Computer Methods in Applied Mechanics and Engineering 456 (2026) 118916

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Multi-fidelity delayed acceptance: Hierarchical MCMC sampling for Bayesian inverse problems combining multiple solvers through deep neural networks

Filippo Zacchei [a,*], Paolo Conti [a,b], Attilio Frangi [c], Andrea Manzoni [a]

[a] MOX - Dept. of Mathematics, p.za Leonardo da Vinci, Politecnico di Milano, 32, 20133, Milano, Italy
[b] The Alan Turing Institute, London, NW1 2DB, UK
[c] Politecnico di Milano, Dept. of Civil and Environmental Engineering, p.za Leonardo da Vinci, 32, 20133, Milano, Italy

ARTICLE INFO

Keywords:
Deep learning
Neural networks
Uncertainty quantification
Bayesian inverse problems
Multi-fidelity methods

ABSTRACT

Inverse uncertainty quantification (UQ) tasks such as parameter estimation are computationally demanding whenever dealing with physics-based models, and typically require repeated evaluations of complex numerical solvers. When partial differential equations are involved, full-order models such as those based on the Finite Element Method can make traditional sampling approaches like Markov Chain Monte Carlo (MCMC) computationally infeasible. Although data-driven surrogate models may help reduce evaluation costs, their utility is often limited by the expense of generating high-fidelity data. In contrast, low-fidelity data can be produced more efficiently, although relying on them alone may degrade the accuracy of the inverse UQ solution. To address these challenges, we propose a Multi-Fidelity Delayed Acceptance scheme for Bayesian inverse problems involving large-scale physics-based models. Extending the Multi-Level Delayed Acceptance framework, the method introduces multi-fidelity neural networks that combine the predictions of solvers of varying fidelity, with high-fidelity evaluations restricted to an offline training stage. During the online phase, likelihood evaluations are obtained by evaluating the coarse solvers and passing their outputs to the trained neural networks, thereby avoiding additional high-fidelity simulations. This construction allows heterogeneous coarse solvers to be incorporated consistently within the hierarchy, providing greater flexibility than standard Multi-Level Delayed Acceptance. The proposed approach improves the approximation accuracy of the low-fidelity solvers, leading to longer sub-chain lengths, better mixing, and accelerated posterior inference. The effectiveness of the strategy is demonstrated on two benchmark inverse problems involving (i) steady isotropic groundwater flow, (ii) an unsteady reaction–diffusion system, for which substantial computational savings are obtained.

1. Introduction

Parameter estimation is a crucial aspect of many engineering applications, playing a key role in the design, optimization, and control of complex systems. In several fields ranging from, e.g., fluid dynamics [1,2] and climate systems [3] to civil structures [4] and microsystems [5,6], the accuracy and reliability of parameter estimates directly influence systems' performances or even safety. Indeed, despite advanced numerical simulations have tremendously improved our ability to predict the behavior of these systems,

* Corresponding author.
E-mail address: filippo.zacchei@polimi.it (F. Zacchei).

https://doi.org/10.1016/j.cma.2026.118916
Received 17 December 2025; Received in revised form 23 February 2026; Accepted 8 March 2026
Available online 24 March 2026

mathematical models often involve Partial Differential Equations (PDEs) [1,2,7] which can be computationally intensive to solve, thus making many-query scenarios, like Uncertainty Quantification (UQ) tasks, prohibitive [8].

To address UQ in parameter estimation (or inverse UQ) a variety of methods have been developed. Bayesian approaches are particularly appealing, as they provide a principled way to update prior beliefs on the parameters using observed data, yielding a posterior distribution that reflects all sources of uncertainty. Since this posterior is rarely available in closed form, a range of computational techniques have been developed to obtain a reliable approximation. Among these, Markov Chain Monte Carlo (MCMC) methods [9–11] provide the most general option among sampling-based methods, as they asymptotically generate samples from the posterior under mild regularity conditions. Other approaches include, for instance, Importance Sampling [12], which estimates expectations by reweighting samples from a proposal distribution, Sequential Monte Carlo methods [13] that rely on ensembles of weighted particles to explore evolving posterior distributions, as well as Kalman filters and their nonlinear extensions [14,15], that are widely used for recursive Bayesian updates in sequential or state-space models.

While more computationally intensive than frequentist techniques such as maximum likelihood estimation, Bayesian methods offer significant advantages: they yield richer probabilistic information, including credibility intervals, sensitivity to outliers, and the ability to capture complex posterior features such as multi-modality and skewness, thus providing a more robust framework for UQ [16].

Variational Inference (VI) [17] has emerged as a scalable alternative, formulating posterior approximation as an optimization problem. Although efficient, VI may introduce bias due to the restricted expressiveness of the variational family. In contrast, MCMC methods impose fewer assumptions but require many forward model evaluations, leading to slow convergence when models are expensive.

To mitigate these limitations, we propose a novel sampling scheme, which we refer to as Multi-Fidelity Delayed Acceptance (MFDA). This method accelerates MCMC-based inference by integrating surrogate models of varying fidelity via multi-fidelity fusion [8] using Neural Networks (NNs) regression, and by incorporating filtering [8] strategies inspired by the Multi-Level Delayed Acceptance (MLDA) framework [18].

1.1. Existing approaches: multi-level methods

The classical MCMC algorithm explores the parameter space through an *outer loop* [8] process: at each iteration, a new candidate parameter is proposed based on the previous sample, and the forward model is evaluated to compute the likelihood of the candidate. The parameter is then accepted (and treated as being a sample drawn from the target posterior distribution) or rejected. Therefore, the effectiveness of MCMC sampling is based fundamentally on *(i)* the rapid and accurate evaluation of the likelihood function and *(ii)* the efficient exploration of the parameter space.

The rapid evaluation of the likelihood function can be addressed by replacing repeated high-fidelity forward solves with low-fidelity surrogates that approximate the parameter-to-observable map (or, equivalently, the log-likelihood computation). While replacing the high-fidelity model ($\mathbf{f}_{\mathrm{HF}}$) with a low-fidelity counterpart ($\mathbf{f}_{\mathrm{LF}}$) substantially reduces computational efforts, this might come with a loss in accuracy that can be quite hard to assess, as it usually depends on the characteristics of the specific surrogate model in use [19,20].

Data-driven surrogates can be highly accurate, but may require substantial high-fidelity training data [6], which limits their practicality when each forward solve is expensive. Common surrogate families range from global spectral approximations, such as polynomial chaos expansions (often coupled with dimension-reduction) [21,22], to kernel-based regressors such as Gaussian-process (kriging) models [23,24], as well as reduced-order models tailored to PDE-based forward operators [25,26]. Complementary surrogate formulations approximate the likelihood (or posterior) directly via spectral likelihood expansions [27,28], or embed local approximations within the transition kernel while retaining asymptotic exactness [29]. Since posterior estimates are primarily sensitive to approximation quality in regions carrying most posterior mass, adaptive and posterior-localised strategies refine the surrogate using information gathered during inference, e.g. sequential enrichment driven by posterior samples or error indicators [30,31]; applications in geophysical imaging likewise exploit posterior-focused refinement to reduce the number of expensive simulations [32]. However, adaptive refinement can still demand frequent high-fidelity evaluations when the posterior is highly concentrated or the forward map is strongly nonlinear. In this context, NN surrogates are attractive for their expressiveness and their ability to handle nonlinear and high-dimensional outputs, but they typically offer weaker a priori error control than classical spectral or projection-based surrogates; explicitly adaptive constructions for NN surrogates also remain comparatively less explored [33], motivating multi-level and multi-fidelity training strategies.

Regarding the exploration of the parameter space, MCMC samples often exhibit high autocorrelation, leading to an effective number of samples being only a small fraction of the total number of forward model evaluations [18,34,35]. Several strategies have been proposed to mitigate this limitation, such as, e.g, updating the proposal distribution using the information obtained during the process, as implemented in the adaptive Metropolis algorithm [36], or estimating gradient information from the forward model, as in the Metropolis-adjusted Langevin algorithm [37], Hamilton Monte Carlo [38] or No-U-Turn-Sampler [39]. However, these methods often pose a series of computational challenges. For instance, the required gradient information can be expensive to obtain, thereby limiting their applicability in complex or high-dimensional settings.

The idea of combining different solvers to enhance sampling-based Bayesian inference has led to several strategies, among which multi-level methods [40,41] have gained substantial attention in the last decade. These techniques were originally introduced in the context of forward UQ, leveraging hierarchies of numerical discretizations, typically generated by varying mesh resolution in a single

solver, to balance accuracy and computational cost [42]. In this setting, a high-fidelity model is used to ensure the required accuracy, while coarser discretizations offer inexpensive but less accurate approximations.

Multi-level strategies have subsequently been adapted to improve the efficiency of sampling methods in inverse UQ, primarily through filtering-based schemes. One prominent class involves Delayed Acceptance algorithms [43], where low-cost models are used to discard unlikely parameter proposals before evaluating them with high-fidelity simulations. Strategies like Multi-level MCMC [44] and MLDA [18] further extend this idea using chains that pass through models of increasing fidelity, retaining proposals only if they pass acceptance tests at all levels: by doing so, the number of expensive model evaluations is reduced by rejecting poor candidates earlier in the chain.

However, the efficiency of MLDA is highly sensitive to the consistency of the posterior approximations across fidelity levels. When the coarse forward models introduce non-negligible bias, proposals that appear acceptable at coarse levels may be rejected at finer levels, resulting in poor mixing. This issue is particularly pronounced when the coarse models differ from the high-fidelity model not only in discretization but also in physical representation or numerical formulation. Consequently, MLDA faces a fundamental limitation: one must either use very short sub-chain lengths, shifting a significant portion of the computational budget back to the high-fidelity model, or employ more accurate (and therefore more expensive) coarse models, which reduces the computational advantage of the multi-level approach.

To improve the alignment between fidelity levels, multi-fidelity fusion methods can be employed. Unlike filtering-based approaches, fusion methods evaluate multiple models simultaneously and combine their outputs to produce better predictions. Classical techniques include control variates [45–47] and co-kriging [48–50], although these approaches may scale poorly with problem dimension. In contrast NNs have proven effective for multi-fidelity fusion in complex, high-dimensional problems. Their capacity to learn nonlinear mappings between low- and high-fidelity models makes them well-suited for improving the consistency of surrogate-based approximations [51–56]. In delayed-acceptance hierarchies, improved cross-level consistency can reduce late-stage rejections and thereby support more effective filtering.

1.2. Our proposed strategy: multi-fidelity delayed acceptance MCMC

In this work, we build upon the filtering structure of the MLDA framework and extend it by incorporating a multi-fidelity fusion strategy based on NNs. The proposed approach consists of two stages. In the offline stage, the NNs are trained to learn a corrective mapping that reduces the discrepancy between the coarse solvers and the high-fidelity model. This is the only stage in which evaluations of the high-fidelity solver are required.

In the online stage, we perform multi-level sampling while retaining the filtering structure of MLDA, but using only the coarse solvers. At each fidelity level, the outputs of all coarser solvers up to that level are provided as input to the corresponding NN, which produces a corrected prediction. This improves the accuracy of the lowest fidelity levels and ensures that the finest level in the hierarchy provides an approximation that is sufficiently consistent with the high-fidelity model.

We refer to this enhanced approach as MFDA. By leveraging the structure and correlation across models of varying fidelity, MFDA provides a scalable tradeoff between two extremes: purely data-driven surrogates, which incur high offline training costs but offer fast online evaluations, and solver-based MLDA schemes, which require no offline phase but heavily rely on costly online high fidelity computations during sampling.

The proposed approach offers several advantages. First, by improving the consistency between posterior approximations across fidelity levels, it enables longer sub-chains at coarse levels, reduces sample autocorrelation, and increases the overall sampling efficiency. Second, the corrective capability of the multi-fidelity NNs allows the use of lower-cost and less accurate coarse models that would otherwise lead to poor performance in standard MLDA schemes. Third, the numerical results indicate that the NNs perform best when trained on a hierarchy of coarse model outputs, rather than relying solely on the most accurate surrogate, highlighting the benefit of exploiting correlations across multiple fidelity levels.

To our knowledge, the proposed MFDA framework represents one of the first systematic integration of multi-fidelity data-fusion using NNs and delayed-acceptance-style filtering within an MCMC-based workflow for inverse uncertainty quantification.

The structure of the paper is as follows. Section 2 presents the inverse UQ framework based on MCMC techniques, beginning with standard methods and progressing to the MLDA approach. This is followed by the introduction of multi-fidelity NNs and the development of the proposed MFDA scheme. Sections 3 and 4 detail the results of two numerical experiments offering an empirical assessment of all the listed advantages of MFDA. Finally, Section 5 concludes the work and outlines prospective avenues for further research.

2. Methodology

In this section, we review the foundations of Bayesian inverse problems and standard MCMC sampling strategies, highlighting their limitations. We then introduce surrogate models and the MLDA method, finally showing how to extend this framework using NN-based multi-fidelity fusion. This will yield our MFDA scheme.

2.1. Bayesian inverse problem for PDEs

Inverse problems deal with the use of actual measurements or observational data to infer the properties of a system described by a mathematical model [11,57]. These properties are often encoded in a vector of input parameters $\boldsymbol{\theta}$, whereas the model $\mathcal{G}(\boldsymbol{\theta})$ allows us

to express all the available knowledge about the way data can be explained in terms of the input parameters. Under the assumption of additive, independent noise, observations are linked to input parameters through the following relationship:

$$\boldsymbol{y}^{\text{obs}} = \mathcal{G}(\boldsymbol{\theta}) + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathcal{G} : \mathbb{R}^m \to \mathbb{R}^d$ denotes the parameter-to-observable map, while $\boldsymbol{\epsilon}$ accounts for observational noise and model mis-specification. The task of computing $\mathcal{G}(\boldsymbol{\theta})$ for a known input $\boldsymbol{\theta}$ defines the so-called *forward problem*, which very often involves the solution of a differential problem, in the form of either a system of Ordinary Differential Equations (ODEs) or PDEs; in this work we focus on the latter. Usually, $\mathcal{G}(\boldsymbol{\theta})$ consists of a set of solution components, whenever mimicking, for instance, data collected at a set of sensors, or more general outputs depending on the PDE solution and involving, e.g., spatial averages, fluxes, or derivatives. In this context, the input $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^m$ typically denotes a vector of model coefficients affecting the differential operator, as well as boundary or initial conditions, and is referred to as the set of model *parameters*.

A common instance of an inverse problem occurs when the model $\mathcal{G}$ is approximated by a high-fidelity simulator $\mathbf{f}_{\text{HF}}$, which provides a highly accurate numerical approximation to the PDE, leading to the reformulation of (1) as:

$$\boldsymbol{y}^{\text{obs}} = \mathbf{f}_{\text{HF}}(\boldsymbol{\theta}) + \boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon}$ is typically assumed to follow a zero-mean Gaussian with covariance $\boldsymbol{\Sigma}_\epsilon \in \mathbb{R}^{d\times d}$. By casting the inverse problem in a Bayesian framework, the goal is to determine the so-called *posterior distribution* $\pi(\boldsymbol{\theta}|\boldsymbol{y}^{\text{obs}})$ of the parameters $\boldsymbol{\theta}$ given $\boldsymbol{y}^{\text{obs}}$, which according to the *Bayes theorem* reads as:

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}^{\text{obs}}) = \frac{\pi(\boldsymbol{y}^{\text{obs}}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{y}^{\text{obs}})}. \tag{3}$$

Here $\pi(\boldsymbol{y}^{\text{obs}}|\boldsymbol{\theta})$ denotes the *likelihood* of the data given the parameters, $\pi(\boldsymbol{\theta})$ is the *prior distribution* of the parameters, encoding all the available knowledge on $\boldsymbol{\theta}$ before acquiring the data, and $\pi(\boldsymbol{y}^{\text{obs}})$ is the *marginal distribution* (or *evidence*), which plays the role of a normalizing constant, and is given by

$$\pi\left(\boldsymbol{y}^{\text{obs}}\right) = \int_{\boldsymbol{\Theta}} \pi(\boldsymbol{\theta})\pi\left(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}\right) d\boldsymbol{\theta}. \tag{4}$$

The integral in Eq. (4) cannot be computed analytically in general, so that a closed-form expression for the posterior distribution is therefore unavailable. A widely used strategy to approximate samples from the posterior distribution relies on MCMC methods [58]. These methods aim at generating a Markov chain in the input parameter space $\boldsymbol{\Theta}$, whose invariant distribution approximates the target posterior. Among the various MCMC algorithms, the Metropolis Hastings (MH) scheme is one of the most widely used options [59]. After initializing the chain, MH proceeds by generating candidate samples $\boldsymbol{\theta}'$, given the previous sample $\boldsymbol{\theta}$, from a proposal distribution $q(\cdot \mid \boldsymbol{\theta})$ and accepting or rejecting each candidate based on the acceptance probability

$$\alpha(\boldsymbol{\theta}', \boldsymbol{\theta}) = \min\left\{1, \frac{\pi(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}')\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}\right\}. \tag{5}$$

For symmetric proposals, $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$, the proposal ratio cancels out and $\alpha(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$ depends only on the posterior ratio. This mechanism ensures convergence of the Markov chain to the target posterior distribution over successive iterations [58]. The full procedure is reported in Algorithm 1.

Algorithm 1 Metropolis-Hastings (MH).

Input: Likelihood $\pi(\boldsymbol{y}^{\text{obs}} \mid \cdot)$, prior distribution $\pi(\cdot)$, proposal distribution $q(\cdot \mid \cdot)$, initial sample $\boldsymbol{\theta}_0$, number of samples N
Output: Chain of samples $\{\boldsymbol{\theta}_j\}_{j=1}^N$.

1: **for** $j = 1$ to N **do**
2: Propose $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}_{j-1})$.
3: Compute acceptance probability:

$$\alpha(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \min\left\{1, \frac{\pi(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}')\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}\right\}.$$

4: Accept $\boldsymbol{\theta}'$ with probability α; set $\boldsymbol{\theta}_j = \boldsymbol{\theta}'$ if accepted, otherwise $\boldsymbol{\theta}_j = \boldsymbol{\theta}_{j-1}$.
5: **end for**

Evaluating the likelihood at each iteration requires solving the high-fidelity model $\mathbf{f}_{\text{HF}}$ as defined in Eq. (2). By assuming that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$ the likelihood takes the form:

$$\pi(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{\Sigma}_\epsilon^{-\frac{1}{2}}\left(\mathbf{f}_{\text{HF}}(\boldsymbol{\theta}) - \boldsymbol{y}^{\text{obs}}\right)\right\|^2\right), \tag{6}$$

where $\|\cdot\|$ denotes the Euclidean norm. Every step of an MCMC algorithm thus requires the solution of the high-fidelity problem: this is the main reason why solving Bayesian inverse problems involving differential models is indeed an intensive task. To overcome this bottleneck, one can introduce a surrogate model $\mathbf{f}_{\text{LF}}$ to approximate $\mathbf{f}_{\text{HF}}$ and define a corresponding approximate likelihood.

This reduces the computational burden at the price of introducing an approximation in the posterior distribution. Unfortunately, the intrinsic ill-posedness of inverse problems might require extremely accurate surrogate models [19,35,60]. When a single surrogate is not sufficiently accurate or efficient, a structured approach based on multi-fidelity model management offers a viable alternative.

2.2. Multi-fidelity model management in Bayesian inverse problems

In many inverse problems involving PDEs, a high-fidelity model $\mathbf{f}_{\mathrm{HF}}$ is often available alongside one or more low-fidelity approximations. The objective of multi-fidelity model management is to design sampling strategies that combine these models efficiently, aiming to retain the accuracy of the high-fidelity model while reducing computational costs through surrogate evaluations [8]. In this work, we leverage both multi-fidelity *filtering* and *fusion* strategies. Following the terminology in the survey by Peherstorfer et al. [8], *filtering* denotes a coarse-to-fine screening mechanism in which inexpensive models are used to early-reject unpromising candidates before invoking the high-fidelity solver, and should not be confused with Bayesian state-estimation filtering. In contrast, *fusion* refers to approaches that combine information from multiple fidelity levels to improve predictive accuracy.

