# A nonparametric belief propagation method for uncertainty quantification with applications to flow in random porous media

Peng Chen[a], Nicholas Zabaras[*,a,b]

[a]*Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 101 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA*

[b]*Center for Applied Mathematics, 657 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA*

**Abstract**

We develop a nonparametric belief propagation method for uncertainty quantification and apply it to model flow in random porous media. The relationship between the high-dimensional input and the multi-output responses is addressed by breaking the global regression problem into smaller local problems using probabilistic links. These links can be well represented in a probabilistic graphical model. The whole framework is designed to be nonparametric (Gaussian mixture) in order to capture the non-Gaussian features of the response. With the known input distribution, a loopy nonparametric belief propagation algorithm is used to find the corresponding marginal distributions of the responses. The probabilistic graphical framework can be used as a surrogate model to predict the responses for new input realizations as well as our confidence on these predictions. Numerical examples are presented to show the accuracy and efficiency of the probabilistic graphical model framework and nonparametric belief propagation method for solving uncertainty quantification problems in flows in porous media using stationary and non-stationary permeability fields.

*Corresponding author at: Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 101 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA.
*Email address:* zabaras@cornell.edu (N. Zabaras).
*URL:* http://mpdc.mae.cornell.edu/ (N. Zabaras).

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **10 DEC 2012** | | **00-00-2012 to 00-00-2012** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **A nonparametric belief propagation method for uncertainty quantification with applications to flow in random porous media** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Cornell University,Materials Process Design and Control Laboratory,101 Frank H.T. Rhodes Hall,Ithaca,NY,14853-3801** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

**We develop a nonparametric belief propagation method for uncertainty quantification and apply it to model flow in random porous media. The relationship between the high-dimensional input and the multi-output responses is addressed by breaking the global regression problem into smaller local problems using probabilistic links. These links can be well represented in a probabilistic graphical model. The whole framework is designed to be nonparametric (Gaussian mixture) in order to capture the non-Gaussian features of the response. With the known input distribution, a loopy nonparametric belief propagation algorithm is used to find the corresponding marginal distributions of the responses. The probabilistic graphical framework can be used as a surrogate model to predict the responses for new input realizations as well as our confidence on these predictions. Numerical examples are presented to show the accuracy and efficiency of the probabilistic graphical model framework and nonparametric belief propagation method for solving uncertainty quantification problems in flows in porous media using stationary and non-stationary permeability fields.**

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT **unclassified** | b. ABSTRACT **unclassified** | c. THIS PAGE **unclassified** | **Same as Report (SAR)** | **69** | |

*Key words:* Uncertainty quantification, Predictive modeling, Probabilistic graphical models, Porous media flow, Nonparametric belief propagation, Loopy belief propagation

## 1. Introduction

Uncertainty Quantification (UQ) of multi-scale and multi-physics systems is a field of great interest and has attracted the attention of many researchers and communities in recent years. However, it is difficult to construct a complete probabilistic model of such problems mainly because they often involve high-dimensional and continuous random variables, which have complex, multi-modal distributions.

Over the past few decades, many methods and algorithms have been developed to address UQ problems. The most widely used method is the Monte Carlo (MC) method. MC's wide acceptance is due to the fact that it can uncover the complete statistics of the solution, while having a convergence rate that is (remarkably) independent of the input dimension. Nevertheless, it quickly becomes inefficient in high dimensional and computationally intensive problems, where only a few samples can be observed. Other methods are attempting to construct a surrogate model for the complex physical system. The idea is to run the deterministic physical solver on a small, well-selected set of inputs and then use these data to learn the response surface, so that the UQ problem can be studied based on the surrogate instead of the computationally expensive simulator. Such kind of methods include, the adaptive sparse grid collocation method (AGSC) [1], the multi-response Gaussian process method (MGP) [2], adaptive locally weighted projection regression methods (ALWPR) [3], and so on. However, all these methods have to face the "curse of dimensionality" problem, when the inputs are high-dimensional.

Recently developed probabilistic graphical models [4] can provide a powerful framework to effectively interpret complex probabilistic problems involving many inter-correlated variables. The two basic elements of a graphical model are its nodes and edges. The nodes represent the random variables and an edge linking two nodes represents the correlation between them. The joint probability distribution can be accessed by decomposing the complex network into local clusters defined by connected subsets of nodes. Then, by applying appropriate inference algorithms, the marginal and conditional probabilities of interest can be effectively calculated.

The probabilistic graphical model has been used in a range of application domains, which include web search [5], medical and fault diagnosis [6], speech recognition [7], robot navigation [8], bioinformatics [9], communications [10], natural language processing [11], computer vision [12], and many more. Most of the above applications have demonstrated the excellent performance of graphical model in the discrete world and the low-dimensional continuous world. However, for problems involving high-dimensional continuous variables, the number of efficient and accurate algorithms is limited.

The simplest way to investigate continuous graphical models is discretization. However, for problems involving high-dimensional variables, exhaustive discretization of the random space is computationally infeasible. Gaussian approximation is a widely used technique to build the continuous model, however, the performance of Gaussian model is not very satisfactory, especially for problems involving non-Gaussian features. Therefore, in this work, we propose to build a nonparametric (non-Gaussian) graphical model.

The general procedure of studying a graphical model problem can be summarized as follows: (1) Prepare the training data for the graphical model; (2) Design the structure of the graphical model; (3) Learn the graphical model with the training data, including designing the potential functions and learning the unknown parameters; (4) Solve an inference problem, that is, find the conditional or marginal probabilities of interest. In this work, we consider a graphical model that interprets the probabilistic relationship between the inputs of a physical system with the corresponding responses. Generally speaking, for a complex physical problem, the input variables are the key factors which determine the characteristics or the performance of the physical system (response variables). For example, in flow in porous media, the input variable is the discretized representation of the permeability field, while the response variables are the physical properties such as velocity and pressure, which are strongly influenced by the inputs. Model reduction techniques often have to be applied first to the input due to its high dimensionality. All the unknown parameters in the graphical model can be learned locally via techniques such as maximum likelihood (MLE), or maximum a posteriori probability (MAP). To address the inference problem several sampling and variational algorithms can be applied. Since the designed graphical model in this work is nonparametric, a sampling based nonparametric belief propagation [13, 14] algorithm is selected to carry out the inference task. In the following sections, we will discuss how we build the graphical model and how to study the problems of interest in detail. In particular, in this work, the

problem under investigation is flow through porous (heterogeneous) media. We are interested in investigating how the uncertainty propagates from the input permeability field to the corresponding response properties field. In addition, we want to build a surrogate model to the deterministic solver that will allows us for any realization of the input permeability to predictive the physical responses as well as our confidence on these predictions (induced by the limited data used to train the graphical model).

In [15], the authors proposed a probabilistic graphical model for multiscale stochastic partial differential equations (SPDEs) that focuses on the correlation between physical responses. The distribution of physical responses conditioned on stochastic input was approximated using conditional random field theories. Different physical responses (such as flux and pressure in flows in heterogeneous media) are correlated in such a way that their interactions are assumed to be conditioned on fine-scale local properties. No model reduction of fine-scale properties was involved in this process. The influence of fine-scale properties on coarse-scale responses was modeled through a set of hidden variables. The approach in this paper is significantly different in multiple fronts: (1) an undirected graph model is introduced rather than a factor graph as in [15]; (2) the graphical model considers output responses that are independent of each other; (3) an explicit model reduction scheme is considered to reduce the dimensionality of the random permeability field without the need for introducing hidden variables; and (4) the graph structure and graph learning scheme are implemented in a completely different algorithmic approach based on the Expectaction/Maximization (EM) algorithm and a sampling based approach to nonparametric belief propagation.

This paper is organized as follows. First, the problem definition is given in Section 2. Then the basic procedure of how to construct an appropriate graphical model and all the associated algorithms are discussed in Sections 3, 4 and 5. In Section 6, we introduce the porous media flow problem and provide various examples demonstrating the efficiency and accuracy of the graphical model approach. Brief discussion and conclusions are finally provided in Section 7.

## 2. Problem definition

Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space, where $\Omega$ is the sample space corresponding to the outcomes of some experiments, $\mathcal{F}$ a $\sigma$-algebra of subsets in $\Omega$ (these subsets are called events) and $\mathcal{P} : \mathcal{F} \to [0, 1]$ the probability mea-

sure. In this framework, let us define $D \subset \mathbb{R}^d, d = 1, 2, 3$ the spatial domain of interest (physical space), $\mathbf{x} \in D$ a spatial point, where $\mathbf{x} = (x_1, \ldots, x_d)$. Consider a random field $\{\mathbf{A_x}\}_{\mathbf{x}\in D}$. We assume $\mathbf{A_x}$ is a function that maps the probability space $\Omega$ to $\mathbb{R}$, i.e.,

$$\mathbf{A_x} : \Omega \to \mathbb{R}, \tag{1}$$

which assigns to each element $\omega \in \Omega$ a real-value (considering isotropic permeability) $\mathbf{A_x}$ at a specific point $\mathbf{x}$. We define $\mathbf{a} = \mathbf{A}(\omega), \omega \in \Omega$, a realization of $\mathbf{A}$.

In practice, the physical domain is decomposed into fine-elements where the permeability is defined. For example, consider a partition, $\mathcal{T}_f$ for the domain $D$ into non-overlapping elements $e_i$, i.e., $\mathcal{T}_f = \bigcup_{i=1}^{N_f} e_i$, where $N_f$ is the number of fine-elements in the domain. The random input field is specified on the fine-grid and $\mathbf{A}(\omega)$ can be written as follows:

$$\mathbf{A}(\omega) = (\mathbf{A}_1(\omega), \mathbf{A}_2(\omega), \ldots, \mathbf{A}_{N_f}(\omega)). \tag{2}$$

For most cases, we are only interested in the responses on a coarser grid than $\mathcal{T}_f$. Therefore, let us define a coarser partition of the same domain $D$. Denote this partition as $\mathcal{T}_c = \bigcup_{i=1}^{N_c} E_i$, where $N_c$ is the number of coarse-elements. Fig. 1 shows a fine-grid (finer lines) and a corresponding coarse-grid (heavier lines). Let $N_G$ denotes the number of nodes on the coarse-grid. Then we can represent the response field $\mathbf{Y}(\omega) = \{\mathbf{Y_x}(\omega) : \mathbf{x} \in D\}$ as a discrete set of responses on the coarse-nodes as follows:

$$\mathbf{Y}(\omega) = (\mathbf{Y}_1(\omega), \mathbf{Y}_2(\omega), \ldots, \mathbf{Y}_{N_G}(\omega)). \tag{3}$$

The values of $\mathbf{Y_x}(\omega)$ on other spatial points can be interpolated.

In uncertainty quantification tasks, one specifies a probability density on the input $\mathbf{A}$, $p(\mathbf{A})$, and is interested to quantify the probability measure induced by it on the response field. This can be achieved by marginalizing $\mathbf{Y}$ from the joint distribution of $p(\mathbf{Y}, \mathbf{A})$ as:

$$\begin{aligned} p(\mathbf{Y}) &= \int p(\mathbf{Y}, \mathbf{A}) d\mathbf{A} \\ &= \int p(\mathbf{Y}|\mathbf{A}) p(\mathbf{A}) d\mathbf{A}. \end{aligned} \tag{4}$$

If we assume a constant permeability at each fine-scale element, then $\mathbf{A}$ is described by a high-dimensional ($N_f$) set of highly correlated random

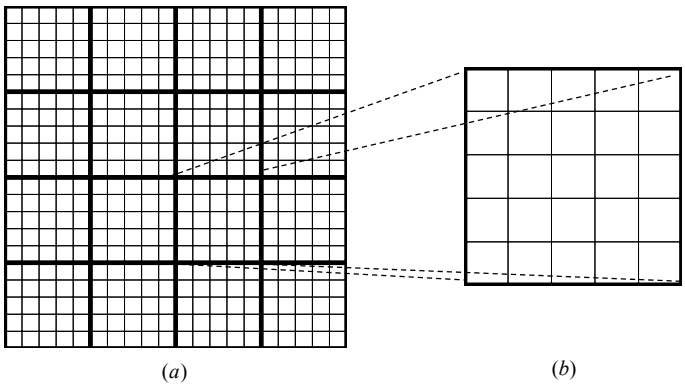


Figure 1: Schematic of the domain partition: (a) fine- and coarse-scale grids and (b) fine-scale local region in one coarse-element.

variables with a distribution that is hard to evaluate. A common way to deal with such problem is creating a reduced-dimensionality input model. For example, one can perform a Karhunen-Loève expansion [16] for a given set of permeability realizations, and then truncate this expansion after $k_\xi$ terms, and use $k_\xi$ ($k_\xi << dim(\mathbf{A})$) random variables $\boldsymbol{\xi}$ to represent the permeability distribution. Therefore, the above uncertainty propagation problem can be re-formulated as:

$$\begin{aligned} p(\mathbf{Y}) &= \int p(\mathbf{Y}, \mathbf{A}) d\mathbf{A} \\ &\approx \int p(\mathbf{Y}, \boldsymbol{\xi}) d\boldsymbol{\xi} \\ &= \int p(\mathbf{Y}|\boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi}. \end{aligned} \tag{5}$$

In many problems, $\boldsymbol{\xi}$ is still high-dimensional which prevents us from efficiently finding the conditional distribution $p(\mathbf{Y}|\boldsymbol{\xi})$. Hence, we propose a way to localize the connection between the inputs and responses, but also link each local connection in a global sense. The localized connection will be introduced by assuming that the response at a give coarse-node is correlated mostly with the random permeability at the neighboring coarse-elements. Thus for each coarse-node $i$ (and thus response $y_i$), we can affiliate a reduced set of random variables $\mathbf{s}_i$ that define the permeability random field at the coarse-elements adjacent to node $i$. This is discussed next.

## 3. Model reduction

Let us assume that a set of realizations of the random input field $\mathbf{A}$ are given. Using the Karhunen-Loève expansion [16] with the given permeability realizations defined over the whole spatial domain $D$, we can compute the global reduced representation of the permeability in terms of the random variables $\boldsymbol{\xi}$. We can symbolically write the model reduction process as follows:

$$\boldsymbol{\xi} = \mathcal{R}_g(\mathbf{a}), \tag{6}$$

where $\mathbf{a}$ is a realization of the random field defined on the fine-grid, and $\mathcal{R}_g$ is the global reduction map. As part of this map, for each realization of the permeability field, we assume that there is a unique realization of the random variables $\boldsymbol{\xi}$. However, the difficulty arises because the dimensionality of $\boldsymbol{\xi}$, $k_\xi$, is high, i.e., $k_\xi >> 1$. Considering that each response on the coarse-grid depends on the permeability field over the whole domain (via the solution of the underlying porous media flow boundary value problem), it is rather difficult to build a probabilistic link between the response $\mathbf{Y}$ and the reduced input $\boldsymbol{\xi}$. Hence, we make a physically reasonable assumption for many problems that the response at one coarse-grid node correlates strongly on the input permeability in the underlying nearest coarse-elements, and that the influence by the permeability at all other coarse-elements can be ignored, as shown in Fig. 2.

Let $\Gamma_i \subset \{1, \ldots, N_c\}$ be the set of coarse-elements corresponding to the coarse-node $i$. Let $\mathbf{a}_{\Gamma_i}$ denote the input permeability field over the corresponding coarse-elements to coarse-node $i$ (see Fig. 2). Based on the given realizations of the permeability $\mathbf{a}_{\Gamma_i}$, one can perform a Karhunen-Loève expansion and obtain the reduced representation $\mathbf{s}_i$ of the permeability field over $\Gamma_i$ that encodes most of the information relevant to the response at coarse-node $i$ as:

$$\mathbf{s}_i = \mathcal{R}_{\Gamma_i}(\mathbf{a}_{\Gamma_i}), \tag{7}$$

where $\mathcal{R}_{\Gamma_i}$ is the reduction map for the permeability in the elements $\Gamma_i$. $\mathbf{s}_i$ should be correlated to its neighboring reduced representation $\mathbf{s}_j$ due to the overlapped inputs considered, as shown in Fig. 2.

The connection between this local model reduction and the global model reduction in Eq. (6) is given as follows: Let $\mathcal{C}_g$ be the global reconstruction map such that:

$$\mathcal{R}_g(\mathcal{C}_g(\boldsymbol{\xi})) = \boldsymbol{\xi}, \tag{8}$$

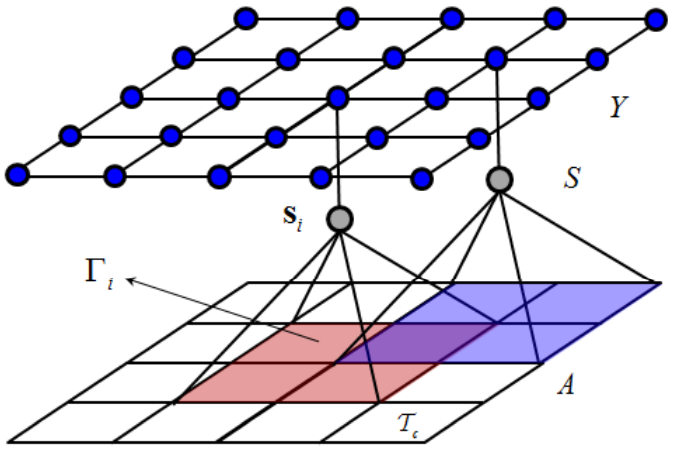


Figure 2: An illustration of the model reduction framework considered in this paper. The response at each coarse-node depends on the permeability field at the neighboring coarse-elements.

and let $\mathcal{C}_{\Gamma_i}$ be the local reconstruction map corresponding to the model reduction for the $i^{th}$ coarse-grid node defined as:

$$\widetilde{\mathbf{a}}_{\Gamma_i} = \mathcal{C}_{\Gamma_i}(\mathbf{s}_i), \tag{9}$$

where $\widetilde{\mathbf{a}}_{\Gamma_i}$ is the reconstructed local random field on the coarse-elements $\Gamma_i$.

We can write the following:

$$\mathbf{s}_i \approx \mathcal{R}_{\Gamma_i}([\mathcal{C}_g(\boldsymbol{\xi})]_{\Gamma_i}), \tag{10}$$

where $[\cdot]_{\Gamma_i}$ is the restriction of $\widetilde{\mathbf{a}} = \mathcal{C}_g(\boldsymbol{\xi})$ over $\Gamma_i$. Similarly, we can write,

$$\boldsymbol{\xi} \approx \mathcal{R}_g\{\mathcal{H}[\mathcal{C}_{\Gamma_1}(\mathbf{s}_1), \ldots, \mathcal{C}_{\Gamma_{N_G}}(\mathbf{s}_{N_G})]\}, \tag{11}$$

where $\mathcal{H}$ is an implicit function that averages permeability realizations obtained from local overlapping reduction models. The above equation implicitly defines how the local reduced input $\mathbf{s}_i$ is correlated to $\boldsymbol{\xi}$ through the reduction/reconstruction maps. The function $\mathcal{H}$ approximates the global reconstruction function $\mathcal{C}_g$, thus the better the choice of $\mathcal{H}$ is, the closer the local/global approximations we obtain. In this work, the function $\mathcal{H}$ is simply chosen as:

$$H(\cdot)|_{E_i} = \frac{1}{N_{\text{overlap}}} \sum_{j=1}^{N_{\text{overlap}}} [\mathcal{C}_{\Gamma_j}(\mathbf{s}_j)]_{E_i}, \tag{12}$$

where $N_{\text{overlap}}$ is the number of overlapped reconstruction over coarse-element $E_i$, and $[\cdot]_i$ is the restriction of $\widetilde{\mathbf{a}}_{\Gamma_j} = \mathcal{C}_{\Gamma_j}(\mathbf{s}_j)$ over the $E_i$ element.

**Remark 1:** The reduced variables $\mathbf{s}_i$ affiliated with different locations $i$ are different random variables. For a stationary permeability random field, local features have the same distribution on coarse-elements, therefore, the localized reduced random variables $\mathbf{s}_i$ are going to follow the same distribution for all $i$ (not considering boundary effects). However, notice that one can not say these $\mathbf{s}_i$ are the same random variables even if they follow the same distribution. Given a realization of stationary stochastic input $\mathbf{a}^{(n)}$, the local features $\mathbf{a}_k^{(n)}$ and $\mathbf{a}_l^{(n)}$ on coarse-elements $E_k$ and $E_l$ are in general different. For nonstationary random permeability field, the $\mathbf{s}_i$ variables at different locations $i$ follow different marginal distributions.

The localization of the input model reduction outlined above can be implemented with literally any model reduction technique including linear and nonlinear dimension reduction algorithms. For linear dimension reduction, the most famous and the most widely used method is the Principal Component Analysis (PCA) [17] method. The first version of PCA method appeared half a century ago and it has been shown since then to be a reliable reduction method forming the basis of many other more advanced mathematical reduction methodologies. In the last decade, a large number of nonlinear dimensional reduction techniques have been proposed (e.g., [18, 19, 20]). Most of the nonlinear techniques are not as well studied and have been shown often to provide better performance than PCA for artificial than physical nonlinear datasets [21]. These methods perform not better (sometimes much poorer) than PCA for natural datasets [21]. Therefore, here for simplicity of the presentation, we choose PCA as the dimension reduction technique. This will allow us to emphasize the graph theoretic approach for solving the underlying stochastic flow problem of interest. The basic algorithm for the PCA method used is summarized for completeness in Appendix A.

Up to now, we have completely defined the relationship between $\boldsymbol{\xi}$ and $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{N_G}\}$. Given the distribution of $\boldsymbol{\xi}$, it is straightforward to use Monte Carlo method to find the distribution of $\mathbf{S}$, $p(\mathbf{S})$. Then computing $p(\mathbf{Y})$ requires to compute the conditional $p(\mathbf{Y}|\mathbf{S})$ where $\mathbf{Y} = \{y_1, y_2, \ldots, y_{N_G}\}$.

Indeed, we can write the following:

$$\begin{aligned} p(\mathbf{Y}) &\approx \int p(\mathbf{Y}|\boldsymbol{\xi})p(\boldsymbol{\xi})d\boldsymbol{\xi} \\ &\approx \int p(\mathbf{Y}|\mathbf{S})p(\mathbf{S}|\boldsymbol{\xi})p(\boldsymbol{\xi})d\mathbf{S}d\boldsymbol{\xi} \\ &= \int p(\mathbf{Y}|\mathbf{S}) \left[\int p(\mathbf{S}|\boldsymbol{\xi})p(\boldsymbol{\xi})d\boldsymbol{\xi}\right] d\mathbf{S} \\ &= \int p(\mathbf{Y}|\mathbf{S})p(\mathbf{S})d\mathbf{S} \\ &= \int p(y_1, y_2, \ldots, y_{N_G}|\mathbf{s}_1, \ldots, \mathbf{s}_{N_G})p(\mathbf{S})d\mathbf{S}. \end{aligned} \tag{13}$$

As discussed in Section 1, probabilistic graphical models [4] can be used to systematically explain the probabilistic relationship between $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{N_G}\}$ and the responses $\mathbf{Y} = \{y_1, y_2, \ldots, y_{N_G}\}$. Their joint distribution can be partitioned in a way the accounts for the local nature of the dependence/correlation of the response to the input variables. The details of such approach are introduced in Section 4.

