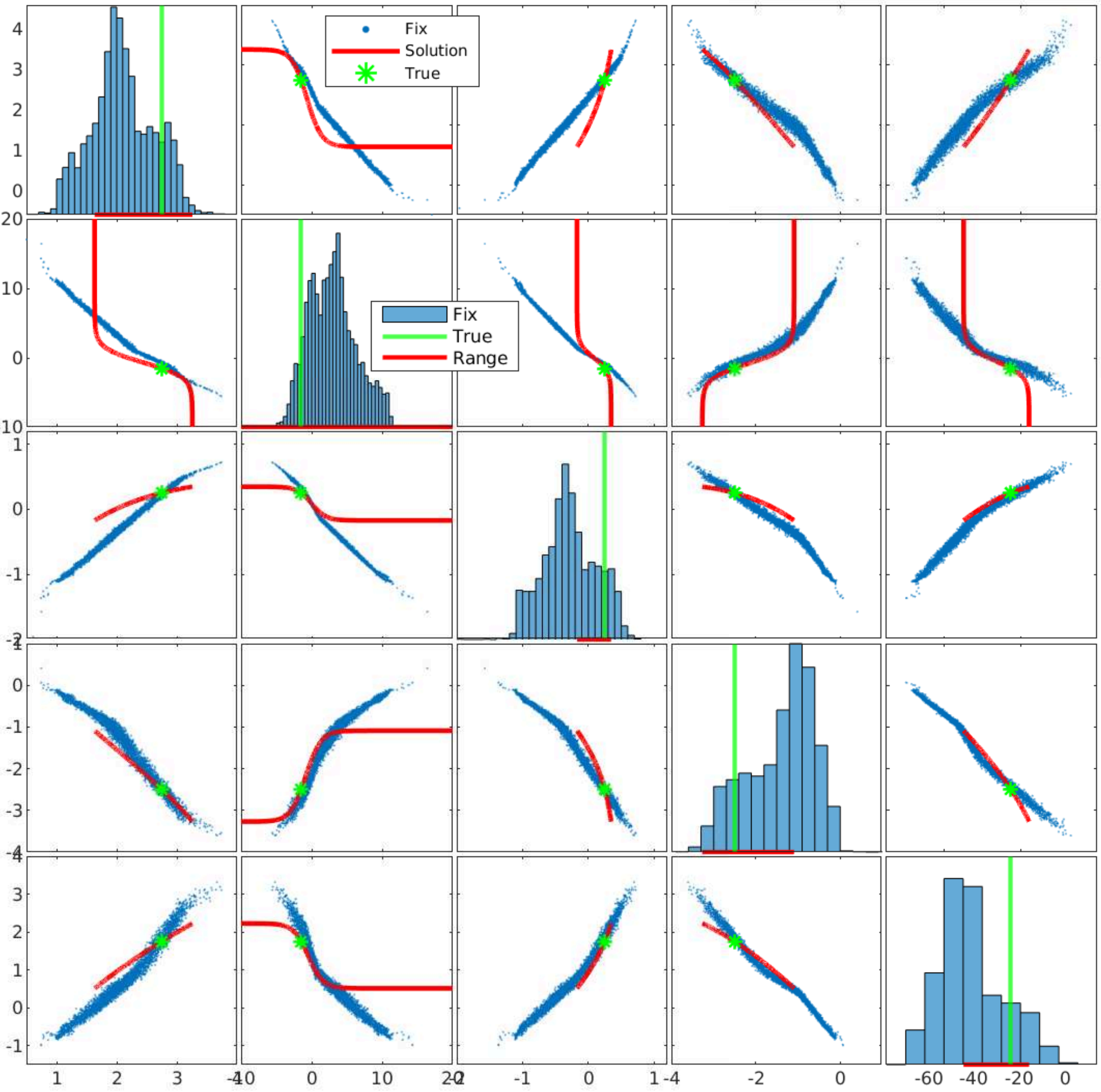


Figure 11: Marginal histograms and pairwise scatter plots of the posterior samples for Experiment 4 obtained via NF with a fixed surrogate. Symbols and colors are consistent with those in Figure 9.

converge and has a smaller computational cost per iteration. In addition, the three stages observed for the posterior samples discussed in Section 2.3 tend to be more evident when using MAF rather than RealNVP.

Acknowledgments

This work was supported by a NSF Big Data Science & Engineering grant #1918692 (PI DES), a NSF CAREER grant #1942662 (PI DES) and used computational resources provided through the Center for Research Computing at the University of Notre Dame.

References

[1] M.G. Kapteyn, J.V.R. Pretorius, and K.E. Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature Computational Science*, 1(5):337–347, 2021.