2.2.1. Multi-fidelity filtering: multi-level delayed acceptance

We consider a setting in which a high-fidelity model $\mathbf{f}_{\mathrm{HF}}$ is available, along with a hierarchy of low fidelity numerical approximations, referred to as *surrogate models*, denoted by $\mathbf{f}_{\mathrm{LF}}^{(1)}, \mathbf{f}_{\mathrm{LF}}^{(2)}, \dots, \mathbf{f}_{\mathrm{LF}}^{(L)}$, ordered by increasing accuracy and computational cost. Each surrogate model $\mathbf{f}_{\mathrm{LF}}^{(l)}$ defines in principle an approximate likelihood:

$$\tilde{\pi}_{\mathrm{LF}}^{(l)}(\boldsymbol{y} \mid \boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{\Sigma}_{\epsilon}^{-\frac{1}{2}}\left(\mathbf{f}_{\mathrm{LF}}^{(l)}(\boldsymbol{\theta}) - \boldsymbol{y}\right)\right\|^2\right), \quad l = 1, \dots, L. \tag{7}$$

These models are integrated into a multi-level MCMC scheme of $L+1$ levels. The levels $l = 1, \dots, L$ correspond to the low-fidelity model $\mathbf{f}_{\mathrm{LF}}^{(l)}$ and level $l = L+1$ corresponds to the high-fidelity model $\mathbf{f}_{\mathrm{HF}}$. Each level $l \leq L$ uses the surrogate $\mathbf{f}_{\mathrm{LF}}^{(l)}$ and its corresponding likelihood $\tilde{\pi}_{\mathrm{LF}}^{(l)}$. At level $L+1$, the high-fidelity model $\mathbf{f}_{\mathrm{HF}}$ and the corresponding likelihood π are employed to ensure accurate sampling.

At each level, a candidate $\boldsymbol{\theta}'$ is proposed by generating a sub-chain of length J_{l-1} (also referred to as sub-sampling rate) from the previous level $l-1$. The sub-chain lengths J_l are tuning parameters that affect efficiency (mixing and cost) but do not alter the high-fidelity invariant distribution for fixed finite values [18].

At the coarsest level $l = 1$, a standard MH algorithm is used employing the cheapest model, and new parameters are proposed using a proposal distribution $q(\cdot \mid \cdot)$, leading to the acceptance probability:

$$\alpha_1(\boldsymbol{\theta}', \boldsymbol{\theta}) = \min\left\{1, \frac{\pi_{\mathrm{LF}}^{(1)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta}')\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{\pi_{\mathrm{LF}}^{(1)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}\right\}. \tag{8}$$

As the level l increases, finer and more expensive models are queried. The acceptance probability at level l is then computed as:

$$\alpha_l(\boldsymbol{\theta}', \boldsymbol{\theta}) = \min\left\{1, \frac{\tilde{\pi}_{\mathrm{LF}}^{(l)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta}')\tilde{\pi}_{\mathrm{LF}}^{(l-1)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta})}{\tilde{\pi}_{\mathrm{LF}}^{(l)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta})\tilde{\pi}_{\mathrm{LF}}^{(l-1)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta}')}\right\}, \qquad l = 2, \dots, L. \tag{9}$$

At the finest level $l = L+1$ we have:

$$\alpha_{L+1}(\boldsymbol{\theta}', \boldsymbol{\theta}) = \min\left\{1, \frac{\pi(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta}')\tilde{\pi}_{\mathrm{LF}}^{(L)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta})}{\pi(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta})\tilde{\pi}_{\mathrm{LF}}^{(L)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta}')}\right\} \tag{10}$$

See Appendix for more details on the derivation of (9) and (10).

This hierarchical design allows for the early rejection of poor candidates using cheaper models, significantly reducing the number of high-fidelity evaluations while maintaining sampling accuracy. Moreover, the nested sub-chains at different levels effectively reduce sample autocorrelation, improving the mixing of the chain.

Importantly, it can be shown that the chain at the finest level satisfies detailed balance with respect to the posterior distribution defined by the high-fidelity model [18,61]. In particular, coarse target distributions must assign positive probability to all states reachable under the proposal (cf. Theorem 1 in [43]). However, poorly aligned low-fidelity targets may still create mixing bottlenecks; if a coarse level strongly down-weights a high-fidelity mode, transitions into that region are rarely promoted to finer levels. Note that the approach remains transparent to the particular choice of the forward model, allowing it to flexibly handle both linear and nonlinear, stationary or transient simulations without significant alterations [1,43,62,63].

In this formulation, the proposal kernel is used only to generate candidate states at the coarsest level. Additional, level-specific proposal kernels at higher levels can be introduced when considering different parameter blocks across distinct fidelity levels [18]. It is worth noting that when the approximation maps do not depend on the current state, the acceptance probabilities for level $l > 1$ in Eqs. (9)–(10) depend only on the targets of the adjacent levels. This is a direct consequence of the delayed-acceptance construction: each coarse-level transition kernel satisfies detailed balance with respect to its own target (see, e.g., Lemma 1 in [18]). As a result, the forward/reverse proposal terms generally present in Delayed Acceptance (DA) acceptance kernel [43] can be substituted with the ratio of the corresponding target distributions. Therefore, at levels $\ell > 1$ the acceptance probability reduces to the ratio of adjacent targets used in MLDA, also when the coarsest-level proposal is not symmetric. In general, the current formulation does not allow the proposal to be adaptive. Enabling this option would require the acceptance ratios at level $l > 1$ in Eqs. (9) and (10) to be adjusted,

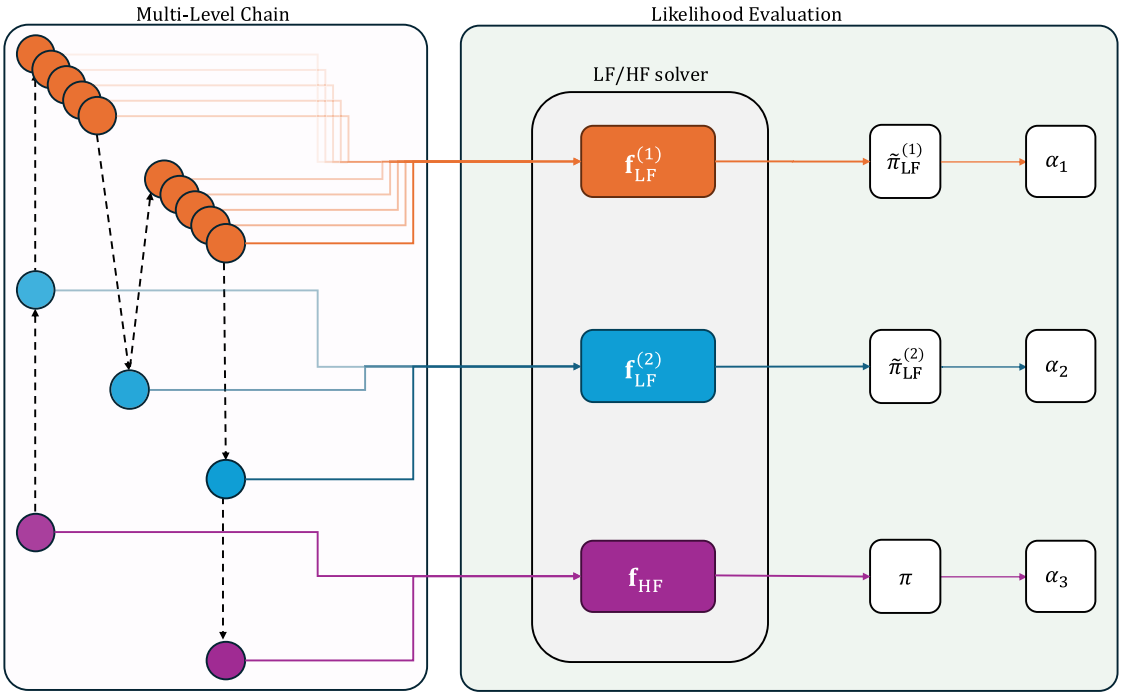


Fig. 1. Schematic representation of the MLDA scheme, for an instance of two low-fidelity solvers and one high-fidelity solver. At each level, a candidate θ' is proposed by generating a sub-chain of length J_{l-1}. Each coarse level l uses the surrogate $\mathbf{f}_{\mathrm{LF}}^{(l)}$ and its corresponding likelihood $\tilde{\pi}_{\mathrm{LF}}^{(l)}$, while the finest level uses $\mathbf{f}_{\mathrm{HF}}$ and its corresponding likelihood π.

since detailed balance with respect to the level target distributions would no longer hold. A straightforward remedy is to keep the adaptation frozen while proposing a new fine-level sample (i.e., to choose an adaptation period that is a multiple of the product of the sub-chain lengths), which is the strategy adopted in the numerical examples.

In practice, J_l is tuned to maintain adequate acceptance between consecutive levels and to de-correlate the coarse sub-chains. A common heuristic is to use larger J_l on cheaper (coarser) levels and smaller J_l on expensive levels, guided by short pilot runs that monitor acceptance rates and finest-level correlation [18]. A schematic representation of the method is provided in Fig. 1, and the detailed algorithm is presented in Algorithm 2.

2.3. Multi-fidelity fusion through neural networks for regression

Although MLDA can significantly reduce reliance on costly high-fidelity evaluations, its practical efficiency is still inhibited by discrepancies among different fidelity levels. Specifically, when low-fidelity approximations poorly represent their high-fidelity counterparts, the acceptance probability at the second stage deteriorates significantly. As a consequence, frequent rejections occur unless extremely low sub-sampling rates are adopted, which in turn result in suboptimal acceptance rates, inadequate mixing, and ultimately diminished effective sample sizes [18]. To mitigate this problem, the current work introduces a multi-fidelity regression strategy leveraging artificial NNs, extending the recent framework of [55,64].

Here, the fundamental assumption is the existence of a nonlinear relationship $\mathcal{F}_{\mathrm{MF}}$ linking n low-fidelity model outputs to their high-fidelity counterpart [65], represented as follows:

$$\mathbf{f}_{\mathrm{HF}}(\boldsymbol{\theta}) = \mathcal{F}_{\mathrm{MF}}\left(\boldsymbol{\theta}, \mathbf{f}_{\mathrm{LF}}^{(1)}(\boldsymbol{\theta}), \mathbf{f}_{\mathrm{LF}}^{(2)}(\boldsymbol{\theta}), \ldots, \mathbf{f}_{\mathrm{LF}}^{(n)}(\boldsymbol{\theta})\right). \tag{11}$$

The key benefit of using NNs with respect to other multi-fidelity regression frameworks lies in the simultaneous and efficient incorporation of multiple low-fidelity approximations as inputs. Traditionally, even in hierarchical settings, multi-fidelity regression methods such as auto-regressive co-kriging [50] incorporate only a single lower-fidelity model at each stage. Although these schemes can be extended to multiple fidelity levels by sequentially stacking such mappings, the resulting structure restricts information flow to a one-directional correction chain. This simplifies implementation and reduces memory requirements, but prevents the method from exploiting complementary information across different solvers. In contrast, the inherent flexibility and nonlinear representational capacity of NNs allow efficient fusion of multiple low-fidelity sources within a unified model, enhancing predictive accuracy without increasing numerical complexity. An example of the effectiveness of NNs in combining multiple sources of information is provided in [64].

Algorithm 2 Multi-level Delayed Acceptance (MLDA).

Input: High-fidelity likelihood $\pi(\boldsymbol{y}^{\text{obs}} \mid \cdot)$, coarse likelihoods $\tilde{\pi}_\ell(\boldsymbol{y}^{\text{obs}} \mid \cdot)$ for $\ell = 1, \dots, L$, prior distribution $\pi(\cdot)$, symmetric proposal distribution $q(\cdot \mid \cdot)$, initial sample $\boldsymbol{\theta}_0$, number of fine-level samples J, sub-chain lengths J_ℓ for $\ell = 1, \dots, L$.

Output: Chain of samples $\{\boldsymbol{\theta}_j\}_{j=1}^{J}$.

1: Initialise $\boldsymbol{\theta}_0^\ell = \boldsymbol{\theta}_0$ and subchains steps counters $n_\ell \leftarrow 0$ for all $\ell = 1, \dots, L$.
2: **for** $j = 0$ to $J - 1$ **do**
3: **Level 1:** Run a sub-chain of length J_1 using Algorithm 1, starting from current state $\boldsymbol{\theta}_0^1$ and proposal $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$
4: Set $\ell = 2$
5: **while** $\ell \leq L$ **do**
6: **Propose from level** $\ell - 1$**:** $\tilde{\boldsymbol{\theta}}^\ell \leftarrow \boldsymbol{\theta}_{J_{\ell-1}}^{\ell-1}$ (last state of the level-$(\ell-1)$ sub-chain).
7: Compute acceptance probability between current state $\boldsymbol{\theta}_{n_\ell}^\ell$ and proposed state $\tilde{\boldsymbol{\theta}}^\ell$:

$$\alpha_\ell = \min\left\{1, \frac{\tilde{\pi}_\ell(\boldsymbol{y}^{\text{obs}} \mid \tilde{\boldsymbol{\theta}}^\ell)\, \tilde{\pi}_{\ell-1}(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}_{n_\ell}^\ell)}{\tilde{\pi}_\ell(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}_{n_\ell}^\ell)\, \tilde{\pi}_{\ell-1}(\boldsymbol{y}^{\text{obs}} \mid \tilde{\boldsymbol{\theta}}^\ell)}\right\}$$

8: With probability α_ℓ, accept: $\boldsymbol{\theta}_{n_\ell+1}^\ell \leftarrow \tilde{\boldsymbol{\theta}}^\ell$; otherwise reject: $\boldsymbol{\theta}_{n_\ell+1}^\ell \leftarrow \boldsymbol{\theta}_{n_\ell}^\ell$
9: Increment ℓ-level sub-chain steps counter $n_\ell \leftarrow n_\ell + 1$
10: **if** $n_\ell = J_\ell$ **then**
11: Set $\ell \leftarrow \ell + 1$
12: **else**
13: Reset $j_k = 0$, $\boldsymbol{\theta}_0^k \leftarrow \boldsymbol{\theta}_{n_\ell}^\ell$ for all $1 \leq k < \ell$, and return to Step 3
14: **end if**
15: **end while**
16: Set final proposal: $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}_{J_L}^L$
17: Compute final acceptance probability between current $\boldsymbol{\theta}_j$ and proposed $\tilde{\boldsymbol{\theta}}$:

$$\alpha_{L+1} = \min\left\{1, \frac{\pi(\boldsymbol{y}^{\text{obs}} \mid \tilde{\boldsymbol{\theta}})\, \tilde{\pi}_L(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}_j)}{\pi(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}_j)\, \tilde{\pi}_L(\boldsymbol{y}^{\text{obs}} \mid \tilde{\boldsymbol{\theta}})}\right\}$$

18: With probability α_{L+1}, accept: $\boldsymbol{\theta}_{j+1} \leftarrow \tilde{\boldsymbol{\theta}}$; otherwise reject: $\boldsymbol{\theta}_{j+1} \leftarrow \boldsymbol{\theta}_j$
19: Reset $n_\ell = 0$, $\boldsymbol{\theta}_0^\ell \leftarrow \boldsymbol{\theta}_{j+1}$ for all $1 \leq \ell \leq L$
20: **end for**

For this reason, to model this multi-fidelity mapping, a densely connected feedforward NN architecture is adopted. Such a network, consisting of N_L layers, defines a parametric function $\mathbf{f}_{\text{MF}}(\cdot; \mathbf{W}, \mathbf{b})$, where $\mathbf{W} = \{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(N_L)}\}$ and $\mathbf{b} = \{\mathbf{b}^{(1)}, \dots, \mathbf{b}^{(N_L)}\}$ denote weight matrices and biases, respectively. Each network layer performs an affine transformation followed by a nonlinear activation function, as follows:

$$\mathbf{z}^{(k)} = \sigma\left(\mathbf{W}^{(k)}\mathbf{z}^{(k-1)} + \mathbf{b}^{(k)}\right), \quad k = 1, \dots, N_L, \tag{12}$$

with $\mathbf{z}^{(0)}$ representing the input vector (i.e., $\mathbf{z}_0 = [\boldsymbol{\theta}, \mathbf{f}_{\text{LF}}^{(1)}(\boldsymbol{\theta}), \dots]^\top$ is the concatenation of parameters $\boldsymbol{\theta}$ and outputs from multiple low-fidelity models), and $\sigma(\cdot)$ indicating a suitable nonlinear activation function.

A schematic representation of a multi-fidelity NN, for a generic instance of two low-fidelity solvers and a reference high-fidelity solver, is presented in Fig. 2. Input parameters $\boldsymbol{\theta}$ are passed to the low-fidelity solvers. The outputs of the solvers and the input parameter $\boldsymbol{\theta}$ are passed to the first hidden layer of the NN. The subsequent hidden layers perform nonlinear transformations to capture complex correlations across fidelities. The final network output serves as an approximation of the high-fidelity model.

We note that a variety of multi-fidelity neural emulator architectures have been proposed in the literature (e.g., decomposed linear/nonlinear corrections [51], residual architecture [66] and auto-regressive formulations [55]), and these alternatives can be integrated within the MFDA workflow without conceptual changes. In this work we focus on a monolithic multi-branch architecture (see B–C for more details), since *(i)* it retains the expressivity of generic NNs while avoiding an explicit linear/nonlinear decomposition (which may make learning more difficult when correlation is strongly nonlinear), and *(ii)* it is easier to train, especially when multiple low fidelity inputs are present. Decomposed formulations (e.g., linear + nonlinear or residual corrections [51,66]) can be seen as introducing additional inductive bias that may improve data-efficiency when their assumptions hold, but they are not required for MFDA. MFDA does not require identical parameterizations across fidelities: it only requires that the information provided to the emulator can be expressed in a common feature space; in particular, low-fidelity outputs can be used as inputs even when the low-fidelity model depends on a reduced parameter set, provided a consistent mapping from the high-fidelity parameters to the low-fidelity inputs is available.

Training of the network parameters is performed offline. Specifically, a set of N_{train} parameter samples $\{\boldsymbol{\theta}_j\}_{j=1}^{N_{\text{train}}}$ is generated, and for each sample, both the high-fidelity and all n low-fidelity solvers are evaluated. The resulting dataset, consisting of N_{train} paired

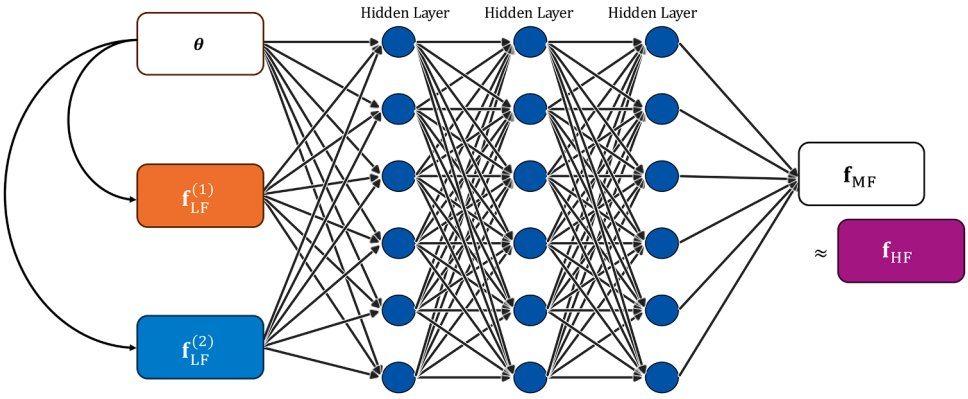


Fig. 2. Schematic representation of a multi-fidelity NN, for a generic instance of two low-fidelity solvers and a reference high-fidelity solver. Input parameters $\boldsymbol{\theta}$ are passed to the low-fidelity solvers. The outputs of the solvers and the input parameter $\boldsymbol{\theta}$ are passed to the first hidden layer of the NN. The subsequent hidden layers perform nonlinear transformations to capture complex correlations across fidelities. The final network output serves as an approximation of the high-fidelity model.

high- and multi-fidelity evaluations, is used to train the NN by minimizing the Mean Squared Error (MSE) between the high-fidelity model outputs and the network predictions. The optimization problem can be expressed as:

$$\mathbf{W}, \mathbf{b} = \arg\min_{\mathbf{W},\mathbf{b}} \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left\| \mathbf{f}_{\mathrm{HF}}(\boldsymbol{\theta}_j) - \mathbf{f}_{\mathrm{MF}}\left(\boldsymbol{\theta}_j, \mathbf{f}_{\mathrm{LF}}^{(1)}(\boldsymbol{\theta}_j), \ldots, \mathbf{f}_{\mathrm{LF}}^{(n)}(\boldsymbol{\theta}_j); \mathbf{W}, \mathbf{b}\right) \right\|^2. \tag{13}$$

This minimization is carried out using standard stochastic gradient–based optimizers, such as stochastic gradient descent and its variants (e.g., Adam [67]), as commonly employed in deep learning frameworks.