## 4. Probabilistic graphical model

We are given a number of realizations of the localized reduced input random variables $\mathbf{S}$ and corresponding responses $\mathbf{Y}$, and also the distribution of $\mathbf{S}$. Our first objective is building a probabilistic graphical model between $\mathbf{S}$ and $\mathbf{Y}$, based on the given set of realizations. We next plan to use an inference algorithm on the probabilistic graph to address uncertainty quantification problems.

### *4.1. Brief introduction to probabilistic graphical models*

A graphical model aims to represent the joint probability distribution of many random variables efficiently by exploiting factorization [4]. The two most common forms of graphical models are directed graphical models and undirected graphical models, based on directed graphs and undirected graphs, respectively. The dependence relationship is visible directly from the graph for the directed graph model, while the dependence relationship is hidden in the undirected graph. In this work, the dependence relationships between the response random variables are not clear, so we focus on

the undirected graph, which is also called pairwise Markov Random Field (MRF) [11].

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ are the nodes (random variables) and $\mathcal{E}$ are the edges of the graph (correlations). Let $\{X_\mathcal{V} : x_i \in \mathcal{V}\}$ be a collection of random variables indexed by the nodes of the graph and let $\mathcal{C}$ denote a collection of cliques of the graph (i.e., fully connected subsects of nodes). Associated with each clique $c \in \mathcal{C}$, let $\phi_c(X_c)$ denote a nonnegative potential function, which implicitly encodes the dependence information among the nodes within the clique. The joint probability $p(X_\mathcal{V})$ is defined by taking the product over these potential functions and normalizing,

$$p(X_\mathcal{V}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi_c(X_c), \tag{14}$$

where $Z$ is a normalization factor.

The graphical model representation makes the inference problem easier. The general algorithm of probabilistic inference is that of computing the marginal probability $p(\mathbf{X}_\mathcal{H})$ or conditional probability $p(\mathbf{X}_\mathcal{H}|\mathbf{X}_\mathcal{O})$, where $\mathcal{V} = \mathcal{O} \cup \mathcal{H}$ for given subsets $\mathcal{O}$ and $\mathcal{H}$. The belief propagation algorithm (inference) is then applied to find the marginal or conditional probabilities of interest. Notice if an event $\{X_\mathcal{O} = \mathbf{x}_\mathcal{O}\}$ is observed, the original clique potentials need to be modified, that is, for $\{X_i, i \in \mathcal{O}\}$, we multiply the potential $\phi_c(X_c)$ by the Kronecker Delta function $\delta_{X_i}(\mathbf{x}_i)$ for any clique $c \in \mathcal{C}$ such that $\{i \in c \cap \mathcal{O}\}$, where $\mathbf{x}_i$ is the observation of node $i$. The detailed inference algorithm will be discussed in Section 5.

*4.2. The structure of the graph*

To find an efficient structure of the graph, let us start from the joint distribution of $(\mathbf{Y}, \mathbf{S}, \boldsymbol{\xi})$,

$$p(\mathbf{Y}, \mathbf{S}, \boldsymbol{\xi}) = p(\mathbf{Y}|\mathbf{S})p(\mathbf{S}|\boldsymbol{\xi})p(\boldsymbol{\xi}). \tag{15}$$

The above decomposition is based on the assumption that $\mathbf{S}$ contains all information $\boldsymbol{\xi}$ contains for the calculation of $\mathbf{Y}$. The probabilistic relationship between $\mathbf{S}$ and $\boldsymbol{\xi}$ is discussed in Section 3. Each variable $\mathbf{s}_i \in \mathbf{S}$ has a deterministic relationship with $\boldsymbol{\xi}$, therefore, all the $\mathbf{s}_i$'s should be directly linked with $\boldsymbol{\xi}$. The correlation among the $\mathbf{s}_i$ variables is then reflected via their connections with $\boldsymbol{\xi}$. The corresponding structure between $\mathbf{S}$ and $\boldsymbol{\xi}$ is given in Fig. 3.

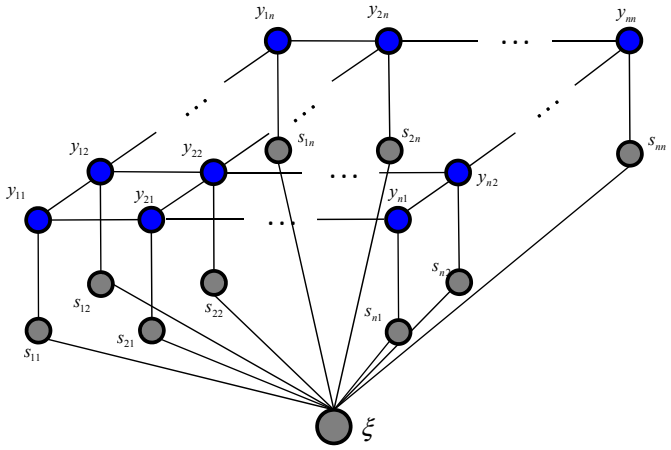


Figure 3: The general graph structure for the problem of interest. The $y$ variables represent the response of the system (velocities and/or pressure on a coarse-grid), $\boldsymbol{\xi}$ represents the reduced set of random variables defining the random permeability over the whole domain $D$ and $\mathbf{s}_i$ is the reduced set of random variables defining the random permeability on the patch of coarse-elements that share the coarse-node $i$.

To find the structure between $\mathbf{Y}$ and $\mathbf{S}$, we need to find an approximate decomposition of $p(\mathbf{Y}|\mathbf{S})$ in Eq. (13),

$$p(\mathbf{Y}|\mathbf{S}) = p(y_1, y_2, \ldots, y_{N_G}|\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N_G}). \tag{16}$$

In this work, we consider only the pairwise correlations between the response random variables $y$, among which, only correlations between neighboring response variables are considered. We further assume that these correlations do not depend on the local features. So the conditional distribution $p(\mathbf{Y}|\mathbf{S})$ can be decomposed as

$$p(\mathbf{Y}|\mathbf{S}) \approx \prod_{i=1}^{N_G} p(y_i|\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N_G}) \prod_{j\in\Gamma(i)} p(y_i, y_j), \tag{17}$$

where $\Gamma(i)$ denotes the set of neighboring nodes of node $i$.

**Remark 2:** This decomposition is inspired by the general treatment to the conditional random field representation of a Gibbs distribution, where in principle the explicit expansion of the conditional distribution involves one-body term and two-body interaction terms to $n$-body interaction term.

In practice, we ignore higher-order interactions and only keep the first two terms [22, 23]. In [15], the authors also apply a similar idea in factorizing the complex conditional probability distribution, however they assume that the correlations between the physical responses are also dependent explicitly on property features.

To further simplify the dependencies in the factorization of $p(\mathbf{Y}|\mathbf{S})$, we further assume that the response at one coarse-grid node is strongly dependent on the inputs in its underlying nearest coarse-elements, thus the influence by all other inputs is ignored. In other words, we are assuming that $y_i$ only depends on its underlying localized reduced input $\mathbf{s}_i$. This is in analogy to various multiscale methods (e.g. the MsFEM method [24]) where in the calculation of the local multiscale basis functions only the local permeability is considered. Hence, the above equation can be further decomposed as

$$p(\mathbf{Y}|\mathbf{S}) \approx \prod_{i=1}^{N_G} p(y_i|\mathbf{s}_i) \prod_{i,j} p(y_i, y_j). \tag{18}$$

The constructed structure of the undirected graph is shown in Fig. 3. If we write Eq. (18) as a product of potential functions in the graphical model, we can obtain

$$p(\mathbf{Y}|\mathbf{S}) \propto \prod_{k \in \mathcal{V}^{(\mathbf{y})}} \varphi_k(y_k, \mathbf{s}_k) \prod_{(i,j) \in \mathcal{E}^{(y)}} \psi_{i,j}(y_i, y_j). \tag{19}$$

There are two kinds of potential functions in Eq. (19), one is pair potential functions, $\psi(*)$, that model the correlation between neighboring response variables, the other one is the cross potential functions, $\varphi(*)$, which interprets the relationship between the reduced input variables $\mathbf{s}_i$ and response variables $y_i$. In the following, we denote with $\mathcal{V}^{(\mathbf{s})}$ the set of the localized reduced input nodes (coarse-grid nodes), and with $\mathcal{E}^{(y)}$ the set of the edges between the response variables (edges of the coarse-elements).

In this work, the potential functions between $\mathbf{s}_i$ and $\boldsymbol{\xi}$ are difficult to model due to their high dimensionality nature. However, $\mathbf{S}$ and $\boldsymbol{\xi}$ are explicitly known to us, so is the relationship between them. Therefore, we do not have to learn these potential functions. In Section 5, we will discuss how we perform the inference problem without using the potential functions between $\mathbf{s}_i$ and $\boldsymbol{\xi}$.

*4.3. Learning the graphical model*

Since we are considering a nonparametric graphical model, the potential functions should have Gaussian mixture forms. As discussed above, there are two types of potential functions, the pairwise potential function for the response variables, $\psi(*)$, and the cross potential function for response variables and localized reduced input variables, $\varphi(*)$, as given in Eq. (19). In this work, both of the potential functions are designed to have the following form,

$$\psi_{i,j}(z_i, z_j) = \sum_{m=1}^{M} \omega^{(m)} \mathcal{N}\left((z_i, z_j); \mu^{(m)}, \Sigma^{(m)}\right), \tag{20}$$

where $z_i$ and $z_j$ denote the random variables on node $i$ and node $j$ and $\mathcal{N}(\cdot)$ is the Gaussian distribution. The unknown parameters in the above potential functions are $\{\omega^{(m)}, \mu^{(m)}, \Sigma^{(m)}, m = 1, \ldots, M\}$, where $\omega^{(m)}$ is the weight (scalar) for component $m$, $\mu^{(m)}$ is the mean for component $m$ (the size of the mean vector is equal to the sum of dimensions of $z_i$ and $z_j$), and $\Sigma^{(m)}$ is the covariance matrix.

In this work, the unknown parameters in the potential functions are learned by maximizing the log-likelihood. Denote $\boldsymbol{\Theta} = \left\{\boldsymbol{\theta}_{i,j} : (i,j) \in \mathcal{E}^{(y)} \text{ and } \boldsymbol{\theta}_i : i \in \mathcal{V}^{(y)}\right\}$ as the set of all the unknown parameters, where $\boldsymbol{\theta}_{i,j} = \left\{\omega_{i,j}^{(m)}, \mu_{i,j}^{(m)}, \Sigma_{i,j}^{(m)}; m = 1, \ldots, M\right\}$ and $\boldsymbol{\theta}_i = \left\{\omega_i^{(m)}, \mu_i^{(m)}, \Sigma_i^{(m)}; m = 1, \ldots, M\right\}$. These parameters can be calculated locally in the coarse-grid. For specific $i, j$ such that $i \in \mathcal{V}^{(\mathbf{y})}$ and $(i,j) \in \mathcal{E}^{(y)}$, let us consider given $N$ observations of $\{(\mathbf{s}_i^{(n)}, \mathbf{s}_j^{(n)}), (y_i^{(n)}, y_j^{(n)})\}$ for $n = 1, \ldots, N$. The log-likelihood can then be calculated as,

$$\begin{aligned} &\mathcal{L}\big(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i,j} | \mathbf{s}_i^{(1)}, \ldots, \mathbf{s}_i^{(N)}, (y_i^{(1)}, y_j^{(1)}), \ldots, (y_i^{(N)}, y_j^{(N)})\big) \\ &= \sum_{n=1}^{N} \left[\log p(y_i^{(n)}, \mathbf{s}_i^{(n)} | \boldsymbol{\theta}_i) + \log p(y_i^{(n)}, y_j^{(n)} | \boldsymbol{\theta}_{i,j})\right]. \end{aligned} \tag{21}$$

By maximizing the log-likelihood, we obtain

$$\big(\widehat{\boldsymbol{\theta}}_i, \widehat{\boldsymbol{\theta}}_{i,j}\big) = \arg\max_{\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i,j}} \mathcal{L}\big(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i,j} | \mathbf{s}_i^{(1)}, \ldots, \mathbf{s}_i^{(N)}, (y_i^{(1)}, y_j^{(1)}), \ldots, (y_i^{(N)}, y_j^{(N)})\big). \tag{22}$$

Notice that maximizing the log-likelihood is equivalent to maximizing each component of Eq. (21) separately, therefore, the graph learning problem can be divided into a number of local learning problems. For example, to

learn $\boldsymbol{\theta}_i$, we only need to maximize $\sum_{n=1}^{N} \log p(y_i^{(n)}, \mathbf{s}_i^{(n)} | \boldsymbol{\theta}_i)$ using the local training data set $\{y_i^{(n)}, \mathbf{s}_i^{(n)}; i = 1, \ldots, N\}$.

**Remark 3:** The parameters $\boldsymbol{\theta}_{i,j}$ define the correlation between the response variables, whereas $\boldsymbol{\theta}_i$ interpret the dependence relation between a response variable and its underlying localized reduced random input. Both of these parameters are computed locally using the training data. Thus the computational cost affiliated with the estimation of the parameters that define the probabilistic dependencies in the graph is minimal. Note that the approach used here is different from that in [15] where local estimation problems are only posed to compute the dependencies on the input permeability permeability of the local potentials. In this work, the effect of the input permeability is introduced via the local random variables $\mathbf{s}_i$ and the potentials considered are Gaussian mixtures with unknown parameters. The potentials in [15] are simple Gaussians. In general case, all the $\boldsymbol{\theta}_{i,j}$ and $\boldsymbol{\theta}_i$ are different across the graph (i.e. for different $i$ and $j$).

**Remark 4:** For a stationary permeability case, the variables $\mathbf{s}_i$ follow the same distribution but this does not imply that $\boldsymbol{\theta}_i$ are the same parameters. Taking simultaneous realizations of $\mathbf{s}_i$ and $\mathbf{s}_j$ leads to different local permeability realizations and thus different response fields $y_i^{(n)}$ and $y_j^{(n)}$, $n = 1, \ldots, N$. In the calculation of $\boldsymbol{\theta}_i$, the training data set $\{y_i^{(n)}, \mathbf{s}_i^{(n)}, i = 1, \ldots, N\}$ that we use vary with the location $i$ and therefore $\boldsymbol{\theta}_i$ differs with location. Similar argument can be made about the location dependence of the parameters $\boldsymbol{\theta}_{i,j}$.

The Expectation Maximization (EM) algorithm is chosen to maximize the local log-likelihood. The details of the EM algorithm are given in Appendix B. Note that in the EM algorithm employed, the number of mixture components $M$ is predefined. A discussion of how to choose $M$ is provided in Section 6.1.

## 5. Inference problem

The general inference problem in a graphical model is to find the marginal or conditional probabilities of interest in the graph. This task is usually performed by using the belief propagation (BP) algorithm [4]. In this work, after all the underlying parameters in the graph are successfully learned, the inference problem can be performed. In the following, we first provide a brief

introduction to the general belief propagation algorithm, and then we discuss in detail how to apply the belief propagation algorithm into our framework.

### *5.1. General belief propagation*

Belief propagation (BP) is a general inference algorithm for graphical models. In the BP algorithm, each node $k$ iteratively solves the global inference problem by integrating information from the local environment, and then transmits a summary message to all its neighbors $l \in \Gamma(k)$ along the edges, where $\Gamma(k)$ denotes all the neighboring nodes of node $k$. The information flow during this process is called the message, or belief, which is a function containing sufficient information of the "influence" that one variable exerts on another.

The BP algorithm begins by randomly initializing all messages $m_{y_k}(y_j)$, where $m_{y_k}(y_j)$ denotes message from node $y_k$ to node $y_j$, and then updating the messages along each edge according to the following recursion [11, 4], as shown in Fig. 4(b):

$$m_{y_k}(y_j) \propto \int \psi(y_k, y_j) \prod_{l \in \Gamma(k) \setminus j} m_{y_l}(y_k) dy_k. \tag{23}$$

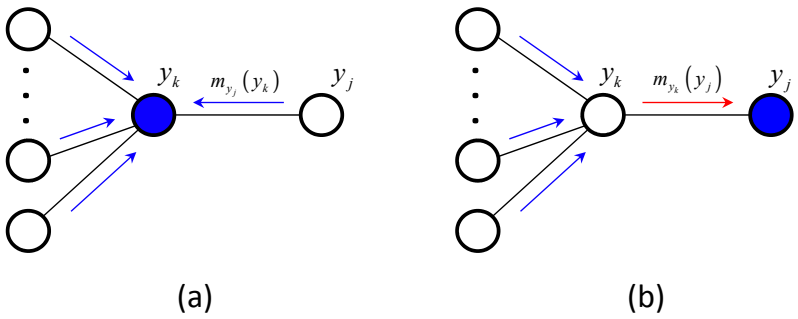


Figure 4: Message-passing recursions underlying the BP algorithm: (a) Approximate estimate of the marginal distribution $p(y_k)$ (Eq. (24)) combines the local input information with messages from neighboring nodes; (b) A new outgoing message $m_{y_k}(y_j)$ from node $y_k$ towards neighboring node $y_j$ is computed from all other incoming messages to node $y_k$ except node $y_j$, as in Eq. (23).

Generally, the marginal distribution of each node (e.g., node $k$) given the observation can be computed by gathering all the messages coming into that

node [11, 4], as shown in Fig. 4(a):

$$p(y_k) \propto \prod_{l \in \Gamma(k)} m_{y_l}(y_k). \tag{24}$$

*5.2. Inference approach for the problem of interest*

In this section, suppose $p(\boldsymbol{\xi})$ is known. The objective is then to find the marginal distribution of $p(\mathbf{Y})$ via the belief propagation algorithm. In the particular problem of interest, there are three main challenges with regards to the inference algorithm: (1) how to represent and calculate the message from $\boldsymbol{\xi}$ to $\mathbf{S}$ (discussed in Section 5.2.1), (2) how to treat the loops in the graph (discussed in Section 5.2.2), and (3) how to calculate the message update in the form of a Gaussian mixture (discussed in Section 5.2.3).

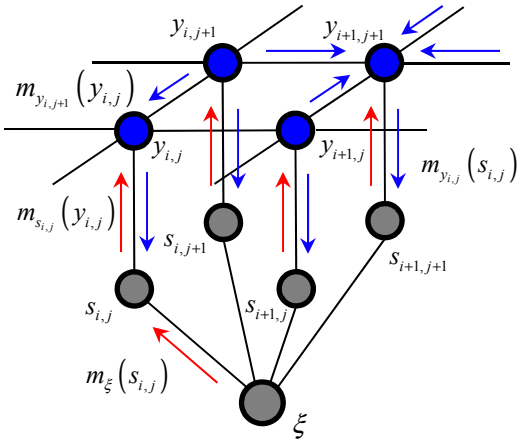


Figure 5: Message flow in the present graphical model framework. We assume a two-dimensional response with response variables (velocity components/pressure indicated by the blue nodes).

*5.2.1. Detailed inference algorithm*

The illustration of the message flow in the current graphical model framework is given in Fig. 5. Note that in this two-dimensional framework, we identify our nodes with two indices $(i, j)$ rather than the single indices $1, 2, , \ldots, N_G$ used in our earlier analysis. There are four types of messages in the figure, (a) message from a response node $y_{(k,l)}$ to its neighboring response node $y_{(i,j)}$, $m_{y_{(k,l)}}(y_{(i,j)})$ (Eq. (25) below), (b) message from a response node $y_{(i,j)}$ to its

neighboring localized reduced input node, $m_{y_{(i,j)}}(\mathbf{s}_{(i,j)})$ (Eq. (26) below), (c) message from a localized reduced input node $\mathbf{s}_{(i,j)}$ to its neighboring response node $y_{(i,j)}$, $m_{\mathbf{s}_{(i,j)}}(y_{(i,j)})$ (Eq. (27) below), and (d) message from the global reduced input node $\boldsymbol{\xi}$ to $\mathbf{s}_{(i,j)}$, $m_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)})$ (calculated by Eq. (29) shown below).

Following Eq. (23), the message from a response node $y_{(k,l)}$ to its neighboring response node $y_{(i,j)}$ can be calculated as:

$$m_{y_{(k,l)}}(y_{(i,j)}) \propto \int \psi(y_{(i,j)}, y_{(k,l)}) m_{\mathbf{s}_{(k,l)}}(y_{(k,l)}) \prod_{y_{(p,q)} \in \Gamma^{(y)}(y_{(k,l)}) \setminus y_{(i,j)}} m_{y_{(p,q)}}(y_{(k,l)}) dy_{(k,l)}, \tag{25}$$

where $\Gamma^{(y)}(y_{(k,l)}) \setminus y_{(i,j)}$ denotes all the nearest neighboring response variables to $y_{(k,l)}$ excluding $y_{(i,j)}$, and $s_{(k,l)}$ is the corresponding reduced input representation that locally influences the response variable $y_{(k,l)}$. $m_{\mathbf{s}_{(k,l)}}(y_{(k,l)})$ is the message sent from $\mathbf{s}_{(k,l)}$ to $y_{(k,l)}$, and it can be calculated by Eq. (27) given below.

The message from a response node $y_{(i,j)}$ to its neighboring localized reduced input node $\mathbf{s}_{(i,j)}$ can be calculated as:

$$m_{y_{(i,j)}}(\mathbf{s}_{(i,j)}) \propto \int \varphi_{(i,j)}(y_{(i,j)}, s_{(i,j)}) \prod_{y_{(k,l)} \in \Gamma^{(y)}(y_{(i,j)})} m_{y_{(k,l)}}(y_{(i,j)}) dy_{(i,j)}. \tag{26}$$

Denote the message sent from $\boldsymbol{\xi}$ to the $\mathbf{s}_{(i,j)}$ variable as $m_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)})$. Assuming that this message is known now (it will be discussed next), then the message from a localized reduced input node $\mathbf{s}_{(i,j)}$ to its neighboring response node $y_{(i,j)}$ can be calculated as:

$$m_{\mathbf{s}_{(i,j)}}(y_{(i,j)}) \propto \int \varphi_{(i,j)}(y_{(i,j)}, s_{(i,j)}) m_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)}) d\mathbf{s}_{(i,j)}. \tag{27}$$

Lastly we discuss the task of calculating the message from $\boldsymbol{\xi}$ to $\mathbf{s}_{(i,j)}$. It is hard to obtain message update from Eq. (23), because both $\boldsymbol{\xi}$ to $\mathbf{s}_{(i,j)}$ are high dimensional, and in addition we need to calculate the messages from all the other $\mathbf{s}$ variables except $\mathbf{s}_{(i,j)}$ to $\boldsymbol{\xi}$. To bypass these difficulties, we use a different way to construct the unknown message from the information that is already known to us as follows. According to Eq. (24), we can write the following:

$$p(\mathbf{s}_{(i,j)}) \propto m_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)}) m_{y_{(i,j)}}(\mathbf{s}_{(i,j)}). \tag{28}$$

Since $\mathbf{S}$ are known to us, we know exactly what $p(\mathbf{s}_{(i,j)})$ is, and also we know $m_{y_{(i,j)}}(\mathbf{s}_{(i,j)})$ from Eq. (26). Then we can write:

$$m_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)}) \propto \frac{p(\mathbf{s}_{(i,j)})}{m_{y_{(i,j)}}(\mathbf{s}_{(i,j)})}. \tag{29}$$

As a result, the message sent from $\boldsymbol{\xi}$ to $\mathbf{s}_{(i,j)}$ is updated using the known marginal distribution of $\mathbf{s}_{(i,j)}$ and $m_{y_{(i,j)}}(\mathbf{s}_{(i,j)})$. In this work, this message is calculated by sampling from $\frac{p(\mathbf{s}_{(i,j)})}{m_{y_{(i,j)}}(\mathbf{s}_{(i,j)})}$ using the Metropolis Hastings algorithm [25]. The details of how to calculate $m_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)})$ are given in Appendix C.