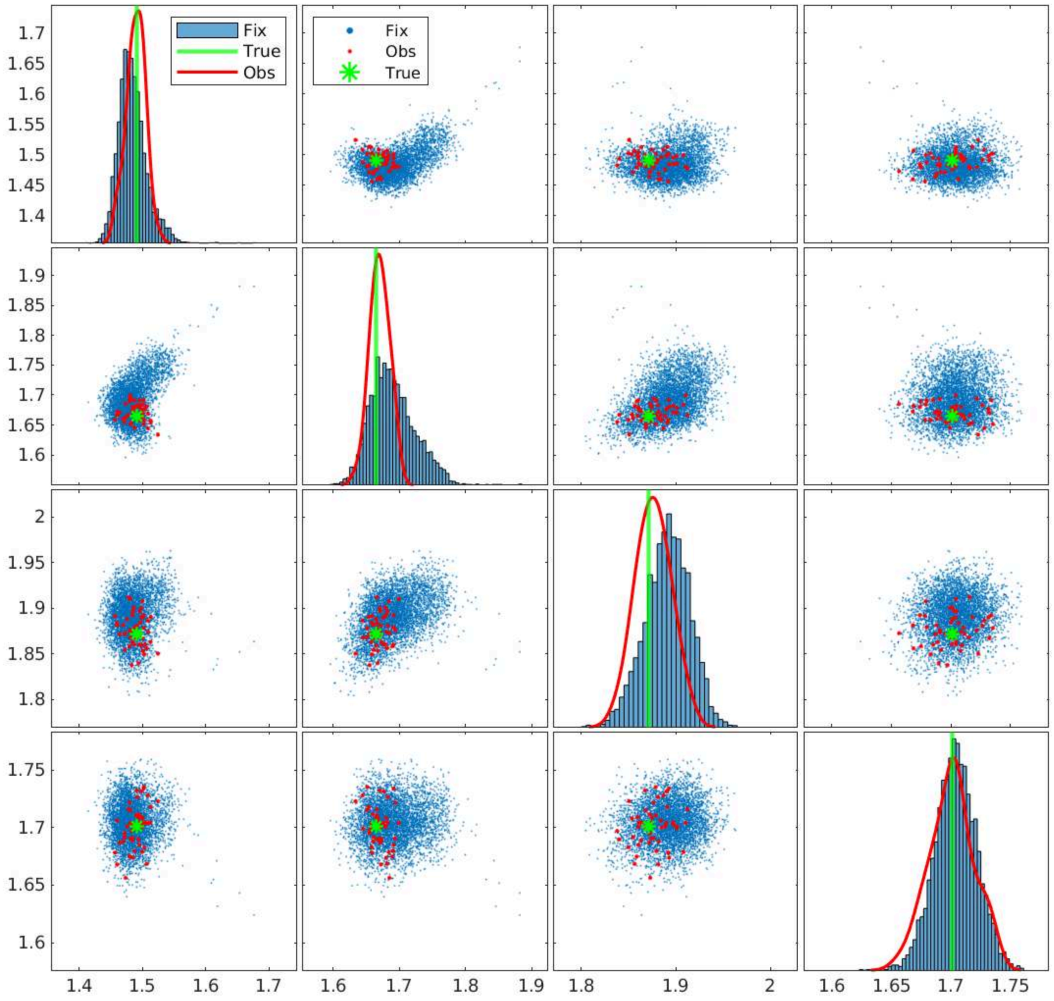


Figure 12: Marginal histograms and pairwise scatter plots for the posterior predictive samples of model outputs in Experiment 4, obtained from NF with a fixed surrogate model. Symbols and colors are consistent with those in Figure 10.

[2] D.E. Schiavazzi and T.J. Juliano. Bayesian network inference of thermal protection system failure in hypersonic vehicles. In *AIAA Scitech 2020 Forum*, page 1652, 2020.

[3] Karlyn K Harrod, Jeffrey L Rogers, Jeffrey A Feinstein, Alison L Marsden, and Daniele E Schiavazzi. Predictive modeling of secondary pulmonary hypertension in left ventricular diastolic dysfunction. *medRxiv*, pages 2020–04, 2021.

[4] M.A. Beaumont. Approximate Bayesian computation. *Annual review of statistics and its application*, 6:379–403, 2019.

[5] W.R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[6] H. Haario, E. Saksman, and J. Tamminen. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001.

[7] H. Haario, M. Laine, A. Mira, and E. Saksman. DRAM: efficient adaptive MCMC. *Statistics and computing*, 16 (4):339–354, 2006.