2.4. A remark on low-fidelity solvers

In standard multi-level frameworks concerning physical problems and PDEs, low-fidelity solvers are typically constructed by systematically reducing the resolution or complexity of a high-fidelity model. Common strategies include mesh coarsening in finite element or finite volume schemes, simplified discretizations, or algebraic multi-level formulations such as multigrid approaches [18,40–42].

However, in many practical scenarios, lower-fidelity models may differ from the high-fidelity solver not only in resolution but also in their underlying physical assumptions, boundary conditions, or numerical formulations [68–71]. In such cases, correlations across fidelity levels can become strongly nonlinear [65]. As a result, the filtering at the lower levels of MLDA scheme fails to reflect the structure of the fine posterior distribution, thereby limiting the effectiveness of MLDA approach.

To address this challenge, non-linear information fusion techniques [64,65], such as the NN–based strategy adopted in this work, provide a flexible solution. By integrating multiple low-fidelity models within a unified nonlinear regression framework, the proposed approach can effectively capture complex dependencies between low- and high-fidelity responses. As a result, the NN-based fusion yields surrogate representations that more closely align the coarse-level approximations with the high-fidelity posterior.

2.5. Multi-fidelity delayed acceptance

The multi-fidelity NNs are embedded into the multi-level MCMC sampling framework of L levels to form the proposed MFDA scheme. For each fidelity level $l = 1, \ldots, L$, we construct a multi-fidelity NN

$$\mathbf{f}_{\mathrm{MF}}^{(l)}(\boldsymbol{\theta}) = \mathbf{f}_{\mathrm{MF}}\left(\boldsymbol{\theta}, \mathbf{f}_{\mathrm{LF}}^{(1)}(\boldsymbol{\theta}), \ldots, \mathbf{f}_{\mathrm{LF}}^{(l)}(\boldsymbol{\theta})\right), \tag{14}$$

which takes as inputs the parameter vector and the outputs of all solvers up to level l. Each $\mathbf{f}_{\mathrm{MF}}^{(l)}$ follows the architecture introduced in Section 2.3. The algorithm operates in two stages:

- Offline phase (training): we generate a set of parameter samples $\{\boldsymbol{\theta}_j\}_{j=1}^{N_{\text{train}}}$ (e.g., via Latin hypercube sampling) and evaluate all available solvers at these points. We train each network $\mathbf{f}_{\mathrm{MF}}^{(l)}$ independently by minimizing the mean squared error between the high-fidelity outputs and the network predictions, as in (13). This is the only part where we rely on the high-fidelity model. The number of training points should be selected so that the surrogate at the finest level $l = L$ attains the desired accuracy, as this model replaces the high-fidelity solver at the final stage of the multi-level chain.
- Online phase (inference): In this phase, parameter samples are drawn using the standard MLDA procedure, with likelihood evaluations replaced by their multi-fidelity NN counterparts:

$$\tilde{\pi}_{\mathrm{MF}}^{(l)}(\boldsymbol{y}^{\mathrm{obs}} \mid \boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-\frac{1}{2}}\left(\mathbf{f}_{\mathrm{MF}}^{(l)}(\boldsymbol{\theta}) - \boldsymbol{y}^{\mathrm{obs}}\right)\right\|^2\right), \quad l = 1, \ldots, L. \tag{15}$$

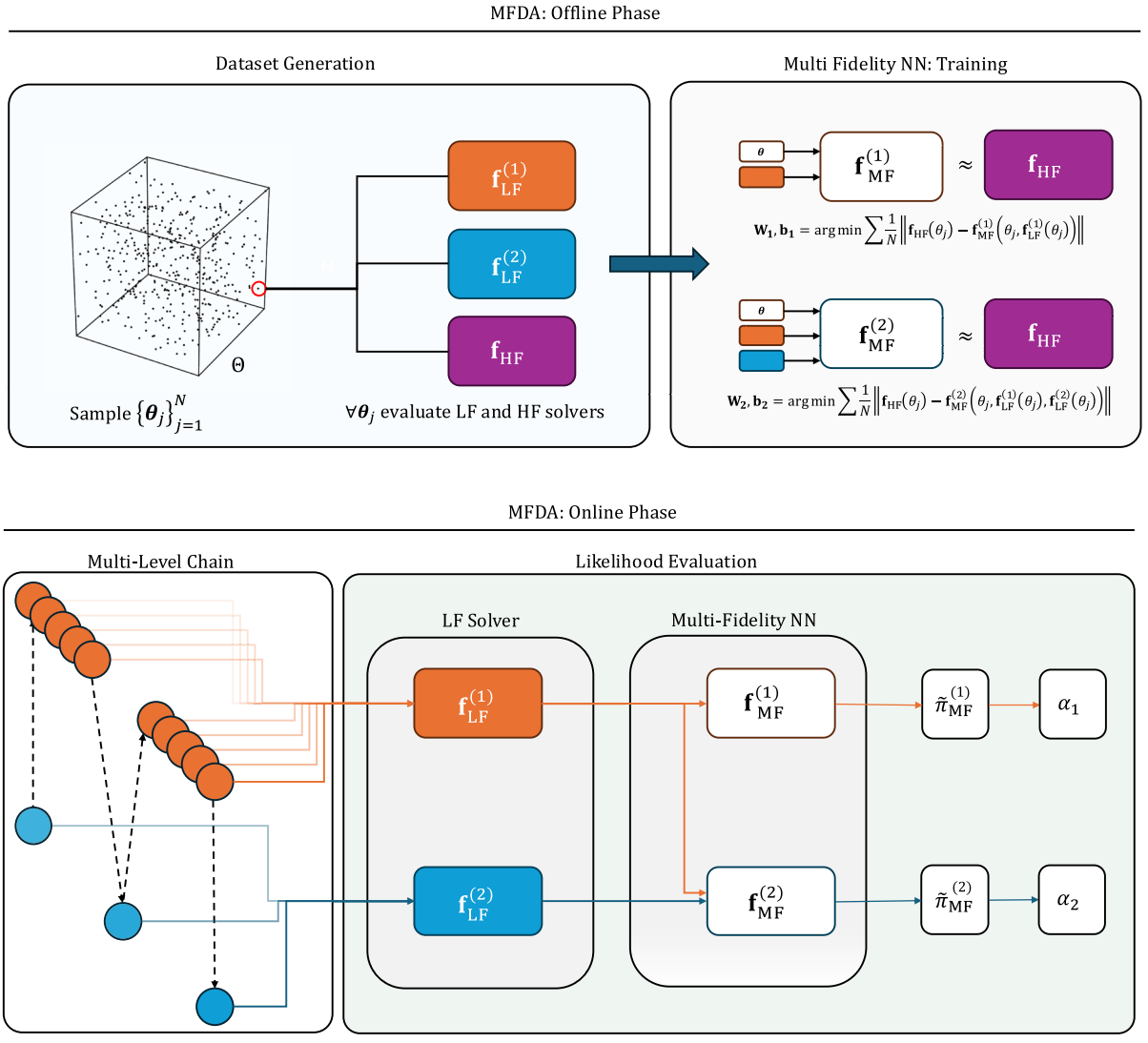


Fig. 3. Schematic representation of the Multi-Fidelity Delayed Acceptance (MFDA) framework for an instance of two low-fidelity solvers and one high-fidelity solver. First row: offline phase. A dataset of parameter samples is generated and each solver is evaluated for every parameter instance. Then multi-fidelity neural networks are trained to approximate the high-fidelity reference. Second row: online phase. A Markov chain of parameter samples is generated using a multi-level structure. At each level we evaluate the corresponding low-fidelity solvers and neural networks to compute the likelihood and acceptance rate.

These adjusted multi-fidelity likelihoods naturally replace standard likelihoods in the acceptance criterion of the MLDA framework. Specifically, the modified acceptance probability at fidelity level l becomes:

$$\alpha_l(\boldsymbol{\theta}', \boldsymbol{\theta}) = \min\left\{1, \frac{\tilde{\pi}_{\text{MF}}^{(l)}(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}')\tilde{\pi}_{\text{MF}}^{(l-1)}(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta})}{\tilde{\pi}_{\text{MF}}^{(l)}(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta})\tilde{\pi}_{\text{MF}}^{(l-1)}(\boldsymbol{y}^{\text{obs}} \mid \boldsymbol{\theta}')}\right\}, \quad l = 2, \ldots, L. \tag{16}$$

At level $l = 1$ we use MH scheme using $\tilde{\pi}_{\text{MF}}^{(1)}$. In the proposed MFDA implementation, the L coarse solvers are used online and their outputs is the input argument of the multi-fidelity NNs and the corresponding likelihoods. The high-fidelity model is not called during sampling. Preserving the filtering structure enables reuse of coarse-level evaluations as inputs to finer-level networks, avoiding redundant computations. Hence, the only additional online cost relative to standard MLDA is the (typically negligible) NN inference, while the high-fidelity solver is not called online.

The Markov chain at the finest level satisfies detailed balance with respect to the posterior associated with the surrogate likelihood $\tilde{\pi}_{\text{MF}}^{(L)}$, rather than with the exact high-fidelity posterior. We expect that the use of multi-fidelity inputs allows one to construct sufficiently accurate surrogate posteriors with substantially fewer high-fidelity training evaluations. The choice between MFDA and

Algorithm 3 Multi-Fidelity Delayed Acceptance (MFDA).

Input: Number of training samples N_{train}; high-fidelity model $\mathbf{f}_{\text{HF}}$; low-fidelity surrogate models $\mathbf{f}_{\text{LF}}^{(\ell)}$ for $\ell = 1, \dots, L$; prior distribution $\pi(\cdot)$, symmetric proposal distribution $q(\cdot \mid \cdot)$; initial state $\boldsymbol{\theta}_0$; sub-chain lengths J_ℓ for $\ell = 1, \dots, L-1$, and number of fine-level samples J.

Output: Sample chain $\{\boldsymbol{\theta}_j\}_{j=1}^{J}$.

Offline Phase (Training):
Collect training data $\{(\boldsymbol{\theta}_i, \mathbf{f}_{\text{HF}}(\boldsymbol{\theta}_i))\}_{i=1}^{N_{\text{train}}}$.
for $\ell = 1$ to L **do**
  Evaluate and store $\mathbf{f}_{\text{LF}}^{(\ell)}(\boldsymbol{\theta}_i)$ for all i.
  Train multi-fidelity surrogate $\mathbf{f}_{\text{MF}}^{(\ell)}$ using inputs $\left(\boldsymbol{\theta}_i, \mathbf{f}_{\text{LF}}^{(1)}, \dots, \mathbf{f}_{\text{LF}}^{(\ell)}\right)$.
end for

Online Phase (Inference):
Initialise $\boldsymbol{\theta}_0^{\ell} = \boldsymbol{\theta}_0$ and subchains steps counters $n_\ell \leftarrow 0$ for all $\ell = 1, \dots, L$.
Initialise cache for low fidelity surrogate models evaluations.
for $j = 0$ to $J - 1$ **do**
  Level 1: Run a sub-chain of length J_1 using Algorithm 1, starting from $\boldsymbol{\theta}_0^1$.
  Set $\ell \leftarrow 2$
  while $\ell < L$ **do**
    Set proposal $\tilde{\boldsymbol{\theta}}^{\ell} \leftarrow \tilde{\boldsymbol{\theta}}^{\ell-1}$ (last state of the level-$(\ell-1)$ sub-chain).
    Evaluate $\mathbf{f}_{\text{LF}}^{(\ell)}(\tilde{\boldsymbol{\theta}}^{\ell})$ and store in cache.
    Retrieve surrogate outputs and compute $\mathbf{f}_{\text{MF}}^{(\ell)}(\tilde{\boldsymbol{\theta}}^{\ell})$ and likelihood $\tilde{\pi}_{\text{MF}}^{(\ell)}(\boldsymbol{y} \mid \tilde{\boldsymbol{\theta}}^{\ell})$ from Eq. (15).
    Compute acceptance probability:

$$\alpha_\ell = \min\left\{1, \frac{\tilde{\pi}_{\text{MF}}^{(\ell)}(\boldsymbol{y} \mid \tilde{\boldsymbol{\theta}}^{\ell})\, \tilde{\pi}_{\text{MF}}^{(\ell-1)}(\boldsymbol{y} \mid \boldsymbol{\theta}_{n_\ell}^{\ell})}{\tilde{\pi}_{\text{MF}}^{(\ell)}(\boldsymbol{y} \mid \boldsymbol{\theta}_{n_\ell}^{\ell})\, \tilde{\pi}_{\text{MF}}^{(\ell-1)}(\boldsymbol{y} \mid \tilde{\boldsymbol{\theta}}^{\ell})}\right\}.$$

    With probability α_ℓ, accept: $\boldsymbol{\theta}_{n_\ell+1}^{\ell} \leftarrow \tilde{\boldsymbol{\theta}}^{\ell}$; else set $\boldsymbol{\theta}_{n_\ell+1}^{\ell} \leftarrow \boldsymbol{\theta}_{n_\ell}^{\ell}$.
    Increment ℓ-level sub-chain steps counter $n_\ell \leftarrow n_\ell + 1$.
    if $n_\ell = J_\ell$ **then**
      Set $\ell \leftarrow \ell + 1$
    else
      Reset $j_k = 0$, $\boldsymbol{\theta}_0^k \leftarrow \boldsymbol{\theta}_{n_\ell}^{\ell}$ for all $1 \le k < \ell$, and return to Step 12
    end if
  end while
  Final Level L**:** Let $\tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}}^{L-1}$.
  Compute $\mathbf{f}_{\text{MF}}^{(L)}(\tilde{\boldsymbol{\theta}})$ and likelihood $\tilde{\pi}_{\text{MF}}^{(L)}(\boldsymbol{y} \mid \tilde{\boldsymbol{\theta}})$.
  Compute final acceptance probability:

$$\alpha_L = \min\left\{1, \frac{\tilde{\pi}_{\text{MF}}^{(L)}(\boldsymbol{y} \mid \tilde{\boldsymbol{\theta}})\, \tilde{\pi}_{\text{MF}}^{(L-1)}(\boldsymbol{y} \mid \boldsymbol{\theta}_j)}{\tilde{\pi}_{\text{MF}}^{(L)}(\boldsymbol{y} \mid \boldsymbol{\theta}_j)\, \tilde{\pi}_{\text{MF}}^{(L-1)}(\boldsymbol{y} \mid \tilde{\boldsymbol{\theta}})}\right\}.$$

  With probability α_L, set $\boldsymbol{\theta}_{j+1} \leftarrow \tilde{\boldsymbol{\theta}}$; else set $\boldsymbol{\theta}_{j+1} \leftarrow \boldsymbol{\theta}_j$.
  Reset $\boldsymbol{\theta}_0^{\ell} \leftarrow \boldsymbol{\theta}_{j+1}$ and $n_\ell \leftarrow 0$ for all $\ell = 1, \dots, L$.
end for

standard MLDA depends on whether the offline training cost is compensated by the improvement in effective sample size and mixing during inference.

The NNs training set is generated offline (prior to inference) and is not adapted to concentrate accuracy around posterior modes. Further improvements could be achieved through the incorporation of adaptive training mechanisms for the NNs during the online phase. This adaptation could substantially reduce the dependence on the offline training phase. However, implementing online learning for NNs remains a non-trivial challenge and may require significant methodological adjustments, potentially involving techniques such as Gaussian process-based sequential design [72–74], deep kernel learning [75], or enforcing physics informed residuals [76]. Additionally, at present, the multi-fidelity surrogate models are not used to select or update the proposal distribution, although in principle offline information could be exploited to guide proposal design. Fig. 3 illustrates the workflow, and Algorithm 3 provides the full procedure.

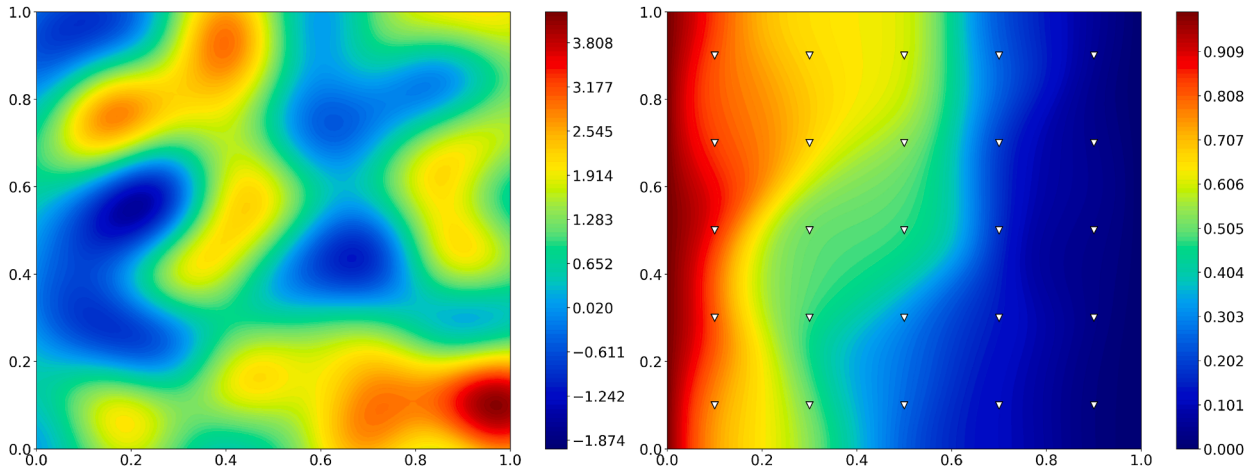


Fig. 4. (a) Transmissivity field corresponding to mesh $\mathcal{T}_{\text{HF}}$ and (b) hydraulic head using high-fidelity solver. Inverted triangles indicate sensor locations.

3. Numerical experiments i: isotropic groundwater flow - transmissivity reconstruction

The first test concerns the reconstruction of a spatially varying subsurface transmissivity field from hydraulic pressure measurements, using the benchmark configuration of [1]. This problem is governed by a linear, stationary elliptic equation and serves as a standard reference in inverse UQ.

The evaluation focuses on two aspects:

1. **Forward accuracy:** The ability of multi-fidelity NNs to enhance the accuracy of coarse models, measured via the Root Mean Squared Error (RMSE) with respect to the high-fidelity solver solution.
2. **Sampling efficiency:** The effectiveness of the MFDA scheme for Bayesian inversion, compared with MH and MLDA algorithms. In particular, sampling efficiency is quantified using the time-to-Effective Sample Size (ESS) ratio. For a Markov chain of N samples, the ESS is usually defined as

$$N_{\text{eff}} = \frac{N}{1 + 2\sum_{t=1}^{\infty} \rho_t}, \tag{17}$$

where ρ_t denotes the autocorrelation at lag t. This reflects the number of effectively independent samples produced by the chain. The objective is to obtain the highest possible ESS for a given computational budget.

3.1. Problem description

Let $\Omega = (0,1)^2$ denote the spatial domain with boundary $\Gamma = \partial\Omega$. The steady-state hydraulic head $h(\mathbf{x})$ satisfies the following diffusion equation

$$-\nabla \cdot (T(\mathbf{x})\nabla h(\mathbf{x})) = g(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{18}$$

where $T(\mathbf{x})$ denotes the transmissivity field and $g(\mathbf{x})$ is the source term. The boundary conditions are

$$h(\mathbf{x}) = h_D(\mathbf{x}) \quad \text{on } \Gamma_D, \qquad (-T(\mathbf{x})\nabla h(\mathbf{x})) \cdot \mathbf{n} = q_N(\mathbf{x}) \quad \text{on } \Gamma_N, \tag{19}$$

where $\Gamma = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$. In particular, $q_N(\mathbf{x}) = 0$ on

$$\Gamma_N = \Gamma_N^{bottom} \cup \Gamma_N^{top} = \{(x_1, 0) \ : \ x_1 \in (0,1)\} \cup \{(x_1, 1) \ : \ x_1 \in (0,1)\}$$

whereas on

$$\Gamma_D = \Gamma_D^{left} \cup \Gamma_D^{right} = \{(0, x_2) \ : \ x_2 \in (0,1)\} \cup \{(1, x_2) \ : \ x_2 \in (0,1)\}$$

we set $h_D(\mathbf{x}) = 1$ on Γ_D^{left} and $h_D(\mathbf{x}) = 0$ on Γ_D^{right}.

