École doctorale 072 : Sciences pour l'Ingénieur

Deep Gaussian Processes for the Analysis and Optimization of Complex Systems - Application to Aerospace System Design

A dissertation submitted by

Ali Hebbal

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
(*Informatics*)

École Doctorale Sciences Pour l'Ingénieur Université de Lille

Lille, France

&

ONERA - The French Aerospace Lab

Palaiseau, France

Defended publicly on the 21th of January 2021 in front of a jury composed of:

Dr. Mathieu BALESDENT	Research engineer, ONERA, Palaiseau	Co-advisor
Dr. Loïc BREVAULT	Research engineer, ONERA, Palaiseau	Co-advisor
Dr. Maurizio FILIPPONE	Associate Professor (HDR), EURECOM, Biot	Reviewer
Dr. Céline HELBERT	Associate Professor, Centrale Lyon, Lyon	Examiner
Pr. Nouredine MELAB	Professor, Université de Lille, Lille	Co-director
Pr. Joseph MORLIER	Professor, ISAE-SUPAERO, Toulouse	Reviewer
Pr. Ann NOWÉ	Professor, Vrije Universiteit, Brussels	Examiner, President
Pr. El-Ghazali TALBI	Professor, Université de Lille, Lille	Co-director

École doctorale 072 : Sciences pour l'Ingénieur

Processus Gaussiens Profonds pour l'Analyse et l'Optimisation des Systèmes Complexes - Application à la Conception des Systèmes Aérospatiaux

Thèse de doctorat présentée et soutenue par

Ali Hebbal

en vue d'obtenir le grade de

Docteur d'université en

Informatiques

École Doctorale Sciences Pour l'Ingénieur Université de Lille

Lille, France

&

ONERA - The French Aerospace Lab

Palaiseau, France

Soutenue publiquement le 21 Janvier 2021 devant un jury composé de:

Dr. Mathieu BALESDENT	Ingénieur de recherche, ONERA, Palaiseau	Co-encadrant
Dr. Loïc BREVAULT	Ingénieur de recherche, ONERA, Palaiseau	Co-encadrant
Dr. Maurizio FILIPPONE	Professeur associé (HDR), EURECOM, Biot	Rapporteur
Dr. Céline HELBERT	Maîtresse de conférences, Centrale Lyon, Lyon	Examinatrice
Pr. Nouredine MELAB	Professeur, Université de Lille, Lille	Co-directeur
Pr. Joseph MORLIER	Professeur, ISAE-SUPAERO, Toulouse	Rapporteur
Pr. Ann NOWÉ	Professeur, Vrije Universiteit, Brussels	Examinatrice, Présidente
Pr. El-Ghazali TALBI	Professeur, Université de Lille, Lille	Co-directeur

To my Parents . . .
In loving memory of my Grandmother . . .

Acknowledgements

J'écris ces remerciements en clôturant mon manuscrit de thèse pour élucider ce qu'on ne peut lire dans les apports scientifiques de ce manuscrit et dans les *et al.* des publications, mais qui sans, ce travail n'aurai jamais pu avoir lieu.

Tout d'abord j'aimerai remercier les personnes qui ont évalué ce manuscrit et ont permis sa publication. Merci au Dr. Maurizio Filippone et au Pr. Joseph Morlier d'avoir accepté d'être rapporteurs de cette thèse et en plus en période des fêtes de fin d'années ! Vos rapports ont permis d'améliorer ce manuscrit et l'ont fait gagner en clarté. Je vous remercie aussi pour les discussions lors de la soutenance de thèse qui m'ont aussi permis de donner plus de sens à mon travail. Merci au Dr. Céline Helbert d'avoir accepté d'examiner cette thèse et pour les questions très intéressantes lors de la soutenance qui mettent en évidence des pistes de recherche prometteuses pour les processus Gaussiens profonds et l'analyse des systèmes complexes. J'adresse finalement mes remerciements à la présidente du jury, Pr. Ann Nowé d'avoir accepté de présider le jury lors de cette soutenance de thèse un peu particulière dans la forme dû aux circonstances du Covid.

Les quatre autres membres du jury m'ont accompagné pendant ces années de thèse et ont été des acteurs actifs de ce travail. Merci à mes deux directeurs de thèse lillois El-Ghazali Talbi et Nouredine Melab pour avoir accepté de diriger cette thèse et aussi pour m'avoir donné l'opportunité d'organiser des conférences et de de mettre en place un cours de master d'optimisation et machine learning et l'enseigner à l'université de Lille.

Ensuite, un remerciement spécial au duo d'encadrants : Mathieu Balesdent et Loïc Brevault, qui ont vraiment encadré ce travail quotidiennement et ont fourni une quantité d'effort bien au-delà des attentes qu'on peut avoir d'un encadrant, et ont assurer la réussite de ce travail. Je remercie particulièrement Mathieu pour sa chaleur et son humour durant nos points ! Et un remerciement tout aussi particulier à Loïc d'avoir proposé les processus Gaussiens profonds comme piste pour l'analyse des systèmes complexes et aussi pour être un modèle de travail et de ténacité.