[8] R.M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

[9] M.D. Hoffman and A. Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

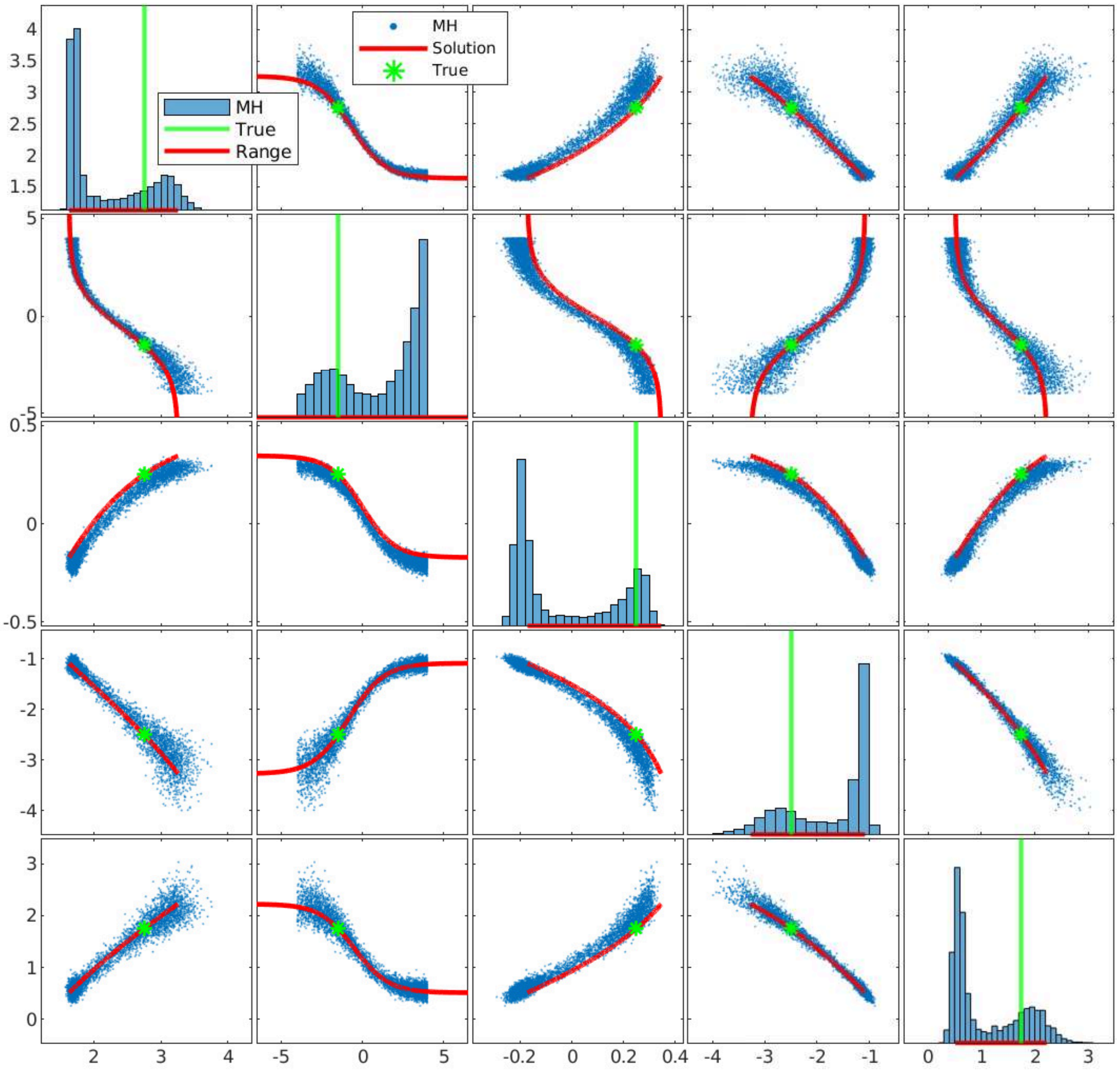


Figure 13: Marginal histograms and pairwise scatter plots of the posterior samples obtained through MH in Experiment 4. Symbols and colors are consistent with those in Figure 9.

[10] J.A. Vrugt, C.J.F. Ter Braak, C.G.H. Diks, B.A. Robinson, J.M. Hyman, and D. Higdon. Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):273–290, 2009.

[11] Jasper A. Vrugt. Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Environmental Modelling & Software*, 75:273–316, 2016.

[12] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[13] M.J. Wainwright and M.I. Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[14] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[15] M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