A widely used model for the prior distribution of aquifer transmissivity in groundwater hydrology is the log-Gaussian random field [1]. In this approach, the logarithm of the transmissivity, $\log T(\mathbf{x})$, is modeled as a Gaussian random field [77] characterized by a specified mean and covariance structure. The mean of $\log T(\mathbf{x})$ is set to $\mu = 1$, and the covariance function is given by

$$C(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2\lambda^2}\right), \quad \mathbf{x}_1, \mathbf{x}_2 \in \Omega, \tag{20}$$

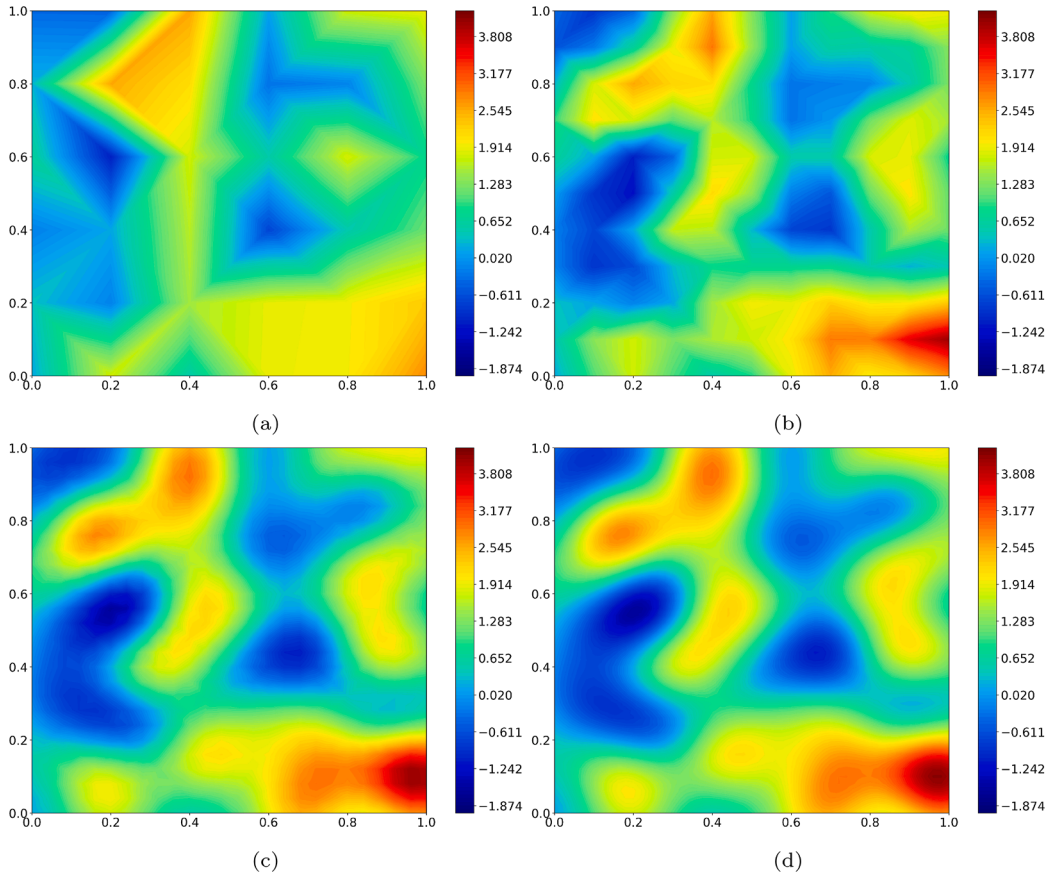


Fig. 5. Transmissivity fields corresponding to low-fidelity meshes: (a) $\mathcal{T}_1$, (b) $\mathcal{T}_2$, (c) $\mathcal{T}_3$, (d) $\mathcal{T}_4$.

where $\sigma = 0.1$ and λ denotes the correlation length. To enable a finite-dimensional parameterization, the log-Gaussian random field is approximated using a truncated Karhunen–Loève (KL) [78] expansion. Specifically,

$$\log T(\mathbf{x}) = \mu + \sum_{i=1}^{m} \sqrt{\lambda_i}\psi_i(\mathbf{x})\,\theta_i, \tag{21}$$

where $\{\lambda_i\}_{i=1}^m$ and $\{\psi_i(\mathbf{x})\}_{i=1}^m$ are the m largest eigenvalues and associated L^2-orthonormal eigenfunctions of the covariance operator with kernel $C(\mathbf{x}_1, \mathbf{x}_2)$, and $\theta_i \sim \mathcal{N}(0,1)$ are independent standard normal random variables. In this study, we deal with $m = 64$ modes to represent the spatially distributed random field, ensuring that approximately 95% of the field variance is retained. The parameter vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)^\top \sim \mathcal{N}(0, I_m)$ serves as the set of uncertain parameters in the stochastic PDE model.

Therefore, our objective is to infer the posterior distribution of $\boldsymbol{\theta}$, given noisy measurements of the hydraulic head h at $d = 25$ discrete sensor locations $\{\mathbf{x}_j\}_{j=1}^d \subset \Omega$. These measurements are collected in the observation vector $\mathbf{y}^{\text{obs}} \in \mathbb{R}^d$.

3.2. MFDA: setting

For this test case we design a MFDA scheme with $L = 4$ levels, defining the following models:

- **High-fidelity solver**: The high-fidelity model employs a finite element discretization on a structured triangular mesh $\mathcal{T}_{\text{HF}}$ consisting of 100 elements per spatial direction, corresponding to 101 nodes along each axis. Linear finite elements are used. For each parameter instance $\boldsymbol{\theta}$, it reconstructs the transmissivity field (see Eq. (21)) and computes the pressure head h at the d sensor locations $\{\mathbf{x}_j\}_{j=1}^d$. We have $\mathbf{f}_{\text{HF}} : \mathbb{R}^m \to \mathbb{R}^d$. The PDE solution requires solving a linear system with 101^2 degrees of freedom. For the solution, we use the GMRES solver with an incomplete LU preconditioner. Fig. 4 illustrates an example of the numerical solution and the corresponding transmissivity field. The high-fidelity solver generates reference data for the offline training. In addition, when solving the inverse problem, synthetic observations are produced by sampling random parameter instances $\boldsymbol{\theta}$, solving the high-fidelity model, and perturbing the resulting pressure fields with additive Gaussian noise, as detailed in Section 3.4.
- **Low-fidelity solvers**: Four low-fidelity models are obtained by uniform mesh coarsening, yielding meshes $\mathcal{T}_l$, $l = 1, \dots, 4$. The corresponding spatial resolutions and degrees of freedom are reported in Table 1. As shown in Fig. 5, the representation of the

Table 1
Spatial discretization, degrees of freedom (DoFs), computational time per forward evaluation, and predictive accuracy (RMSE) of the high-fidelity, low-fidelity, and multi-fidelity surrogate models.

Model	Mesh Size [#elements]	DoFs	Time / eval [s]	RMSE
f_{HF}	100×100	10,201	1.27×10^{-1}	0
$f_{\text{LF}}^{(1)}$	5×5	36	2.53×10^{-3}	5.1×10^{-2}
$f_{\text{LF}}^{(2)}$	10×10	121	3.22×10^{-3}	1.8×10^{-2}
$f_{\text{LF}}^{(3)}$	25×25	676	4.11×10^{-3}	4.4×10^{-3}
$f_{\text{LF}}^{(4)}$	50×50	2,601	1.09×10^{-2}	6.7×10^{-4}
$f_{\text{MF}}^{(1)}$	—	—	4.0×10^{-4}	7.0×10^{-3}
$f_{\text{MF}}^{(2)}$	—	—	5.6×10^{-4}	4.9×10^{-3}
$f_{\text{MF}}^{(3)}$	—	—	6.6×10^{-4}	6.7×10^{-4}
$f_{\text{MF}}^{(4)}$	—	—	7.8×10^{-4}	1.9×10^{-4}

transmissivity field becomes progressively smoother as the mesh is refined. All low-fidelity models employ linear finite elements and GMRES for the solution of the resulting linear systems. Each solver $\mathbf{f}_{\text{LF}}^{(l)} : \mathbb{R}^m \to \mathbb{R}^d$ maps the parameter vector $\boldsymbol{\theta}$ to the pressure head evaluated at the 25 sensor locations. The outputs of $\mathbf{f}_{\text{LF}}^{(l)}$ are used as inputs to the multi-fidelity NN at level l, and, when the end of the sub-chain is reached, are stored for use at subsequent finer levels.

- **Multi-fidelity NN:** At each level l, a multi-fidelity NN $\mathbf{f}_{\text{MF}}^{(l)} : \mathbb{R}^m \times [\mathbb{R}^d]^l \to \mathbb{R}^d$ is trained to approximate the high-fidelity pressure head solution, given the parameter vector $\boldsymbol{\theta}$ and the outputs of all low-fidelity solvers up to level l, according to (14). Each multi-fidelity surrogate is then used during inference to compute an approximate likelihood $\pi_{\text{MF}}^{(l)}$. Additional details for the training are provided in the next section.

All finite element simulations are performed using the FEniCS library[79], based on code adapted from Lykkegaard et al. [1] (available at https://ore.exeter.ac.uk/repository/handle/10871/125704). In the following, the neural network surrogates are implemented in `TensorFlow` [80], and the multi level MCMC are implemented using the `TinyDA` library.

3.3. MFDA: offline training and models accuracy

During the offline phase, N_{train} parameter samples $\{\boldsymbol{\theta}_j\}_{j=1}^{N_{\text{train}}}$ are drawn from the prior, and all finite element solvers are evaluated at these instances. Each multi-fidelity network $\mathbf{f}_{\text{MF}}^{(l)}$, $l = 1, \dots, 4$, is trained to minimize the mean squared error with respect to the high-fidelity output:

$$\mathcal{L}(\mathbf{W}_l, \mathbf{b}_l) = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left\| \mathbf{f}_{\text{HF}}(\boldsymbol{\theta}_j) - \mathbf{f}_{\text{MF}}^{(l)}\left(\boldsymbol{\theta}_j, \mathbf{f}_{\text{LF}}^{(1)}(\boldsymbol{\theta}_j), \dots, \mathbf{f}_{\text{LF}}^{(l)}(\boldsymbol{\theta}_j); \mathbf{W}_l, \mathbf{b}_l\right) \right\|^2, \quad l = 1, \dots, 4. \tag{22}$$

where $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weights and biases of the multi-fidelity NN at level l, respectively. The Adam optimizer is used for minimization. Network architecture details are provided in Appendix B.

We assess the accuracy of the multi-fidelity models incrementally, from level $l = 1$ to the level $l = 4$. For each level, we evaluate whether the corresponding multi-fidelity NN enhances the predictive accuracy of the associated low-fidelity solver, and whether incorporating information from coarser levels contributes positively to the prediction performance.

In Fig. 6a, the multi-fidelity surrogate $\mathbf{f}_{\text{MF}}^{(1)}$, which receives as input $\boldsymbol{\theta}$ and evaluations from the low-fidelity solver $\mathbf{f}_{\text{LF}}^{(1)}$, achieves a RMSE nearly one order of magnitude lower than that of the solver, for $N_{\text{train}} = 64{,}000$. By comparison, a standard (or single-fidelity) NN $\mathbf{f}_{\text{SF}}$ trained using only $\boldsymbol{\theta}$, without low-fidelity inputs, yields a RMSE almost three times higher, confirming the advantage of incorporating low-fidelity information.

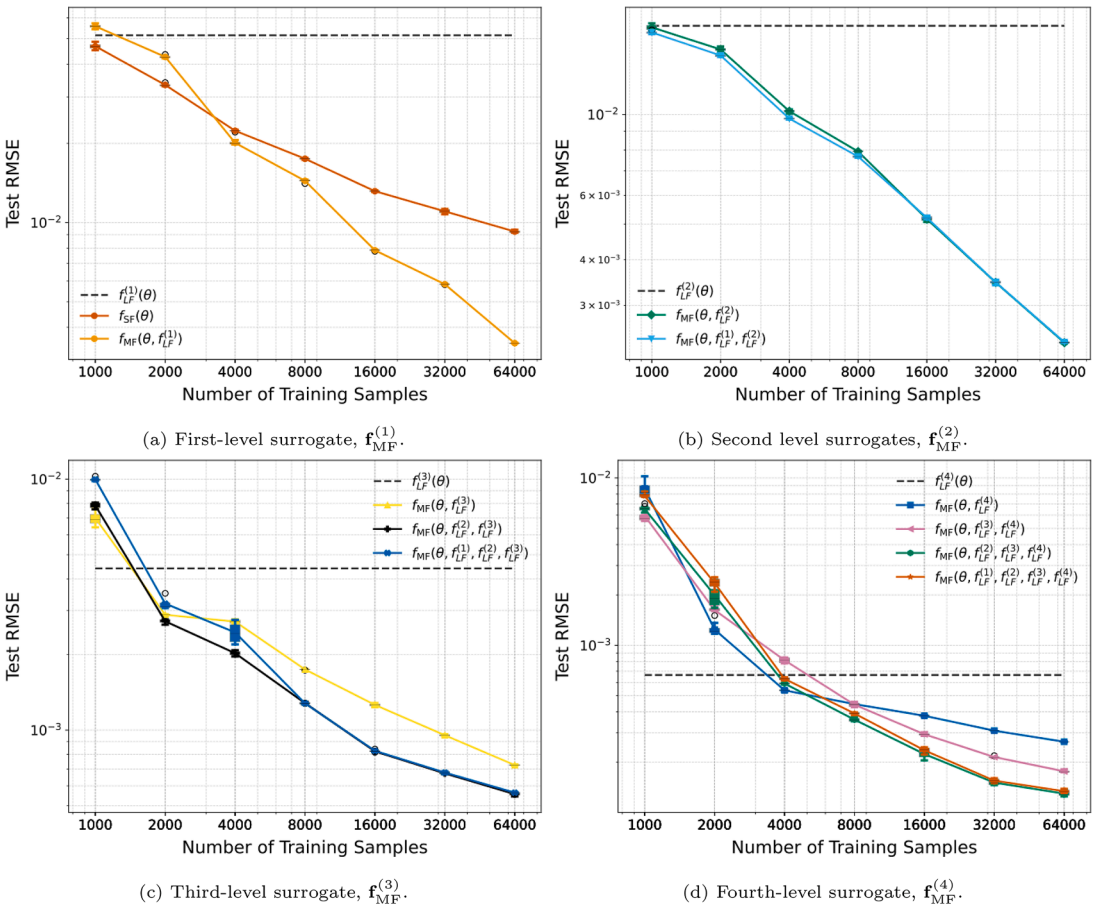
Fig. 6b reports the results for the second-level surrogate $\mathbf{f}_{\text{MF}}^{(2)}$, which takes as input both $\mathbf{f}_{\text{LF}}^{(1)}$ and $\mathbf{f}_{\text{LF}}^{(2)}$. Also in this case, the multi-fidelity network surpasses the predictive accuracy of $\mathbf{f}_{\text{LF}}^{(2)}$. Notably, a version using only $\mathbf{f}_{\text{LF}}^{(2)}$ as input provides nearly equivalent performance, suggesting no benefits when $\mathbf{f}_{\text{LF}}^{(1)}$ is added.

The third-level surrogate $\mathbf{f}_{\text{MF}}^{(3)}$ (Fig. 6c), which combines the outputs from $\mathbf{f}_{\text{LF}}^{(1)}$, $\mathbf{f}_{\text{LF}}^{(2)}$, and $\mathbf{f}_{\text{LF}}^{(3)}$, again shows improved accuracy over $\mathbf{f}_{\text{LF}}^{(3)}$. In contrast to the second level, here the inclusion of $\mathbf{f}_{\text{LF}}^{(2)}$ proves beneficial, showing the usefulness of multi-fidelity fusion.

Finally, Fig. 6d illustrates the performance of the fourth-level surrogate $\mathbf{f}_{\text{MF}}^{(4)}$, which aggregates all available low-fidelity solver evaluations. The resulting model achieves a RMSE approximately four times lower than that of the best-performing low-fidelity solver at $N_{\text{train}} = 64{,}000$. The figure also compares variants using only subsets of the low-fidelity inputs, confirming that including coarser solvers as inputs systematically enhances accuracy, with the only exclusion of model $\mathbf{f}_{\text{LF}}^{(1)}$. This however might be reasonable, given the extremely coarse mesh the surrogate $\mathbf{f}_{\text{LF}}^{(1)}$ has been built on.

Overall, the results consistently show that multi-fidelity NNs outperform the corresponding low-fidelity solvers at all levels. The integration of information from multiple fidelities contributes significantly to predictive performance.

We retain 16 000 samples because it is sufficient to bring the error of the coarsest (and cheapest) level to the order of the observation noise, which is the accuracy regime required for efficient filtering in MLDA. Following the modelling-error aware likelihood function



(a) First-level surrogate, $\mathbf{f}_{\mathrm{MF}}^{(1)}$.

(b) Second level surrogates, $\mathbf{f}_{\mathrm{MF}}^{(2)}$.

(c) Third-level surrogate, $\mathbf{f}_{\mathrm{MF}}^{(3)}$.

(d) Fourth-level surrogate, $\mathbf{f}_{\mathrm{MF}}^{(4)}$.

Fig. 6. Accuracy of the multi-fidelity surrogate models for different levels and training sample sizes, varying the amount of input coarse solver information.

discussed by Meles et al. [32], one may account for the residual surrogate discrepancy by augmenting the likelihood with an additional error term, which may reduce the amount of data needed for enabling long sub-chain at the coarsest levels.

A comprehensive summary of the spatial discretization, accuracy, and computational cost of the high-fidelity, low-fidelity, and multi-fidelity surrogate models is provided in Table 1.

3.4. MFDA: online inference

Synthetic observations at the sensor locations $\{\mathbf{x}_j\}_{j=1}^{d}$ are generated by evaluating the high-fidelity model and adding independent Gaussian noise,

$$\mathbf{y}^{\mathrm{obs}} = \mathbf{f}_{\mathrm{HF}}(\boldsymbol{\theta}) + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, 0.01^2 I_d).$$

The aim is to characterize the posterior distribution $\pi(\boldsymbol{\theta} \mid \mathbf{y}^{\mathrm{obs}})$ and to compare the sampling efficiency of MFDA with that of MH and MLDA.

In MFDA, the high-fidelity model is evaluated only during the offline stage to produce the training dataset. During inference, the likelihood is evaluated using the multi-fidelity surrogates $\mathbf{f}_{\mathrm{MF}}^{(l)}$, which combine $\boldsymbol{\theta}$ with low-fidelity forward evaluations. In contrast, MLDA evaluates a hierarchy of low- and high-fidelity solvers online, while MH relies solely on the high-fidelity model.

The sampling schemes, the forward models used at every level and the corresponding sub-chain lengths are summarized in Table 2. Longer sub-chains are assigned to coarser levels so that most evaluations occur at inexpensive models. Five independent MCMC chains are run, and convergence is monitored every 100 iterations using the Gelman–Rubin statistic $\hat{R}$ [81]. Sampling is terminated once the maximum $\hat{R}$ across all components of $\boldsymbol{\theta}$ falls below 1.01.