Toujours au niveau scientifique, je remercie mes ex-professeurs qui m'ont mis sur la voie de la recherche. Un remerciement particulier au Pr. Loubna Benabbou et au Pr. Najiba Sbihi.

Ces 4 dernières années (en comptant aussi mon stage de master à l'Onera) ont été une expérience de croissance très importante aussi au niveau personnel. Pour cela je dois remercier un groupe de personnes très particulier.

Je remercie d'abord les doctorants et stagiaires de l'Onera DTIS qui ont été mes compagnons de thèse durant ses dernières années et qui ont assuré une bonne ambiance au quotidien. Par ordre d'ancienneté, je remercie Ricardo et sa passion pour les mathématiques avec qui c'était un véritable plaisir d'échanger sur les sujets scientifiques. Je remercie ensuite Irina de nous avoir montrer qu'on peut être une personne très chill et réussir à la Nasa ! Je remercie aussi Romain qui était un voisin de bureau très chaleureux et pour les débats politiques auxquels j'assistais certes en tant que spectateur mais qui étaient très instructifs (et amusants huh) ! Et bien sûr quand on parle de sujet politique et de Romain, Vincent n'est jamais très loin ! Un merci particulier à cette personne, que je considère comme un grand frère de thèse, et qui était l'une des personnes qui m'ont convaincu de faire un doctorat et qui était toujours là pour me conseiller avant, durant et après la thèse !

Ensuite, je souhaite remercier la golden generation de doctorants. Emilien d'abord pour son humour très spécial (alalala... ayayayay), son enthousiasme pour le foot qui a tellement manqué au groupe après son départ, et les fameux débats fausse/vrai Optim. Nathan, merci encore pour tes suggestions culinaires, surtout comment manger un kiwi, ça a changé ma vie, et tu le sais très bien au fond aoe 3 » aoe 2. Merci à Léon, l'une des personnes les plus gentilles du groupe et n'oublie jamais pas de pâtes avant le sport ! Merci aussi à Sergio et Camille (la grande).

Un remerciement très particulier pour Julien, mon deuxième frère de thèse et mon cobureau durant 3 ans. Merci Julien d'abord pour m'avoir introduit aux problématiques aérospatiales et pour les échanges sur nos thèses jumelles qui a permis vraiment d'enrichir cette thèse. Merci aussi pour ta bonne humeur au quotidien et ta culture des memes. J'aurais aimé faire plus d'escalade avec toi mais maudite crampe au mollet (huh).

Je remercie aussi la dernière génération de doctorants, et particulièrement le trio (Esteban, Enzo et Camille). Merci Esteban, enfant du soleil et enfant prodige pour ta bonne humeur et d'être facile à faire rire, toujours bien de t'avoir à coté pour la confiance en notre humour. Merci à Camille, ma deuxième co-bureau pour sa gentillesse hors borne, ses petits déjeuner, nos échanges sur les animes, et surtout pour ta volonté sans cesse de créer une bonne ambiance au sein du groupe, et l'une de mes réussites durant mon temps à l'Onera c'est de t'avoir converti au foot huh (oui je sais je te dois une séance de boxe). Merci à Enzo, the turning point au sein du groupe, il y a vraiment eu un avant et après Enzo et je suis très content de l'après ! Merci aussi pour m'avoir introduit à Tikz/gnuplot sans quoi les figures de ce manuscrit ne seraient pas aussi réussies ! Dommage pour toi que je finis la thèse avec la balle d'or au baby !

En dehors de l'Onera, je remercie mon groupe d'amis EMIste pour les voyages, les soirées de jeux vidéo et les expériences que nous avons partagé durant toutes ces années. Un remerciement ensuite tout aussi particulier que la personne elle-même à FZZ qui a été et qui est la personne sur qui je me repose.

Finalement, je remercie ma famille, mes deux frères Ahmed et Abdel, ma sœur Fatima-Zohra et mes parents sans qui rien de tout cela ne serait possible et qui sont ma source de motivation au quotidien. Merci d'avoir toujours été présents pour moi, d'avoir su me guider par l'exemple, de m'avoir permis de suivre ma passion pour la science et de m'avoir permis de donner le meilleur de moi-même. Merci pour tout !

Table of contents

List of figures

List of tables

Notations:

A scalar is represented by a lower case character: $y \in \mathbb{R}$

usual iterators: i, j, t

usual constants: n, d, m, l

A function with scalar values is represented by a lower case character with an argument $f(\cdot)$

A vector is represented by a bold character: $\mathbf{x} \in \mathbb{R}^d, \mathbf{x} = [x_1, \dots, x_d]^\top$

A multi-output function is represented by a lower case bold character with an argument $\mathbf{f}(\cdot)$

A matrix is represented by upper case bold character:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,d} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,j} & \cdots & x_{n,d} \end{bmatrix} \in \mathcal{R}^{n \times d}$$

The i^{th} row of a matrix $\mathbf{X}$ is noted $\mathbf{x}^{(i)\top}$

The j^{th} column of a matrix $\mathbf{X}$ is noted $\mathbf{x}_j$

Blackboard bold characters are used for usual measures such as: $\mathbb{E}, \mathbb{KL}, \mathbb{P}$ and for usual spaces: $\mathbb{R}^d$ real space of dimension d, $\mathbb{X}$ input design space, $\mathbb{Y}$ objective space.