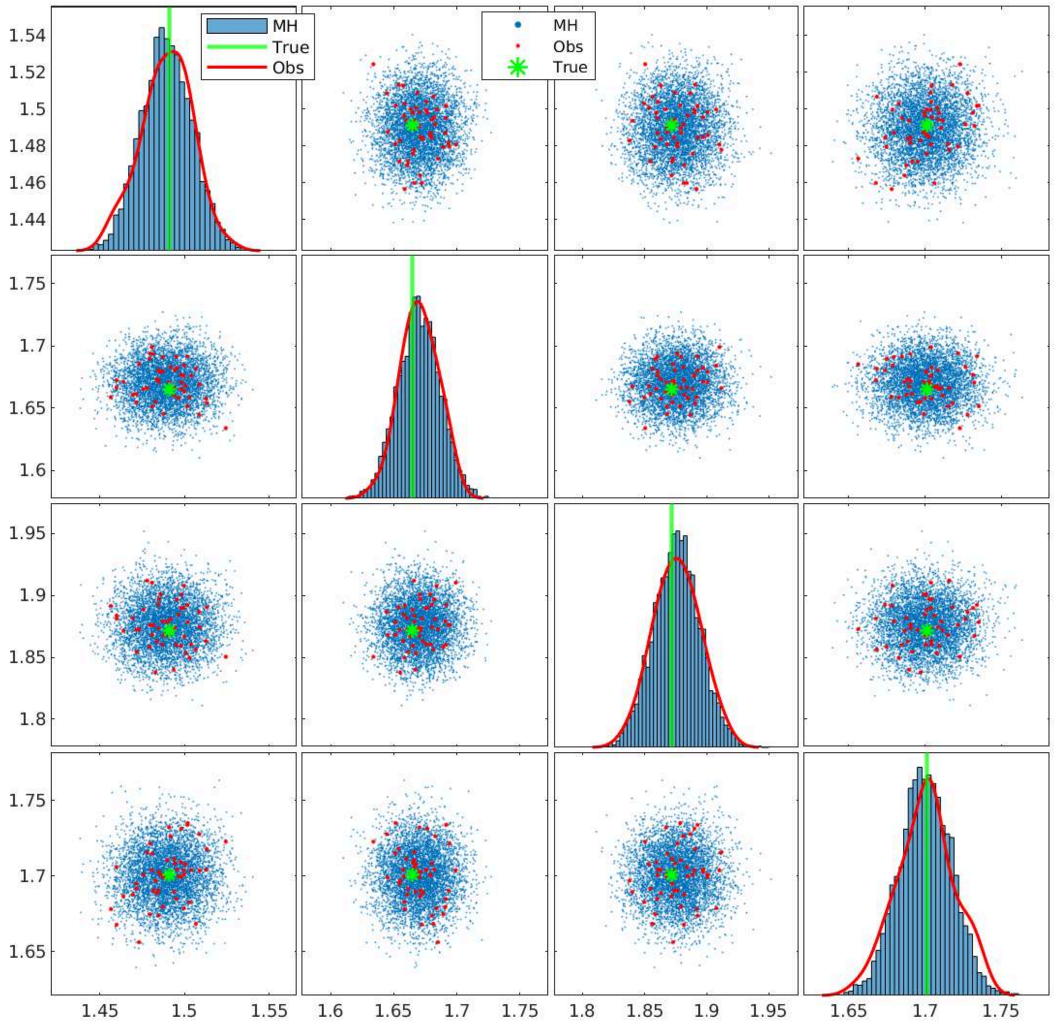


Figure 14: Marginal histograms and pairwise scatter plots for the posterior predictive samples of the model solutions in Experiment 4 computed by MH. Symbols and colors are consistent with those in Figure 10.

[16] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[17] T. Salimans and D.A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

[18] D.P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] D.J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[20] F.J.R. Ruiz, M.K. Titsias, and D. Blei. The generalized reparameterization gradient. *Advances in neural information processing systems*, 29:460–468, 2016.

[21] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333. PMLR, 2016.

[22] Ghazal Fazelnia and John Paisley. Crvi: Convex relaxation for variational inference. In *International Conference on Machine Learning*, pages 1477–1485. PMLR, 2018.

[23] Dustin Tran, David Blei, and Edo M Airoldi. Copula variational inference. In *Advances in Neural Information Processing Systems*, pages 3564–3572, 2015.

[24] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

[25] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

[26] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29:4743–4751, 2016.

[27] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[28] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017.

[29] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

[30] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[31] Reza Alizadeh, Janet K Allen, and Farrokh Mistree. Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design*, 31(3):275–298, 2020.

[32] Huanhuan Gao, Piotr Breitkopf, Rajan Filomeno Coelho, and Manyu Xiao. Categorical structural optimization using discrete manifold learning approach and custom-built evolutionary operators. *Structural and Multidisciplinary Optimization*, 58(1):215–228, 2018.

[33] Hyunkyoo Cho, Sangjune Bae, KK Choi, David Lamb, and Ren-Jye Yang. An efficient variable screening method for effective surrogate models for reliability-based design optimization. *Structural and multidisciplinary optimization*, 50(5):717–738, 2014.

[34] Patrick N Koch, Timothy W Simpson, Janet K Allen, and Farrokh Mistree. Statistical approximations for multidisciplinary design optimization: the problem of size. *Journal of Aircraft*, 36(1):275–286, 1999.

[35] Bert Bettonvil and Jack PC Kleijnen. Searching for important factors in simulation models with many factors: Sequential bifurcation. *European Journal of Operational Research*, 96(1):180–194, 1997.

[36] JC Helton. Uncertainty and sensitivity analysis in performance assessment for the waste isolation pilot plant. *Computer Physics Communications*, 117(1-2):156–180, 1999.

[37] Ilya M Sobol. Sensitivity analysis for non-linear mathematical models. *Mathematical modelling and computational experiment*, 1:407–414, 1993.

[38] Herschel Rabitz. Systems analysis at the molecular scale. *Science*, 246(4927):221–226, 1989.

[39] James L Beck and Siu-Kui Au. Bayesian updating of structural models and reliability using markov chain monte carlo simulation. *Journal of engineering mechanics*, 128(4):380–391, 2002.

[40] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[41] Richard F Gunst and Robert L Mason. Fractional factorial design. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):234–244, 2009.

[42] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[43] A Samad Hedayat, Neil James Alexander Sloane, and John Stufken. *Orthogonal arrays: theory and applications*. Springer Science & Business Media, 2012.

[44] André I Khuri and Siuli Mukhopadhyay. Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149, 2010.

[45] Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

[46] Songqing Shan and G Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and multidisciplinary optimization*, 41(2):219–241, 2010.