Fig. 7 shows the posterior mean reconstructions of the transmissivity field obtained with MH, MLDA, and MFDA, together with the ground truth, for one random instance. All schemes recover the main spatial features with comparable accuracy. To assess robustness, posterior sampling is repeated for ten independent realizations of $\boldsymbol{\theta}$ and corresponding synthetic observations. The distribution of the resulting RMSE values is reported in Fig. 8, confirming that the reconstruction quality is similar across the three methods. In particular,

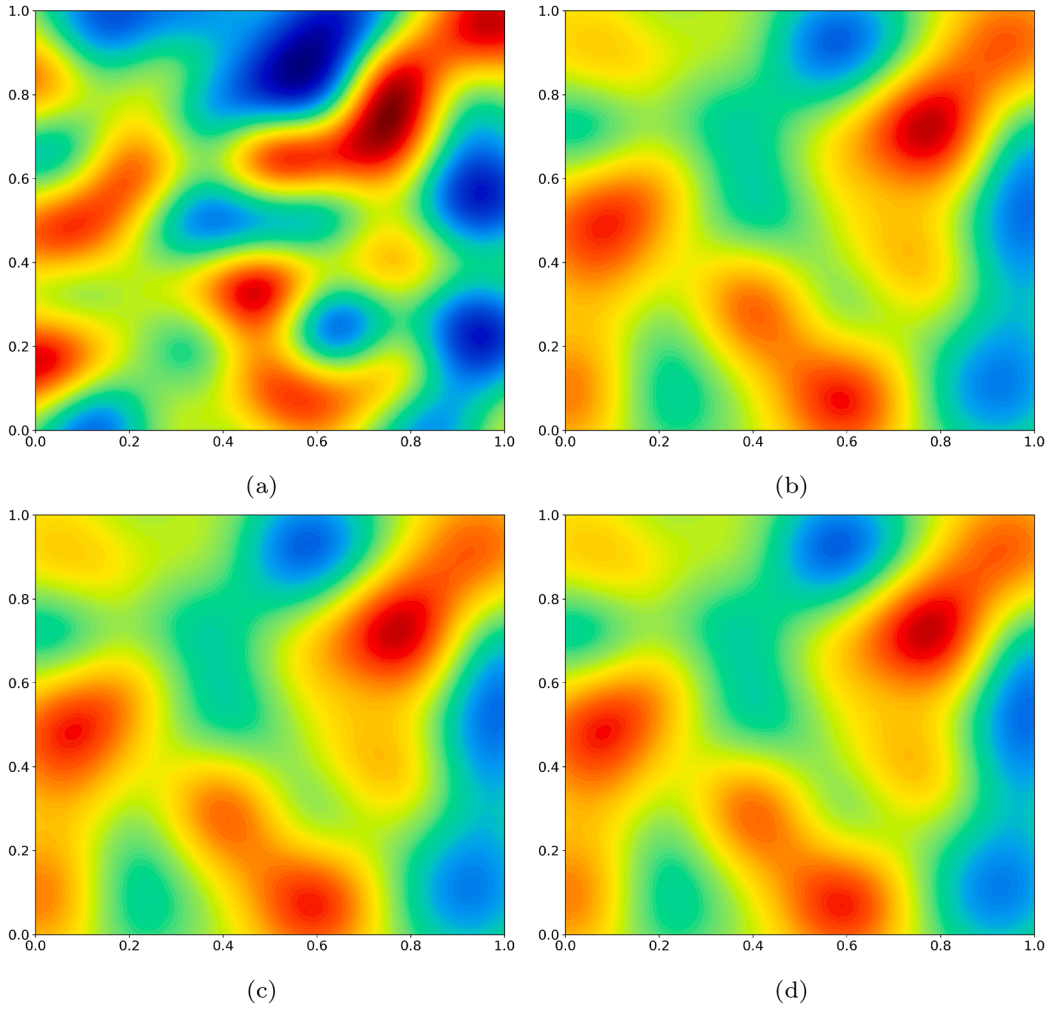


Fig. 7. (a) Exact transmissivity field θ to be identified. (b–d) Posterior mean transmissivity fields reconstructed using (b) MH, (c) MLDA, and (d) MFDA, respectively.

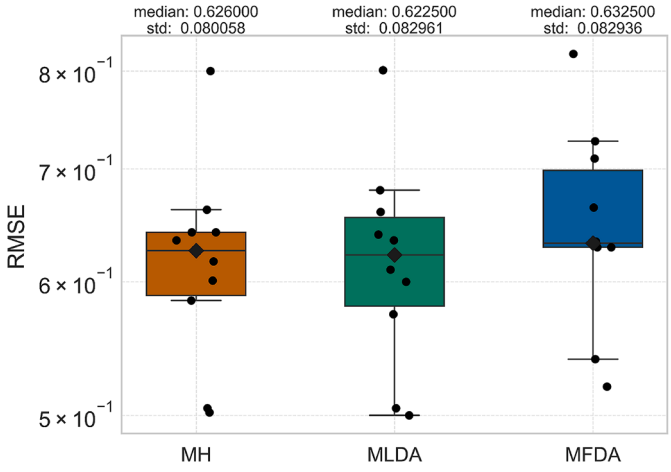


Fig. 8. Root mean square error (RMSE) of reconstructed parameters across 10 instances for all sampling schemes.

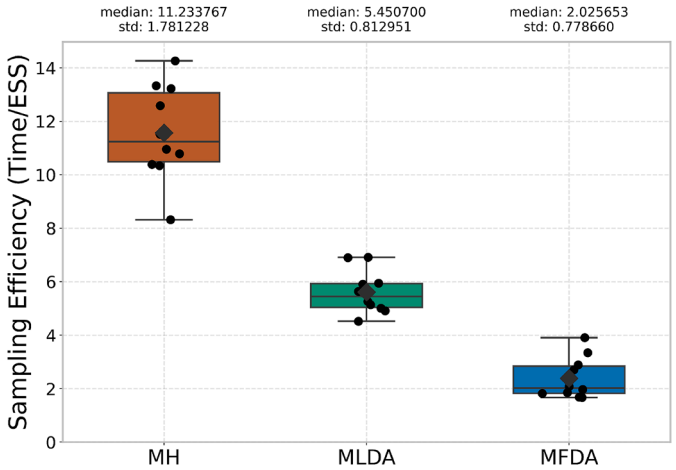


Fig. 9. Sampling efficiency (computation time per ESS) across 10 instances for all sampling schemes.

Table 2
Likelihood evaluation strategies for each scheme and level during the online phase. MFDA entries specify the surrogate and its required inputs.

Scheme	Level	Forward model	Inputs	Sub-chain Length
MH	only 1 level	$\mathbf{f}_{\text{HF}}$	$\boldsymbol{\theta}$	-
MLDA	1	$\mathbf{f}_{\text{LF}}^{(1)}$	$\boldsymbol{\theta}$	5
	2	$\mathbf{f}_{\text{LF}}^{(2)}$	$\boldsymbol{\theta}$	2
	3	$\mathbf{f}_{\text{LF}}^{(3)}$	$\boldsymbol{\theta}$	2
	4	$\mathbf{f}_{\text{LF}}^{(4)}$	$\boldsymbol{\theta}$	1
	5	$\mathbf{f}_{\text{HF}}$	$\boldsymbol{\theta}$	-
MFDA	1	$\mathbf{f}_{\text{MF}}^{(1)}$	$\boldsymbol{\theta}, \mathbf{f}_{\text{LF}}^{(1)}$	10
	2	$\mathbf{f}_{\text{MF}}^{(2)}$	$\boldsymbol{\theta}, \mathbf{f}_{\text{LF}}^{(1)}, \mathbf{f}_{\text{LF}}^{(2)}$	2
	3	$\mathbf{f}_{\text{MF}}^{(3)}$	$\boldsymbol{\theta}, \mathbf{f}_{\text{LF}}^{(1)}, \mathbf{f}_{\text{LF}}^{(2)}, \mathbf{f}_{\text{LF}}^{(3)}$	1
	4	$\mathbf{f}_{\text{MF}}^{(4)}$	$\boldsymbol{\theta}, \mathbf{f}_{\text{LF}}^{(1)}, \mathbf{f}_{\text{LF}}^{(2)}, \mathbf{f}_{\text{LF}}^{(3)}, \mathbf{f}_{\text{LF}}^{(4)}$	-

Table 3
Computational cost comparison among sampling schemes. MFDA incurs a one-time offline cost due to dataset generation and surrogate training, but achieves a substantially lower online and total inference cost. Online cost values report mean ± standard deviation over 10 independent inverse problems (i.e., 10 realizations of $\boldsymbol{\theta}$ and $\mathbf{y}^{\text{obs}}$).

Scheme	Data Gen. [s]	Training [s]	Online Cost [s]	Total Cost [s]
MH	0	0	12078 ± 1600	12078 ± 1600
MLDA	0	0	8794 ± 2044	8794 ± 2044
MFDA	2367	1655	2722 ± 928	6744 ± 928

MH and MLDA provide almost identical accuracy, while MFDA provide a median RMSE 1% larger, which is fully acceptable given the computational gains. Sampling efficiency, measured as computation time per ESS, is shown in Fig. 9. The MFDA scheme achieves the best performance, reducing the cost per effective sample by about a factor of five relative to MH and by about a factor of 3 relative to MLDA.

To quantify the overall computational gains, Table 3 reports the offline and online wall-clock costs associated with each scheme. The mean online wall-clock time per inference is reduced by about 75% compared to MH and by about 70% compared to MLDA. Although MFDA requires a one-time offline effort for data generation and surrogate training, the overall cost per inference remains roughly 50% lower than MH and about 30% lower than MLDA, even when this offline stage is included. Moreover, the offline cost is incurred only once: in scenarios requiring repeated inference, MFDA reuses the trained surrogates, and its effective cost reduces to the online stage alone.

In summary, the MFDA approach attains posterior reconstruction accuracy comparable to that of MLDA and standard MH, while delivering substantially greater computational efficiency in Bayesian inference for the considered groundwater flow model.

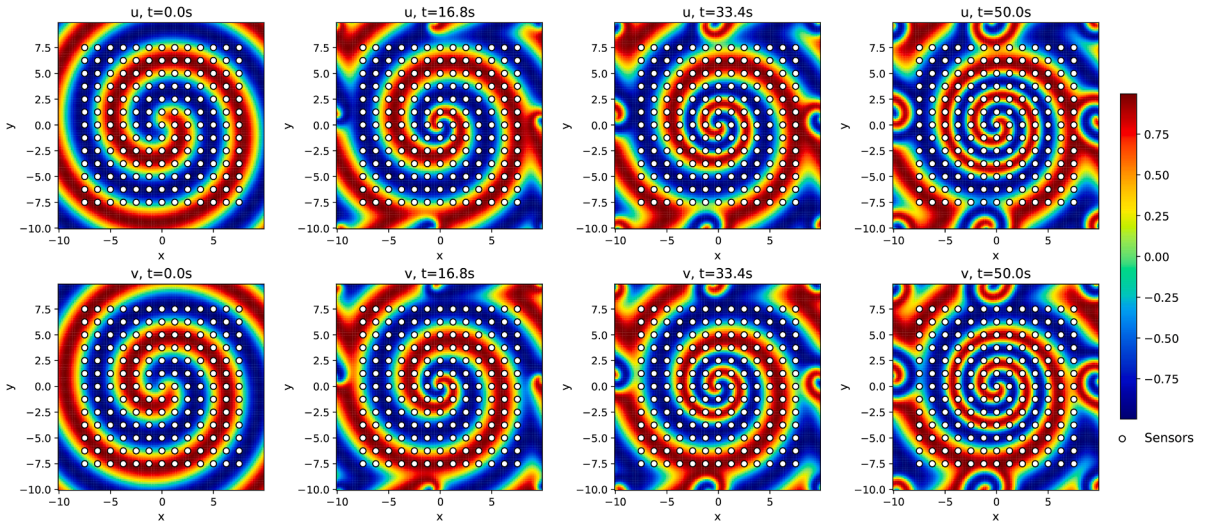


Fig. 10. Reference solution for the diffusion reaction problem obtained using high fidelity model at four distinct time instants along with the observation sensor locations.

4. Numerical experiments II: reaction–diffusion equation

To further assess the performance of the MFDA algorithm, we consider a nonlinear, time-dependent reaction-diffusion system as a second benchmark [7,82]. This case study is characterized by higher-dimensional output, offering a challenging test for data-driven models and posterior inference. The governing equations are:

$$\begin{aligned} \dot{u} &= \big(1-(u^2+v^2)\big)u + \mu_1(u^2+v^2)v + \mu_2\big(u_{xx}+u_{yy}\big), \\ \dot{v} &= -\mu_1(u^2+v^2)u + \big(1-(u^2+v^2)\big)v + \mu_2\big(v_{xx}+v_{yy}\big), \end{aligned} \tag{23}$$

where $u(x,y,t)$ and $v(x,y,t)$ represent two interacting species. The parameters of interest are the nonlinear reaction coefficient $\mu_1 \in (0.5, 1.5)$ and the diffusion coefficient $\mu_2 \in (0.01, 0.1)$, therefore $\boldsymbol{\mu} = [\mu_1, \mu_2] \in \mathcal{M} = (0.5, 1.5) \times (0.01, 0.1) \subseteq \mathbb{R}^2$. The domain is $(x, y) \in (-L, L)^2$ with $L = 20$ and periodic boundary conditions. The system is integrated in time over $t \in [0, 50]$ with initial conditions:

$$u(x,y,0) = v(x,y,0) = \tanh\left(\sqrt{x^2+y^2}\cos\left((x+iy) - \sqrt{x^2+y^2}\right)\right).$$

Observations are collected at 13×13 spatial sensor locations for both u and v. We set $N_s = 13 \times 13 \times 2$. The measurements are taken every 0.2 seconds, resulting in $T = 250$ time steps and a total observation vector of size $d = N_s \times T = 13 \times 13 \times 2 \times 250$.

4.1. MFDA: setting

For this test case, we design a MFDA scheme with $L = 3$ levels. Analogously to before, we define:

- **High-fidelity solver:** The high-fidelity forward model solves problem (23) numerically using a pseudo-spectral method on a 128×128 spatial grid, with a time step $\Delta t = 0.2$. We denote by $\mathbf{g}_{\mathrm{HF}}(\boldsymbol{\mu})$ the corresponding full-field solution (for both u and v) on this grid. The dominant computational cost comes from the two-dimensional Fast Fourier Transform [83], with per-step complexity $\mathcal{O}(N \log N)$, where $N = 128^2$. Given $T = 50/\Delta t = 250$ time steps, the total complexity of the solver is $\mathcal{O}(T \cdot N \log N)$. As before, this high-fidelity solver is used to generate reference data for training the NNs. Evaluating $\mathbf{g}_{\mathrm{HF}}(\boldsymbol{\mu})$ at the $N_s = 13 \times 13 \times 2$ grid nodes corresponding to the sensor locations yields the sensor-level output $\mathbf{f}_{\mathrm{HF}}(\boldsymbol{\mu}) \in \mathbb{R}^{N_s \times T}$. In addition, it is used to generate the synthetic observations employed in the inverse problem, as outlined in Section 4.3. The reference solution obtained using the high-fidelity model for a random instance of the parameters $\boldsymbol{\mu}$ is shown in Fig. 10 along with the sensor configuration at four distinct time instances.
- **Low-fidelity solvers:** A hierarchy of three low-fidelity forward models is constructed by uniformly coarsening both the spatial and temporal discretisations of the high-fidelity solver. Each low-fidelity solver computes a full-field solution $\mathbf{g}^{(l)}_{\mathrm{LF}}(\boldsymbol{\mu})$ on a coarser spectral grid and with a larger time step, $l = 1, 2, 3$. All solvers employ the same pseudo-spectral scheme; only the mesh resolution and time-step size vary across levels. The spatial resolutions, number of time steps, forward evaluation times, and associated errors relative to the high-fidelity solution are summarised in Table 4. Restricting $\mathbf{g}^{(l)}_{\mathrm{LF}}(\boldsymbol{\mu})$ to the sensor locations yields the corresponding sensor-level outputs $\mathbf{f}^{(l)}_{\mathrm{LF}}(\boldsymbol{\mu}) \in \mathbb{R}^{N_s \times T}$.
- **Dimensionality reduction:** To make NN training tractable and mitigate overfitting, we perform Proper Orthogonal Decomposition (POD) on the full-field high-fidelity solutions. Define the snapshot matrix

$$\mathbf{S} = \left[\mathbf{g}_{\mathrm{HF}}(\boldsymbol{\mu}^{(1)}), \ldots, \mathbf{g}_{\mathrm{HF}}(\boldsymbol{\mu}^{(20)})\right] \in \mathbb{R}^{N \times (T \times 20)}, \tag{24}$$

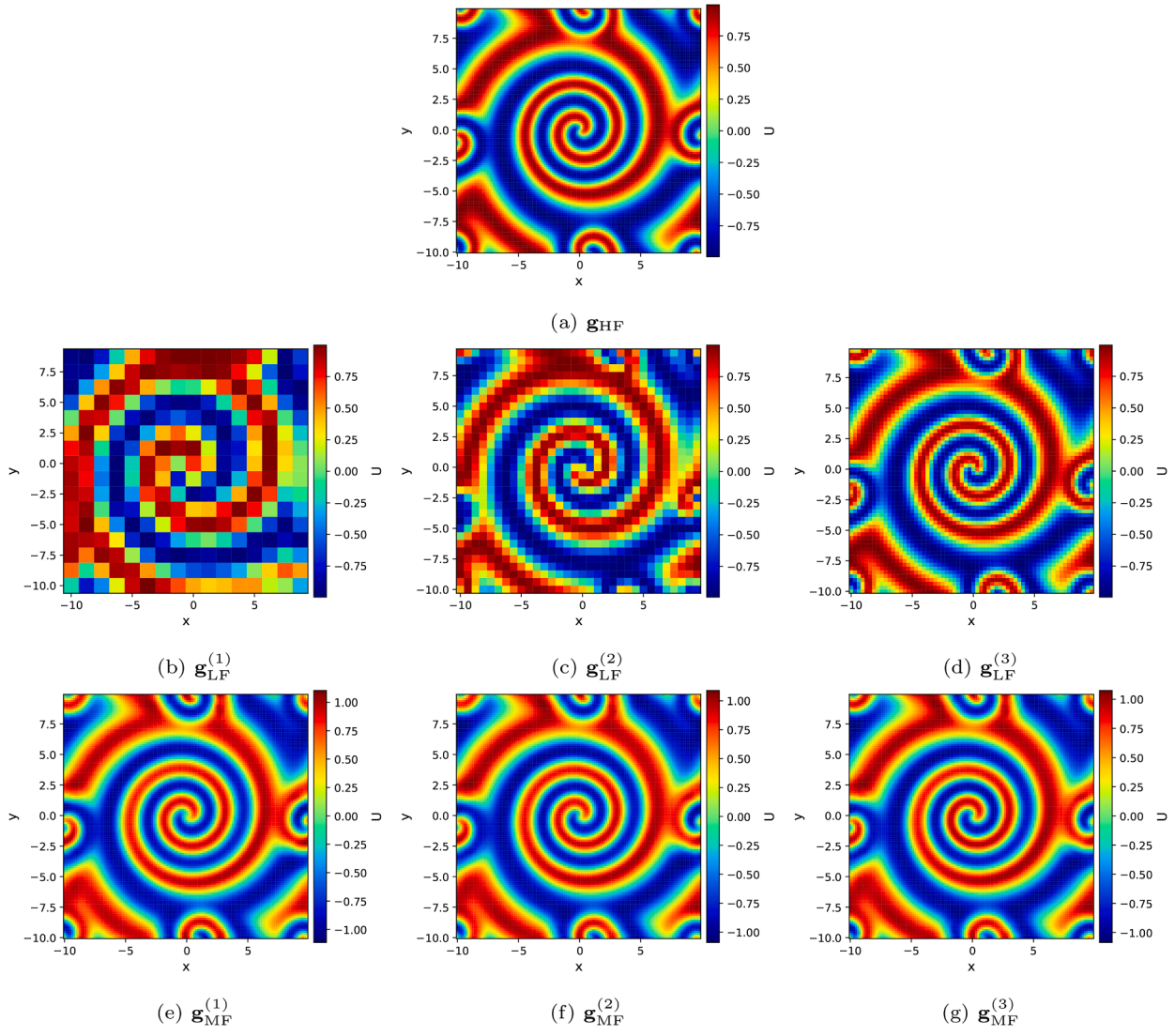


Fig. 11. Final-time full field numerical solutions for u. (a) HF reference; (b–d) LF solver outputs at increasing spatial resolution; (e–g) MF surrogate predictions corresponding to the LF levels.

where each column corresponds to a vectorised full-field solution at a given time instance and N is the number of spatial degrees of freedom of the high-fidelity discretisation. We compute the singular value decomposition

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\top}, \tag{25}$$

and select the smallest r such that

$$\frac{\sum_{i=1}^{r}\sigma_i^2}{\sum_i \sigma_i^2} \geq 0.95, \tag{26}$$

where σ_i denote the singular values. The resulting POD basis on the full grid is

$$\boldsymbol{\Phi} = \mathbf{U}_{[:,1:r]} \in \mathbb{R}^{N\times r}, \qquad r \ll N. \tag{27}$$

In this case, the above criterion yields $r = 25$ retained modes.