Calligraphic characters are used for special objects such as: $\mathcal{GP}$ for Gaussian processes, $\mathcal{P}$ for a defined problem.

Specific sets are represented with upper case characters: P for Pareto set, S for the set of solutions, H for the dominated hyper-volume.

For hierarchical/deep models $\mathbf{H}_{(i)}$ denotes the i^{th} level/layer in the structure.

$\mathbf{H}_{[i],t}^{(j)}$ denotes the t^{th} coordinate of the j^{th} observation in the i^{th} level/layer in the considered structure.

Probability notations:

$\mathbb{E}[\mathbf{w}]$: Expected value of $\mathbf{w}$.

$\mathbb{E}_q[\mathbf{w}]$: Expected value of $\mathbf{w}$ with respect to distribution q.

$p(\mathbf{w})$: Probability density function.

$p(\mathbf{w}|\mathbf{y})$: Conditional probability density function of $\mathbf{w}$ given $\mathbf{y}$.

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: Multivariate Gaussian distribution of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

$\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$: Multivariate Gaussian density function of $\mathbf{w}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

$\phi_{\mathcal{N}(0,1)}, \Phi_{\mathcal{N}(0,1)}$ Probability density function and cumulative density function of the standard Gaussian distribution.

Usual notations:

$\mathbf{X}$: training input data.

$\mathbf{y}$: training output data.

$f(\cdot)$: objective function.

$g_i(\cdot)$: constraint function i.

n: training set size.

d: dimension of the input space.

n_c: number of constraint functions.

n_o: number of objective functions.

n_p: number of solutions in the approximated Pareto front.

n_{fi}: number of fidelity levels.

$k(\cdot,\cdot)$: kernel function.

$\boldsymbol{\theta}$: kernel hyper-parameters.

$k_*(\cdot)$: kernel as a function of distance.

Δ: distance (*e.g.* $\Delta_{\text{Mahalanobis}}$, Δ_{NS}).

Σ: $d \times d$ positive definite matrix.

$\Sigma_{\mathbf{x}}$: $d \times d$ positive definite matrix depending on the position $\mathbf{x}$.

$\hat{f}(\cdot)$: posterior mean function of GP f.

$\hat{s}^2_f(\cdot)$: posterior variance function of GP f.

$\boldsymbol{\psi}(\cdot)$: mapping function.

$\phi(\cdot)$: basis function (*e.g.*, ϕ_{rbf} rbf basis function)

y_{min}: current feasible minimum in the DoE.

S: set of solutions.

P: Pareto front.

$H_{\mathbf{Y}}$: dominated hyper-volume by $\mathbf{Y}$.

$\mathbf{I}_n$: Identity matrix of size n.

$\mathbb{I}[\cdot]$: Indicator function.

$\mathcal{I}$: Improvement

Chapter 1

Introduction

1.1 The challenges in the design of complex systems

In engineering, the design of complex systems, such as spacecraft and aircraft, consists of a long process of analysis and optimization that allows the designer to specify the design variables most adequate to the purpose of the designed system. On of the critical challenges arises in the late phase of this process where high-fidelity models are deployed to integrate all the disciplines of interest simultaneously [Balesdent et al., 2012b]. For instance, in the design of aerospace systems, the multi-disciplinary models include multiple disciplines such as structure, propulsion, aerodynamics, trajectory, and costs (Fig. 1.2). The simultaneous integration of all these disciplines makes it possible to interestingly exploit the different interactions between them. However, this comes with an important computational burden. In fact, one evaluation of such a model comes back to an internal loop between the different disciplines. Furthermore, some disciplines are inherently computationally expensive. For instance, the structure and aerodynamic disciplines can constitute a computational bottleneck due to methods such as finite element analyses [Segerlind, 1976] and computational fluid dynamics [Anderson and Wendt, 1995]. Moreover, these disciplines typically rely on legacy codes that may not provide analytical forms of the functions involved. Therefore, the design of complex systems is based on the analysis and optimization of computationally intensive black-box functions. The optimization is performed with respect to different constraints that express the physics to which the design variables are subjected or some specifications imposed by the designer. Another characteristic of these optimization problems is that usually multiple objectives are optimized. In fact, considering only one