[47] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[48] Jaime R Carbonell. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE transactions on man-machine systems*, 11(4):190–202, 1970.

[49] Dirk Gorissen, Ivo Couckuyt, Piet Demeester, Tom Dhaene, and Karel Crombecq. A surrogate modeling and adaptive sampling toolbox for computer based design. *The Journal of Machine Learning Research*, 11:2051–2055, 2010.

[50] Dongbin Xiu and George Em Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

[51] O.G. Ernst, A. Mugler, H-J. Starkloff, and E. Ullmann. On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(2):317–339, 2012.

[52] Ivo Babuška, Fabio Nobile, and Raúl Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[53] Fabio Nobile, Raul Tempone, and Clayton G Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2411–2442, 2008.

[54] Xiaoliang Wan and George Em Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, 2005.

[55] Xiang Ma and Nicholas Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics*, 228(8):3084–3113, 2009.

[56] J.A.S. Witteveen and G. Iaccarino. Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces. *SIAM Journal on Scientific Computing*, 34(2):A814–A838, 2012.

[57] Alireza Doostan and Houman Owhadi. A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.

[58] Daniele Schiavazzi, Alireza Doostan, and Gianluca Iaccarino. Sparse multiresolution regression for uncertainty propagation. *International Journal for Uncertainty Quantification*, 4(4), 2014.

[59] DE Schiavazzi, A Doostan, G Iaccarino, and AL Marsden. A generalized multi-resolution expansion for uncertainty propagation with application to cardiovascular modeling. *Computer methods in applied mechanics and engineering*, 314:196–221, 2017.

[60] Rohit K Tripathy and Ilias Bilionis. Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of computational physics*, 375:565–588, 2018.

[61] P.R. Conrad, Y.M. Marzouk, N.S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516): 1591–1607, 2016.

[62] P.R. Conrad, A.D. Davis, Y.M. Marzouk, N.S. Pillai, and A. Smith. Parallel local approximation MCMC for expensive models. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1):339–373, 2018.

[63] A. Davis, Y. Marzouk, A. Smith, and N. Pillai. Rate-optimal refinement strategies for local approximation MCMC. *arXiv preprint arXiv:2006.00032*, 2020.

[64] L. Mihaela Paun and Dirk Husmeier. Markov chain monte carlo with gaussian processes for fast parameter estimation and uncertainty quantification in a 1d fluid-dynamics model of the pulmonary circulation. *International Journal for Numerical Methods in Biomedical Engineering*, 37(2):e3421, 2021.

[65] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889. PMLR, 2015.

[66] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[67] Josef Dick and Friedrich Pillichshammer. *Digital nets and sequences: discrepancy theory and quasi–Monte Carlo integration*. Cambridge University Press, 2010.

[68] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[70] O. Frank. The basic shape of the arterial pulse. first treatise: mathematical analysis. *Journal of molecular and cellular cardiology*, 22(3):255–277, 1990.

[71] Ilya M Sobol'. Theorems and examples on high dimensional model representation. *Reliability Engineering and System Safety*, 79(2):187–193, 2003.

[72] Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

Appendix

A. Experiments with Black-Box Variational Inference

Black-Box Variational Inference (BBVI) [16] approximates a target density $p(\boldsymbol{x}, \boldsymbol{z})$ with a distribution $q(\boldsymbol{z}|\lambda)$ from a parametric family, by minimizing the Evidence Lower BOund (ELBO) defined as $\mathcal{L} = \mathbb{E}_{q(z|\lambda)}[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\lambda)]$ (or the KL-Divergence $D(q(\boldsymbol{z}|\lambda)\|p(\boldsymbol{x}, \boldsymbol{z}))$) and by reducing the variance of the stochastic gradient of the ELBO by Rao-Blackwellization or control variate estimators [16].

We applied BBVI on two ODE hemodynamic models (experiments 2 and 3) assuming a Gaussian variational distribution and minimizing the ELBO with RMSprop. Results for RC model are shown in Figure 6 and RCR model in Figure 8. The BBVI estimates for the RC model appear accurate. However, this is not the case for the RCR model. Not only the correlations between R_d and R_p could not be captured due to the mean-field assumption in BBVI, but the posterior parameter estimates were found to be significantly biased.

B. Experiments with Metropolis Hastings

Metropolis Hastings (MH) is a Markov Chain Monte Carlo (MCMC) method [5], where a candidate sample $\boldsymbol{z'}$ is first generated given $\boldsymbol{z}_t$ from a pre-specified and fixed proposal distribution $q(\boldsymbol{z'}|\boldsymbol{z}_t)$, and then accepted as $\boldsymbol{z}_{t+1} = \boldsymbol{z'}$ with probability $A(\boldsymbol{z'}, \boldsymbol{z}_t) = \min(1, p(\boldsymbol{z'})\ q(\boldsymbol{z}_t|\boldsymbol{z'})/\ (p(\boldsymbol{z}_t)q(\boldsymbol{z'}|\boldsymbol{z}_t)))$ or rejected ($\boldsymbol{z}_{t+1} = \boldsymbol{z}_t$) with probability $1 - A(\boldsymbol{z'}, \boldsymbol{z}_t)$.