Since the low-fidelity solvers are defined on coarser spatio-temporal grids, their full-field outputs are first mapped to the high-fidelity spatio-temporal resolution using an interpolation operator

$$\mathcal{I} : \mathbb{R}^{N_l\times T_l} \longrightarrow \mathbb{R}^{N\times T}, \tag{28}$$

where N_l is the number of spatial degrees of freedom of the corresponding low-fidelity discretisation. The operator $\mathcal{I}$ applies linear interpolation in space and cubic spline resampling in time . For a given parameter vector $\boldsymbol{\mu}$ and fidelity level l, the reduced-order

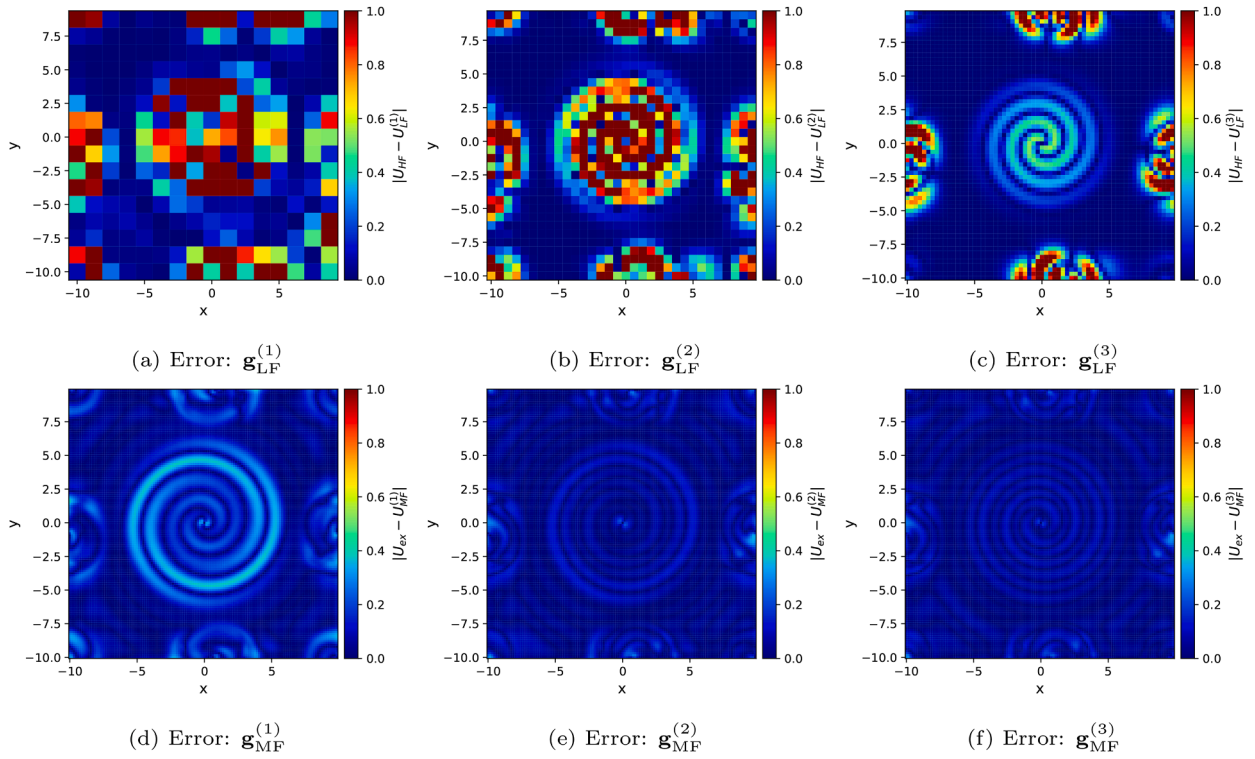


Fig. 12. Absolute error fields at final time for the u quantity. (a–c) LF solver errors; (d–f) MF surrogate errors for the corresponding LF levels. Identical color limits are used within each row for direct comparison.

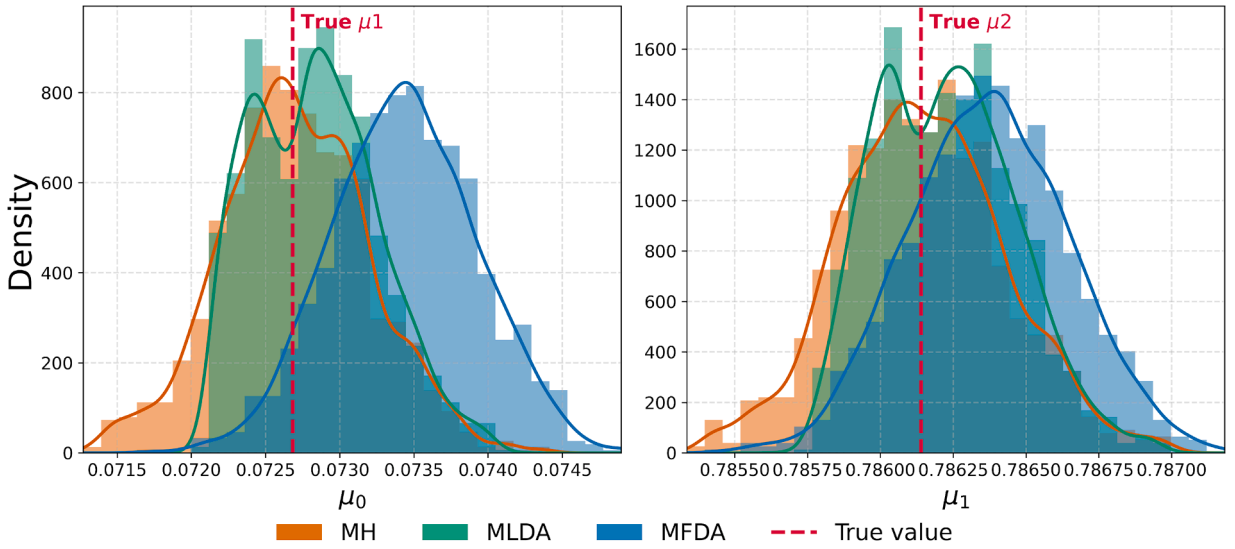


Fig. 13. Posterior distributions for the reaction–diffusion parameters using MH, MLDA, and MFDA.

representation (POD coefficients) of the low-fidelity model is then defined as

$$\mathbf{z}_{\mathrm{LF}}^{(l)}(\boldsymbol{\mu}) = \boldsymbol{\Phi}^{\top}\, \mathcal{I}\left(\mathbf{g}_{\mathrm{LF}}^{(l)}(\boldsymbol{\mu})\right) \in \mathbb{R}^{r\times T}. \tag{29}$$

An analogous projection applied to the high-fidelity full-field solution yields the corresponding high-fidelity POD coefficients $\mathbf{z}_{\mathrm{HF}}(\boldsymbol{\mu})$. In this way, all models share a consistent reduced representation in the POD coefficient space.

- **Multi-fidelity NN:** At each level $l = 1, 2, 3$, a multi-fidelity NN $\mathbf{f}_{\mathrm{MF}}^{(l)}$ is trained to predict the high-fidelity POD coefficients using the parameter vector $\boldsymbol{\mu}$ and the reduced outputs from all low-fidelity solvers up to level l. The multi-fidelity NN acts in the reduced

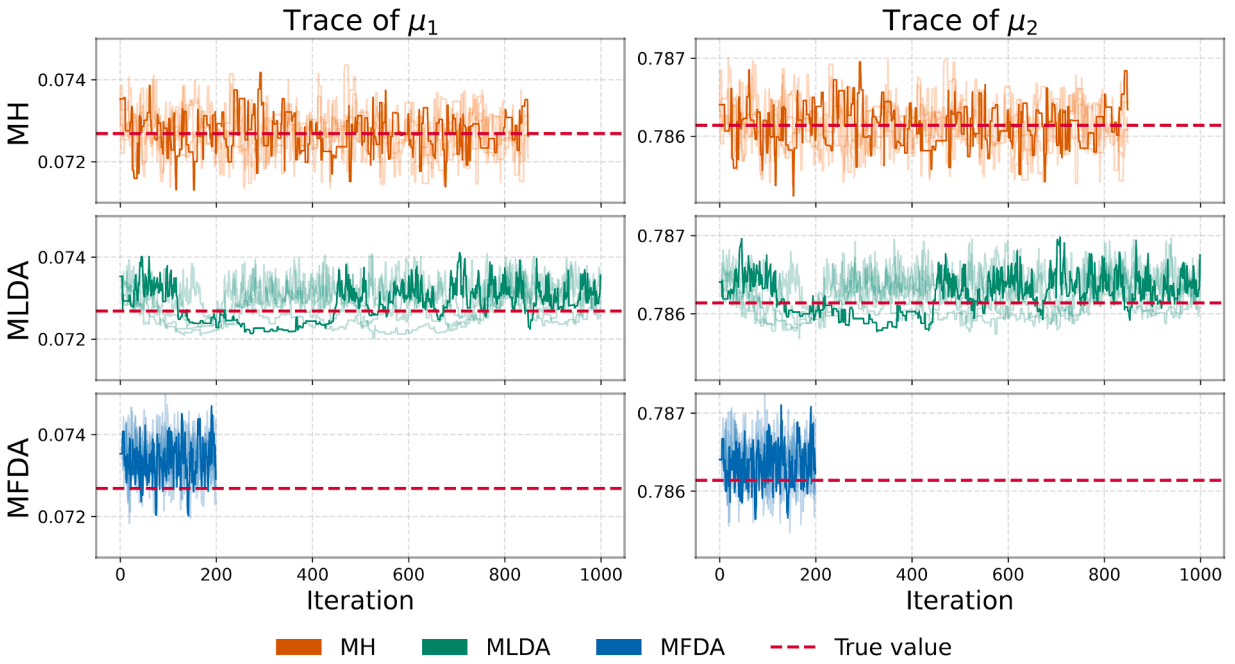


Fig. 14. Trace plots for the reaction–diffusion posterior distributions of the parameters obtained using MH, MLDA, and MFDA.

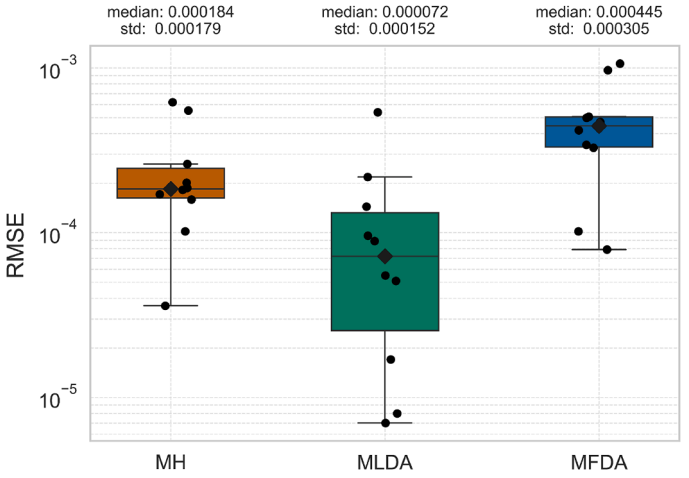


Fig. 15. Root mean square error (RMSE) of reconstructed parameters across 10 test instances for all methods.

space and has the form

$$\hat{\mathbf{g}}_{\mathrm{MF}}^{(l)} : \mathbb{R}^2 \times \left[\mathbb{R}^{r\times T}\right]^l \to \mathbb{R}^{r\times T}, \tag{30}$$

where the output is the time sequence of POD coefficients associated with the high-fidelity solution. The corresponding reconstructed full-field prediction is obtained by

$$\mathbf{g}_{\mathrm{MF}}^{(l)}(\boldsymbol{\mu}) = \boldsymbol{\Phi}\,\hat{\mathbf{g}}_{\mathrm{MF}}^{(l)}\left(\boldsymbol{\mu}, \left\{\mathbf{z}_{\mathrm{LF}}^{(j)}(\boldsymbol{\mu})\right\}_{j=1}^{l}\right) \in \mathbb{R}^{N\times T}. \tag{31}$$

Evaluating $\mathbf{g}_{\mathrm{MF}}^{(l)}(\boldsymbol{\mu})$ at the sensor grid nodes yields the sensor-level surrogate output

$$\mathbf{f}_{\mathrm{MF}}^{(l)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_s\times T}, \tag{32}$$

which is the quantity used in the likelihood evaluation. This architecture efficiently leverages low-fidelity evaluations to approximate the high-fidelity response, combining model reduction with hierarchical information fusion. It can be regarded as a multi-fidelity extension of the POD–NN framework [84]. The architecture details are provided in Appendix C.

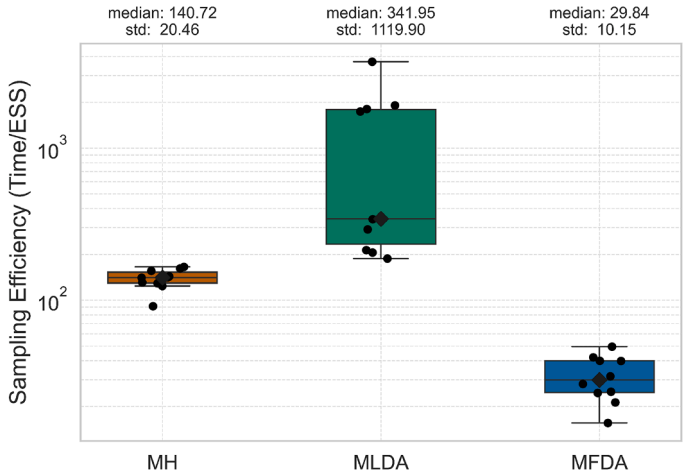


Fig. 16. Sampling efficiency: computation time per ESS across 10 test instances for all methods.

Table 4
Computation time and RMSE for high-fidelity, low-fidelity, and multi-fidelity surrogate models.

Model	Mesh Size [#elements]	Time Steps	Computational Time [s]	RMSE
f_{HF}	128×128	250	19.34	0
$f_{\text{LF}}^{(1)}$	16×16	50	0.407	1.74×10^{-1}
$f_{\text{LF}}^{(2)}$	32×32	100	0.809	8.72×10^{-2}
$f_{\text{LF}}^{(3)}$	64×64	250	3.293	1.34×10^{-2}
$f_{\text{MF}}^{(1)}$	–	–	0.0588	4.0×10^{-2}
$f_{\text{MF}}^{(2)}$	–	–	0.5180	3.9×10^{-2}
$f_{\text{MF}}^{(3)}$	–	–	1.556	2.8×10^{-2}

4.2. MFDA: offline training and models accuracy

We sample $N_{train} = 500$ parameter instances, denoted by $\{\boldsymbol{\mu}_i\}_{i=1}^{N_{train}}$, using Latin Hypercube Sampling within the admissible parameter domain μ. For each sampled instance, all the high-fidelity and low-fidelity models are evaluated. A dimensionality reduction is applied using $r = 25$ modes. The three multi-fidelity neural networks are then trained using the Adam optimizer by minimizing the MSE:

$$\mathcal{L}(\mathbf{W}_l, \mathbf{b}_l) = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \left\| \mathbf{f}_{\text{HF}}(\boldsymbol{\mu}_j) - \hat{\mathbf{f}}_{\text{MF}}^{(l)}\left(\boldsymbol{\mu}_j, \boldsymbol{\Phi}^\top \mathcal{I}(\mathbf{f}_{\text{LF}}^{(1)}(\boldsymbol{\mu}_j)), \ldots, \boldsymbol{\Phi}^\top \mathcal{I}(\mathbf{f}_{\text{LF}}^{(l)}(\boldsymbol{\mu}_j)); \mathbf{W}_l, \mathbf{b}_l\right) \right\|^2, \quad l = 1, 2, 3. \tag{33}$$

An additional set of 20 parameter instances is reserved for testing. Table 4 reports the computational cost and predictive accuracy, measured in terms of RMSE in the testing set, for each solver and surrogate model. The results indicate that the multi-fidelity NNs consistently achieve significantly lower RMSE compared to any individual low-fidelity model, while maintaining comparable computational costs. The performance gain is particularly notable when using the coarsest solvers: for instance, the surrogate $f_{\text{MF}}^{(1)}$ reduces the RMSE by approximately one order of magnitude relative to its corresponding low-fidelity model $f_{\text{LF}}^{(1)}$.

Fig. 11 presents the predicted solutions for the low-fidelity solvers and the multi-fidelity neural networks at all three levels, evaluated at the final time step for the u quantity in a benchmark case. The corresponding errors are presented in Fig. 12. As shown, the multi-fidelity neural networks substantially enhance the reconstruction accuracy. The resulting error is significantly reduced and primarily concentrated in the central region of the domain, indicating the effectiveness of the surrogate models in capturing the underlying dynamics even at coarse resolutions. The source of the errors is mostly related to higher frequencies modes, not included in dimensionality reduction.

4.3. MFDA: online inference

Synthetic observations are generated by evaluating the high-fidelity solver at the 13×13 sensor locations and adding independent Gaussian noise,

$$\mathbf{y}^{\text{obs}} = \mathbf{f}_{\text{HF}}(\boldsymbol{\mu}) + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_d), \tag{34}$$

where $\sigma_\varepsilon = 0.2$, measurements are collected every 0.2 time units. The sensor configuration is shown in Fig. 10. The aim is to characterize the posterior distribution $\pi(\boldsymbol{\mu} \mid \mathbf{y}^{\text{obs}})$ and compare the sampling efficiency of MFDA with that of MH and MLDA.

Table 5
Likelihood evaluation strategies during online inference for the reaction–diffusion test case. MFDA uses POD-based surrogate evaluations with reduced-model inputs, see Eq. (29) for definition of $\mathbf{z}_{LF}^{(\cdot)}$; MLDA sequentially filters proposals across low- to high-fidelity solvers; MH directly evaluates the high-fidelity model.

Scheme	Level	Forward Model	Inputs	Subchain Length
MFDA	1	$\mathbf{f}_{\mathrm{MF}}^{(1)}$	$\boldsymbol{\mu}, \mathbf{z}_{\mathrm{LF}}^{(1)}$	5
	2	$\mathbf{f}_{\mathrm{MF}}^{(2)}$	$\boldsymbol{\mu}, \mathbf{z}_{\mathrm{LF}}^{(1)}, \mathbf{z}_{\mathrm{LF}}^{(2)}$	5
	3	$\mathbf{f}_{\mathrm{MF}}^{(3)}$	$\boldsymbol{\mu}, \mathbf{z}_{\mathrm{LF}}^{(1)}, \mathbf{z}_{\mathrm{LF}}^{(2)}, \mathbf{z}_{\mathrm{LF}}^{(3)}$	–
MLDA	1	$\mathbf{f}_{\mathrm{LF}}^{(1)}$	$\boldsymbol{\mu}$	5
	2	$\mathbf{f}_{\mathrm{LF}}^{(2)}$	$\boldsymbol{\mu}$	5
	3	$\mathbf{f}_{\mathrm{LF}}^{(3)}$	$\boldsymbol{\mu}$	1
	4	$\mathbf{f}_{\mathrm{HF}}$	$\boldsymbol{\mu}$	–
MH	1	$\mathbf{f}_{\mathrm{HF}}$	$\boldsymbol{\mu}$	–

In MFDA, the high-fidelity model is used only offline to both construct the POD basis and train the multi-fidelity surrogates. During inference, the likelihood is evaluated exclusively using $\mathbf{f}_{\mathrm{MF}}^{(l)}$, which combines $\boldsymbol{\mu}$ with reduced low-fidelity model outputs. In contrast, MLDA evaluates a hierarchy of low- and high-fidelity solvers online, while MH relies solely on the high-fidelity solver.