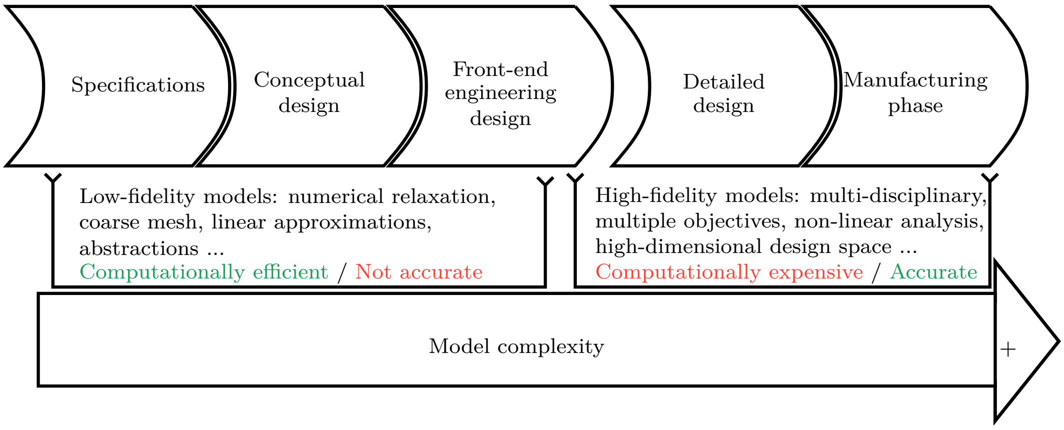

Specifications
Conceptual design
Front-end engineering design
Detailed design
Manufacturing phase
Low-fidelity models: numerical relaxation, coarse mesh, linear approximations, abstractions ...
Computationally efficient / Not accurate
High-fidelity models: multi-disciplinary, multiple objectives, non-linear analysis, high-dimensional design space ...
Computationally expensive / Accurate
Model complexity
+

Fig. 1.1 The different phases for the design of a complex system.

objective may result in limited performance in other disciplines. Hence, the objectives have to be taken into account simultaneously within the optimization problem. The resolution of these optimization problems is difficult in the context of complex systems. In fact, due to the black-box aspect, the exact optimization approaches based on the analytical form of the functions and the gradients is challenging. Moreover, the high-computational cost makes the use of meta-heuristics that require a large number of evaluations not suitable.

In addition to the computationally intensive and black-box aspects, the design of such systems takes into account complex physical phenomena inducing abrupt change of physical properties, here referred as non-stationary behavior. This is usually the case in the modeling of constrained optimization problems. In fact, the objective function and the constraints may have an inconsistent behavior and discontinuities between the feasible and non-feasible regions of the design space [Gramacy and Lee, 2008].

The design of complex systems goes through different phases. The computationally expensive black-box physical models are usually in the late phases of design that are the detailed phase and manufacturing phase (Fig. 1.1). These models are accurate to the detriment of computational cost. The early phases of design, however, are in general characterized by models that are not sufficiently representative of the final system. Such models are called low-fidelity models and have the advantage to be computationally efficient. One of the challenges of the design engineer is to use these different levels of fidelity obtained throughout the design phases to obtain a trade-off between computational costs and accuracy. Considering these different levels of fidelity in a given framework is called multi-fidelity modeling [Fernández-Godino et al., 2016].

Other challenges arise in the design of complex systems. For instance, the imperfect knowledge of the different physics behaviors makes it necessary for the designer to