**Remark 5:** If a realization of the stochastic input, $\mathbf{a}$, is given, then there is no need to calculate Eqs. (26) and (29) because $\mathbf{S}$ and $\boldsymbol{\xi}$ can be exactly known from the model reduction scheme. The message from $\mathbf{s}_{(i,j)}^{(n)}$ to $y_{(i,j)}$, as given in Eq. (27), can be calculated as: $m_{\mathbf{s}_{(i,j)}}(y_{(i,j)}) \propto \int \varphi_{(i,j)}(y_{(i,j)}, s_{(i,j)})\delta_{s_{(i,j)}}(\mathbf{s}_{(i,j)}^{(n)})d\mathbf{s}_{(i,j)}$, where the Delta function only takes value when $\mathbf{s}_{(i,j)}^{(n)}$ equals to the given realization.

After the belief propagation algorithm is completed, we obtain the marginal distribution of the physical responses conditioned on the given input, e.g. $p(y_{(i,j)}|\mathbf{a})$. Let the expectation $\mathbb{E}[y_{(i,j)}|\mathbf{a}]$ be the predicted values of the physical responses and the variance be the corresponding error bars. We thus obtain a surrogate model based on the graphical model that for an any input realization provides us the response of the system as well as our confidence on this prediction.

In the uncertainty quantification problem, one exerts a known distribution on the input $\mathbf{A}$, $p(\mathbf{A})$, and is interested to quantify the probability induced by it on the response. Using the model reduction techniques discussed in Section 3, we can explicitly compute the distribution of $\boldsymbol{\xi}$ and $\mathbf{S}$, $p(\boldsymbol{\xi})$ and $p(\mathbf{S})$, respectively, given $p(\mathbf{A})$ or a set of realizations of $\mathbf{A}$. Then, by executing the inference problem discussed above, we can obtain an explicit representation of the marginal distribution of all the responses by gathering all the messages coming from the neighbors (as in Eq. (24)). The statistics of interest can be calculated directly from the marginal distribution.

*5.2.2. Loopy belief propagation (LBP)*

For graphs with loops, there are two major types of treatment. The first approach is to break the loops, by either cutting the cycles or finding an

equivalent surrogate graph, such as in the graph cut [26] or the junction tree algorithms [27]. The second approach is to find an efficient way to recursively update the messages until convergence. Although until now there is no strict mathematical justification that the loopy belief propagation converges to the true marginals, in many applications, the resulting LBP algorithm exhibits excellent performance [28, 29, 30, 31]. Recently, several theoretical studies have provided insights into the approximations made by LBP, establishing connections to other variational inference algorithms and partially justifying its application to graphs with cycles [32, 33]. In this work, we simply follow the second approach, by finding an efficient way to calculate all the messages in one iteration. The complete procedure is given in Algorithm 1 and depicted in Fig. 6. The messages are considered as converged if their change is less than a threshold in two successive iterations as in Eq. (30).

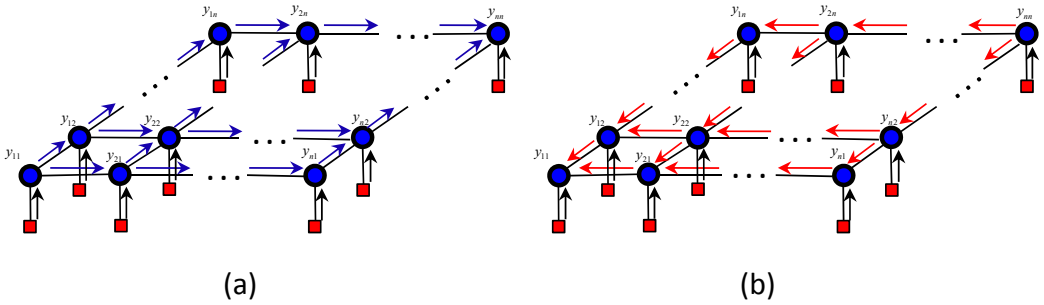


Figure 6: Illustration of the Loopy Belief Algorithm used in this paper. (a) Message flow from the root node to the end node, as in Step 2.1.b in Algorithm 1. (b) Message flow from the end node to the root node, as in step 2.1.c in Algorithm 1.

*5.2.3. Nonparametric belief propagation*

In the nonparametric graphical model, each message is represented by a Gaussian mixture. Then the belief update (Eqs. (25)-(27)) becomes analytically intractable. Currently there are two possible approximations for performing the belief update. The first one is using a variational method [34]. The basic idea of the variational method is using a much simpler form (user defined) to obtain an approximation that is as close as possible to the target message. The second approach is using a sampling method [13]. The idea of the sampling method comes from particle filters. In [14], it was extended to graphs containing continuous, non-Gaussian variables leading to the so called "Nonparametric belief propagation (NBP) method". In this work, we utilize

**Algorithm 1** The complete inference algorithm

1: Initialization: With given $p(\boldsymbol{\xi})$ and the deterministic relationship between $\mathbf{S}$ and $\boldsymbol{\xi}$, $p(\mathbf{S})$ can be obtained via MC method as discussed in Section 3. We set the initial message as $m^{(0)}_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)}) = p(\mathbf{s}_{(i,j)})$, and all other messages as a standard Gaussian $\mathcal{N}(0,1)$.

2: Iterate: At step $t$,

(1) Update $m^{(t)}_{y_{(k,l)}}(y_{(i,j)})$ as in Eq. (25),

a) Set one node as the root node, and another node as the end node.

b) Calculate all the messages from the root node to the end node, as shown in Fig. 6(a).

c) Calculate all the messages from the end node to the root node, as shown in Fig. 6(b).

(2) Update $m^{(t)}_{y_{(i,j)}}(\mathbf{s}_{(i,j)})$ as in Eq. (26).

(3) Update $m^{(t)}_{\boldsymbol{\xi}}(\mathbf{s}_{(i,j)})$ as in Eq. (29).

(4) Update $m^{(t)}_{\mathbf{s}_{(i,j)}}(y_{(i,j)})$ as in Eq. (27).

3: Convergence: the algorithm stops when,

$$\epsilon = \frac{1}{N_{\mathcal{V}^{(y)}}} \sum_{y_{(i,j)} \in \mathcal{V}^{(y)}} \left[ \frac{1}{N_{\Gamma(y_{(i,j)})}} \sum_{y_{(k,l)} \in \Gamma(y_{(i,j)})} \left( v_{y_{(k,l)}}(y_{(i,j)})^{(t)} - v_{y_{(k,l)}}(y_{(i,j)})^{(t-1)} \right)^2 \right] < \delta, \tag{30}$$

where $v_{y_{(k,l)}}(y_{(i,j)})^{(t)}$ denotes the estimated variance of the message from the neighboring node $y_{(k,l)}$ of $y_{(i,j)}$ to node $y_{(i,j)}$, which can be calculated as $v_{y_{(k,l)}}(y_{(i,j)})^{(t)} = \sum_{m=1}^{M} \left( \omega^{(m)}_{y_{(k,l)} \to y_{(i,j)}} \sigma^{(m)}_{y_{(k,l)} \to y_{(i,j)}} \right)^2$ (see Appendix D for the proof), where $\omega^{(m)}_{y_{(k,l)} \to y_{(i,j)}}$ and $\sigma^{(m)}_{y_{(k,l)} \to y_{(i,j)}}$ are the weight and variance for component $(m)$ in the representation of message $m^{(t)}_{y_{(k,l)}}(y_{(i,j)}) = \sum_{m=1}^{M} \omega^{(m)}_{y_{(k,l)} \to y_{(i,j)}} \mathcal{N}\left( \mu^{(m)}_{y_{(k,l)} \to y_{(i,j)}}, (\sigma^{(m)}_{y_{(k,l)} \to y_{(i,j)}})^2 \right)$. Notice that in this work, all the messages are normalized after being updated. The subscript $t$ denotes the step number.

the NBP algorithm to perform the inference problem. Specifically, we use the NBP algorithm to approximately find the update of Eqs. (25), (26) and (27) (corresponding to Steps 2.1, 2.2, and 2.4 in Algorithm 1, respectively). In the following, we clearly demonstrate how to use the NBP algorithm to compute the message update in Eq. (25). The message update in Eqs. (26) and (27) is similar.

The NBP algorithm approximates the belief update Eq. (25) using a sampling method [14]. It circumvents sampling directly from Eq. (25) (which is rather difficult task) by decomposing the process into two steps. In the first step, we draw $N$ independent samples $\widetilde{y}_{(k,l)}^{(n)}$ from a partial belief estimate combining the marginal influence function of the pairwise potential function $\psi(y_{(i,j)}, y_{(k,l)})$ on $y_{(k,l)}$, cross potential $\varphi_{(k,l)}(y_{(k,l)}, s_{(k,l)})$, and all the other incoming messages (Eq. (32)). The marginal influence function $\zeta(y_{(k,l)})$ is defined by

$$\zeta(y_{(k,l)}) = \int \psi(y_{(i,j)}, y_{(k,l)}) dy_{(i,j)}. \tag{31}$$

In this work, $\psi(y_{(i,j)}, y_{(k,l)})$ is a Gaussian mixture, so $\zeta(y_{(k,l)})$ is simply the Gaussian mixture obtained by marginalizing each component.

In the second step, for each of these auxiliary particles $\widetilde{y}_{(k,l)}^{(n)}$, we make samples $\widetilde{y}_{(i,j)}^{(n)}$ from the normalized conditional potentials proportional to $\psi(y_{(i,j)}, y_{(k,l)} = \widetilde{y}_{(k,l)}^{(n)})$ (Eq. (33)). The detailed algorithm of NBP is summarized in Algorithm 2.

**Remark 6:** Since we are using a Gaussian mixture to represent all messages, an inevitable problem will arise with the increase of the number of mixture components. For example, assume that all the messages are $M$-component Gaussian mixtures, and the BP belief update of Eq. (25) is defined by a product of $h$ mixtures. The product of $h$ Gaussian mixtures, each containing $M$ components, is itself a mixture of $M^h$ Gaussian distributions. While in principle this belief update could be performed exactly, the exponential growth in the number of mixture components quickly becomes intractable. Therefore, in this work, we use Gaussian mixture reduction via clustering (GMRC) method to reduce the number of mixture components whenever the number of mixture components of the beliefs exceed $M$. The details of GMRC algorithm are given in Appendix E.

**Algorithm 2** The detailed algorithm for nonparametric belief propagation

**Given:** Input message $m_{y_{(p,q)}}(y_{(k,l)}) = \{\omega^{(m)}_{y_{(p,q)}\to y_{(k,l)}}, \mu^{(m)}_{y_{(p,q)}\to y_{(k,l)}}, \Sigma^{(m)}_{y_{(p,q)}\to y_{(k,l)}}\}_{m=1}^{M}$ for each $y_{(p,q)} \in \Gamma(y_{(k,l)}) \setminus y_{(i,j)}$.

**Objective:** Construct an output message $m_{y_{(k,l)}}(y_{(i,j)})$.

1: Determine the marginal influence $\zeta(y_{(k,l)})$ by Eq. (31).

2: Draw $N$ independent, weighted samples from the product,

$$\widetilde{y}^{(n)}_{(k,l)} \sim \zeta(y_{(k,l)})\varphi_{(k,l)}(y_{(k,l)}, s_{(k,l)}) \prod_{y_{(p,q)}\in\Gamma(y_{(k,l)})\setminus y_{(i,j)}} m_{y_{(p,q)}}(y_{(k,l)}). \tag{32}$$

GMRC (A Gaussian mixture reduction technique discussed in Appendix E) is first adopted to reduce the components of the product and then exact sampling method [35] is applied.

3: For each $\widetilde{y}^{(n)}_{(k,l)}$, sample from,

$$\widetilde{y}^{(n)}_{(i,j)} \sim \psi(y_{(i,j)}, y_{(k,l)} = \widetilde{y}^{(n)}_{(k,l)}). \tag{33}$$

4: Construct $m_{y_{(k,l)}}(y_{(i,j)})$ from $\widetilde{y}^{(n)}_{(i,j)}$ by taking $\widetilde{y}^{(n)}_{(i,j)}$ as realizations of message $m_{(k,l)}(y_{(i,j)})$. Specifically, assume $m_{y_{(k,l)}}(y_{(i,j)}) \propto \sum_{m=1}^{M} \omega^{(m)}_{y_{(k,l)}\to y_{(i,j)}} \mathcal{N}(\mu^{(m)}_{y_{(k,l)}\to y_{(i,j)}}, \Sigma^{(m)}_{y_{(k,l)}\to y_{(i,j)}})$, where the unknowns are $\{\omega^{(m)}_{y_{(k,l)}\to y_{(i,j)}}, \mu^{(m)}_{y_{(k,l)}\to y_{(i,j)}}, \Sigma^{(m)}_{y_{(k,l)}\to y_{(i,j)}}\}_{m=1}^{M}$. These can be learned using the EM algorithm discussed in Appendix B by taking $\widetilde{y}^{(n)}_{(i,j)}$ as training samples.

## 6. Numerical examples

In this paper, we construct a probabilistic graphical model to study two-dimensional, single phase, steady-state fluid flow through random heterogeneous porous media. A review of the mathematical models of flow through porous media can be found in [36]. The spatial domain $D$ is chosen to be the unit square $[0,1]^2$, representing an idealized oil reservoir. Let us denote with $p$ and $\mathbf{u}$ the pressure and the velocity fields of the fluid, respectively. These are connected via the Darcy law:

$$\mathbf{u} = -\mathbf{K}\nabla p, \text{in } D, \tag{34}$$

where $\mathbf{K}$ is the permeability tensor that models the easiness with which the liquid flows through the reservoir. Combining the Darcy law with the continuity equation, it is easy to show that the governing PDE for the pressure is:

$$-\nabla \cdot (\mathbf{K}\nabla p) = f, \text{in } D, \tag{35}$$

where the source term $f$ may be used to model injection/production wells. In this example, we consider square wells: an injection well on the left-bottom corner of $D$ and a production well on the top-right corner. The particular mathematical form of the source term $f$ is as follows:

$$f(\mathbf{x}) = \begin{cases} -r, & \text{if } |x_i - \frac{1}{2}w| < \frac{1}{2}w, \text{ for } i = 1,2, \\ r, & \text{if } |x_i - 1 + \frac{1}{2}w| < \frac{1}{2}w, \text{ for } i = 1,2, \\ 0, & \text{otherwise,} \end{cases} \tag{36}$$

where $r$ specifies the rate of the wells, $w$ their size (chosen here to be $r = 10$ and $w = 1/8$), and $\mathbf{x} = (x_1, x_2) \in D$. Furthermore, we impose no-flux boundary conditions on the walls of the reservoir:

$$\mathbf{u} \cdot \tilde{\mathbf{n}} = 0, \text{on } \partial D, \tag{37}$$

where $\tilde{\mathbf{n}}$ is the unit normal vector to the boundary. These boundary conditions specify the pressure $p$ up to an additive constant. To assure uniqueness of the boundary value problem defined by Eqs. (34), (35) and (37), we impose the constraint [37]:

$$\int_D p(\mathbf{x}) d\mathbf{x} = 0.$$

The boundary value problem is solved using a mixed finite element formulation. We use first-order Raviart-Thomas elements for the velocity [38], and zero-order discontinuous elements for the pressure [39]. The permeability is defined on a $64 \times 64$ fine-grid and we are interested in the physical responses on a $8 \times 8$ coarse-grid. The solver was implemented using the Dolfin C++ library [40]. The eigenfunctions of the exponential random field used to model the permeability were calculated via Stokhos which is part of Trilinos [41].

In this work, the final responses taken into consideration include $x-$velocity, $u_x$, $y-$velocity, $u_y$ and pressure, $p$. We assume independence of the multiple output responses so we can build an independent graphical model for each of them. This is typical of many uncertainty quantification methods but methodologies where correlations are accounted can be considered as well [15]. As previously discussed, the constructed graphical model is a non-parametric model. Then naturally an important question arises as to what the proper number is of the mixture components considered. One should avoid to choose a large number due to the exponential increase for the computational cost, especially at the loop belief propagation step. A large number of mixture components does not necessarily lead to better results and for some cases may lead to over-fitting. In the context of the examples presented below, three mixture components were shown to provide an adequate choice for the accuracy desired.

### *6.1. Stationary random field*

In this example, the log-permeability is considered as a stationary random field. We restrict ourselves to an isotropic permeability tensor:

$$K_{ij} = K\delta_{ij}.$$

$K$ is modeled as

$$K(\mathbf{x}) = \exp\{G(\mathbf{x})\},$$

where $G$ is a Gaussian random field:

$$G(\cdot) \sim \mathcal{N}(m, c_G(\cdot,\cdot)),$$

with constant mean $m$ and an exponential covariance function given by

$$c_G(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = v_G^2 \exp\left\{ -\frac{|x_1^{(1)} - x_1^{(2)}|}{l} - \frac{|x_2^{(1)} - x_2^{(2)}|}{l} \right\}. \tag{38}$$

The parameter $l$ represents the correlation lengths of the field, while $v_G > 0$ is its variance. In order to obtain a finite dimensional representation of $G$, we employ the Karhunen-Loève expansion [16] and truncate it after $k_\xi$ terms:

$$G(\boldsymbol{\xi}; \mathbf{x}) = m + \sum_{k=1}^{k_\xi} \xi_k \phi_k(\mathbf{x}), \tag{39}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{k_\xi})$ is a vector of independent, zero mean and unit variance Gaussian random variables and $\phi_k(\mathbf{x})$ are the eigenfunctions of the exponential covariance given in Eq. (38) (suitably normalized).

The values we choose for the parameters are $m = 0, l = 0.1$ and $v_G = 1$ in Eq. (38), and $k_\xi = 50$ in Eq. (39). In the following, we first verify the model reduction framework in Section 6.1.1 and then we move to the inference tasks on the graph: 1) given the input distribution of $\boldsymbol{\xi}$, investigate how the uncertainty propagates to the response in Section 6.1.2; 2) given a new permeability field, find the prediction of unobserved responses with proper error bars in Section 6.1.3.

*6.1.1. Model reduction*

As discussed in Section 3, PCA model reduction technique is applied to reduce the dimensionality of the input permeability field. Fig. 7 shows the normalized eigen-plot and energy-plot for the PCA reduction for the input permeability over the corresponding coarse-elements to two random coarse-nodes. Here, "normalized" means that each eigenvalue is divided by the sum of all the eigenvalues. As shown on these plots, by using less than ten eigenvectors, the cumulative preserved energy is almost one, which means a ten-dimensional random variable representation is enough to describe the original data set. In addition, we compare the reconstructed input permeability field with the original one (the dimension of the original permeability field is $64 \times 64 = 4096$) with different $k$, where $k$ is the dimensionality of the reduced space, as shown in Figs. 8 and 9. The major differences occur along the coarse-grid boundaries. This is indeed expected by using the reconstruction strategy discussed in Section 3. It is clear from these figures that as the reduced dimensionality $k$ increases, the reconstructed permeability field gets closer to the original microstructure. Also as the number of training data $N$ used increases, the reconstructed permeability becomes closer to the original realization.

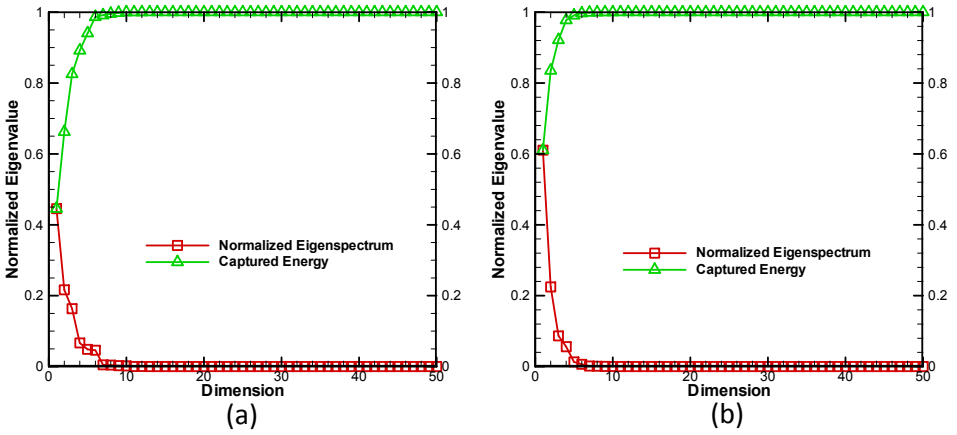


Figure 7: Stationary random field - Normalized eigenspectrum and energy plot for the input permeability in two random subdomains.