The sampling schemes, together with the forward models and sub-chain lengths at each level, are summarized in Table 5. Sub-chain lengths are chosen so that the number of coarse evaluations per fine-level proposal in MLDA matches that of MFDA. Analogously to the previous test case, five independent MCMC chains are run, and convergence is monitored every 100 iterations using the Gelman–Rubin diagnostic; sampling is terminated once $\widehat{R} < 1.01$ across all components of $\boldsymbol{\mu}$. For all cases, the chains are initialized by solving a least-squares problem for the observations using the highest-level model and setting the resulting estimate as the initial parameter sample. The proposal distribution is a Gaussian random walk with covariance given by a Gauss-Newton approximation of the least-squares Hessian at this point and adaptive step size (adapted every fine level iteration).

Computed posterior distributions for a given reference value of the parameter vector are shown in Fig. 13. The corresponding trace plots are reported in Fig. 14. The MH and MLDA schemes yield posterior distributions that are broadly consistent with the reference parameters. However, MLDA distribution is corrupted by spurious narrow peaks, which we attribute to the discrepancy between fidelity levels: long stretches of rejections lead to highly correlated segments, as also visible in the trace plots. This let the MLDA posterior to be slightly misaligned with the reference MH distribution, especially in the left tail for both parameters. Thanks to the least-squares initialization the samples remain close to the true parameters. The MFDA scheme shows the best mixing behaviour. Its posterior is slightly shifted with respect to MH/MLDA, which is consistent with the residual approximation error introduced by removing the high-fidelity model from the online hierarchy. To avoid relying solely on parameter RMSE (which assumes practical identifiability), we also report a posterior predictive check in Appendix C.1, where replicated observations generated from posterior samples are compared against the measurements via predictive credible bands and a coverage diagnostic.

The procedure is repeated over ten independent realizations of the parameter vector $\boldsymbol{\mu}$. Fig. 15 reports the distribution of the RMSE between the posterior means and the corresponding true parameters. All methods recover the unknown parameters with small errors. The MH reference yields consistently low RMSE, indicating that the considered setup is practically identifiable at the adopted noise level. MLDA exhibits the smallest RMSE and variability; however, this behaviour is influenced by the least-squares initialization and by reduced mobility of the chain due to inter-level discrepancies, which can lead to frequent rejections and strong autocorrelation. MFDA shows a modest increase in RMSE, consistent with the residual approximation error introduced by removing the high-fidelity model from the online hierarchy. Nonetheless, the estimation accuracy of MFDA remains acceptable for this inverse UQ task, with a typical RMSE around 4×10^{-4}. The higher error relative to MH/MLDA is likely dominated by the approximation introduced by the reduced-order representation rather than by sampling variability.

Finally, the sampling efficiency of the schemes is investigated in Fig. 16. The MFDA scheme demonstrates the highest efficiency, achieving approximately a fourfold speedup relative to standard MH. Conversely, MLDA provides no computational gain in this setting and results in the poorest performance, over an order of magnitude less efficient than MFDA. This loss of efficiency can be attributed to the limited accuracy of the coarsest-level model.

To quantify the computational gains, Table 6 reports the offline and online wall-clock costs associated with each sampling scheme. The offline cost includes the high-fidelity data generation used to construct the POD basis and the surrogate training phase (MFDA only). The online cost corresponds to the total wall-clock time required to complete one inference run, averaged over the five MCMC chains. The MFDA scheme achieves a substantial reduction in online cost relative to standard MH, and is significantly more efficient than MLDA. When the offline stage is included, the total cost of MFDA remains lower than both MH and MLDA. As in the previous test case, the offline cost is incurred only once: in settings where inference must be repeated for multiple observational datasets, the effective cost per inference for MFDA reduces to its online cost alone.

Overall, these results indicate that MFDA yields posterior estimates that are statistically consistent with those obtained by the high-fidelity and MLDA samplers, exhibiting only a negligible loss in accuracy. At the same time, MFDA achieves a substantial gain in sampling efficiency.

Table 6
Computational cost comparison among sampling schemes for the reaction–diffusion test case. MFDA incurs a one-time offline cost for POD construction and surrogate training, but provides a substantially lower online and total inference cost. Online cost values correspond to mean ± standard deviation over the independent MCMC chains.

Scheme	Data Gen. [s]	Training [s]	Online Cost [s]	Total Cost [s]
MH	0	0	91181 ± 12115	91181 ± 12115
MLDA	0	0	62591 ± 12066	62591 ± 12066
MFDA	14615	13200	9220 ± 2753	38035 ± 2753

5. Conclusions

In this work we have presented a novel MFDA framework for Bayesian inverse problems governed by partial differential equations. The proposed approach integrates the hierarchical sampling strategy of MLDA with neural network–based multi-fidelity fusion, enabling the combination of information from multiple low-fidelity solvers to construct enhanced surrogates that approximate high-fidelity model evaluations with high accuracy and low cost. In doing so, the MFDA framework retains the sampling accuracy of high-fidelity models while reducing the computational cost traditionally associated with Markov chain Monte Carlo (MCMC)–based inference in large-scale PDE settings.

From a methodological perspective, the MFDA framework offers three key advances. First, it introduces multi-fidelity neural networks capable of incorporating outputs from multiple low-fidelity models simultaneously, thereby overcoming the limitations of traditional auto-regressive or pairwise multi-fidelity regression schemes. Second, it confines the use of computationally expensive high-fidelity solvers to an offline training phase, making the online sampling stage entirely reliant on low fidelity solvers and neural network corrections. Third, it strongly improves the accuracy of the coarsest levels, allowing to rely on the cheapest models for long sub-chains without diverging too much from fine level posterior distributions.

The performance of MFDA has been demonstrated on two benchmark problems: the reconstruction of transmissivity fields in a groundwater flow model, and the parameter inference for a nonlinear, time-dependent reaction–diffusion system. Across both case studies, MFDA achieved posterior estimates statistically comparable from those obtained using high-fidelity MH and MLDA samplers, while substantially improving sampling efficiency. In the groundwater flow example, MFDA reduced the computation time per effective sample by factors of approximately five relative to MH and 3 relative to MLDA. In the reaction–diffusion case, MFDA delivered similar gains w.r.t. to MH and much higher w.r.t. to MLDA, despite the added complexity of nonlinear dynamics and high-dimensional spatio-temporal observations. The results also confirmed that the neural network surrogates consistently improved the predictive accuracy of all low-fidelity solvers, with the benefits being most pronounced at coarser resolutions.

The proposed methodology is general and can be applied to a broad class of Bayesian inverse problems involving expensive forward models, provided that lower-fidelity approximations are available. The modular design allows flexibility in the choice of surrogate architectures and subchain configurations, making it extendable to problem-specific constraints and computational budgets. Overall, the MFDA framework provides an effective and scalable tool for accelerating Bayesian inference in PDE-constrained inverse problems, achieving a favorable balance between computational efficiency and posterior accuracy.

Future research directions include the development of adaptive online training strategies to update neural network surrogates during sampling, potentially reducing offline training costs. Additionally, investigating MFDA to settings where low-fidelity models differ in physics or dimensionality, rather than solely in numerical resolution, could further highlight its applicability.

Code availability

The source code implementing the MFDA framework and supporting the numerical experiments of this work is available at https://github.com/filippozacchei/MFDA.

CRediT authorship contribution statement

Filippo Zacchei: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Paolo Conti:** Writing – review & editing, Visualization, Supervision, Methodology; **Attilio Frangi:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization; **Andrea Manzoni:** Writing – review & editing, Visualization, Supervision, Methodology, Funding acquisition, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

FZ acknowledges the support of the JRC STEAM STM-Politecnico di Milano agreement. PC and AM acknowledges the PRIN 2022 Project “Numerical approximation of uncertainty quantification problems for PDEs by multi-fidelity methods (UQ-FLY)” (No. 202222PACR), funded by the European Union - NextGenerationEU.

AM acknowledges the project “Dipartimento di Eccellenza” 2023–2027 funded by MUR, the project FAIR (Future Artificial Intelligence Research), funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence), and the Project “Reduced Order Modeling and Deep Learning for the real-time approximation of PDEs (DREAM)” (Starting Grant No. FIS00003154), funded by the Italian Science Fund (FIS) - Ministero dell’Università e della Ricerca.

AF acknowledges the PRIN 2022 Project “DIMIN- DIgital twins of nonlinear MIcrostructures with iNnovative model-order-reduction strategies” (No. 2022XATLT2) funded by the European Union - NextGenerationEU.

The authors acknowledge Luca Caroselli for useful discussions and preliminary testing of the methodology proposed in this paper.

Appendix A. Derivation of Eq. (9)

We highlight here the derivation of the acceptance rate of Eq. (9), following and summarizing the key results of [18]. In the *original* delayed-acceptance Metropolis-Hastings [43], the proposal $q(\cdot \mid \theta)$ appears explicitly only in the first stage acceptance rate. Suppose we want to sample from a distribution π and we have access to an approximation $\pi_{\mathrm{LF}}^{(1)}$, given the current state θ, draw $\psi \sim q(\cdot \mid \theta)$ and define the stage-1 acceptance

$$\alpha_1(\psi,\theta) = \min\left\{1, \frac{q(\theta \mid \psi)}{q(\psi \mid \theta)} \frac{\pi_{\mathrm{LF}}^{(1)}(\psi)}{\pi_{\mathrm{LF}}^{(1)}(\theta)}\right\}, \tag{A.1}$$

which induces the effective proposal

$$q^*(\psi \mid \theta) = \alpha_1(\psi,\theta)\, q(\psi \mid \theta) + (1 - r(\theta))\, \delta_\theta(\psi), \qquad r(\theta) = \int \alpha_1(\psi,\theta)\, q(\psi \mid \theta)\, d\psi, \tag{A.2}$$

and the stage-2 acceptance

$$\alpha_2(\psi,\theta) = \min\left\{1, \frac{q^*(\theta \mid \psi)}{q^*(\psi \mid \theta)} \frac{\pi(\psi)}{\pi(\theta)}\right\}. \tag{A.3}$$

In particular, [18] (Lemma 1 and Eq. 2.10) shows that when the coarse transition kernel q^* is in detailed balance with $\pi_{\mathrm{LF}}^{(1)}$, which is the case when using Metropolis-Hastings kernel, and the forward map is not state dependent, as in our case, the proposal terms cancel in the promotion step, yielding the compact form:

$$\alpha_2(\psi,\theta) = \min\left\{1, \frac{\pi(\psi)\, \pi_{\mathrm{LF}}^{(1)}(\theta)}{\pi_{\mathrm{LF}}^{(1)}(\theta)\, \pi(\psi)}\right\}. \tag{A.4}$$

When we are targeting a posterior, since the prior is the same for both levels they cancel each other and only likelihood appears in the ratio, and the proposal is present only at level 1.

Since the composition of transition kernel keeps the same property (Lemma 2, [18]) we can extend this to multiple coarse steps. Thus, by induction, introducing multiple approximations $\pi_{\mathrm{LF}}^{(\ell)}, \ell = 1, \dots, L$, we have for the multi-level case (Theorem 5, [18]):

$$\alpha_1(\psi,\theta) = \min\left\{1, \frac{\pi_{\mathrm{LF}}^{(1)}(\psi) q(\psi \mid \theta)}{\pi_{\mathrm{LF}}^{(1)}(\theta) q(\theta \mid \psi)}\right\}, \tag{A.5}$$

$$\alpha_\ell(\psi,\theta) = \min\left\{1, \frac{\pi_{\mathrm{LF}}^{(\ell)}(\psi)\, \pi_{\mathrm{LF}}^{(\ell-1)}(\theta)}{\pi_{\mathrm{LF}}^{(\ell)}(\theta)\, \pi_{\mathrm{LF}}^{(\ell-1)}(\psi)}\right\}, \quad \ell = 2, \dots, L \tag{A.6}$$

$$\alpha_{L+1}(\psi,\theta) = \min\left\{1, \frac{\pi(\psi)\, \pi_{\mathrm{LF}}^{(L)}(\theta)}{\pi(\theta)\, \pi_{\mathrm{LF}}^{(L)}(\psi)}\right\}. \tag{A.7}$$

Appendix B. Neural Network Architecture Selection for the Groundwater Flow Test Case

A comparative analysis of neural network architectures was performed for the test cases in Section 3, using 16 000 samples and focusing on $\mathbf{f}_{\mathrm{MF}}^{(3)}$. Because of the multi-input, multi-fidelity setting, the search was restricted to architectures with one branch per input source: each branch processes a single input (e.g. equation parameters or low-fidelity data), the resulting latent representations are concatenated in a fusion layer, and a final fully connected output block maps the concatenated representation to the output.

Table B.7
Schematic architecture of the multi-fidelity NN $\mathbf{f}_{\mathrm{MF}}^{(3)}$ for the groundwater flow test case. The fusion layer concatenates the outputs of the input branches, which are then processed by the output block to produce the final prediction.

Component	Layer type	Neurons	Activation
Parameter branch (θ)	Input	64	–
	Dense	128	GeLU
	Dense	128	GeLU
	Dense	128	GeLU
	Dense	128	GeLU
	Dense	128	Linear
Low-fidelity branch 1 ($\mathbf{f}_{\mathrm{LF}}^{(1)}$)	Input	25	–
	Dense	128	Linear
Low-fidelity branch 2 ($\mathbf{f}_{\mathrm{LF}}^{(2)}$)	Input	25	–
	Dense	128	Linear
Low-fidelity branch 3 ($\mathbf{f}_{\mathrm{LF}}^{(3)}$)	Input	25	–
	Dense	128	Linear
Fusion layer	Concatenation	512	–
Output block	Dense	128	GeLU
	Dense	128	GeLU
	Dense	25	Linear

Architectures within this family were compared using Bayesian optimisation [85] as implemented in Python package *Optuna* [86]. Each candidate model was trained for a fixed duration of one minute, favouring compact and computationally efficient networks. The architectures for $\mathbf{f}_{\mathrm{MF}}^{(1)}$ and $\mathbf{f}_{\mathrm{MF}}^{(2)}$ were obtained by reusing the same design as for $\mathbf{f}_{\mathrm{MF}}^{(3)}$, while removing the unused low-fidelity input branches.

The hyperparameter search varied the number of hidden layers in each input branch and in the shared/output branch between 0 and 6. The special case with 0 hidden layers in all input branches corresponds to a standard fully connected network acting on the concatenation of all inputs, without any branch structure. The number of neurons per layer and the activation function were explored over $\{32, 64, 128, 256\}$ and $\{\mathrm{GeLU}, \tanh, \mathrm{ReLU}, \mathrm{sigmoid}\}$, respectively, and a regularisation coefficient was also tuned.

The final architecture for $\mathbf{f}_{\mathrm{MF}}^{(3)}$ employs a high-fidelity branch (Input 1) with four fully connected layers of 128 neurons, and three additional low-fidelity branches (Inputs 2–4), each mapped to 128 neurons. The resulting latent representations are concatenated and passed through a shared processing block with two fully connected layers and a 25-neuron linear output layer.

The complete configuration of this architecture is reported in Table B.7. The networks used for $\mathbf{f}_{\mathrm{MF}}^{(1)}$ and $\mathbf{f}_{\mathrm{MF}}^{(2)}$ are obtained from the same structure by retaining only the branches associated with their available inputs.

We emphasize that the resulting architecture is not universal: for different training-set sizes, an optimal bias–variance trade-off may require adjusting the network capacity and/or regularisation. In this work, the architecture was selected at 16 000 samples and, for the learning-curve study, we re-tuned the regularisation coefficient for each dataset size while keeping the architecture fixed. For unseen problems, we recommend a short validation-driven search that starts from the suggested multi-branch models and gradually increases depth/width, monitoring validation error. When data are limited, smaller networks combined with stronger regularisation (early stopping, weight decay, dropout) are preferable; when more data are available, increasing capacity can improve accuracy with reduced overfitting risk.

Appendix C. Neural Network Architecture for the Reaction–Diffusion Test Case

For the results in Section 4, a hand-tuning phase was performed, building on the insights from the groundwater flow case. The goal was not to identify a single globally optimal architecture, but to show that a limited, targeted tuning effort already leads to substantial accuracy gains for the low-fidelity models without excessive offline cost.

All architectures process sequential inputs with a length equal to the number of time steps. All the branches are concatenated and further processed by a fully connected layer, followed by two stacked LSTM layers and two additional fully connected layers. The network concludes with a fully connected output layer.

The detailed architecture for $\mathbf{f}_{\mathrm{MF}}^{(3)}$ (using all inputs) is reported in Table C.8. The models for $\mathbf{f}_{\mathrm{MF}}^{(1)}$ and $\mathbf{f}_{\mathrm{MF}}^{(2)}$ share the same design, restricted to the branches corresponding to their available inputs.

C.1. Posterior predictive validation

To complement the parameter-based error metric reported in Fig. 8, we include a posterior predictive validation. Given the approximate posterior samples $\{\boldsymbol{\theta}^{(s)}\}_{s=1}^{S}$, we generate replicated observations

$$\boldsymbol{y}_{\mathrm{rep}}^{(s)} = \mathbf{f}(\boldsymbol{\theta}^{(s)}) + \boldsymbol{\varepsilon}^{(s)}, \qquad \boldsymbol{\varepsilon}^{(s)} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\epsilon}\right), \tag{C.1}$$

Table C.8
Schematic architecture of the multi-fidelity NN $\mathbf{f}_{\mathrm{MF}}^{(3)}$ for the reaction–diffusion test case. The fusion layer concatenates the outputs of the input branches, which are then processed by the output block to produce the final prediction.

Component	Layer type	Neurons	Activation
Parameter branch ($\boldsymbol{\mu}$)	Input	3	–
	Dense	64	GeLU
	Dense	64	GeLU
	Dense	64	GeLU
	Dense	64	Linear
Low-fidelity branch 1 ($\mathbf{z}_{\mathrm{LF}}^{(1)}$)	Input	25	–
	Dense	64	Linear
Low-fidelity branch 2 ($\mathbf{z}_{\mathrm{LF}}^{(2)}$)	Input	25	–
	Dense	64	Linear
Low-fidelity branch 3 ($\mathbf{z}_{\mathrm{LF}}^{(3)}$)	Input	25	–
	Dense	64	Linear
Fusion layer	Concatenation	256	–
Output block	Dense	64	GeLU
	LSTM	64	Tanh
	LSTM	64	Tanh
	Dense	64	GeLU
	Dense	25	Linear

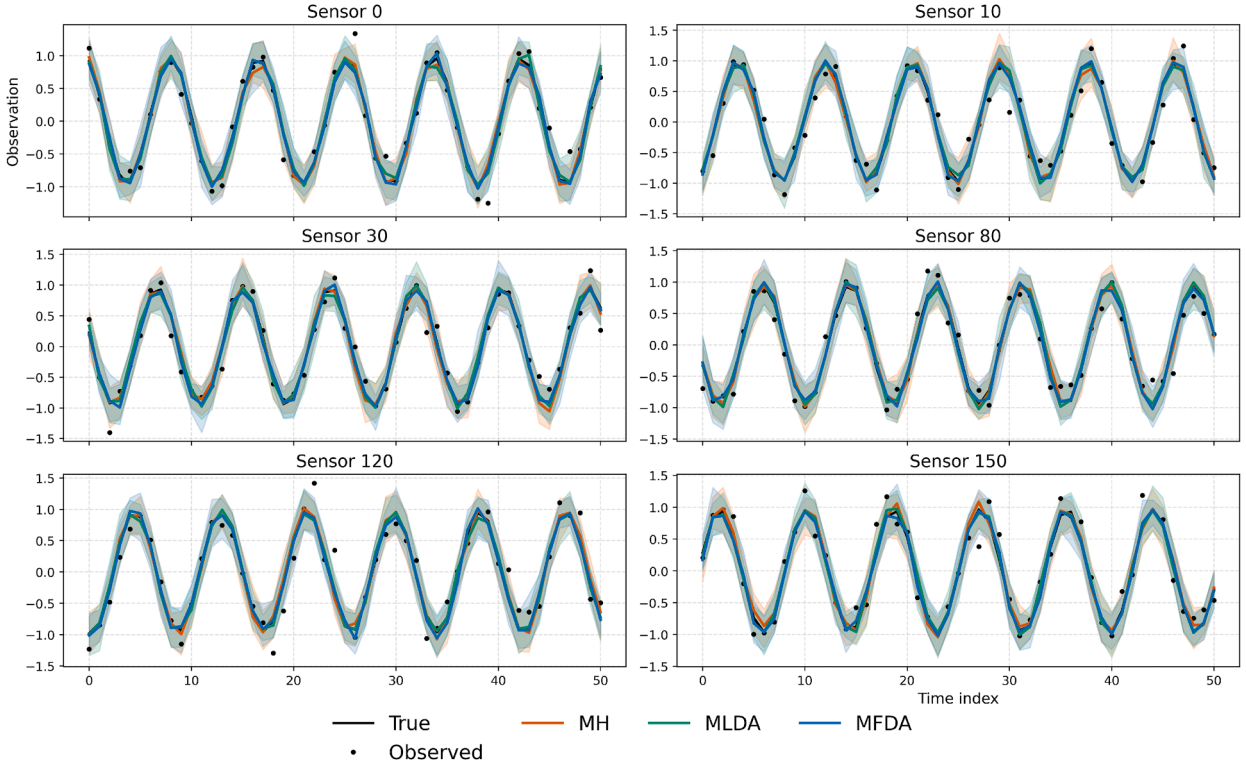


Fig. C.17. Pointwise predictive credible bands (5–95% quantiles) of replicated observations at 6 of the 169 sensor locations, compared with the observed data.

and compare the distribution of $\boldsymbol{y}_{\mathrm{rep}}^{(s)}$ with the observed data $\boldsymbol{y}^{\mathrm{obs}}$. Fig. C.17 reports pointwise predictive credible bands (5–95% quantiles) for the replicated observations at 6 of the 169 sensor locations. Overall, the posterior predictive distributions are consistent with the measurements and the assumed aleatoric uncertainty for all three schemes.