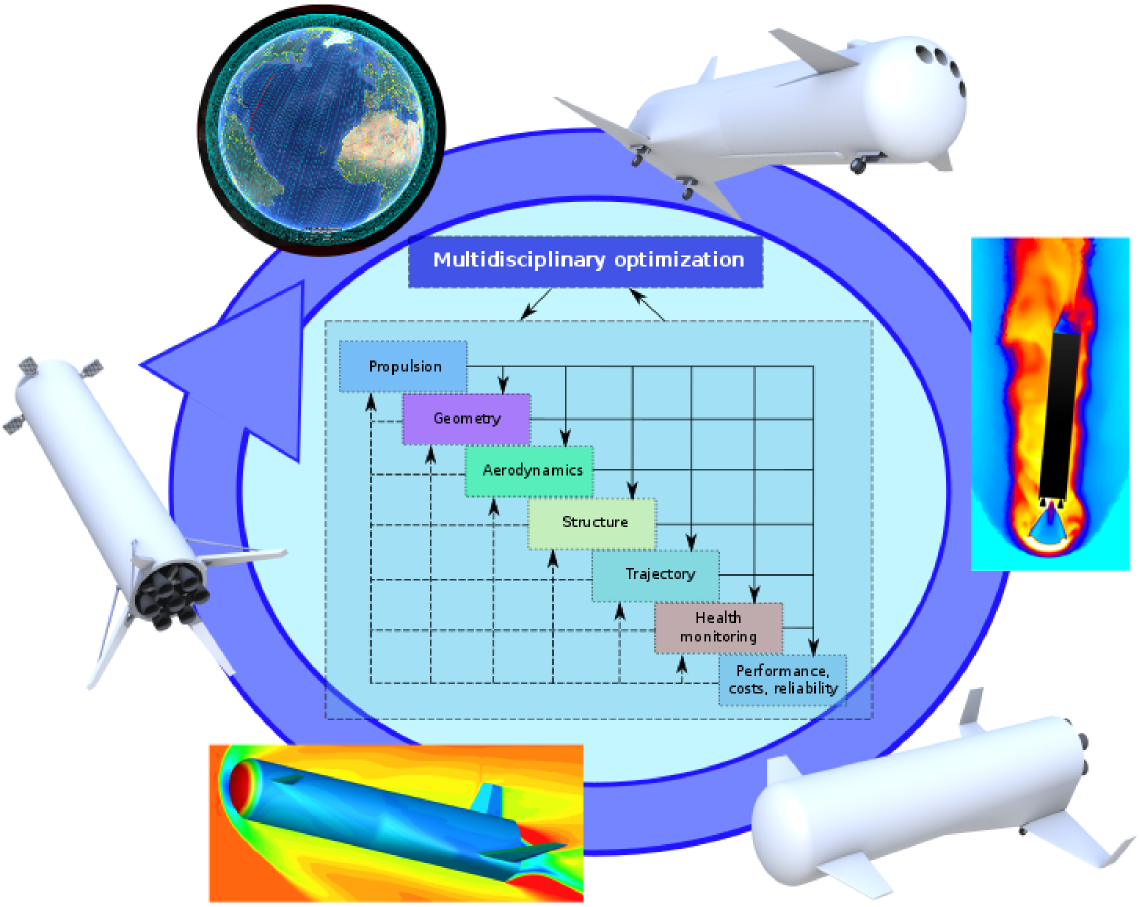


Fig. 1.2 Example of a launch vehicle multi-disciplinary design.

account for different sources of uncertainty. Uncertainty quantification is a key topic in reliability and risk analysis as well as sensitivity analysis [Sudret, 2007; Ghanem et al., 2017; Brevault et al., 2020b]. Another challenge comes from the nature of the design variables. In fact, the design of complex systems may include discrete technological and architectural choices, hence, increasing the difficulty of the optimization. Moreover, the number of design variables involved in the modeling of a complex system is typically large, inducing a high-dimensional design space.

Recently, these various challenges have been partially addressed by machine learning methods. In fact, the recent advances in machine learning have brought design engineering into an era of data-driven approaches.

1.2 Machine learning for the analysis and optimization of complex systems

Machine learning encompasses different methods that allow to discover statistical relationships in observed data and to use these found patterns to predict unobserved data [Bishop, 2006; Alpaydin, 2020]. During this last decade, machine learning has gained large popularity across diverse fields including engineering [Fuge et al., 2014; Mosavi et al., 2017; Bock et al., 2019; Brunton and Kutz, 2019]. This gain in popularity is mainly due to deep learning [Goodfellow et al., 2016] and the astonishing possibilities that it now offers thanks to the advances in high-performance computing and the use of large data sets. In fact, problems that seem not possible to be solved by traditional machine learning models are now classic routines for deep learning models [Lee et al., 2018].

In engineering, machine learning models have been widely used even before the deep learning revolution. Actually, due to the computationally expensive black-box aspect of the physical models involved in the design of complex systems, machine learning models are used to avoid an excessive number of expensive evaluations [Simpson et al., 2001; Wang and Shan, 2007; Forrester et al., 2008]. Regression and classification machine learning models, called also in this context *surrogate models* or *meta-models*, perform supervised learning to predict the behavior of the physical model in new design variables. This is performed by inferring from a set of observations the statistical relationship between the design variables, considered as the inputs of the model, and their evaluations through the engineering model, considered as the outputs. The set of evaluations used to train the machine learning model is obtained *via* design of experiments approaches [Anderson and Whitcomb, 2000]. Diverse machine learning models have been used as surrogate models in the literature. Simple regression models such as linear regression and their polynomial expansion [Ostertagová, 2012], kernel methods as support vector machines [Filippone et al., 2008; Cholette et al., 2017], decision trees [Agrawal et al., 2014], ensemble approaches including random forests and gradient boosting [Moore et al., 2018], artificial neural networks [Rafiq et al., 2001; Simpson et al., 2001], Bayesian models such as Gaussian processes [Kleijnen, 2017], and also recently the deep learning generalization of these models as deep neural networks [Dimiduk et al., 2018; Hegde, 2019] and deep Gaussian processes [Hebbal et al., 2020; Radaideh and Kozlowski, 2020]. The deep learning models, thanks to their increased power of representation emulate highly varying and non-stationary functions.

These surrogate models are used for optimization purposes of the physical model in unconstrained or constrained cases [Jones et al., 1998; Sasena, 2002] and single or multi-objective configurations [Emmerich et al., 2006]. For analysis tasks, surrogate models also provide interesting applications. Actually, machine learning models with multi-source input information called multi-task models have been used to handle the multi-fidelity information obtained during the design process [Kennedy and O'Hagan, 2000; Le Gratiet and Garnier, 2014; Perdikaris et al., 2017; Cutajar et al., 2019]. Moreover, machine learning models and specifically probabilistic machine learning models including Gaussian processes and Bayesian neural networks, are well-suited for representing multiple sources of uncertainty (*e.g.*, the uncertainty on the design variables and the noise in measurement), by physically realistic priors and likelihood distributions. This makes Bayesian models interesting to use for reliability and sensitivity analysis [Dubourg et al., 2011; Sudret, 2012; Nanty et al., 2016]. Some machine learning models have been extended to handle categorical design variables that occur in the design of complex systems [Pelamatti et al., 2019].