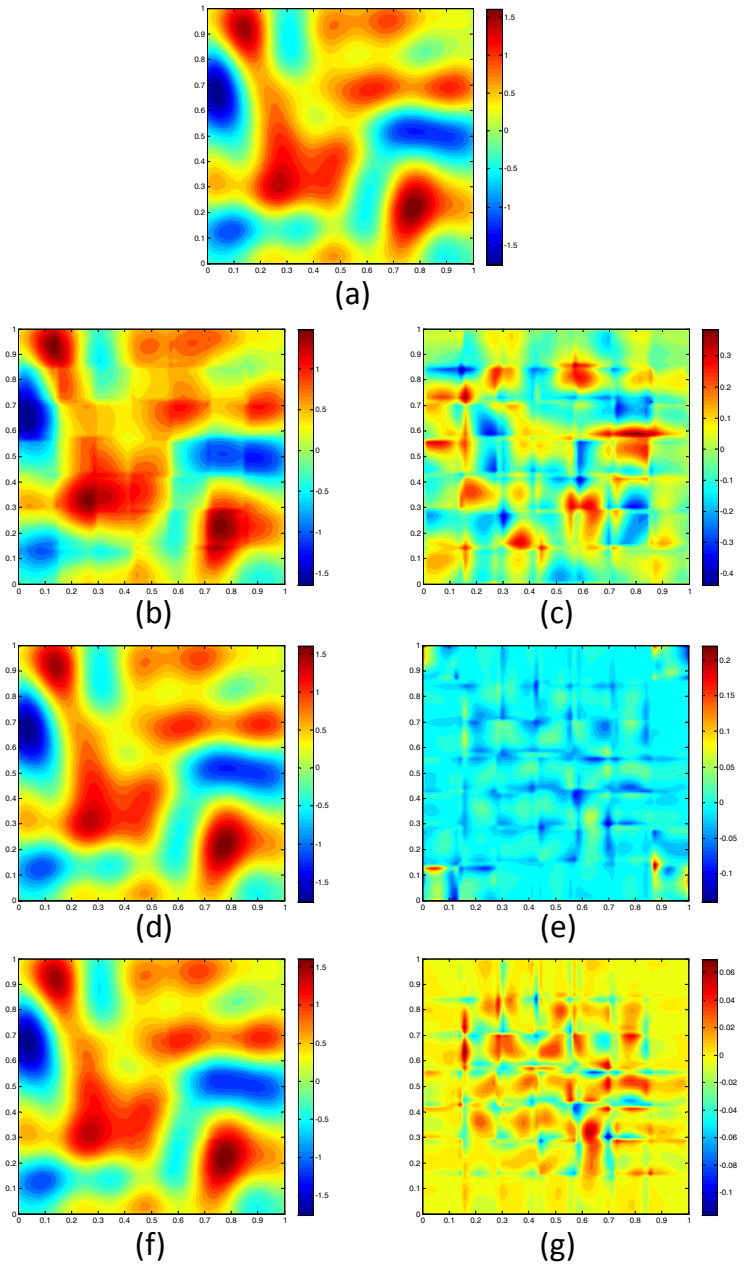


Figure 8: Stationary random field - Comparison of the reconstructed input permeability field with the original given sample (a) with different number of training data for $k = 10$, where $k$ is the dimensionality of the reduced space; (b)(d)(f) The reconstructed input permeability using $N = 200, 1000$, and 4000 training data, respectively; (c)(e)(g) The error between the reconstructed permeability field and the original sample.

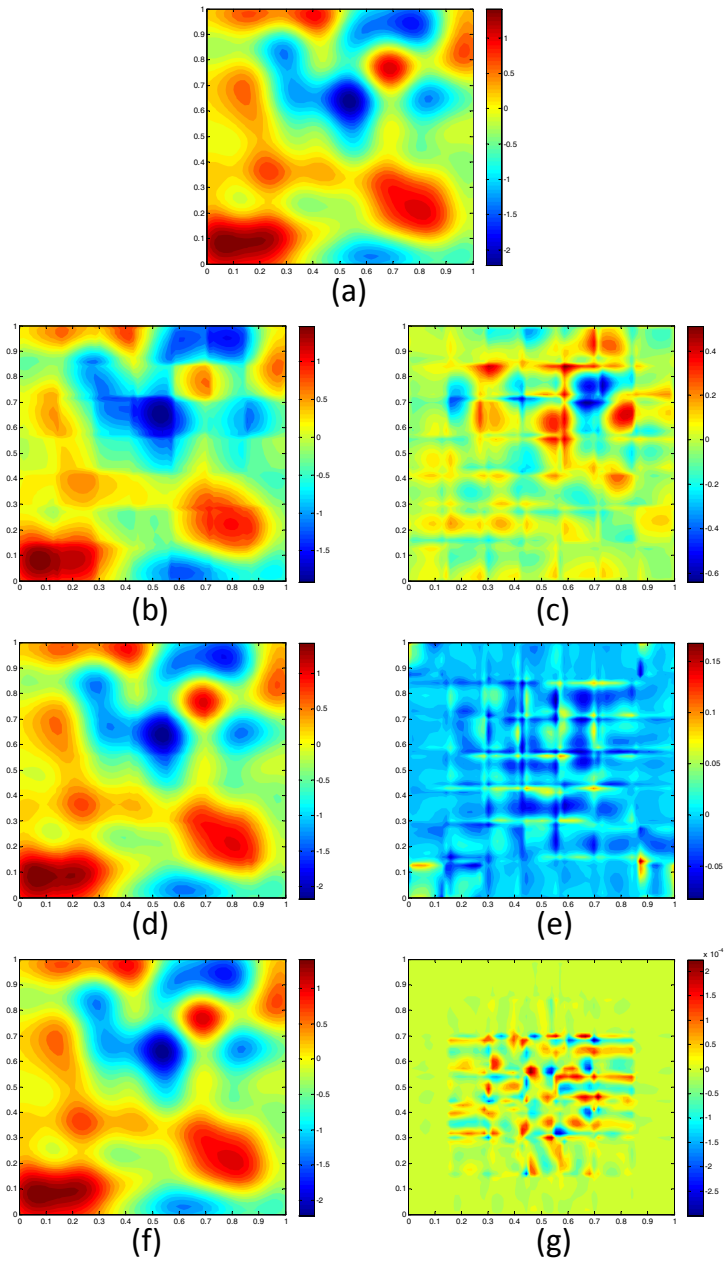


Figure 9: Stationary random field - Comparison of the reconstructed input permeability field with the original given sample (a) with different $k$ for $N = 1000$; (b)(d)(f) The reconstructed input permeability using $k = 5, 10$, and 30, respectively; (c)(e)(g) The error between the reconstructed permeability field and the original sample.

Finally, we compare the reconstruction error of the input permeability field with different number of training data and different reduced dimensionality $k$ (Fig. 10). The reconstruction error is computed by

$$e = \frac{1}{N_g N} \sum_{i=1}^{N} \|\mathbf{A}^{(i)} - \widetilde{\mathbf{A}}^{(i)}\|^2, \tag{40}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{N_f}]$, $N_f$ is the number of elements on the fine-mesh and $\widetilde{\mathbf{A}} = [\widetilde{\mathbf{a}}_1, \widetilde{\mathbf{a}}_2, \ldots, \widetilde{\mathbf{a}}_{N_f}]$ where $\widetilde{\mathbf{a}}_i$ is the reconstructed permeability on the fine-element $e_i$. The superscript $(i)$ denotes the $i-th$ sample and $N$ is the total number of samples used. The given figure indicates that the number of reduced dimensionality $k$ has a higher impact on the final performance of the reconstruction than the number of training data. The reduced dimensionality $k$ is chosen as 10 in this problem.

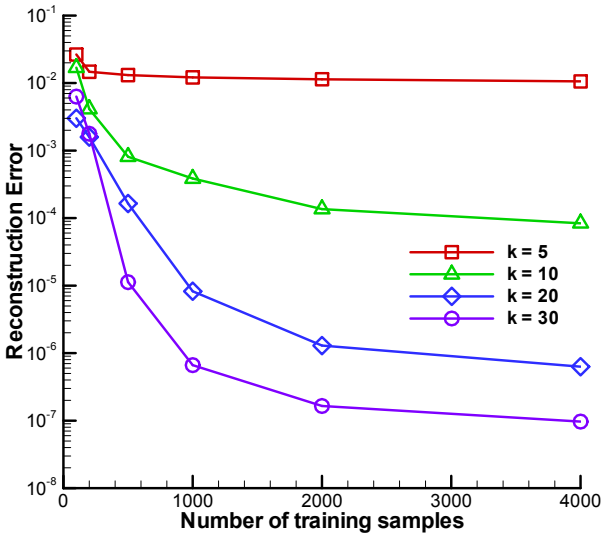


Figure 10: Stationary random field - Comparison of the reconstruction error of the input permeability field with different number of training data, and different reduced dimensionality $k$.

*6.1.2. Uncertainty propagation*

In this section, we are going to investigate how the uncertainties propagate from the input permeability to the output (velocity and pressure) response. After the graphical model is completely learnt by the training data, we send

the distribution of $\mathbf{S}$, $p(\mathbf{S})$, which is calculated from the known input distribution $p(\boldsymbol{\xi})$, to the graphical model as the input message, and then run the nonparametric belief propagation algorithm. After all the messages in the graph converge, we compute the marginal distribution of the response variables by combining all the messages coming into the response variable as in Eq. (24).

Fig. 11 compares the predicted mean of $u_x$ with a Monte Carlo estimate using $10^5$ observations. We can clearly see that as the number of training data increases, the prediction gets more and more accurate. The same statistic for $u_y$ and $p$ is reported in Figs. 12 and 13, respectively.

Fig. 14 compares the predicted variance of $u_x$ to a Monte Carlo estimate using $10^5$ observations. Also, the predicted variance converges to the MC results with the increase of the number of the training data. The same statistic for $u_y$ and $p$ is given in Figs. 15 and 16, respectively. We can see that $N = 1000$ training samples can already give rather accurate predictions for the marginal mean and variance of the responses. Notice that in this work, the predicted marginal probability is given in a Gaussian mixture form with three components. For example, the response at one coarse-grid node, $y_{(i,j)}$ (random variable) can be represented as $y_{(i,j)} = \sum_{m=1}^{M} \omega_{(i,j)}^{(m)} y_{(i,j)}^{(m)}$, where $y_{(i,j)}^{(m)} \sim \mathcal{N}(\mu_{(i,j)}^{(m)}, (\sigma_{(i,j)}^{(m)})^2)$. As discussed in Appendix D, the first-order and second-order statistics can be obtained by $\mathbb{E}[y_{(i,j)}] = \sum_{m=1}^{M} \omega_{(i,j)}^{(m)} \mu_{(i,j)}^{(m)}$ and $Var[y_{(i,j)}] = \sum_{m=1}^{M} (\omega_{(i,j)}^{(m)})^2 (\sigma_{(i,j)}^{(m)})^2$, respectively.

The error of the statistics is evaluated using the (normalized) $L_2$ norm of the error in variance defined by:

$$E_{L_2} = \sqrt{\frac{1}{N_G} \sum_{i=1}^{N_G} (v_{i,MC} - \widetilde{v}_i)^2}, \tag{41}$$

where $v_{i,MC}$ is the Monte Carlo estimate of the variance of the response on the $i$-th coarse-node using $10^5$ samples, and $\widetilde{v}_i$ is the predictive variance given by the graphical model. In Fig. (17), we plot the $L_2$ norm of the error as a function of the number of samples for $u_x$, $u_y$ and $p$ and a comparison with the MC results is shown. In addition, we compare the predicted probability densities of $u_x$, $u_y$ and $p$ at physical positions $(0.429, 0.429)$ and $(0.571, 0.571)$, with the PDFs obtained from the MC estimate using $10^5$ observations, as shown in Figs. 18 and Fig. 19, respectively. From the figures, we can see that the PDFs do not have symmetric tails, so obviously, they are not Gaussian

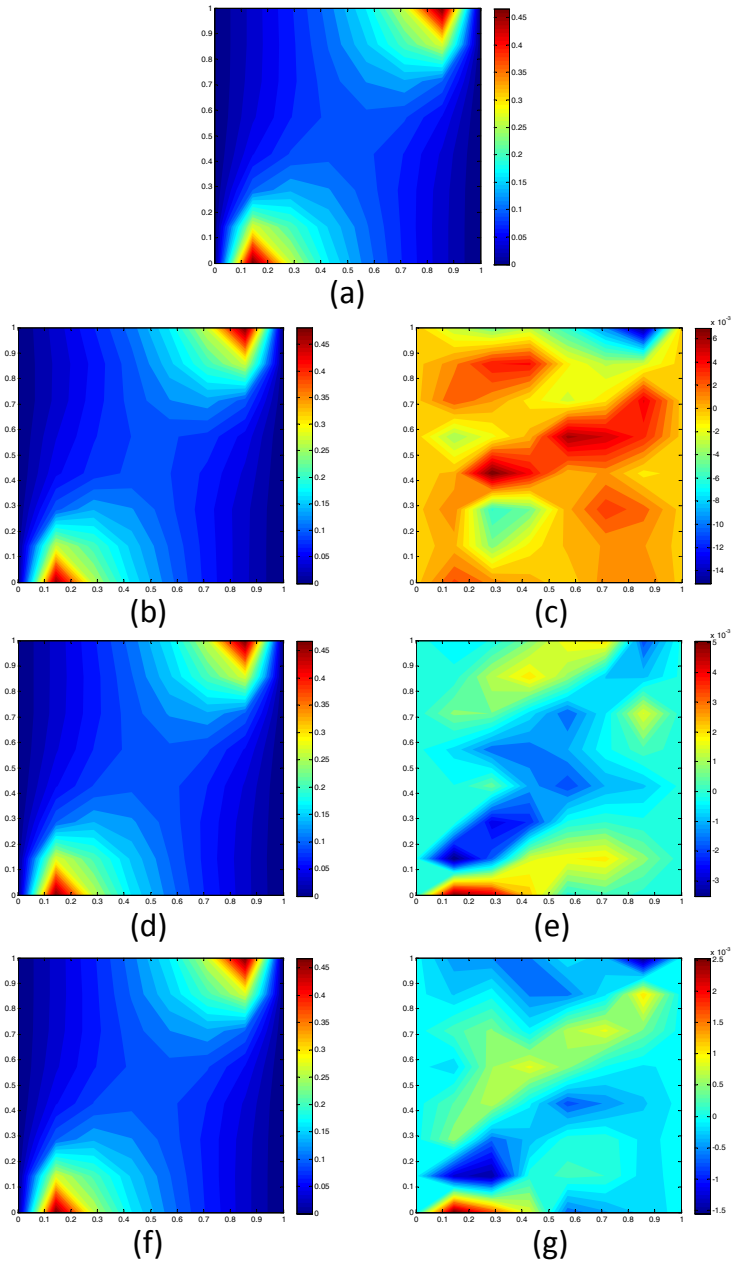


Figure 11: Stationary random field - Mean of $u_x$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted mean of $u_x$ using $100, 400$, and $1000$ training samples, respectively; (c)(e)(g) The error between the predicted mean and the MC mean for $N = 100, 400$, and $1000$, respectively.

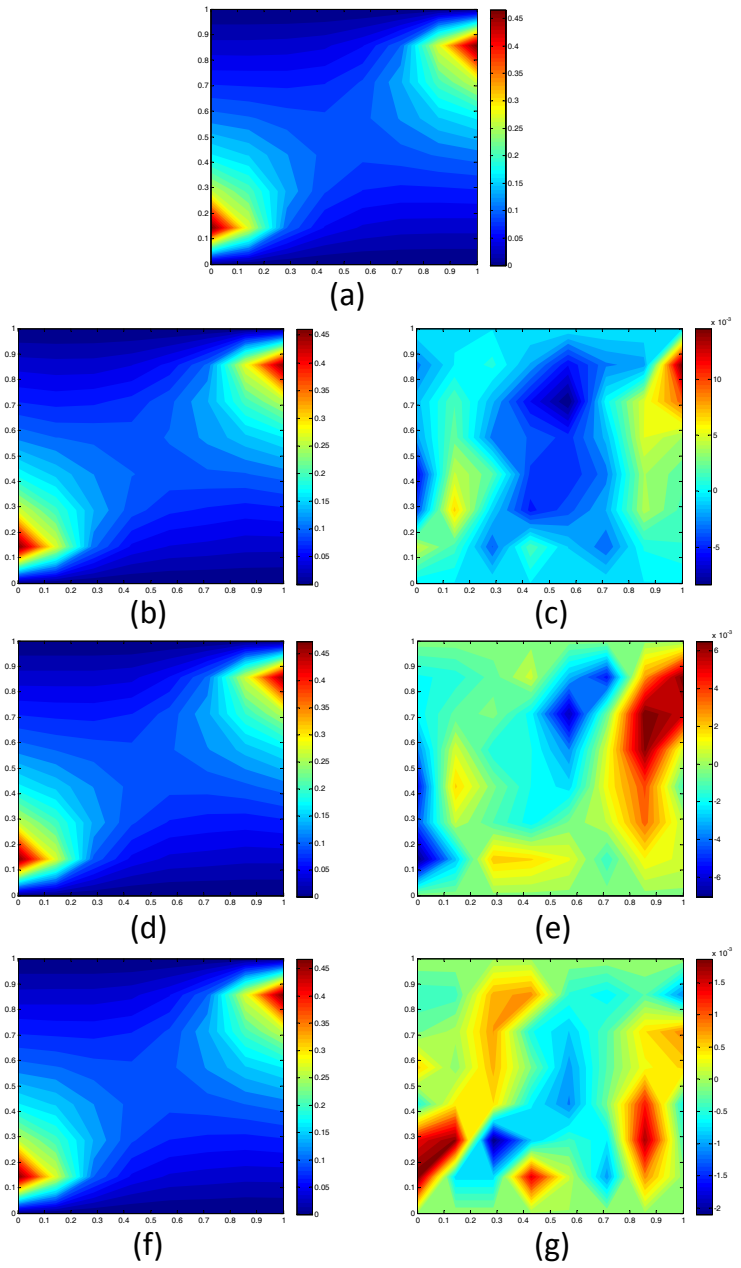


Figure 12: Stationary random field - Mean of $u_y$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted mean of $u_y$ using $100, 400$, and $1000$ training samples, respectively; (c)(e)(g) The error between the predicted mean and the MC mean for $N = 100, 400$, and $1000$, respectively.

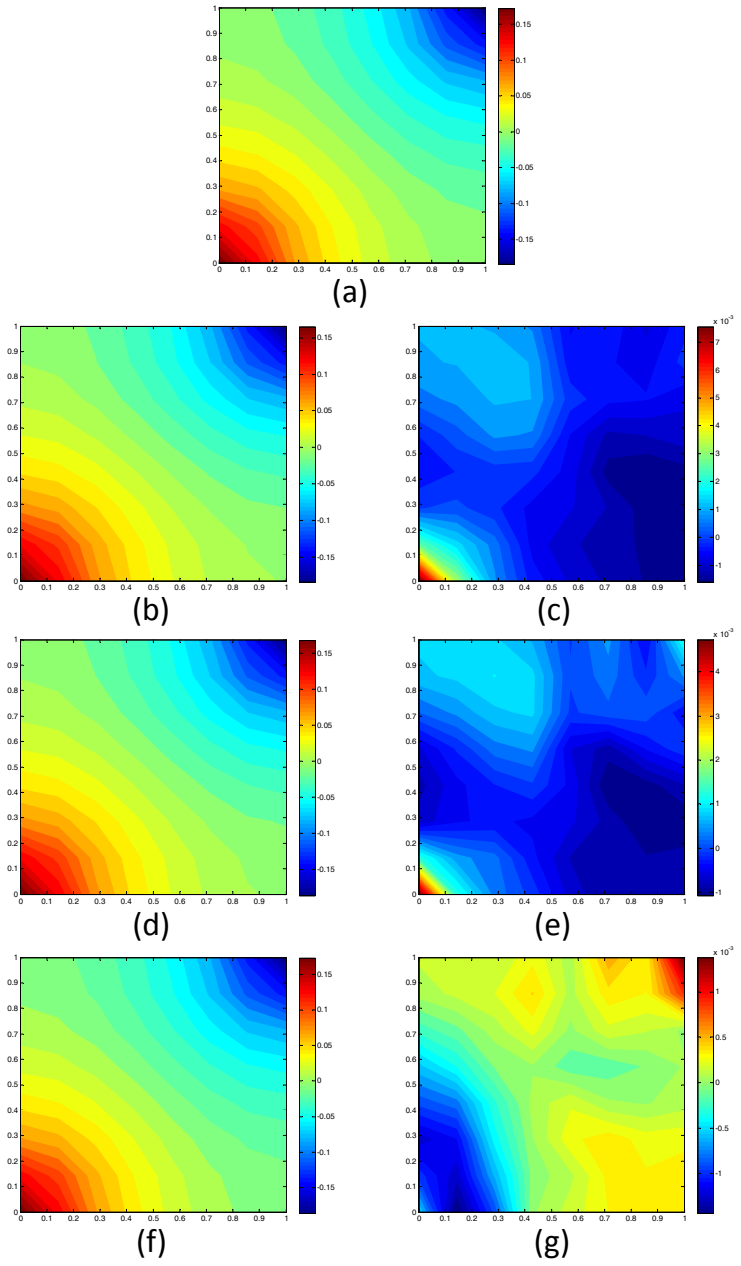


Figure 13: Stationary random field - Mean of $p$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted mean of $p$ using $100, 400$, and $1000$ training samples, respectively; (c)(e)(g) The error between the predicted mean and the MC mean for $N = 100, 400$, and $1000$, respectively.

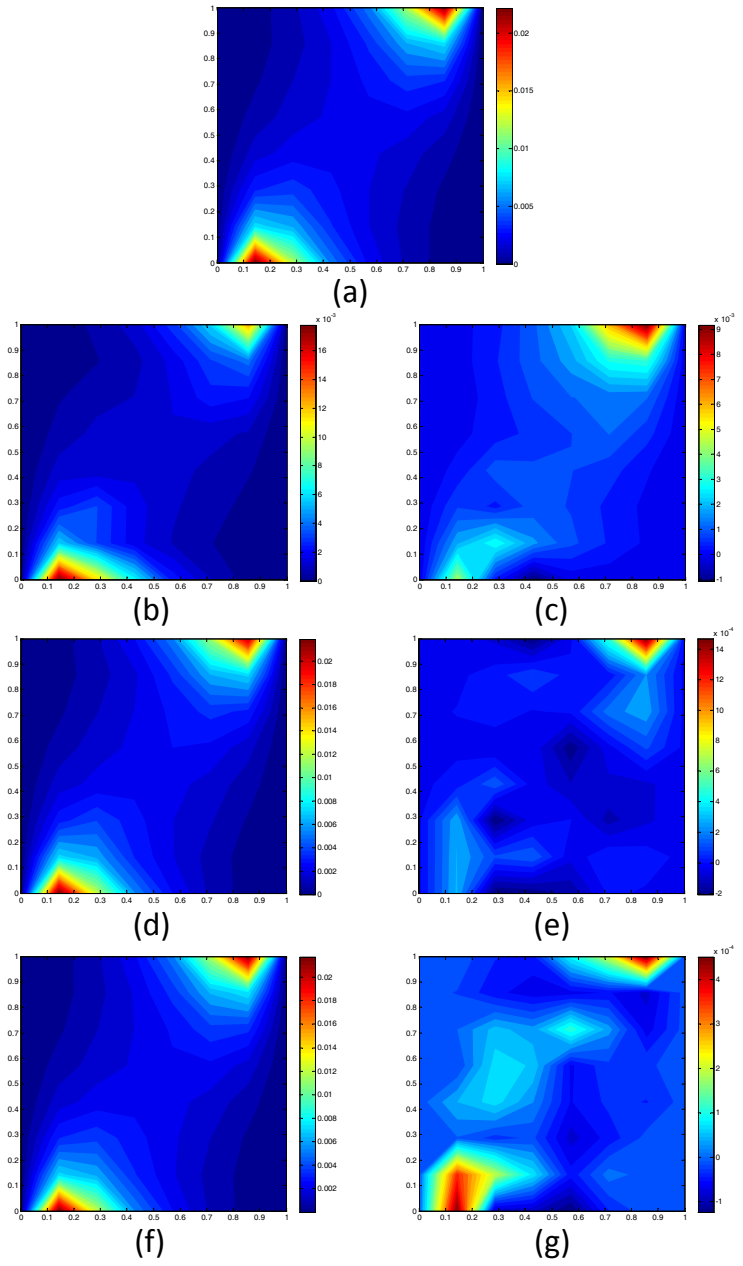


Figure 14: Stationary random field - Variance of $u_x$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted variance of $u_x$ using $100, 400$, and $1000$ training samples, respectively; (c)(e)(g) The error between the predicted variance and the MC variance for $N = 100, 400$, and $1000$, respectively.