We applied MH in all four experiments using a multivariate Gaussian proposal distribution with a diagonal precision matrix. Additional details on the hyper-parameters and convergence metric are presented in Table 2. The results provided by MH are shown in Figures 3, 6, 8 and 13 for Experiment 1, 2, 3 and 4, respectively. In addition, fixed surrogate models are employed for the RC and RCR models (experiments 2 and 3) to speed up the inference task. Specifically, a uniform $30 \times 30 = 900$ grid is used for the RC surrogate and a uniform $20 \times 20 \times 20 = 4000$ grid for RCR. The true model is used in Experiments 1 and 4.

In Experiment 1, MH performs as well as NoFAS but requires the computation of millions of true model solutions. Bias in the estimated parameters and model solutions is observed for the RC and RCR model respectively, likely due to the use of surrogate models. For Experiment 4, the parameter z_2 becomes unimportant at a certain distance from its true value, and further changes in z_2 leave the posterior distribution unaltered, resulting in bad mixing and compromising the convergence of MH. Introduction of a more informative uniform prior on $[-4, 4]^5$ mitigates this problem, resulting in a similar performance to NoFAS.

Experiment	Var-Cov matrix	# of iterations	Burn-in	Thinning	Accept rate	Gelman-Rubin metric
1: Closed-Form	$0.01\ \boldsymbol{I}_2$	4×10^6	10%	1000	45.53%	(1.0020, 1.0014)
2: RC	$\text{diag}(0.01, 0.1)$	2×10^6	10%	1000	60.87%	(0.9996, 1.0013)
3: RCR	$0.025\ \boldsymbol{I}_3$	4×10^6	10%	2000	31.42%	(1.0605, 1.0428, 1.0307)
4: Sobol	$0.03\ \boldsymbol{I}_5$	6×10^8	10%	100000	43.23%	(0.9984, 0.9983, 0.9982, 0.9985, 0.9983)

Table 2: Details for the Metropolis-Hastings algorithm in the four proposed numerical experiments.

C. NoFAS Hyperparameters

All experiments used the RMSprop optimizer and an exponential scheduler with decay factor 0.9999. All normalizing flows use ReLU activations and the maximum number of iterations is set to 25001. MADE autoencoders or linear masked coupling layers contain 1 hidden layer with 100 nodes. In addition, we use $\beta_0 = 0.5$ and $\beta_1 = 0.1$ in all experiments. The recommended values for the batch size are reported in Table 3. Based on our experiments with different batch sizes, larger batch sizes lead to more stable results.

Experiment	NF type	NF layers	Batch size	Budget	Updating size	Updating interval	Learning rate
Closed-form	RealNVP	5	200	64	2	1000	0.002
RC	MAF	5	250	64	2	1000	0.003
RCR	MAF	15	500	216	2	300	0.003
Sobol	RealNVP	15	250	1023	12	250	0.0005