Certainly not as much as supervised learning, unsupervised machine learning is also used in design engineering. One of its most popular application is for dimensionality reduction to avoid the curse of dimension. This is achieved by methods such as principal component analysis [Ivosev et al., 2008], factor analysis [Yu et al., 2008] where the design space is projected to a low-dimensional sub-space that is sufficient to explain the statistical relationship between the design variables and their evaluations.

The third class of approaches in machine learning that is semi-supervised learning has been applied to engineering design problems through reinforcement learning [Lee et al., 2019]. In fact, reinforcement learning has successfully been applied to single and multi-objective optimization [Van Moffaert and Nowé, 2014]. Through trial-and-errors, it explores the design space and obtains feed-back on the performance evaluation to find the optimal long-term policy.

These different applications of machine learning to the analysis and optimization problems occurring in the design of complex systems are summarized in Fig. 1.3.

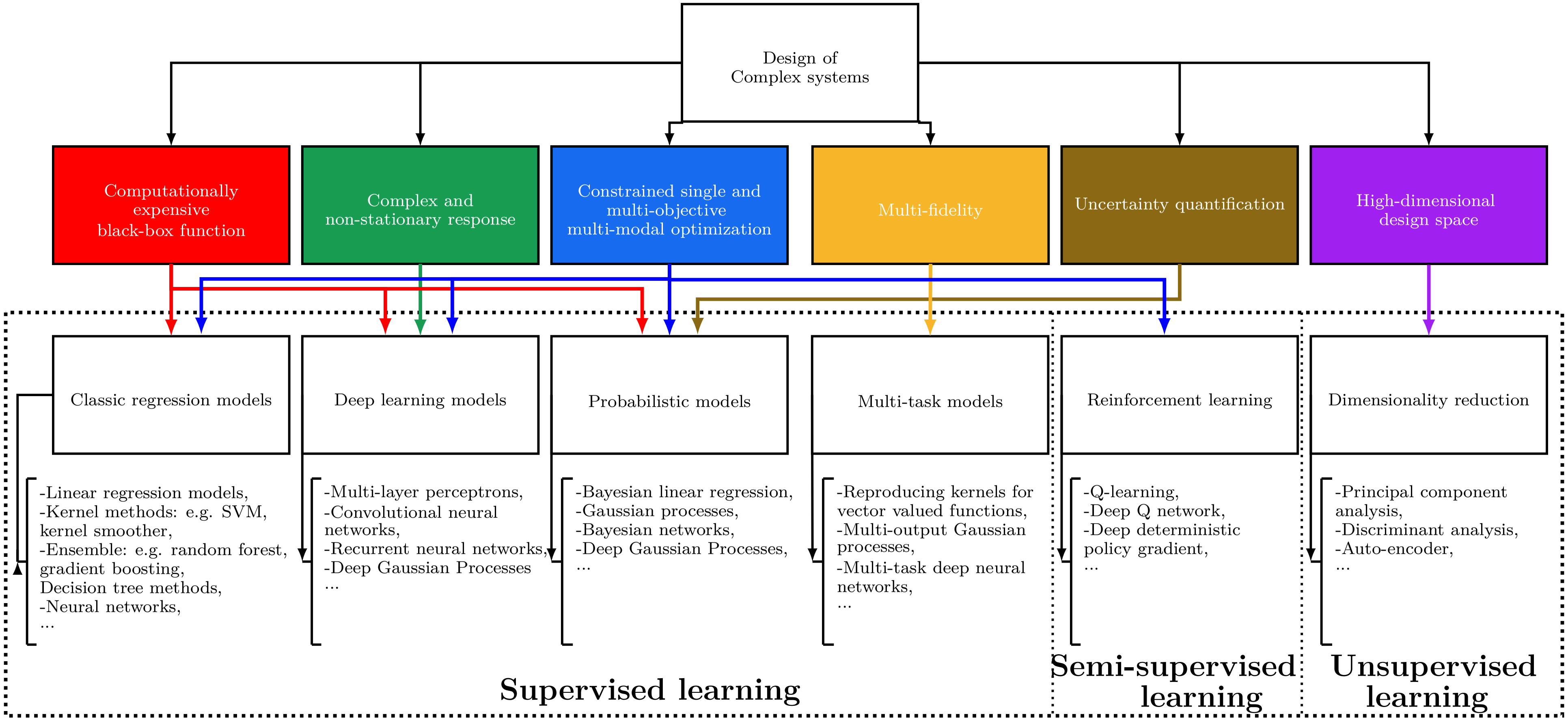


Fig. 1.3 The different machine learning applications in the analysis and optimization of complex systems.