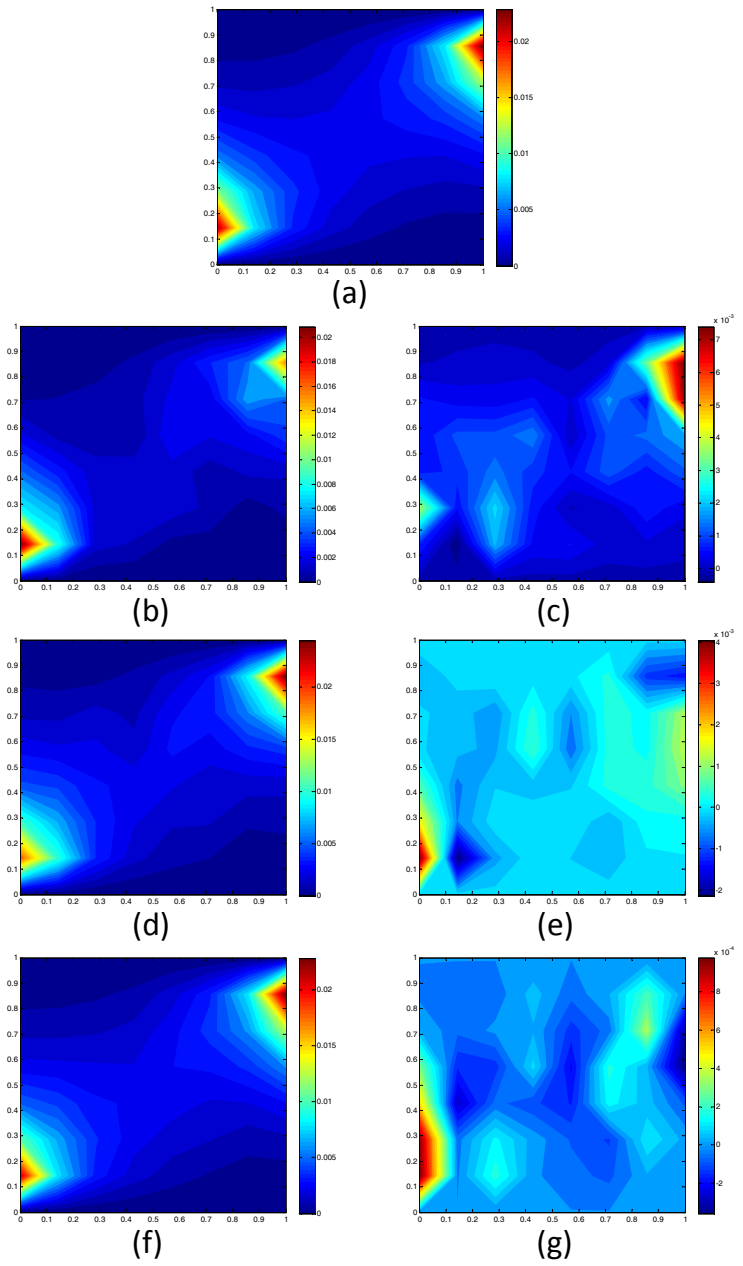


Figure 15: Stationary random field - Variance of $u_y$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted variance of $u_y$ using $100, 400$, and $1000$ training samples, respectively; (c)(e)(g) The error between the predicted variance and the MC variance for $N = 100, 400$, and $1000$, respectively.

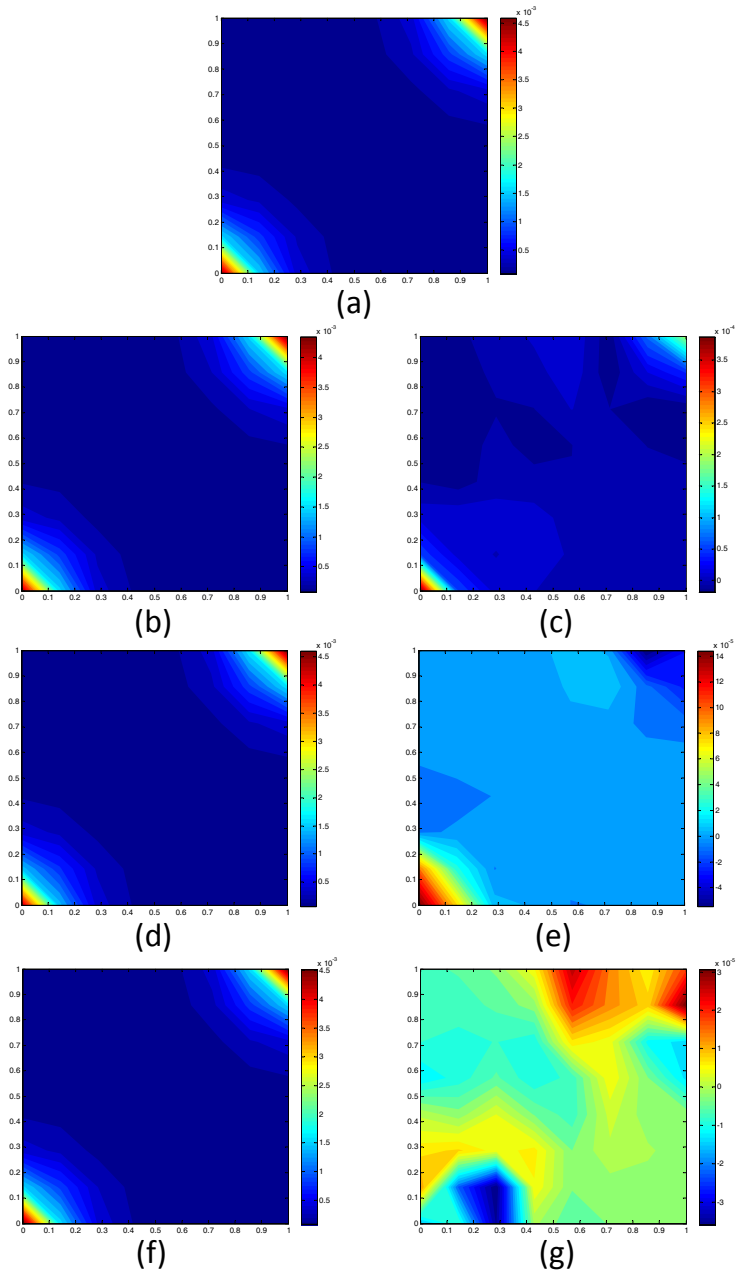


Figure 16: Stationary random field - Variance of $p$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted variance of $p$ using $100, 400$, and $1000$ training samples, respectively; (c)(e)(g) The error between the predicted variance and the MC variance for $N = 100, 400$, and $1000$, respectively.

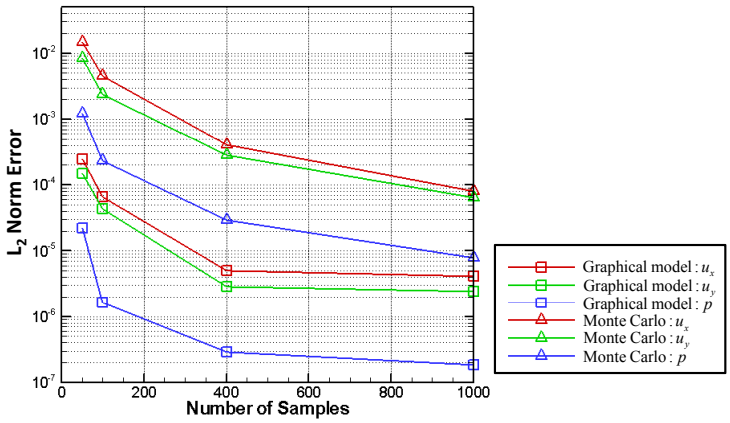


Figure 17: Stationary random field - The $L_2$ norm of the error as a function of the number of samples observed for graphical model framework.

distributions. This is especially true for the velocity components that should be positive. As the number of observations increases, we can observe that the graphical model prediction gradually captures the major key features of the PDFs.

*6.1.3. Response Prediction*

In this section, we will show that the constructed graphical model is also capable of acting as a surrogate model of the deterministic solver. The problem can be described as follows: Given a new observation of the permeability field, $\mathbf{a}$, the objective is to obtain the conditional distribution $p(\mathbf{Y}|\mathbf{A} = \mathbf{a})$. With a new realization of the permeability field $\mathbf{a}$, we first compute the localized reduced input variables $\mathbf{S}(\mathbf{a})$. After that, instead of using the distribution of $\mathbf{S}$, $p(\mathbf{S})$, we use a Kronecker Delta function $\delta_{\mathbf{S}}(\mathbf{S}(\mathbf{a}))$ as the input message, and we send it to the pre-learned graphical model to execute the nonparametric belief propagation algorithm. Notice that now, all the potential functions involving the $\mathbf{S}$ variable need to be multiplied by the Kronecker Delta function $\delta_{\mathbf{S}}(\mathbf{S}(\mathbf{a}))$, as discussed in Section 5.2.1.

Figs. 20, 21, and 22 show a comparison of the predicted $u_x$, $u_y$ and $p$ fields, respectively, with the results of the deterministic solver for given a new input permeability field $\mathbf{a}$. This permeability sample was generated from the same process as the training data. The predictions are given by the nonparametric model using different number of training data. As the number of training data ($N$) increases, the predictions gradually converge to the true response

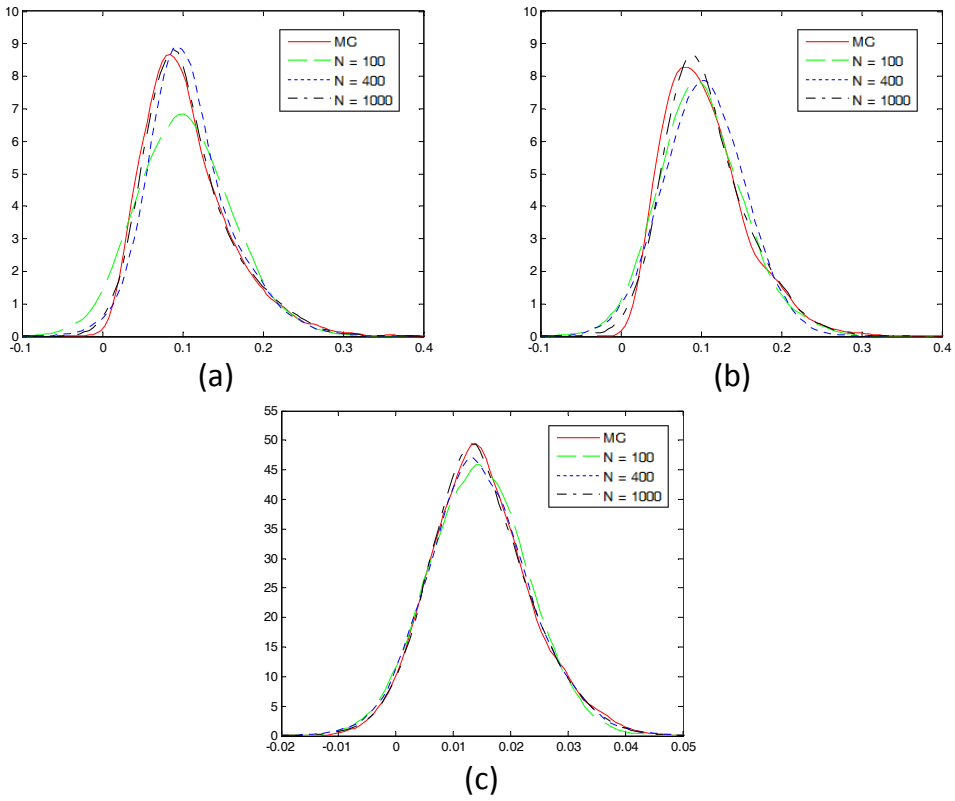


Figure 18: Stationary random field - Comparison of the predicted PDFs using different training data with the MC estimate at physical position ($(0.429, 0.429)$): (a) $u_x$, (b) $u_y$, (c) $p$.

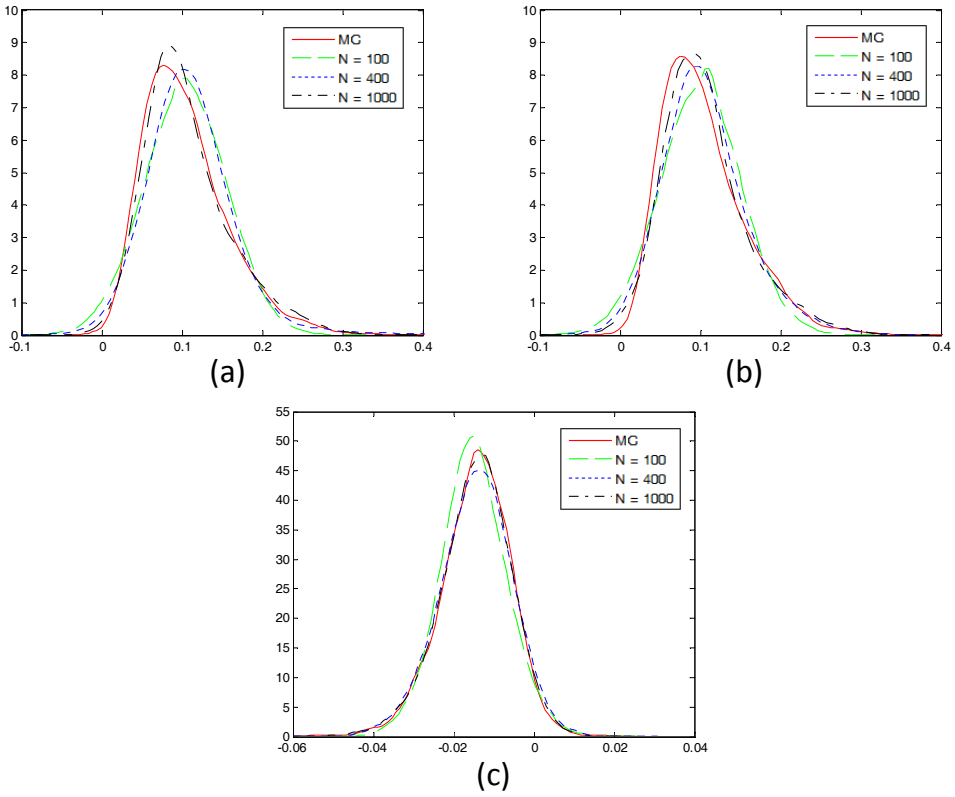


Figure 19: Stationary random field - Comparison of the predicted PDFs using different training data with the MC estimate at physical position $(0.571, 0.571)$: (a) $u_x$, (b) $u_y$, (c) $p$.

fields. Note that in the learning process, the unknown parameters in the potential functions converge after some $N$ and using more data does not improve considerably the final predictions.

*6.2. Non-stationary random field*

In the previous example, it was assumed that the permeability field considered was stationary such that the covariance between any two points in the domain depends on their distance rather than their actual locations. However, hydraulic properties may exhibit spatial variations at various scales. Therefore, it is important to extend the probabilistic graphical model to non-stationary random fields. In this example, we use a non-stationary random field as stochastic input. The log-permeability on the $k$-th coarse-element is still a Gaussian random field with mean zero and an exponential covariance function, as given in Eq. (38):

$$c_G(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = v_G^2 \exp\left\{ -\frac{|x_1^{(1)} - x_1^{(2)}|}{l_{k,1}} - \frac{|x_2^{(1)} - x_2^{(2)}|}{l_{k,2}} \right\}. \tag{42}$$

However, the correlation length in the non-stationary case is not a constant anymore. Since the coarse-grid has $N_x = 8$ rows and $N_y = 8$ columns of elements, we define the coordinate of the $k$-th element as $(i_k, j_k)$ where $i_k$ is the index in row and $j_k$ is the index in column. Then the correlation length is set to be $l_{k,1} = 0.1 + \frac{0.4}{N_y - 1} j_k$ and $l_{k,2} = 0.1 + \frac{0.4}{N_x - 1} i_k$. The source term $f$ is set to zero. Flow is induced from left to right side with Dirichlet boundary conditions $\bar{p} = 1$ on $x = 0$, $\bar{p} = 0$ on $y = 1$. No−flow Neumann boundary conditions are applied on the other two sides of the square domain.

In this example, we investigate the non-stationary problem in a similar way as the previous stationary case. First, we verify the model reduction framework in Section 6.2.1 and then we investigate the uncertainty propagation from the inputs to the responses in Section 6.2.2. Finally, we use the graphical model to predict the responses given a new permeability field, in Section 6.2.3.

*6.2.1. Model reduction*

In this section, we first compare the reconstructed input permeability field with the original one for different $k$, where $k$ is the dimensionality of the reduced space, as shown in Fig. 23. Fig. 24 shows the comparison of the

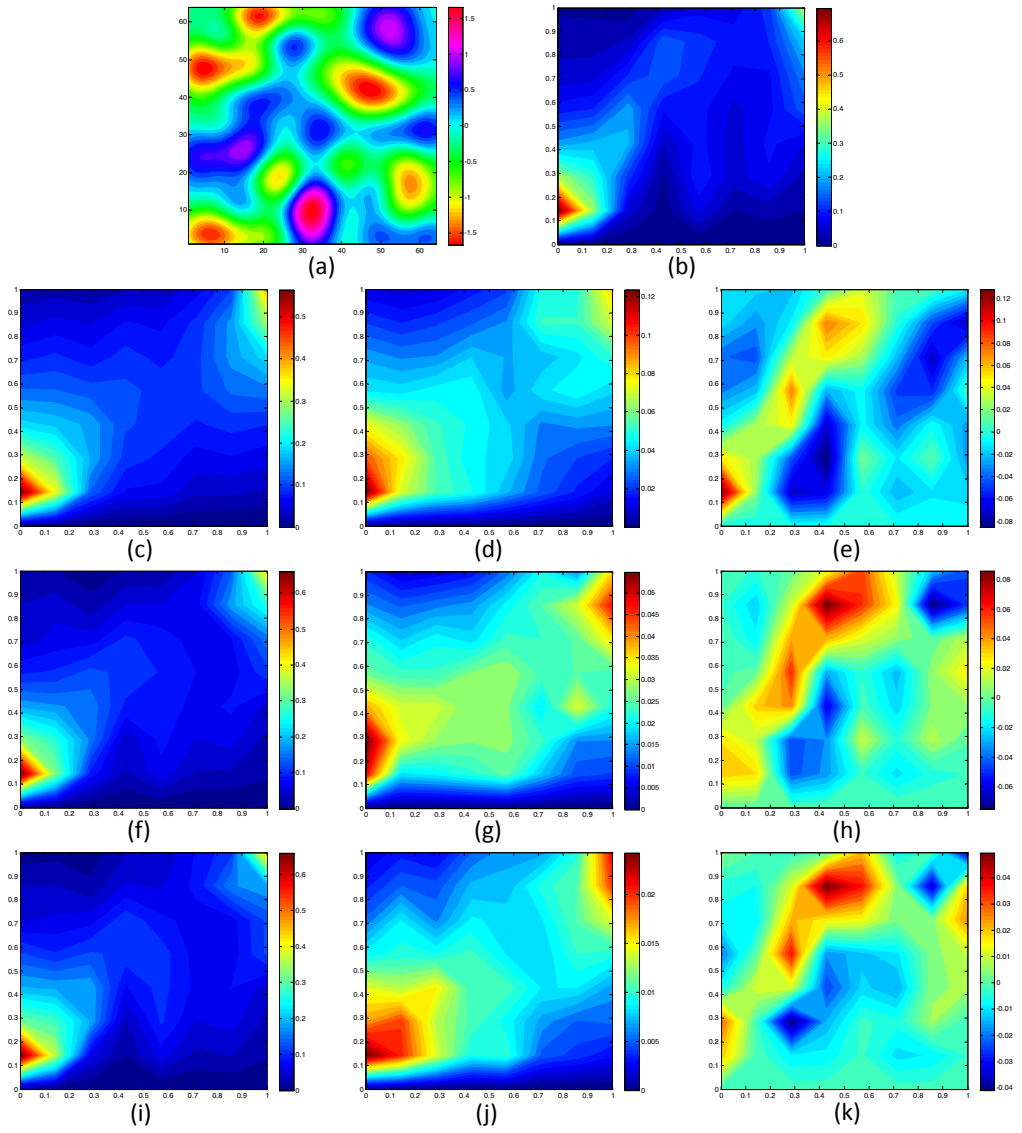


Figure 20: Stationary random field - Comparison of the prediction of $u_x$ given a new input permeability field using the nonparametric model versus the true $u_x$ from the deterministic solver: (a) The new observed input permeability field; (b) The contour plot of the true $u_x$ field; (c)(f)(i) The predicted mean by the nonparametric graphical model using $N = 400, 1000$ and 4000 training data, respectively; (d)(g)(j) The corresponding predictive variance for $N = 400, 1000$ and 4000 cases, respectively; (e)(h)(k) The error between the predicted mean and the true response field for $N = 400, 1000$ and 4000 cases, respectively.