References

[1] M.B. Lykkegaard, T.J. Dodwell, D. Moxey, Accelerating uncertainty quantification of groundwater flow modelling using a deep neural network proxy, Comput.

Methods Appl. Mech. Eng. 383 (2021) 113895. https://doi.org/10.1016/j.cma.2021.113895
[2] T. Bui-Thanh, K. Wilcox, O. Ghattas, Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications, AIAA J. 46 (2008) 2520–2529. https://doi.org/10.2514/1.35850
[3] S. Sugimoto, Y. Takakura, H. Kajiro, J. Fujiki, H. Dashti, T. Yajima, Y. Kawajiri, Modeling, parameter estimation, and uncertainty quantification for CO2 adsorption process using flexible metal-organic frameworks by bayesian monte carlo methods, J. Adv. Manuf. Process. 5 (2023). https://doi.org/10.1002/amp2.10165
[4] L. Rosafalco, M. Torzoni, A. Manzoni, S. Mariani, A. Corigliano, Online structural health monitoring by model order reduction and deep learning algorithms, Comput. Struct. 255 (2021) 106604. https://doi.org/10.1016/j.compstruc.2021.106604
[5] O. Dürr, P.-Y. Fan, Z.-X. Yin, Bayesian calibration of MEMS accelerometers, IEEE Sens. J. 23 (12) (2023) 13319–13326. https://doi.org/10.1109/jsen.2023.3272907
[6] F. Zacchei, F. Rizzini, G. Gattere, A. Frangi, A. Manzoni, Neural networks based surrogate modeling for efficient uncertainty quantification and calibration of MEMS accelerometers, Int. J. Non Linear Mech. 167 (2024) 104902. https://doi.org/10.1016/j.ijnonlinmec.2024.104902
[7] P. Conti, M. Guo, A. Manzoni, A. Frangi, S.L. Brunton, J. Nathan Kutz, Multi-fidelity reduced-order surrogate modelling, Proc. R. Soc. A: Math., Phys. Eng. Sci. 480 (2283) (2024) 20230655. https://doi.org/10.1098/rspa.2023.0655
[8] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Rev. 60 (3) (2018) 550–591. https://doi.org/10.1137/16M1082469
[9] J.P. Kaipio, E. Somersalo, Statistical and Computational Inverse Problems, Applied Mathematical Sciences, Springer New York, NY, 1 edition, 2005. Mathematics and Statistics, Mathematics and Statistics (R0), https://doi.org/10.1007/b138659
[10] C.J. Geyer, Introduction to markov chain monte carlo, in: Handbook of Markov Chain Monte Carlo, Chapman and Hall/CRC, 2011. https://doi.org/10.1201/b10905-2
[11] A. Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, Society for Industrial and Applied Mathematics, 2005. https://doi.org/10.1137/1.9780898717921
[12] S.T. Tokdar, R.E. Kass, Importance sampling: a review, WIREs Comput. Stat. 2 (1) (2010) 54–60. https://doi.org/10.1002/wics.56
[13] A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, 2001.
[14] L. Rosafalco, P. Conti, A. Manzoni, S. Mariani, A. Frangi, EKF–SINDy: Empowering the extended kalman filter with sparse identification of nonlinear dynamics, Comput. Methods Appl. Mech. Eng. 431 (2024) 117264. https://doi.org/10.1016/j.cma.2024.117264
[15] D. Simon, Nonlinear kalman filtering, in: Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches, John Wiley & Sons, Ltd, 2006, pp. 393–431. Accessed via Wiley Online Library, https://doi.org/10.1002/0470045345.ch13
[16] R. van de Schoot, D. Kaplan, J. Denissen, J.B. Asendorpf, F.J. Neyer, M.A.G. van Aken, A gentle introduction to bayesian analysis: applications to developmental research, Child Dev. 85 (3) (2014) 842–860. https://doi.org/10.1111/cdev.12169
[17] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: a review for statisticians, J. Am. Stat. Assoc. 112 (518) (2017) 859–877. https://doi.org/10.1080/01621459.2017.1285773
[18] M.B. Lykkegaard, T.J. Dodwell, C. Fox, G. Mingas, R. Scheichl, Multilevel delayed acceptance MCMC, SIAM/ASA J. Uncertain. Quantific. 11 (1) (2023) 1–30. https://doi.org/10.1137/22M1476770
[19] S. Pagani, A. Manzoni, A. Quarteroni, Efficient state/parameter estimation in nonlinear unsteady PDEs by a reduced basis ensemble kalman filter, SIAM/ASA J. Uncertain. Quantific. 5 (1) (2017) 890–921.
[20] H. Zheng, W. Hongqiao, Z. Qingping, A MCMC method based on surrogate model and gaussian process parameterization for infinite bayesian PDE inversion, J. Comput. Phys. 507 (2024) 112970. https://doi.org/10.1016/j.jcp.2024.112970
[21] Y.M. Marzouk, H.N. Najm, L.A. Rahn, Stochastic spectral methods for efficient bayesian solution of inverse problems, J. Comput. Phys. 224 (2) (2007) 560–586. https://doi.org/10.1016/j.jcp.2006.10.010
[22] Y.M. Marzouk, H.N. Najm, Dimensionality reduction and polynomial chaos acceleration of bayesian inference in inverse problems, J. Comput. Phys. 228 (6) (2009) 1862–1902. https://doi.org/10.1016/j.jcp.2008.11.024
[23] J. Zhang, A.A. Taflanidis, Accelerating MCMC via kriging-based adaptive independent proposals and delayed rejection, Comput. Methods Appl. Mech. Eng. 355 (2019) 1124–1147. https://doi.org/10.1016/j.cma.2019.07.016
[24] T.J. Santner, B.J. Williams, W.I. Notz, B.J. Williams, The design and analysis of computer experiments, 1, Springer, 2003.
[25] T. Cui, Y.M. Marzouk, K.E. Willcox, Data-driven model reduction for the bayesian solution of inverse problems, Int. J. Numer. Methods Eng. 102 (5) (2015) 966–990. https://doi.org/10.1002/nme.4748
[26] M. Frangos, Y. Marzouk, K. Willcox, B. van Bloemen Waanders, Surrogate and Reduced-Order Modeling: A Comparison of Approaches for Large-Scale Statistical Inverse Problems, John Wiley & Sons, Ltd, 2010, pp. 123–149. https://doi.org/10.1002/9780470685853.ch7
[27] J.B. Nagel, B. Sudret, Spectral likelihood expansions for bayesian inference, J. Comput. Phys. 309 (2016) 267–294. https://doi.org/10.1016/j.jcp.2015.12.047
[28] P.-R. Wagner, S. Marelli, B. Sudret, Bayesian model inversion using stochastic spectral embedding, J. Comput. Phys. 436 (2021) 110141. https://doi.org/10.1016/j.jcp.2021.110141
[29] P.R. Conrad, Y.M. Marzouk, N.S. Pillai, A. Smith, Accelerating asymptotically exact MCMC for computationally intensive models via local approximations, J. Am. Stat. Assoc. 111 (516) (2016) 1591–1607. https://doi.org/10.1080/01621459.2015.1096787
[30] J. Li, Y.M. Marzouk, Adaptive construction of surrogates for the bayesian solution of inverse problems, SIAM J. Sci. Comput. 36 (3) (2014) A1163–A1186. https://doi.org/10.1137/130938189
[31] L. Yan, T. Zhou, Adaptive multi-fidelity polynomial chaos approach to bayesian inference in inverse problems, J. Comput. Phys. 381 (2019) 110–128. https://doi.org/10.1016/j.jcp.2018.12.025
[32] G.A. Meles, N. Linde, S. Marelli, Bayesian tomography with prior-knowledge-based parametrization and surrogate modelling, Geophys. J. Int. 231 (1) (2022) 673–691. https://doi.org/10.1093/gji/ggac214
[33] H. Wang, H. Wang, J. Ying, Q. Zhou, Sequential bayesian design for efficient surrogate construction in the inversion of darcy flows, J. Comput. Phys. 553 (2026) 114723. https://doi.org/10.1016/j.jcp.2026.114723
[34] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, Bayesian Data Analysis, Chapman and Hall/CRC, 3 edition, 2013. https://doi.org/10.1201/b16018
[35] T. Cui, J. Martin, Y.M. Marzouk, A. Solonen, A. Spantini, Likelihood-informed dimension reduction for nonlinear inverse problems, Inverse Probl. 30 (11) (2014) 114015. https://doi.org/10.1088/0266-5611/30/11/114015
[36] H. Haario, E. Saksman, J. Tamminen, An adaptive metropolis algorithm, Bernoulli 7 (2) (2001) 223 – 242.
[37] M. Girolami, B. Calderhead, Riemann manifold langevin and hamiltonian monte carlo methods, J. R. Stat. Soc. Ser. B: Stat. Methodol. 73 (2) (2011) 123–214.
[38] M. Betancourt, A Conceptual Introduction to Hamiltonian Monte Carlo, 2018, arXiv:1701.02434.
[39] M. Hoffman, A. Gelman, The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo, J. Mach. Learn. Res. 15 (2011).
[40] M.B. Giles, Multilevel monte carlo methods, Acta Numer. 24 (2015) 259–328.
[41] A. Beskos, A. Jasra, K. Law, R. Tempone, Y. Zhou, Multilevel sequential monte carlo samplers, Stoch. Process. Their Appl. 127 (5) (2017) 1417–1440.
[42] K.A. Cliffe, M.B. Giles, R. Scheichl, A.L. Teckentrup, Multilevel monte carlo methods and applications to elliptic PDEs with random coefficients, Comput. Vis. Sci. 14 (2011) 3–15.
[43] J.A. Christen, C. Fox, Markov chain monte carlo using an approximation, J. Comput. Graph. Stat. 14 (4) (2005) 795–810. https://doi.org/10.1198/106186005X76983
[44] T.J. Dodwell, C. Ketelsen, R. Scheichl, A.L. Teckentrup, A hierarchical multilevel markov chain monte carlo algorithm with applications to uncertainty quantification in subsurface flow, SIAM/ASA J. Uncertain. Quantific. 3 (1) (2015) 1075–1108. https://doi.org/10.1137/130915005
[45] P. Bratley, B.L. Fox, L.E. Schrage, A Guide to Simulation, Springer, New York, NY, 1987. https://doi.org/10.1007/978-1-4419-8724-2
[46] J.M. Hammersley, D.C. Handscomb, Monte Carlo Methods, Springer Netherlands, Dordrecht, 1964. https://doi.org/10.1007/978-94-009-5819-7

[47] B.L. Nelson, On control variate estimators, Comput. Oper. Res. 14 (3) (1987) 219–225. https://doi.org/10.1016/0305-0548(87)90024-4
[48] A.E. Annels, Geostatistical ore-reserve estimation, in: A.E. Annels (Ed.), Mineral Deposit Evaluation: a Practical Approach, Springer Netherlands, Dordrecht, 1991, pp. 175–245. https://doi.org/10.1007/978-94-011-9714-4_4
[49] D.E. Myers, Matrix formulation of co-kriging, J. Int. Assoc. Math. Geol. 14 (3) (1982) 249–257. https://doi.org/10.1007/BF01032887
[50] P. Perdikaris, D. Venturi, J.O. Royset, G.E. Karniadakis, Multi-fidelity modelling via recursive co-kriging and gaussian-markov random fields, Proc. R. Soc. A: Math., Phys. Eng. Sci. 471 (2179) (2015) 20150018. https://doi.org/10.1098/rspa.2015.0018
[51] X. Meng, G.E. Karniadakis, A composite neural network that learns from multi-fidelity data: application to function approximation and inverse PDE problems, J. Comput. Phys. 401 (2020) 109020. https://doi.org/10.1016/j.jcp.2019.109020
[52] X. Meng, H. Babaee, G.E. Karniadakis, Multi-fidelity bayesian neural networks: algorithms and applications, J. Comput. Phys. 438 (2021) 110361. https://doi.org/10.1016/j.jcp.2021.110361
[53] D. Liu, Y. Wang, Multi-fidelity physics-constrained neural network and its application in materials modeling, J. Mech. Des. 141 (121403) (2019). https://doi.org/10.1115/1.4044400
[54] M. Motamed, A multi-fidelity neural network surrogate sampling method for uncertianty quantification, Int. J. Uncertain. Quantif. 10 (4) (2020). https://doi.org/10.1615/Int.J.UncertaintyQuantification.2020031957
[55] M. Guo, A. Manzoni, M. Amendt, P. Conti, J.S. Hesthaven, Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities, Comput. Methods Appl. Mech. Eng. 389 (2022) 114378. https://doi.org/10.1016/j.cma.2021.114378
[56] A.N. Krouglova, H.R. Johnson, B. Confavreux, M. Deistler, P.J. Gonçalves, Multifidelity Simulation-based Inference for Computationally Expensive Simulators, 2025, arXiv:2502.08416v3.
[57] A. Kirsch, An Introduction to the Mathematical Theory of Inverse Problems, 120 of *Applied Mathematical Sciences*, Springer International Publishing, Cham, 2021. https://doi.org/10.1007/978-3-030-63343-1
[58] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer Texts in Statistics, Springer, New York, NY, 2 edition, 2004. Springer Science + Business Media New York. https://doi.org/10.1007/978-1-4757-4145-2
[59] W.K. Hastings, Monte carlo sampling methods using markov chains and their applications, Biometrika 57 (1) (1970) 97–109.
[60] A. Manzoni, S. Pagani, T. Lassila, Accurate solution of bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models, SIAM/ASA J. Uncertain. Quantific. 4 (1) (2016) 380–412.
[61] J.H. Lee, J. Kim, H. Lee, J. Park, A Delayed Acceptance Auxiliary Variable MCMC for Spatial Models with Intractable Likelihood Function, (2025). arXiv:2504.17147.
[62] L. Seelinger, A. Reinarz, M.B. Lykkegaard, R. Akers, A.M.A. Alghamdi, D. Aristoff, W. Bangerth, J. Bénézech, M. Diez, K. Frey, et al., Democratizing uncertainty quantification, J. Comput. Phys. 521 (2025) 113542.
[63] J.P. Madrigal-Cianci, F. Nobile, R. Tempone, Analysis of a class of multilevel markov chain monte carlo algorithms based on independent metropolis–hastings, SIAM/ASA J. Uncertain. Quantific. 11 (1) (2023) 91–138.
[64] P. Conti, M. Guo, A. Frangi, A. Manzoni, Progressive multi-fidelity learning for physical system predictions, (2025). arXiv:2510.13762.
[65] P. Perdikaris, M. Raissi, A. Damianou, N.D. Lawrence, G.E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, Proc. R. Soc. A: Math., Phys. Eng. Sci. 473 (2198) (2017) 20160751.
[66] O. Davis, M. Motamed, R. Tempone, Residual multi-fidelity neural network computing, BIT Numer. Math. 65 (2) (2025) 15.
[67] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015.
[68] R. Behrou, H. Mansourifar, Y. Zhou, S. Wang, P.D. Schmalenberg, C. Ling, E.M. Dede, Physics-informed multi-fidelity surrogate modeling of fluid flow in porous media, APL Mach. Learn. 3 (3) (2025) 036116. https://doi.org/10.1063/5.0279064
[69] C. Jeremy, A. Kian, N. Evan, R. Malhotra, Partial-physics-informed multi-fidelity modeling of manufacturing processes, J. Mater. Process. Technol. 320 (2023) 118125. https://doi.org/10.1016/j.jmatprotec.2023.118125
[70] M.R. Ebers, K.M. Steele, J.N. Kutz, Discrepancy modeling framework: learning missing physics, modeling systematic residuals, and disambiguating between deterministic and random effects, SIAM J. Appl. Dyn. Syst. 23 (1) (2024) 440–469. https://doi.org/10.1137/22M148375X
[71] M. Kim, J. Lim, S. Jee, D. Park, Multi-fidelity approach for transitional boundary layer, Int. J. Heat Fluid Flow 102 (2023) 109163. https://doi.org/10.1016/j.ijheatfluidflow.2023.109163
[72] J. Zhang, W. Li, L. Zeng, L. Wu, An adaptive gaussian process-based method for efficient bayesian experimental design in groundwater contaminant source identification problems, Water Resour. Res. 52 (8) (2016) 5971–5984. https://doi.org/10.1002/2016WR018598
[73] P. Lartaud, P. Humbert, J. Garnier, Sequential design for surrogate modeling in Bayesian inverse problems, 2025, arXiv:2402.16520.
[74] P. Villani, J. Unger, M. Weiser, Adaptive Gaussian Process Regression for Bayesian inverse problems, 2024, arXiv:2404.19459.
[75] N. Botteghi, M. Guo, C. Brune, Deep kernel learning of dynamical models from high-dimensional noisy data, Sci. Rep. 12 (1) (2022) 21530.
[76] Y. Li, Y. Wang, L. Yan, Surrogate modeling for bayesian inverse problems based on physics-informed neural networks, J. Comput. Phys. 475 (2023) 111841. https://doi.org/10.1016/j.jcp.2022.111841
[77] R.J. Adler, J.E. Taylor, Random fields and geometry, Springer, 2007.
[78] M. Loève, Probability Theory. Volume II, 46 of *Graduate Texts in Mathematics*, Springer-Verlag, New York, 4 edition, 1978.
[79] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M.E. Rognes, G.N. Wells, The FEnics project version 1.5, Arch. Numer. Softw. 3 (100) (2015). Number: 100. https://doi.org/10.11588/ans.2015.100.20553
[80] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015, Software available from tensorflow.org, https://www.tensorflow.org/.
[81] A. Gelman, D.B. Rubin, Inference from iterative simulation using multiple sequences, Stat. Sci. 7 (4) (1992) 457–472. https://doi.org/10.1214/ss/1177011136
[82] K. Champion, B. Lusch, J.N. Kutz, S.L. Brunton, Data-driven discovery of coordinates and governing equations, Proc. Natl. Acad. Sci. 116 (45) (2019) 22445–22451. https://doi.org/10.1073/pnas.1906995116
[83] M.T. Heideman, D.H. Johnson, C.S. Burrus, Gauss and the history of the fast fourier transform, IEEE ASSP Mag. 1 (4) (1984) 14–21. https://doi.org/10.1109/MASSP.1984.1162257
[84] J. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018). https://doi.org/10.1016/j.jcp.2018.02.037
[85] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. de Freitas, Taking the human out of the loop: a review of bayesian optimization, Proc. IEEE 104 (1) (2016) 148–175.
[86] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2623–2631.