1.3 Motivations and outline of the thesis

A wide range of machine learning methods can be applied in the analysis and optimization of complex systems. In this thesis, the focus is on Gaussian processes [Williams and Rasmussen, 2006], a popular class of Bayesian models, that is extensively used in engineering design. In fact, for optimization of computationally black-box function, multi-fidelity analysis, and uncertainty quantification, Gaussian Processes are one of the most used approaches in the literature [Forrester et al., 2008; Sullivan, 2015; Fernández-Godino et al., 2016]. However, methods based on Gaussian processes still present limitations for handling some specific problems that occur in the design of complex systems. This thesis addresses three of these limitations:

- **Bayesian optimization of non-stationary problems**

  Bayesian optimization [Močkus, 1975] is an iterative algorithm that starts with an initial design of experiments, then, the most promising data-points are added iteratively using an acquisition function. This acquisition function is based on the predictive distribution obtained by a Bayesian model trained on the data set. Gaussian processes as Bayesian models are the classic approach for Bayesian optimization [Jones et al., 1998]. However, Gaussian processes are not suitable to handle non-stationary problems due to stationary covariance functions [Xiong et al., 2007; Gramacy and Lee, 2008]. The existing approaches to overcome this limitation of Gaussian processes such as parametric non-linear mapping, direct formulation of a non-stationary covariance function, and local stationary covariance functions may not be adapted to the scarce data context and the high-dimensionality problems that occur in the design of complex systems.

- **Multi-objective Bayesian optimization with correlated objectives**

  Multi-objective Bayesian optimization is the extension of Bayesian optimization in the case of multiple objectives [Emmerich et al., 2006]. It considers for each objective an independent Bayesian model, then a multi-objective infill criterion such as the expected hyper-volume improvement [Emmerich and Klinkenberg, 2008], also computed with the assumption of independence between objectives, is used to add the most promising data-point. However, training the models independently does not take into account a potential correlation between the objectives. Actually, in a multi-objective context, the different objectives are usually antagonistic. For instance, in the design of a space launch vehicle, the gross lift-off weight and the change in velocity ΔV are negatively correlated.

Therefore, modeling independently each objective in the context of multi-objective Bayesian optimization may not take into account all the information provided by the data.

- **Multi-fidelity analysis for problems with different input space domain definitions**

  Different multi-fidelity models are based on multi-source Gaussian processes that model jointly the different levels of fidelity [Fernández-Godino et al., 2016]. This allows enriching the high-fidelity model with low-fidelity data to improve its prediction accuracy. These models consider the design space identically defined for the different fidelity physical models. However, in the design of complex systems, this is not usually the case. Actually, in the low-fidelity levels for the sake of simplification of the physical model, some design variables may be abstracted or a different parameterization may be used. This yields to a different input space for each fidelity physical model, hence, the classic Gaussian process multi-fidelity models can not be directly used.

Thesis objective

This thesis aims to develop new algorithms and models based on the hierarchical generalization of Gaussian processes called deep Gaussian processes [Damianou and Lawrence, 2013] to overcome the limitations of regular Gaussian processes in the analysis and optimization of complex systems

This is accomplished through contributions at three levels:

- A framework for the coupling between Bayesian optimization and deep Gaussian processes is proposed to handle non-stationary problems. This framework adapts deep Gaussian processes from a training and architecture perspectives to the iterative structure and infill criteria of Bayesian optimization. This framework has been initially proposed in a conference paper and developed more thoroughly in a journal article:

  - Efficient global optimization using deep Gaussian processes. Hebbal, A., Brevault, L., Balesdent, M., Taibi, E. G., & Melab, N. In IEEE Congress on evolutionary computation (CEC) 2018 (pp. 1-8).

  - Bayesian optimization using deep Gaussian processes with applications to aerospace system design. Hebbal, A., Brevault, L., Balesdent, M., Talbi, E. G., & Melab, N. In Optimization and Engineering, 1-41. Springer, 2020.

  A generalization of this coupling to non-stationary problems in the context of multiple objectives has been addressed in a conference paper:

  - Multi-objective optimization using deep Gaussian processes: application to aerospace vehicle design. Hebbal, A., Brevault, L., Balesdent, M., Talbi, E. G., & Melab, N. In AIAA Scitech 2019 Forum (p. 1973).

- A multi-objective deep Gaussian process-based model is developed to model jointly multiple objectives in the context of multi-objective optimization. This model exhibits complex correlations between the different objectives in order to improve the prediction accuracy in the objective space. Moreover, a computation approach is proposed to compute the expected hyper-volume improvement without the assumption of independence between the objectives and also for non-Gaussian predictive distributions. This contribution has been partly communicated in a conference:

  - A deep Gaussian process based model for multi-objective optimization. Hebbal, A., Brevault, L., Balesdent, M., Talbi, E., & Melab, N. In the 13th International Conference on Multiple Objective Programming and Goal Programming (MOPGP) 2019.