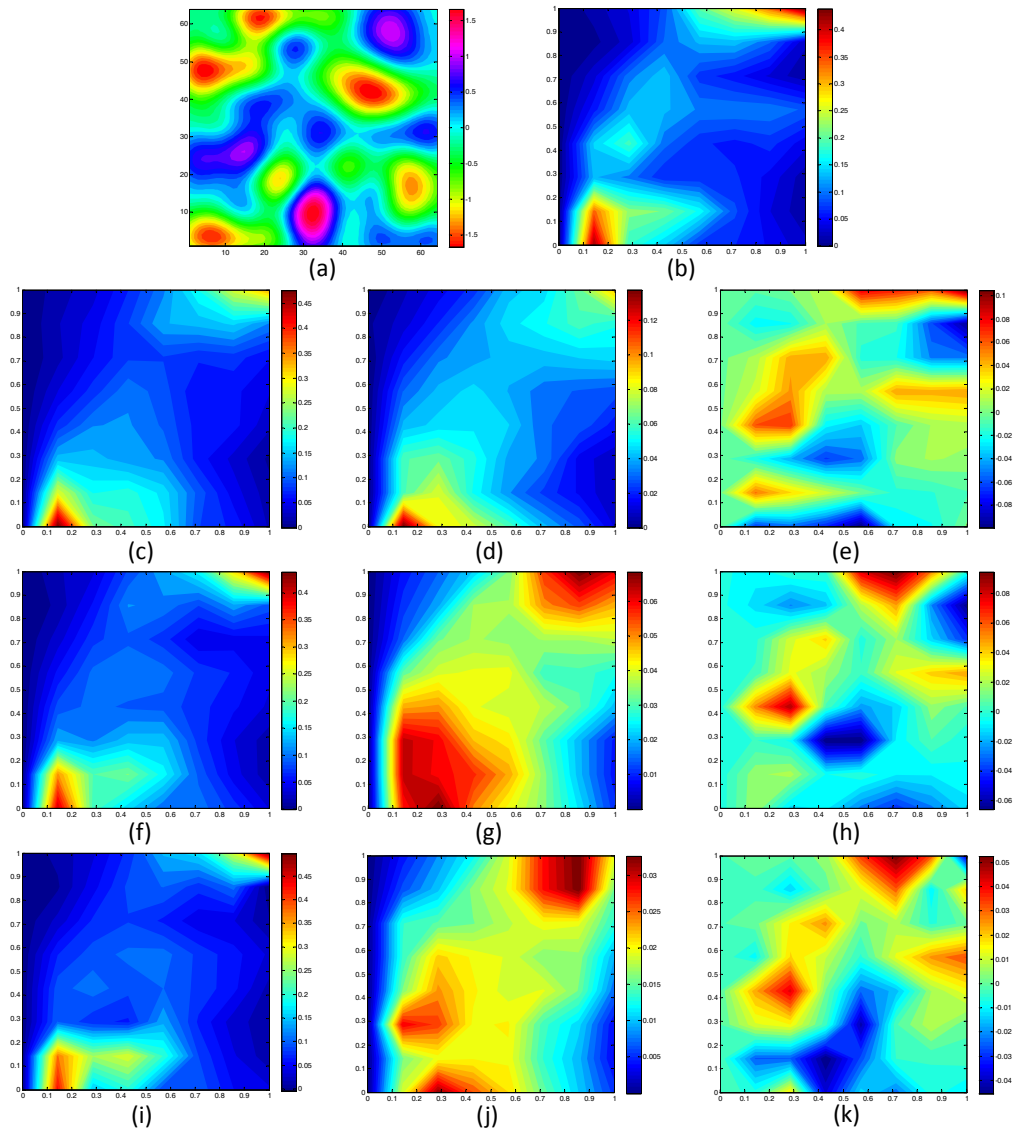


Figure 21: Stationary random field - Comparison of the prediction of $u_y$ given a new input permeability field using the nonparametric model versus the true $u_y$ from the deterministic solver: (a) The new observed input permeability field; (b) The contour plot of the true $u_y$ field; (c)(f)(i) The predicted mean by the nonparametric graphical model using $N = 400, 1000$ and 4000 training data, respectively; (d)(g)(j) The corresponding predictive variance for $N = 400, 1000$ and 4000 cases, respectively; (e)(h)(k) The difference between the predicted mean and the true response field for $N = 400, 1000$ and 4000 cases, respectively.

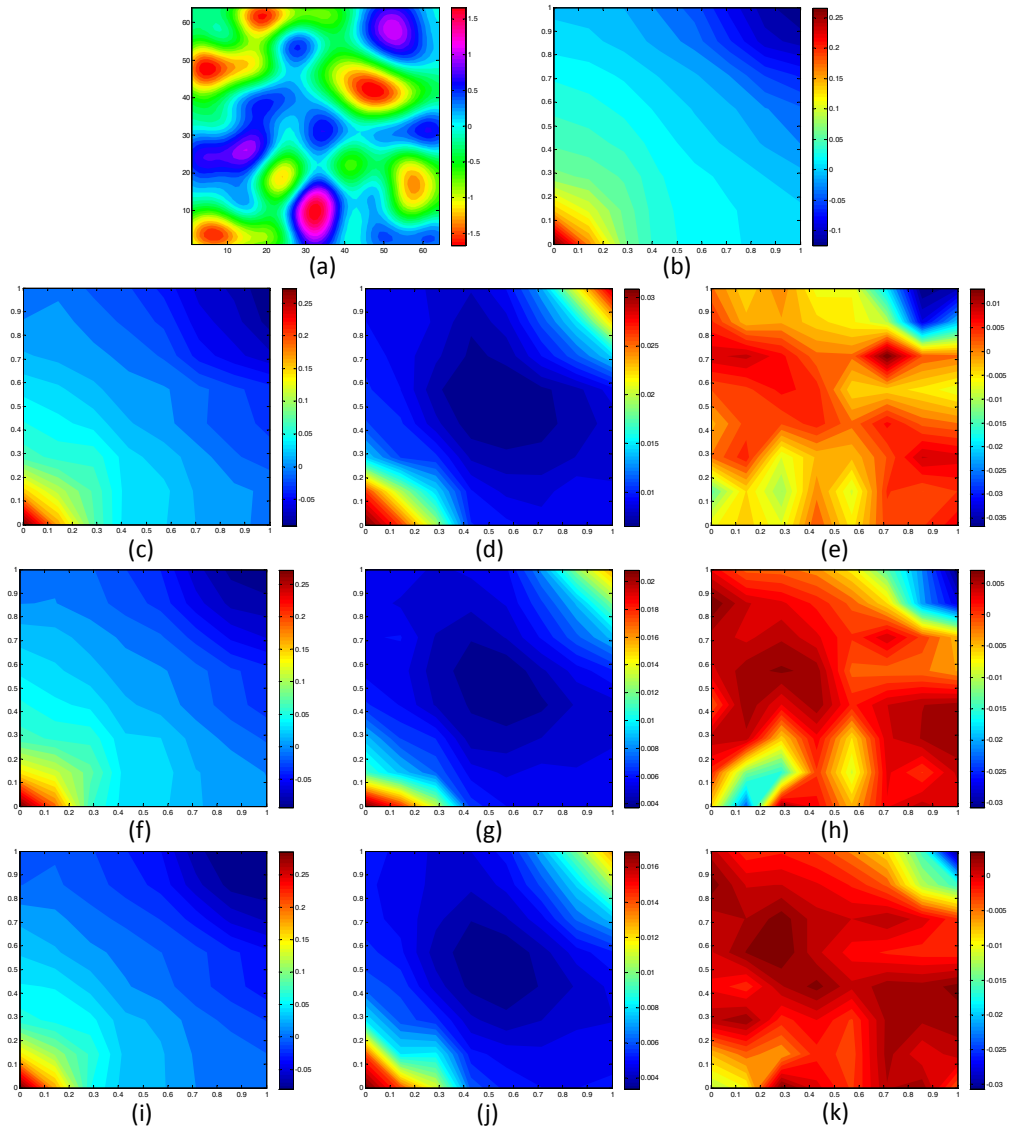


Figure 22: Stationary random field - Comparison of the prediction of $p$ given a new input permeability field using the nonparametric model versus the true $p$ from the deterministic solver: (a) The new observed input permeability field; (b) The contour plot of the true $p$ field; (c)(f)(i) The predicted mean by the nonparametric graphical model using $N = 400, 1000$ and 4000 training data, respectively; (d)(g)(j) The corresponding predictive variance for $N = 400, 1000$ and 4000 cases, respectively; (e)(h)(k) The difference between the predicted mean and the true response field for $N = 400, 1000$ and 4000 cases, respectively.

reconstructed input permeability field with the original given sample for different number of training samples $N$. In comparison with the reconstruction results in the previous example in Section 6.1.1, a higher $k$ is needed here to obtain a relatively good reconstruction. This is expected because the non-stationary permeability field is much more complicated than the stationary case. We obtain the similar conclusions from these figures as in the earlier example. As the reduced dimensionality $k$ increases, the reconstructed permeability field gets closer to the original permeability. Also as the number of training data $N$ used increases, the reconstructed permeability becomes closer to the original realization.

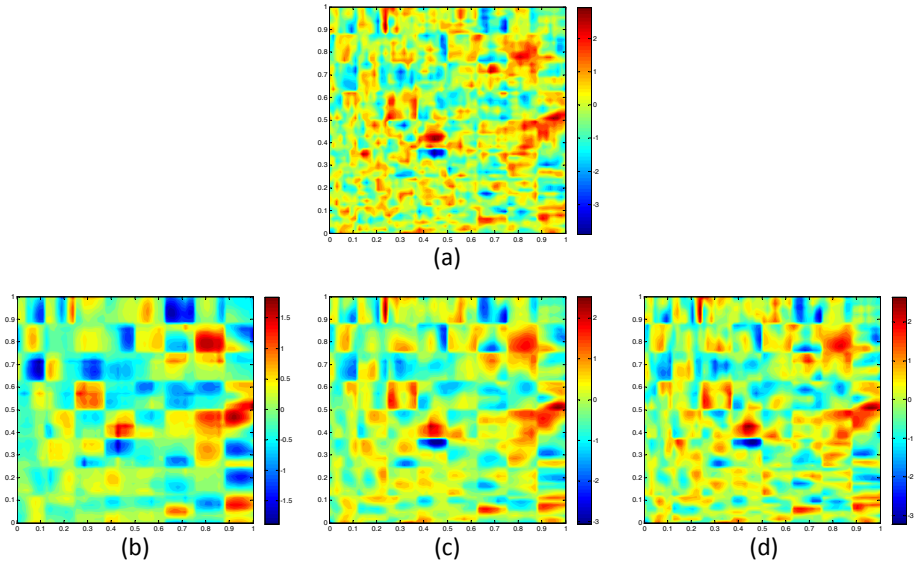


Figure 23: Non-stationary random field - Comparison of the reconstructed input permeability field with the original given sample: (a) With different $k$ for $N = 2000$; (b)(c)(d) The reconstructed input permeability using $k = 10, 30$, and 50, respectively.

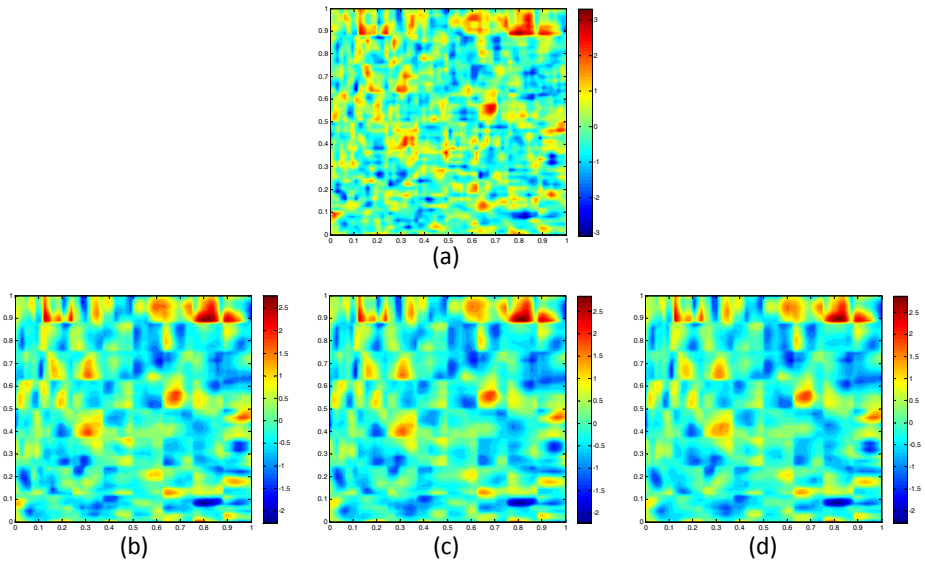


Figure 24: Non-stationary random field - Comparison of the reconstructed input permeability field with the original given sample: (a) With different number of training data for $k = 30$, where $k$ is the dimensionality of the reduced space; (b)(d)(f) The reconstructed input permeability using $N = 1000, 2000$, and 4000 training data, respectively.

Finally, we compare the reconstruction error of the input permeability field with different number of training data $N$ and different reduced dimensionality $k$ using Eq. (40) in section 6.1.1, as shown in Fig. 25. In this example, $k$ is chosen as 20, that is, the dimensionality of each $\mathbf{s}$ variable is 20.

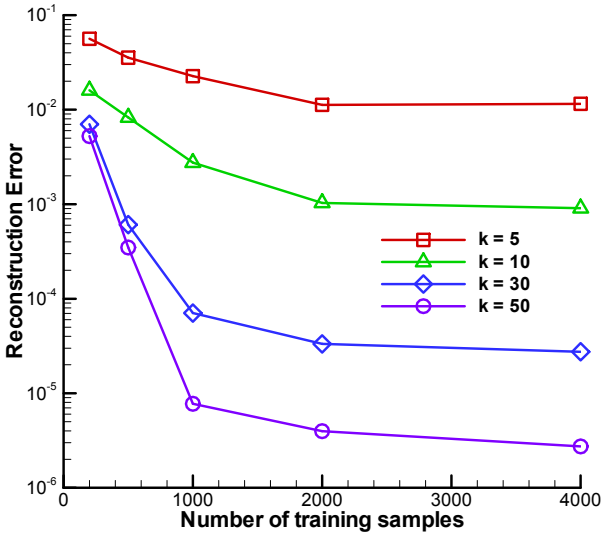


Figure 25: Non-stationary random field - Comparison of the reconstruction error of the input permeability field with different number of training data, and different reduced dimensionality $k$.

*6.2.2. Uncertainty propagation*

In this section, we are also going to investigate how the uncertainty propagate from the input permeability to the output (velocity and pressure) response, as in Section 6.1.2. Figure 26 compares the predicted mean of $u_x$ with a Monte Carlo estimate using $10^5$ observations. We can clearly see that as the number of training data increases, the prediction gets more and more accurate. The same statistic for $u_y$ and $p$ is reported in Figs. 27 and 28, respectively.

Fig. 29 compares the predicted variance of $u_x$ to a Monte Carlo estimate using $10^5$ observations. Also, the predicted variance converges to the MC results with the increase of the number of the training data. The same statistic for $u_y$ and $p$ is given in Figs. 30 and 31, respectively. We can see that in this example, $N = 4000$ training samples can only provide a reasonable predictions for the marginal mean and variance of the responses, while in the previous stationary example in section 6.1.2, $N = 1000$ training samples can already give rather accurate predictions. The predictions of the pressure, compared to the predictions of velocity, are much more accurate, this is because the pressure has less variability than the velocity in porous media flow problem.

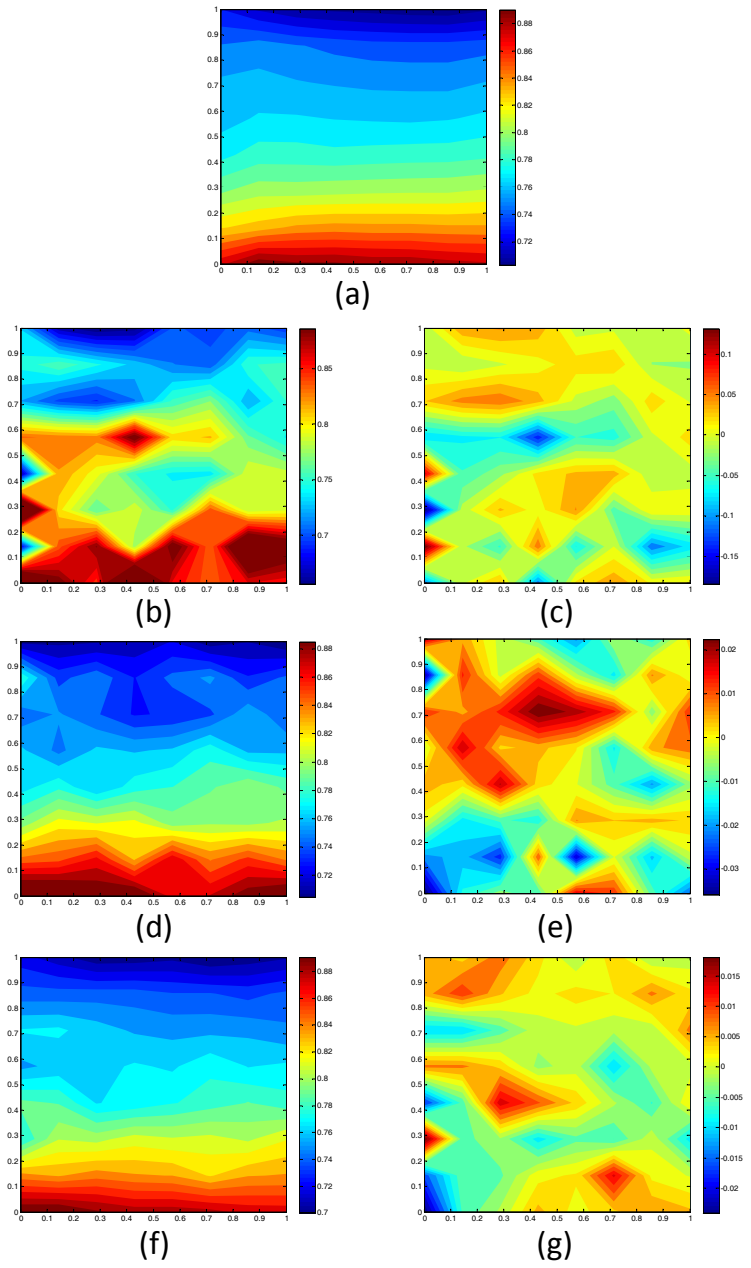


Figure 26: Non-stationary random field - Mean of $u_x$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted mean of $u_x$ using $400, 1000$, and $4000$ training samples, respectively; (c)(e)(g) The error between the predicted mean and the MC mean for $N = 400, 1000$, and $4000$, respectively.

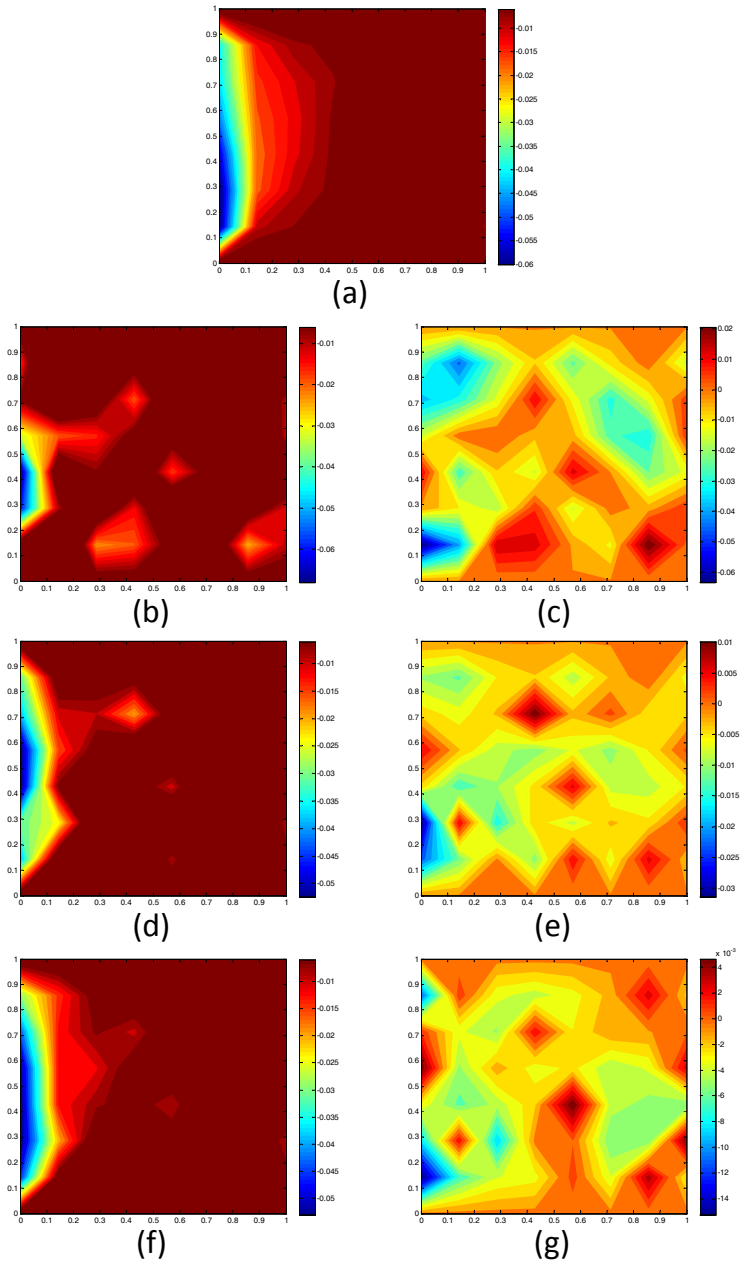


Figure 27: Non-stationary random field - Mean of $u_y$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted mean of $u_x$ using $400, 1000$, and $4000$ training samples, respectively; (c)(e)(g) The error between the predicted mean and the MC mean for $N = 400, 1000$, and $4000$, respectively.

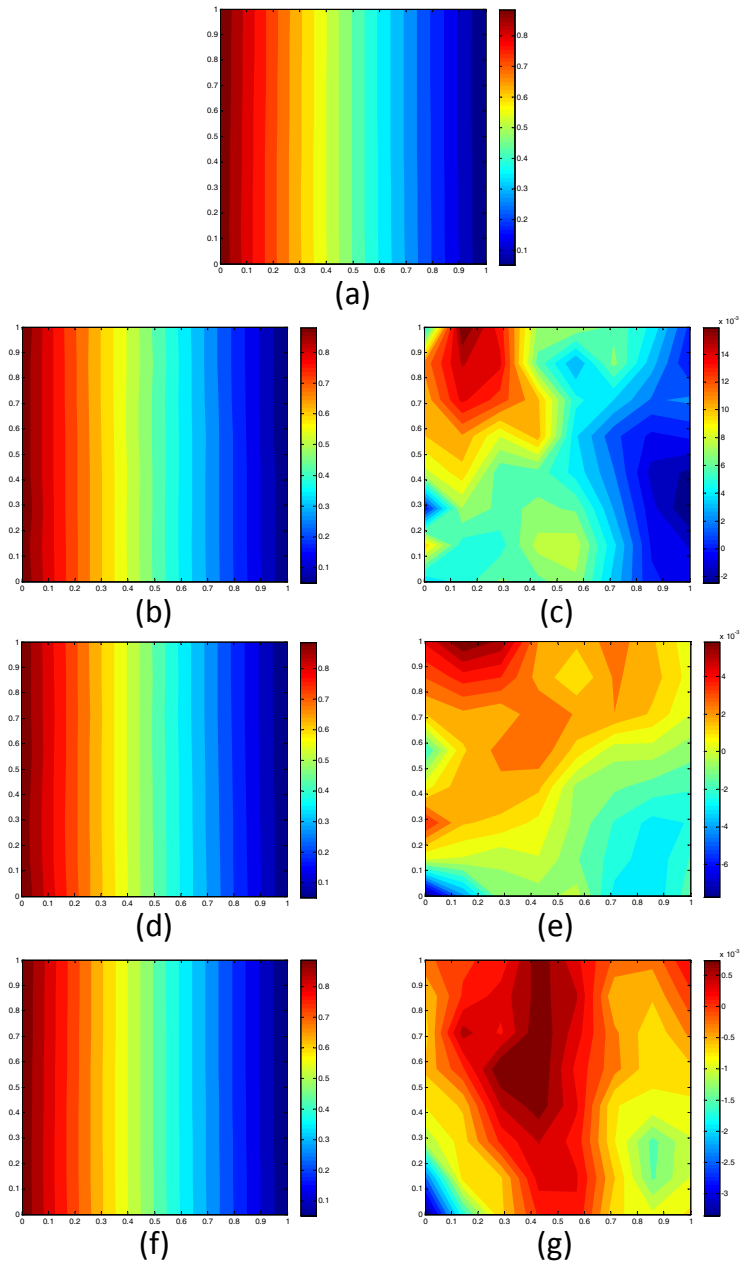


Figure 28: Mean of $p$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted mean of $u_x$ using $400, 1000$, and $4000$ training samples, respectively; (c)(e)(g) The error between the predicted mean and the MC mean for $N = 400, 1000$, and $4000$, respectively.