Table 3: Hyper parameters of NoFAS used in all four experiments

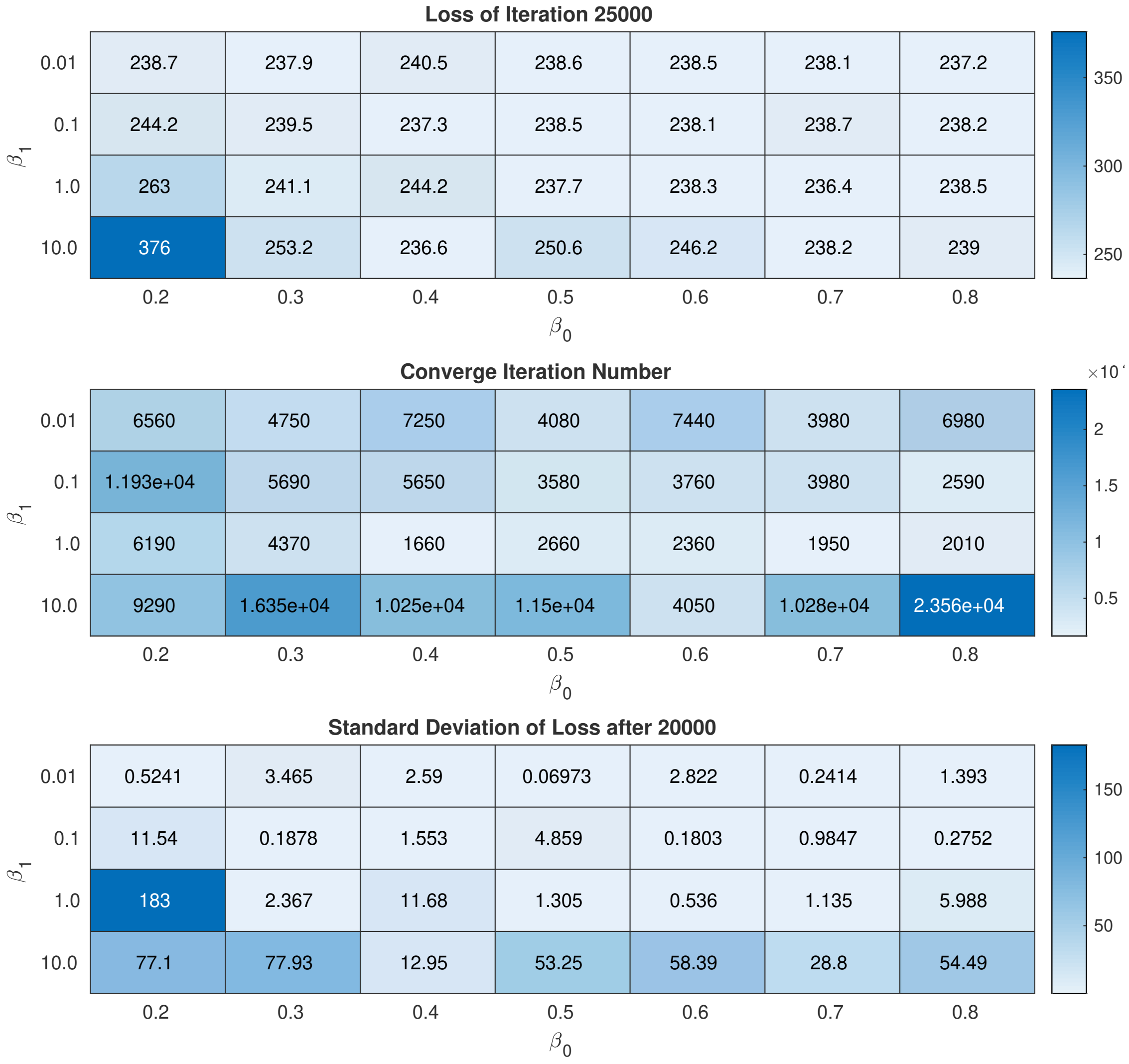
D. Sensitivity Analysis

D1. Pre-grid weight factor β_0 and memory decay factor β_1

We examined the performance of NoFAS varying β_0 from 0.2 to 0.8 and $\beta_1 \in \{0.01, 0.1, 1, 10\}$. The results are shown in Figure 15. The top heat map suggests that all the examined β_0 and β_1 values lead to similar loss function values given a sufficiently large number of iterations, but they converge with different speeds (heat map in the middle). Assigning extreme values to β_0 would slow down the convergence significantly, preventing the model to achieve sufficient local accuracy; on the other hand, too much emphasis on local regions of the parameter space would result in a surrogate model that largely ignores the global structure of f, possibly producing biased estimates. Using a large β_1 would put too much weight on the most recent training samples leading to slow convergence (see heat map for the convergence iteration number) and instability in the loss function (bottom heat map). In conclusion, we recommend setting $\beta_0 \in [0.5, 0.7]$ and $\beta_1 \in [0.01, 1.0]$.

D2. NF Paramater Initialization

We used the RCR example (Section 3.2.2) to examine how different NF parameter initializations may affect NoFAS results. We compared the Glorot, Kaiming Uniform, and Kaiming Normal initializations in 10 repeats with different random seeds with results shown in Figure 16. The Glorot initially performs slightly better than the other two initializations but overall the performance is similar for all 3 choices. Glorot and Kaiming Normal achieve slightly better quality and more stable convergence compared to Kaiming Uniform.



Loss of Iteration 25000

β_1 \ β_0	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.01	238.7	237.9	240.5	238.6	238.5	238.1	237.2
0.1	244.2	239.5	237.3	238.5	238.1	238.7	238.2
1.0	263	241.1	244.2	237.7	238.3	236.4	238.5
10.0	376	253.2	236.6	250.6	246.2	238.2	239