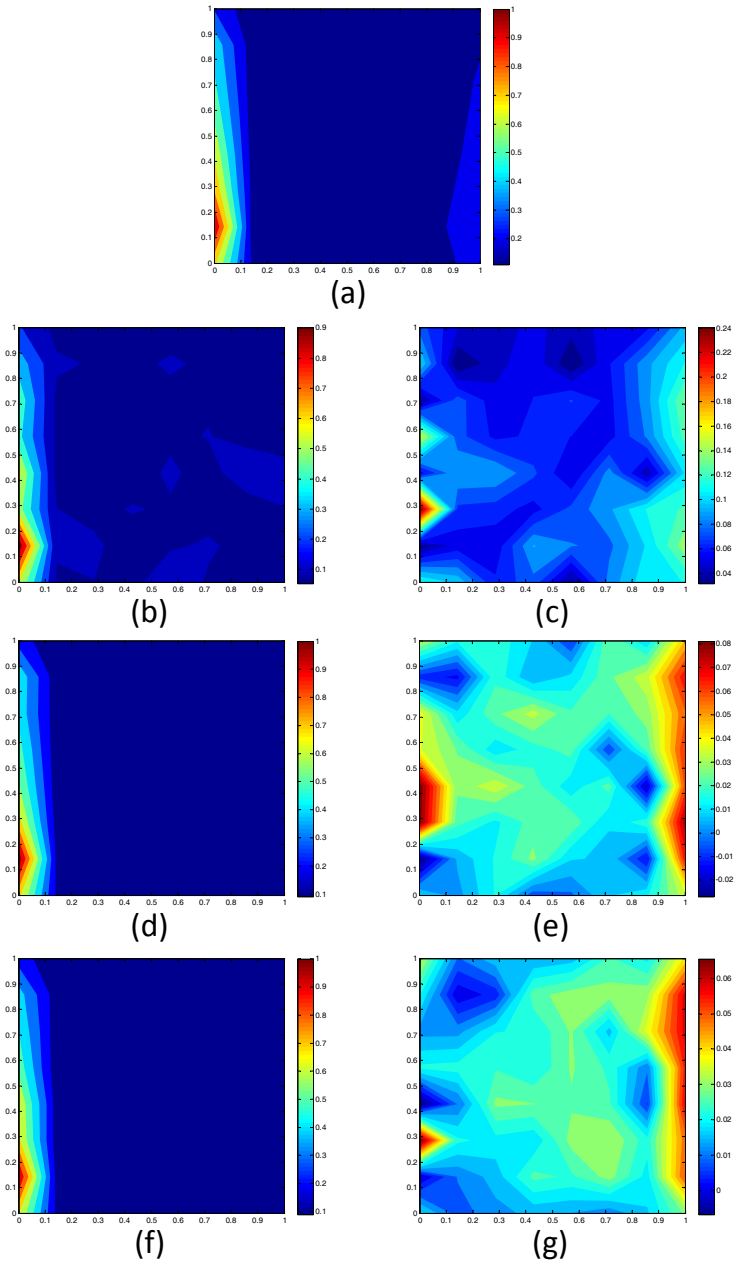


Figure 29: Non-stationary random field - Variance of $u_x$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted variance of $u_x$ using $400, 1000$, and $4000$ training samples, respectively; (c)(e)(g) The error between the predicted variance and the MC variance for $N = 400, 1000$, and $4000$, respectively.

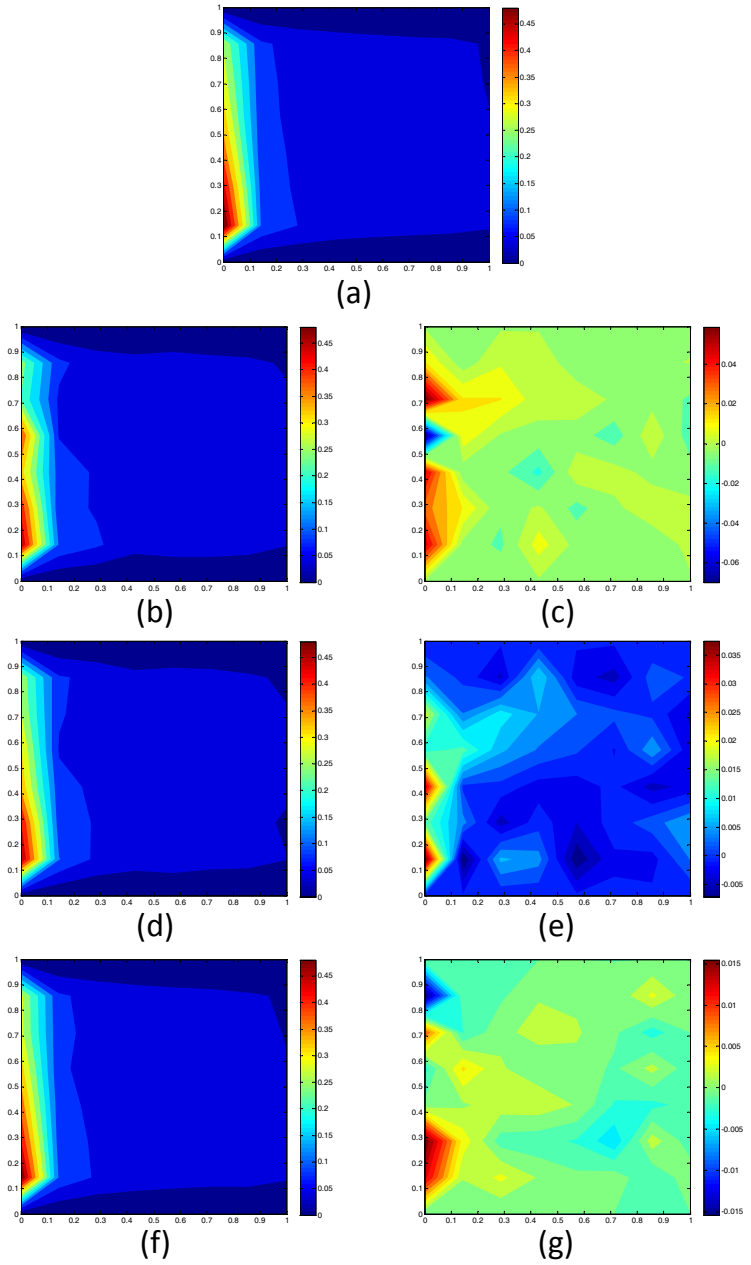


Figure 30: Non-stationary random field - Variance of $u_y$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted variance of $u_x$ using $400, 1000$, and $4000$ training samples, respectively; (c)(e)(g) The error between the predicted variance and the MC variance for $N = 400, 1000$, and $4000$, respectively.

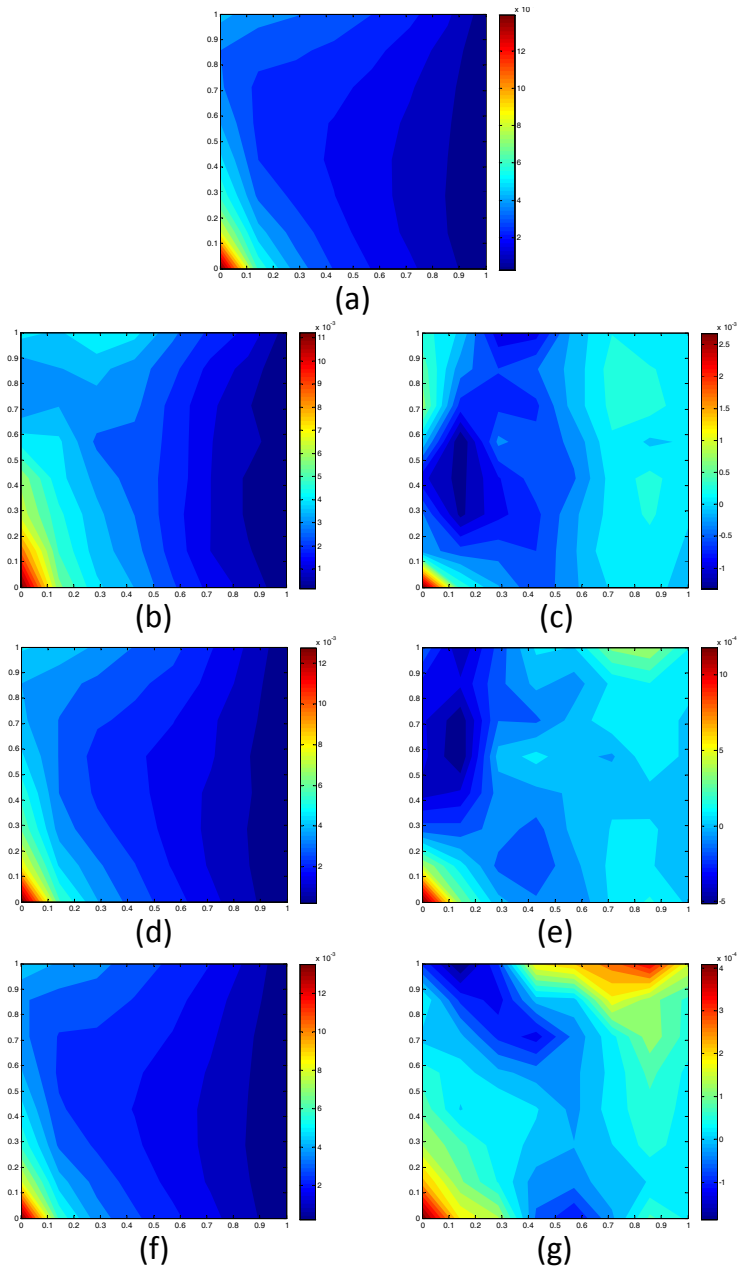


Figure 31: Non-stationary random field - Variance of $p$: (a) MC estimate using $10^5$ observations; (b)(d)(f) The predicted variance of $u_x$ using $400, 1000$, and $4000$ training samples, respectively; (c)(e)(g) The error between the predicted variance and the MC variance for $N = 400, 1000$, and $4000$, respectively.

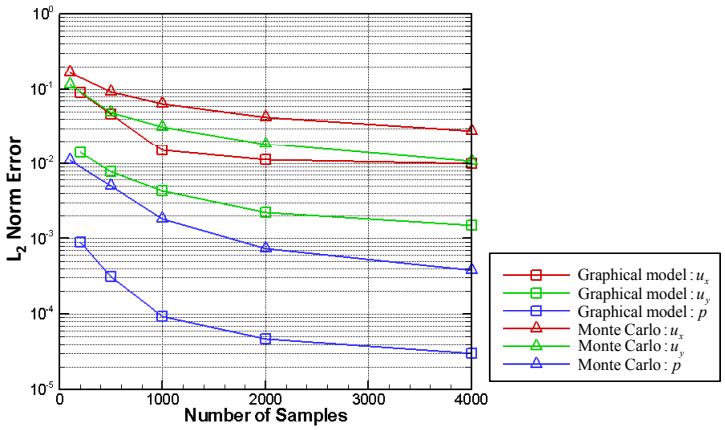


Figure 32: Non-stationary random field - The $L_2$ norm of the error as a function of the number of samples observed for graphical model framework.

Similarly, in Fig. 32, we plot the $L_2$ norm (Eq. (41)) of the error as a function of the number of samples for $u_x$, $u_y$ and $p$ and compare with the MC results. In addition, we compare the predicted probability densities of $u_x$, $u_y$ and $p$ at physical positions $(0.429, 0.429)$ and $(0.571, 0.571)$, with the PDFs obtained from the MC estimate using $10^5$ observations, as shown in Figs. 33 and Fig. 34, respectively. From the figures, we can see that as the number of observations increases, the graphical model prediction gradually captures the major key features of the PDFs.

*6.2.3. Response Prediction*

In this section, we also provide an example to demonstrate that the constructed graphical model is capable of acting as a surrogate model for the deterministic solver, as in section 6.1.3. Fig. 35 shows a comparison of the predicted $u_x$, $u_y$ and $p$ fields, with the results of the deterministic solver for given a new input permeability field $\mathbf{a}$, using $N = 4000$ training samples. Fig. 36 gives the comparison results for another realization. As shown from the figures, the predictions capture the main features of the responses.

## 7. Discussion and Conclusions

A probabilistic graphical model framework was developed to address the uncertainty propagation problem for flows in porous media. The framework could quantify the uncertainties propagating from the random input to the

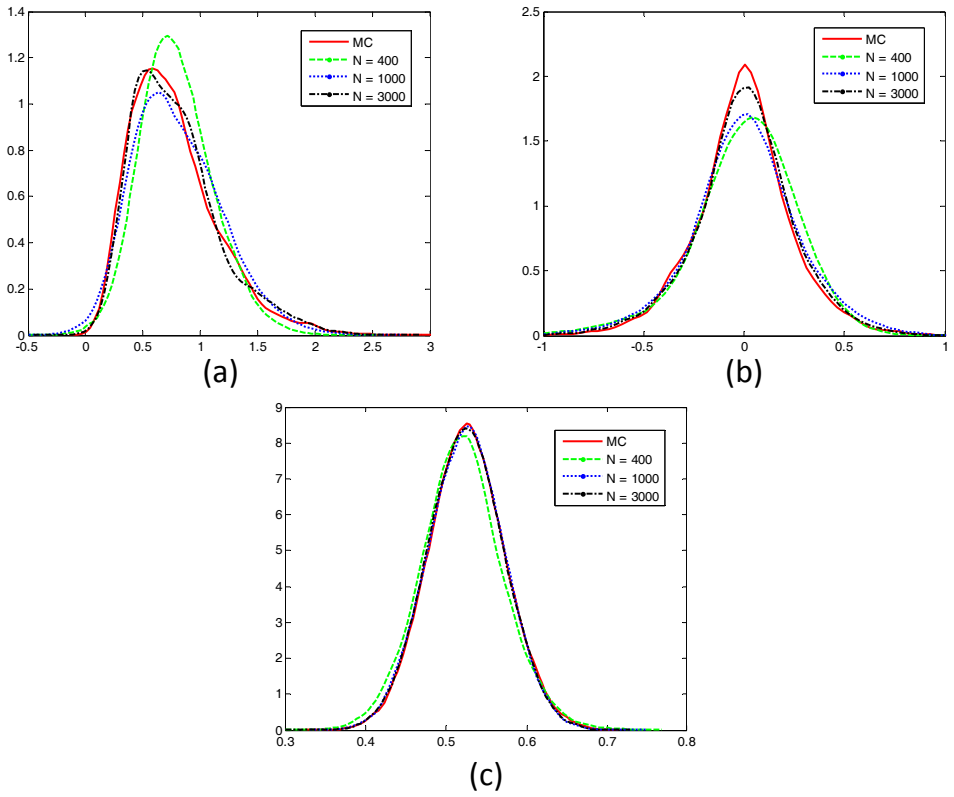


Figure 33: Non-stationary random field - Comparison of the predicted PDFs using different training data with the MC estimate at physical position ($(0.429, 0.429)$): (a) $u_x$, (b) $u_y$, (c) $p$.

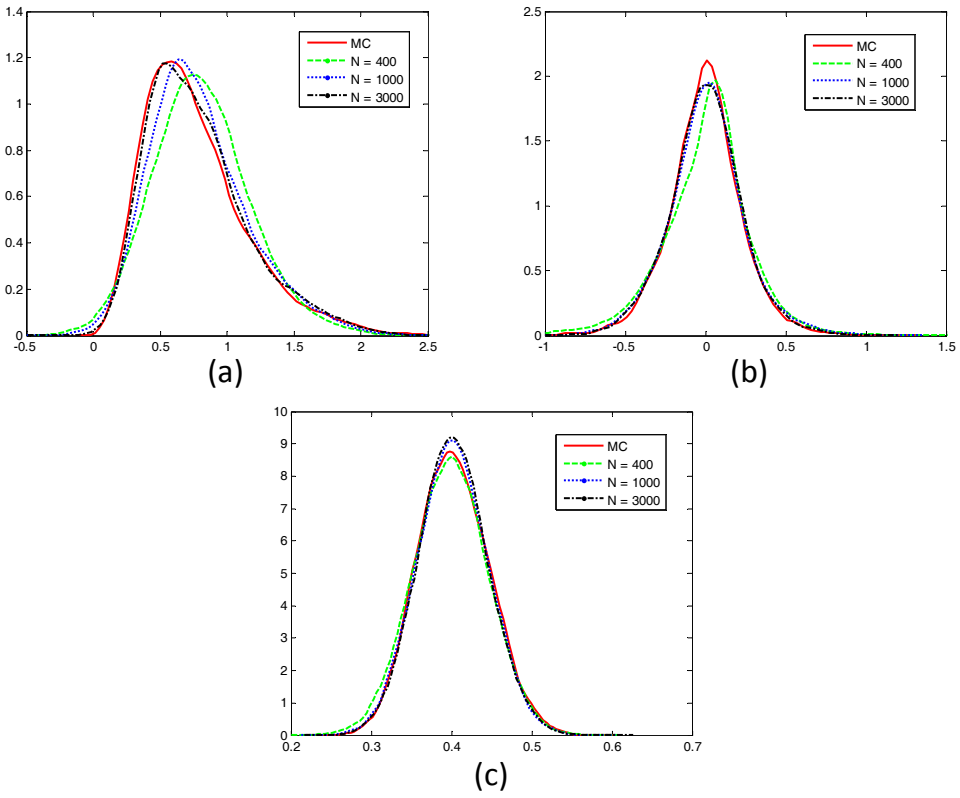


Figure 34: Non-stationary random field - Comparison of the predicted PDFs using different training data with the MC estimate at physical position $(0.571, 0.571)$: (a) $u_x$, (b) $u_y$, (c) $p$.

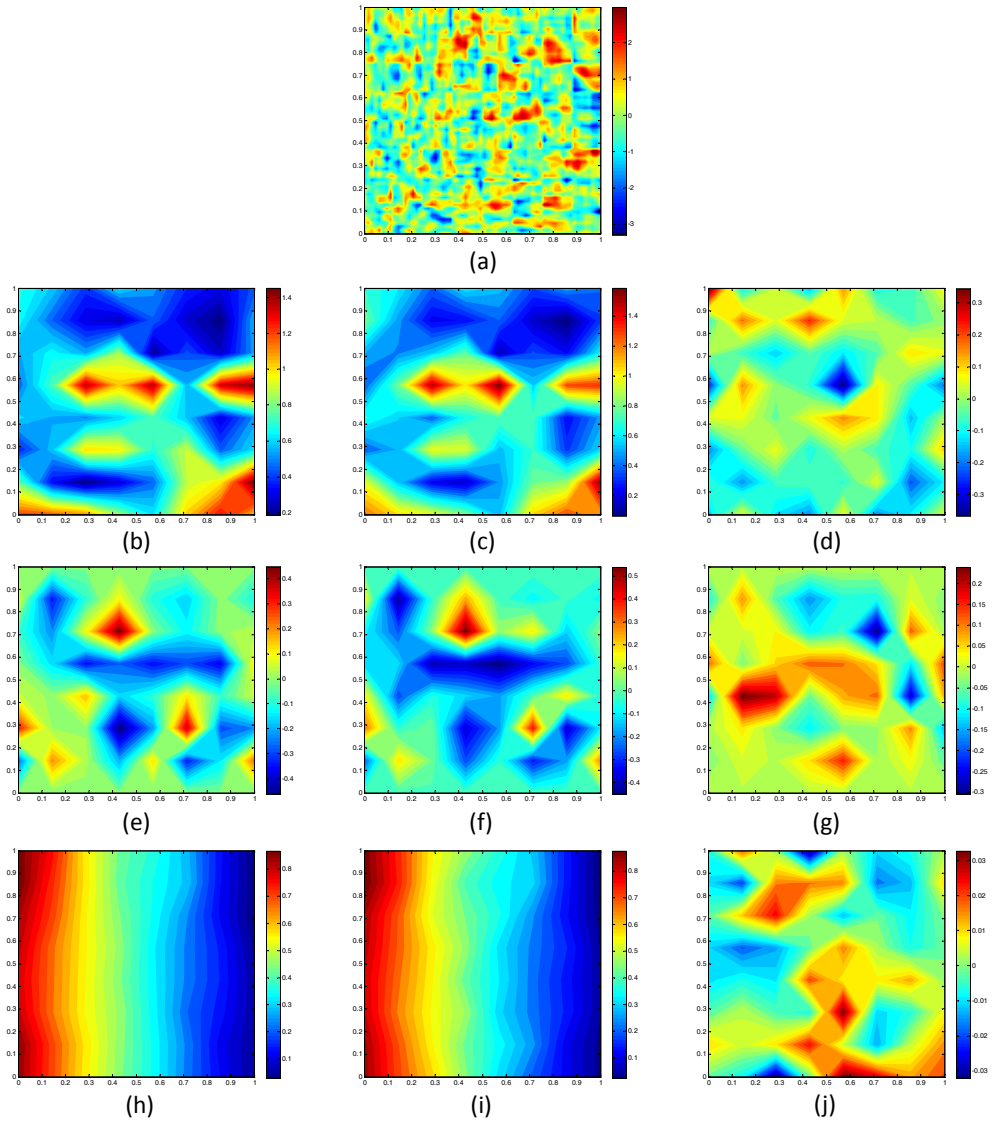


Figure 35: Non-stationary random field - Comparison of the predicted physical responses given a realization of stochastic input permeability with the true response: (a) The new observed input permeability field; (b)(e)(h) The true responses for the given permeability realization, from top to bottom, $u_x$, $u_y$ and $p$, respectively; (c)(f)(i) The predicted means for $u_x$, $u_y$ and $p$ by graphical model using $N = 4000$ training data, respectively; (d)(g)(j) The difference between the predicted mean and the true response field for $u_x$, $u_y$ and $p$, respectively.