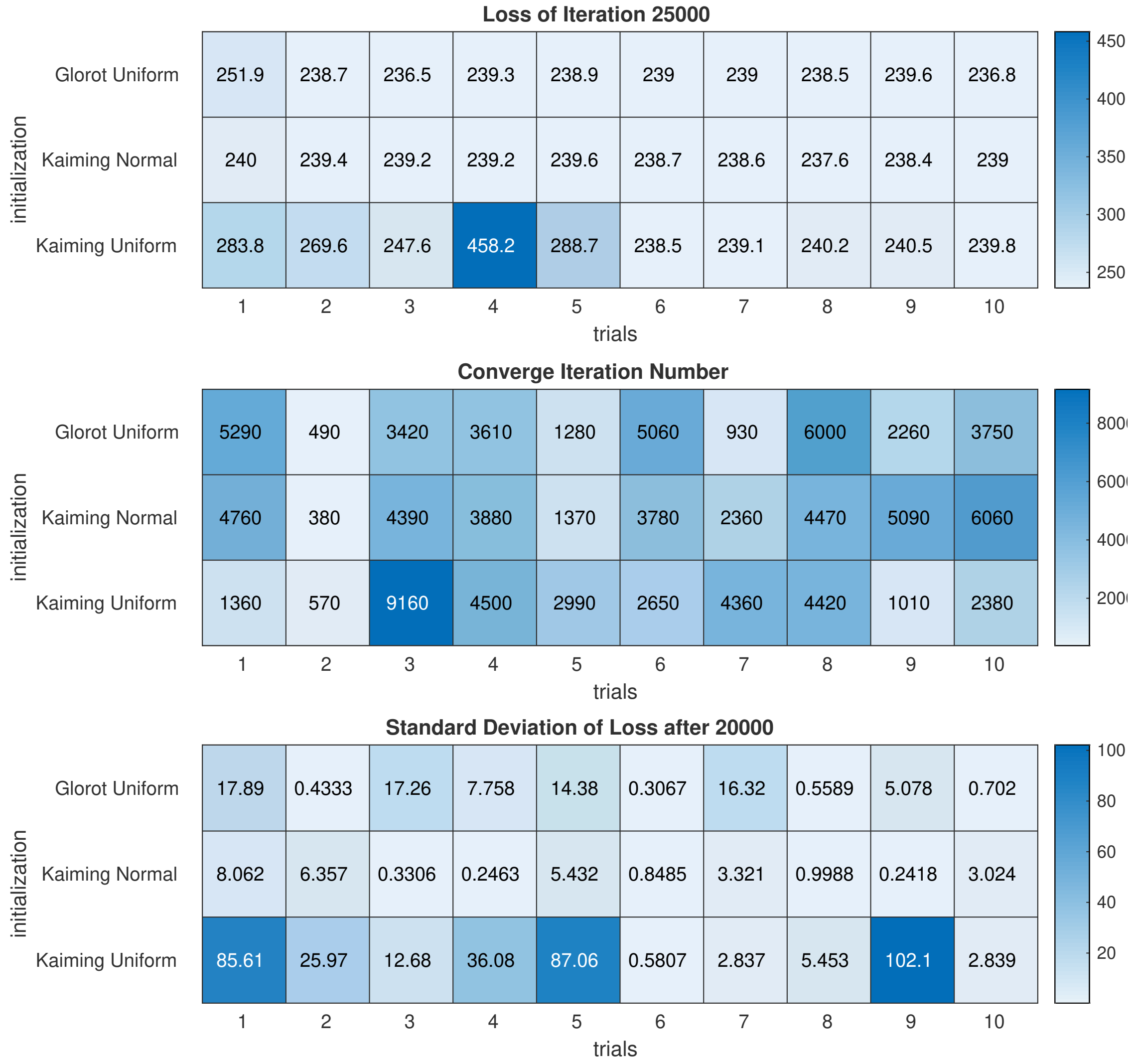
Converge Iteration Number

β_1 \ β_0	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.01	6560	4750	7250	4080	7440	3980	6980
0.1	1.193e+04	5690	5650	3580	3760	3980	2590
1.0	6190	4370	1660	2660	2360	1950	2010
10.0	9290	1.635e+04	1.025e+04	1.15e+04	4050	1.028e+04	2.356e+04

Standard Deviation of Loss after 20000

β_1 \ β_0	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.01	0.5241	3.465	2.59	0.06973	2.822	0.2414	1.393
0.1	11.54	0.1878	1.553	4.859	0.1803	0.9847	0.2752
1.0	183	2.367	11.68	1.305	0.536	1.135	5.988
10.0	77.1	77.93	12.95	53.25	58.39	28.8	54.49

Figure 15: Loss function values at iteration 25000 (top), iteration number at convergence (middle) and standard deviation of loss function values after iteration 20000 (bottom) for different combinations of β_0 and β_1 in the RCR experiment (see Section 3.2.2). Convergence is identified when the change in the moving average of the loss function within a 100 iteration window is less than 0.05% with respect to the previous window.



Loss of Iteration 25000

initialization	1	2	3	4	5	6	7	8	9	10
Glorot Uniform	251.9	238.7	236.5	239.3	238.9	239	239	238.5	239.6	236.8
Kaiming Normal	240	239.4	239.2	239.2	239.6	238.7	238.6	237.6	238.4	239
Kaiming Uniform	283.8	269.6	247.6	458.2	288.7	238.5	239.1	240.2	240.5	239.8

Converge Iteration Number

initialization	1	2	3	4	5	6	7	8	9	10
Glorot Uniform	5290	490	3420	3610	1280	5060	930	6000	2260	3750
Kaiming Normal	4760	380	4390	3880	1370	3780	2360	4470	5090	6060
Kaiming Uniform	1360	570	9160	4500	2990	2650	4360	4420	1010	2380

Standard Deviation of Loss after 20000

initialization	1	2	3	4	5	6	7	8	9	10
Glorot Uniform	17.89	0.4333	17.26	7.758	14.38	0.3067	16.32	0.5589	5.078	0.702
Kaiming Normal	8.062	6.357	0.3306	0.2463	5.432	0.8485	3.321	0.9988	0.2418	3.024
Kaiming Uniform	85.61	25.97	12.68	36.08	87.06	0.5807	2.837	5.453	102.1	2.839

Figure 16: Loss function values at iteration 25000 (top), iteration number at convergence (middle) and standard deviation of loss function values after iteration 20000 (bottom) for different weight initializations in the RCR experiment (see Section 3.2.2). Convergence is identified when the change in the moving average of the loss function within a 100 iteration window is less than 0.05% with respect to the previous window.