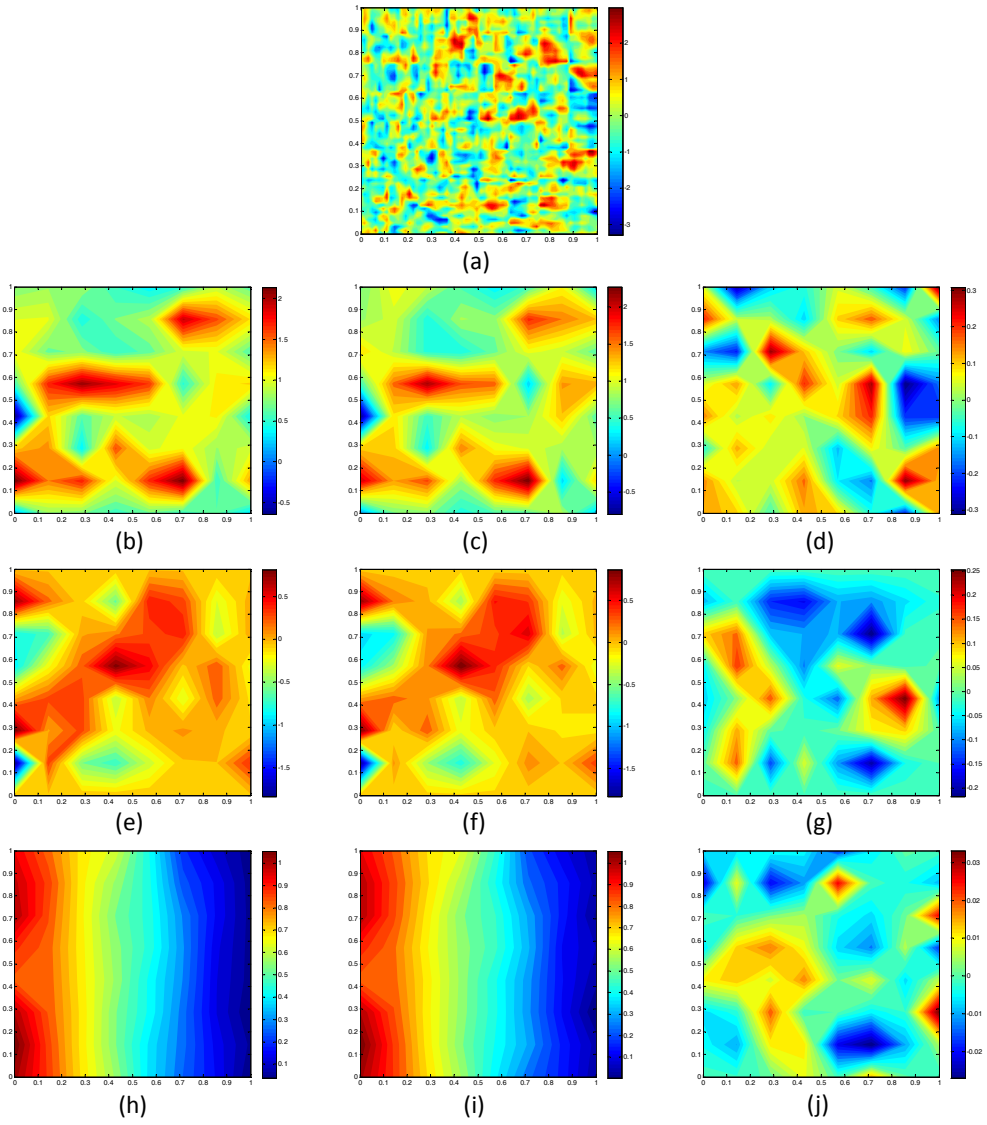


Figure 36: Non-stationary random field - Comparison of the predicted physical responses given a realization of stochastic input permeability with the true response: (a) The new observed input permeability field; (b)(e)(h) The true responses for the given permeability realization, from top to bottom, $u_x$, $u_y$ and $p$, respectively; (c)(f)(i) The predicted means for $u_x$, $u_y$ and $p$ by graphical model using $N = 4000$ training data, respectively; (d)(g)(j) The difference between the predicted mean and the true response field for $u_x$, $u_y$ and $p$, respectively.

multi-output system response. The high dimensionality nature of the relationship between the inputs and responses was addressed by breaking the global problem into small local problems posed over coarse-elements. The whole framework was designed to be nonparametric (Gaussian mixture), so it was capable of capturing non-Gaussian features and thus it should have a wider applicability to other multiscale problems. The graphical model was shown that it can serve as a surrogate model for predicting the responses for any new observed permeability input.

Various examples were considered to study the accuracy and efficiency of the probabilistic graphical model framework and inference algorithms. It was shown that this framework is capable of predicting the correct output statistics with rather limited number of observations. In the provided examples, it was shown to capture well the first- and second-order statistics, and also provided reasonable predictions of the PDFs of the outputs. The framework can be used to address inverse problems (e.g. from limited output data predict unobservable permeability information). Such inverse problems and extending the applicability of the framework to other critical engineering applications are topics of current research interest.

## A. PCA model reduction

The basic algorithm for the PCA method used is briefly illustrated as follows: Consider a set of $D$ dimensional data $\mathbf{X} := \{\mathbf{x}^{(n)}; n = 1, \ldots, N\}$ where $N$ is the number of data. We first compute the mean as $\bar{\mathbf{x}} = \mathbb{E}[\mathbf{x}] = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}^{(n)}$ and the covariance matrix as $\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}^{(n)} - \bar{\mathbf{x}})(\mathbf{x}^{(n)} - \bar{\mathbf{x}})^T$. The eigenvalues $\lambda_i$ and eigenvectors $\mathbf{v}_i$ of the covariance matrix $\mathbf{C}$ are then computed. We sort the eigenvalues $\lambda_i$ in a descending order, and take the first corresponding $K$ eigenvectors to assemble the transform matrix $\mathbf{T}$. Finally, we map the original data set to a $K$-dimensional space via the transform matrix $\mathbf{T}$ as

$$\boldsymbol{\xi}^{(n)} = \mathcal{R}(\mathbf{x}^{(n)}) = \mathbf{T}_{K\times D}(\mathbf{x}^{(n)} - \bar{\mathbf{x}}), n = 1, \ldots, N. \tag{43}$$

The random vector $\boldsymbol{\xi} := [\xi_1, \ldots, \xi_K]$ satisfies the following two conditions:

$$\mathbb{E}[\xi_i] = 0, \quad \mathbb{E}[\xi_i \xi_j] = \delta_{ij}\frac{\lambda_i}{N}, \quad i, j = 1, \ldots, K. \tag{44}$$

Therefore, the random coefficients $\xi_i$ are uncorrelated but not independent.

The reconstruction is a reverse process to the reduction process:

$$\widetilde{\mathbf{x}}_K^{(n)} = \mathcal{C}(\boldsymbol{\xi}^{(n)}) = \mathbf{T}_{K\times D}^T \boldsymbol{\xi}^{(n)} + \bar{\mathbf{x}}, n = 1, \ldots, N. \tag{45}$$

Here, the subscript $K$ is used to emphasize that the realization $\widetilde{x}_K^{(n)}$ is constructed using only the first $K$ eigenvectors, and we use $\mathbf{X}_K$ to denote the reconstructed random variable.

## B. Expectation Maximization Algorithm

EM [42, 43] is an elegant and powerful method to find maximum likelihood solutions for mixture models. The log-likelihood function is given by

$$L(\omega^{(m)}, \mu^{(m)}, \Sigma^{(m)}, m = 1, .., M|Z) = \sum_{i=1}^{N} \log \left[ \sum_{m=1}^{M} \omega^{(m)} \mathcal{N}(z_i|\mu^{(m)}, \Sigma^{(m)}) \right]. \tag{46}$$

The log-likelihood is calculated based as in [44]. The detailed steps are described in Algorithm 3. Notice that in our implementation of the EM algorithm, the number of mixture components $M$ was kept fixed.

## C. Metropolis Hastings algorithm

The Metropolis Hastings (MH) algorithm can draw samples from any probability distribution, especially, it can generate samples without knowing the normalization constant [25]. Therefore, in this work, we use MH algorithm to generate samples from $\frac{p(\mathbf{s}_{(i,j)})}{m_{y_{(i,j)}}(\mathbf{s}_{(i,j)})}$ in Eq. (29). In the following, for mathematical convenience, we use $x$ to denote the random variable $\mathbf{s}_{(i,j)}$, and $P(x)$ to denote the distribution $\frac{p(\mathbf{s}_{(i,j)})}{m_{y_{(i,j)}}(\mathbf{s}_{(i,j)})}$.

The detailed steps are given in Algorithm 4. The main disadvantage of the MH algorithm is that the samples generated are correlated. Even though over the long term they do correctly follow $P(x)$, a set of nearby samples will be correlated with each other and not correctly reflect the distribution. This means that if we want a set of independent samples, we have to throw away the majority of samples and only take every $n - th$ sample, for some value of $n$ (in this work, we set $n = 5$). In addition, although the Markov chain eventually converges to the desired distribution, the initial samples may follow a very different distribution, especially if the starting point is in a region of low density. So a "burn-in" period is needed, where an initial number of samples (e.g. the first $1,000$) are thrown away.

**Algorithm 3** The EM Algorithm

1: Initialize the means $\mu^{(m)}$, covariance $\Sigma^{(m)}$ and mixing coefficient $\omega^{(m)}$, and evaluate the initial value of the log likelihood.

2: E-step: Evaluate the responsibilities using the current parameter values

$$\gamma(r_{im}) = \frac{\omega^{(m)}\mathcal{N}(z_i|\mu^{(m)}, \Sigma^{(m)})}{\sum_{j=1}^{M} \omega^{(j)}\mathcal{N}(z_i|\mu^{(j)}, \Sigma^{(j)})}. \tag{47}$$

3: M-step: Re-estimate the parameters using the current responsibilities:

$$\mu_{new}^{(m)} = \frac{1}{N_m}\sum_{i=1}^{N}\gamma(r_{im})z_i, \tag{48}$$

$$\Sigma_{new}^{(m)} = \frac{1}{N_m}\sum_{i=1}^{N}\gamma(r_{im})(z_i - \mu_{new}^{(m)})(z_i - \mu_{new}^{(m)})^T, \tag{49}$$

$$\omega_{new}^{(m)} = \frac{N_m}{N}, \tag{50}$$

where

$$N_m = \sum_{i=1}^{N}\gamma(r_{im}). \tag{51}$$

4: Evaluate the log likelihood given in Eq. (46) using the current parameters and check for convergence of the log-likelihood. If the convergence criterion is not satisfied, then return to Step 2.

**Algorithm 4** The Metropolis Hastings Algorithm

1: Pick an arbitrary probability density $Q(x'|x_t)$, where $Q$ is the proposal jumping distribution, which suggests a new sample value $x'$ given a sample value $x_t$. Here, we choose a widely used symmetric jumping distribution – Gaussian distribution centered at $x_t$.

2: Start with some arbitrary point $x_0$ as the first sample.

3: To generate a new sample $x_{t+1}$ given the most recent sample $x_t$, proceed as follows:

(1) Generate a proposed new sample value $x'$ from the jumping distribution $Q(x'|x_t)$.

(2) Calculate the acceptance ratio as:

$$r = \frac{P(x')}{P(x_t)}. \tag{52}$$

(3) If $r \geq 1$, accept $x'$ by setting $x_{t+1} = x'$.

(4) Else, accept $x'$ with probability $r$. That is, pick a uniformly distributed random number $u \sim \mathcal{U}[0, 1]$, and if $u \leq r$ set $x_{t+1} = x'$, else set $x_{t+1} = x_t$.

## D. Proof of estimated variance for Gaussian mixture

Assume $y$ is a random variable that can be represented as $y = \sum_{m=1}^{M} \omega_m y_m$, where $y_m \sim \mathcal{N}(\mu_m, \sigma_m^2)$. The mean of $y$ can be calculated as

$$\mathbb{E}[y] = \sum_{m=1}^{M} \omega_m \mu_m. \tag{53}$$

And the variance of $y$ can be calculated as

$$\begin{aligned}
Var[y] &= \mathbb{E}[y^2] - (\mathbb{E}[y])^2 \\
&= \mathbb{E}[(\sum_{m=1}^{M} \omega_m y_m)^2] - (\sum_{m=1}^{M} \omega_m \mu_m)^2 \\
&= \mathbb{E}[\sum_{m=1}^{M} \omega_m^2 y_m^2 + \sum_{m,n=1,m\neq n}^{M} \omega_m \omega_n y_m y_n] - (\sum_{m=1}^{M} \omega_m^2 \mu_m^2 + \sum_{m,n=1,m\neq n}^{M} \omega_m \omega_n \mu_m \mu_n) \\
&= \left(\sum_{m=1}^{M} \omega_m^2 \mathbb{E}[y_m^2] + \sum_{m,n=1,m\neq n}^{M} \omega_m \omega_n \mathbb{E}[y_m]\mathbb{E}[y_n]\right) \\
&\quad - \left(\sum_{m=1}^{M} \omega_m^2 \mu_m^2 + \sum_{m,n=1,m\neq n}^{M} \omega_m \omega_n \mu_m \mu_n\right) \\
&= \sum_{m=1}^{M} \omega_m^2 (\mathbb{E}[y_m^2] - \mu_m^2) \\
&= \sum_{m=1}^{M} \omega_m^2 \sigma_m^2.
\end{aligned} \tag{54}$$

## E. Gaussian Mixture Reduction

Given a $N$-components Gaussian mixture, we want to find an effectively reduced Gaussian mixture form without losing too much information from the original Gaussian mixture. The problem can be defined as follows:

$$\widetilde{f}(x) = \sum_{i=1}^{N} \widetilde{w}_i \cdot \mathcal{N}(x; \widetilde{\mu}_i, \widetilde{\Sigma}_i) \Longrightarrow f(x) = \sum_{j=1}^{M} w_j \cdot \mathcal{N}(x; \mu_j, \Sigma_j). \tag{55}$$

General approaches dealing with the problem of Gaussian mixture reduction can be classified into two fields. Bottom-up approaches start with a

single Gaussian function and iteratively add additional components until the original mixture density is approximated appropriately (e.g. PGMR [45]). Top-down approaches take the original Gaussian mixture density and iteratively decrease the number of mixture components, either by removing single unimportant components or by merging similar components (e.g. Salmond's algorithm [46]). In addition, these algorithms can be further divided into local and global methods. Gaussian mixture reduction via clustering (GMRC [47]) method can be classified as a top-down algorithm using global information. The interested reader can refer to [47] for the detailed algorithm.

## Acknowledgements

The research at Cornell was supported by an OSD/AFOSR MURI09 award on uncertainty quantification, the US Department of Energy, Office of Science, Advanced Scientific Computing Research and the Computational Mathematics program of the National Science Foundation (NSF) (award DMS-1214282). This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Additional computing resources were provided by the NSF through TeraGrid resources provided by NCSA under Grant No. TG-DMS090007.

## References

[1] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, Journal of Computational Physics 228 (2009) 3084–3113.

[2] I. Bilionis, N. Zabaras, Multi-output Local Gaussian Process Regression: Applications to Uncertainty Quantification, Journal of Computational Physics 231 (2012) 5718–5746.

[3] P. Chen, N. Zabaras, Adaptive Locally Weighted Projection Regression Method for Uncertainty Quantification, Communications in Computational Physics (CiCP) (under review).

[4] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.

[5] M. Koutsokeras, I. Pratikakis, G. Miaoulis, A web-based 3D graphical model search engine, In Proceedings of the Eighth International Conference on Computer Graphics and Artificial Intelligence, May 11-12, Limoges, France (2005) 79–90.

[6] G. K. Baah, A. Podgurski, M. J. Harrold, The Probabilistic Program Dependence Graph and Its Application to Fault Diagnosis, IEEE Transactions on Software Engineering 36 (2010) 528–545.

[7] J. A. Bilmes, C. Bartels, Graphical model architectures for speech recognition, Signal Processing Magazine, IEEE 22 (2005) 89–100.

[8] J. van de Ven, F. Ramos, G. D. Tipaldi, An integrated probabilistic model for scan-matching, moving object detection and motion estimation, 2010 IEEE International Conference on Robotics and Automation (ICRA), 3-7 May 2010 887–894.

[9] P. Baldi, Probabilistic Graphical Models in Computational Molecular Biology, Journal of the Italian Association for Artificial Intelligence 1 (2000) 8–12.

[10] J. F. Brendan, Graphical Models for Machine Learning and Digital Communication, 1998.

[11] M. I. Jordan, Graphical models, Statistical Science 19 (1) (2004) 140–155.

[12] M. Isard, PAMPAS: Real-Valued Graphical Models for Computer Vision, Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition (2003) 613–620.

[13] J. Schiff, E. B. Sudderth, K. Goldberg, Nonparametric belief propagation for distributed tracking of robot networks with noisy inter-distance measurements, IEEE International Conference on Intelligent Robots and Systems (2009) 1369 – 1376.

[14] E. B. Sudderth, A. T. Ihler, W. T. Freeman, A. S. Willsky, Nonparametric belief propagation, Communications of the ACM 53 (2010) 95–103.

[15] J. Wan, N. Zabaras, A probabilistic graphical model approach to stochastic multiscale partial differential equations, Journal of Computational Physics (under review).

[16] R. G. Ghanem, P. D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, New York, 1991.

[17] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics and Intelligent Laboratory Systems 2 (1) (1987) 37–52.

[18] C. J. C. Burges, Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, chapter Geometric Methods for Feature Selection and Dimensional Reduction: A Guided Tour., Kluwer Academic Publishers, 2005.

[19] B. Scholkopf, A. J. Smola, K. R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.

[20] J. Wang, Z. Zhang, H. Zha, Adaptive manifold learning. In Advances in Neural Information Processing Systems, The MIT Press 17 (2005) 1473–1480.

[21] L. J. P. van der Maaten, E. O. Postma, H. J. van den Herik, Dimensionality Reduction: A Comparative Review, Tech. rep., Tilburg University Technical Report, TiCC-TR 2009-005 (2009).

[22] M. J. Wainwright, M. I. Jordan, Graphical models, exponential families, and variational inference, Found. Trends Mach. Learn. 1 (1-2) (2008) 1–305.

[23] H. Rue, L. Held, Gaussian Markov Random Fields: Theory and Applications, Vol. 104 of Monographs on Statistics and Applied Probability, Chapman & Hall, London, 2005.

[24] Y. Efendiev, T. Y. Hou, Multiscale Finite Element Methods: Theory and Applications (Surveys and Tutorials in the Applied Mathematical Sciences, Vol. 4), Springer, 2009.

[25] W. K. Hastings, Monte Carlo Sampling Methods Using Markov Chains and Their Applications, Biometrika 57 (1970) 97–109.

[26] A. Delong, A Scalable graph-cut algorithm for N-D grids, IEEE Conference on Computer Vision and Pattern Recognition, June, 2008 (2008) 1–8.

[27] D. Koller, U. Lerner, D. Anguelov, A General Algorithm for Approximate Inference and its Application to Hybrid Bayes Nets, in: Proceedings of the Fifteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-99), Morgan Kaufmann, San Francisco, CA, 1999, pp. 324–333.

[28] J. Sun, H. Shum, N. Zheng, Stereo matching using belief propagation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 787–800.

[29] W. T. Freeman, E. C. Pasztor, O. T. Carmichael, Learning low-level vision, IJCV 40 (1) (2000) 25–47.

[30] J. L. Williams, R. A. Lau, Convergence of loopy belief propagation for data association, Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference (2010) 175–180.

[31] J. Mooij, H. Kappen, Sufficient conditions for convergence of Loopy Belief Propagation, in: Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), AUAI Press, Arlington, Virginia, 2005, pp. 396–403.

[32] M. J. Wainwright, T.S.Jaakkola, A. Willsky, Tree-based reparameterization analysis of sum-product and its generalizations, IEEE Transactions on Information Theory 49 (5) (2003) 1120–1146.

[33] Y. Weiss, W. T. Freeman, Correctness of belief propagation in Gaussian graphical models of arbitrary topology, Neural Computation 13 (2001) 2173–2200.

[34] K. Hackl, IUTAM symposium on variational concepts with applications to the mechanics of materials : Proceedings of the IUTAM Symposium on Variational Concepts with Applications to the Mechanics of Materials, Bochum, Germany, September 22-26, 2008, Dordrecht, New York, 2010.

[35] A. T. Ihler, E. B. Sudderth, W. T. Freeman, A. S. Willsky, Efficient multiscale sampling from products of gaussian mixtures, in: S. Thrun,

L. K. Saul, B. Schlkopf (Eds.), Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada], MIT Press, 2003.

[36] J. E. Aarnes, V. Kippe, K.-A. Lie, A. B. Rustad, Modelling of Multiscale Structures in Flow Simulations for Petroleum Reservoirs, in: G. Hasle, K.-A. Lie, E. Quak (Eds.), Geometric Modelling, Numerical Simulation, and Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[37] P. Bochev, R. B. Lehoucq, On Finite Element solution of the pure Neumann problem, SIAM Rev 47 (2001) 50–66.

[38] P. Raviart, J. Thomas, A mixed finite element method for 2-nd order elliptic problems, in: I. Galligani, E. Magenes (Eds.), Mathematical Aspects of Finite Element Methods, Vol. 606 of Lecture Notes in Mathematics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1977.

[39] F. Brezzi, T. J. R. Hughes, L. D. Marini, A. Masud, Mixed discontinuous Galerkin methods for Darcy flow, SIAM J. Sci. Comput. 22-23 (2005) 119–145.

[40] J. H. A. Logg, G. N.Wells, DOLFIN: a C++/Python Finite Element Library, 2012.

[41] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, K. S. Stanley, An overview of the TRILINOS project, ACM Trans. Math. Softw. 31 (2005) 397–423.

[42] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1) (1977) 1–38.

[43] G. J. McLachlan, K. E. Basford, Mixture Models: Inference and Applications to Clustering, Marcel Dekker, New York, 1988.

[44] K. V. Mardia, J. T. Kent, J. M. Bibby, Multivariate Analysis, Academic Press, 1979.

[45] M. Huber, U. Hanebeck, Progressive Gaussian Mixture Reduction, In 11th International Conference on Information Fusion (2008) 1–8.

[46] D. J. Salmond, Mixture reduction algorithms for target tracking in clutter, In Proceedings of SPIE 1305 (1990) 434–445.

[47] D. Schieferdecker, M. Huber, Gaussian Mixture Reduction via Clustering, In 12th International Conference on Information Fusion (2009) 1536–1543.