- At the multi-fidelity level, firstly, a more elaborated training approach is developed for the existing multi-fidelity deep Gaussian process model [Cutajar et al., 2019] that improves its learning capacity. Next, an extensive analytical and aerospace benchmark is used to evaluate the different Gaussian process-based multi-fidelity approaches. The second part of the multi-fidelity contributions addresses the issue of different input space domain definitions for the different fidelities. For that, a multi-fidelity deep Gaussian process model for different input space domain definitions is developed. This novel model embeds a Bayesian non-parametric mapping between the input spaces within the multi-fidelity model, allowing a joint optimization of the multi-fidelity model and the input mapping. These contributions have been proposed through one NeurIPS workshop and two journal papers:

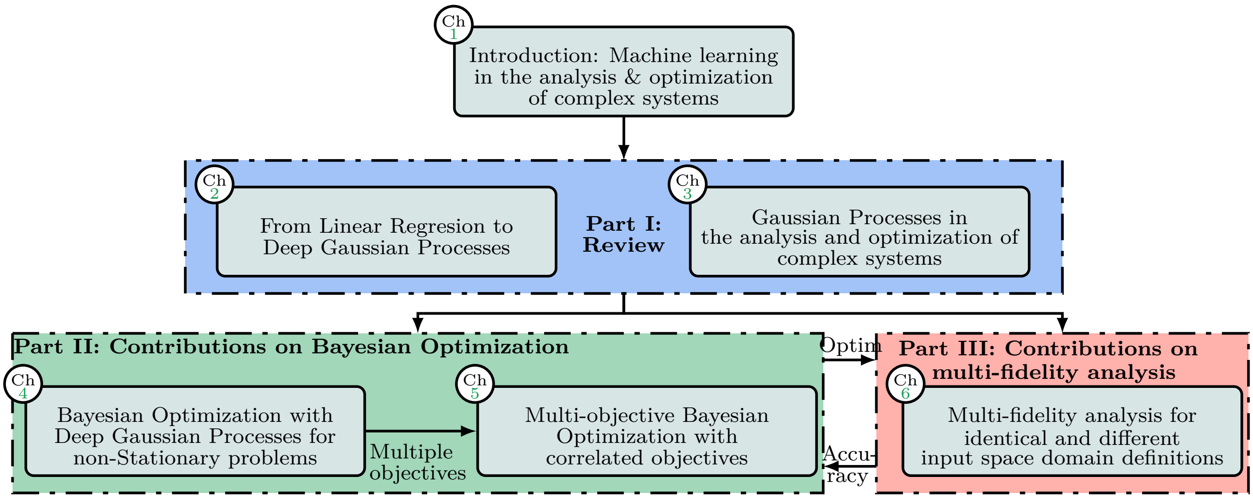

Ch 1
Introduction: Machine learning in the analysis & optimization of complex systems
Ch 2
From Linear Regresion to Deep Gaussian Processes
Part I: Review
Ch 3
Gaussian Processes in the analysis and optimization of complex systems
Part II: Contributions on Bayesian Optimization
Ch 4
Bayesian Optimization with Deep Gaussian Processes for non-Stationary problems
Multiple objectives
Ch 5
Multi-objective Bayesian Optimization with correlated objectives
Optim
Accuracy
Part III: Contributions on multi-fidelity analysis
Ch 6
Multi-fidelity analysis for identical and different input space domain definitions

Fig. 1.4 Thesis structure.

- Multi-fidelity modeling using DGPs : Improvements and a generalization to varying input space dimensions. Hebbal, A., Brevault, L., Balesdent, M., Talbi, E., & Melab, N. 4th workshop on Bayesian Deep Learning (NeurIPS 2019).
- Multi-fidelity modeling with different input domain definitions using deep Gaussian processes. Hebbal, A., Brevault, L., Balesdent, M., Talbi, E., & Melab, N. Structural and Multidisciplinary Optimization Journal. (*Accepted*).
- Overview of Gaussian process based multi-fidelity techniques with variable relationship between fidelities, application to aerospace systems. Brevault, L., Balesdent, M., & Hebbal, A. Aerospace Science and Technology, 106339. Vol.107, Elsevier (2020)

The efficiency of each algorithm and model developed is assessed and compared to the existing approaches on analytical and aerospace engineering design problems in a numerical experiment section.

The structure of this manuscript (illustrated in Fig. 1.4) is organized in three main parts. The first part is concerned with reviewing the background on which the contributions are based and also the state-of-the-art approaches in the optimization and analysis of complex systems. This first part is constituted of two chapters. Its first chapter (Chapter 2) introduces the cornerstone of the thesis that is the deep Gaussian process model through a pedagogical path starting from Bayesian linear regression until reaching the different inference approaches and applications of deep

Gaussian processes in the literature. The next chapter (Chapter 3) is devoted to the Gaussian process-based approaches in the optimization and analysis of complex systems. This chapter displays the existing approaches based on Gaussian processes to handle non-stationarity, multi-objective Bayesian optimization, and multi-fidelity analysis.

Part II and part III of this manuscript are the contribution parts of the thesis. Part II is organized into two chapters devoted to the contributions on Bayesian optimization. Its first chapter (Chapter 4) proposes a framework for coupling deep Gaussian processes and Bayesian optimization in order to address the non-stationarity limitations of regular Gaussian processes. While its second chapter (Chapter 5) concerns the contributions on multi-objective Bayesian optimization taking into account the potential correlation between objectives.

Part III is devoted to the contributions on multi-fidelity analysis. A single chapter (Chapter 6) constitutes this part. It is organized into two main sections that are the contributions on multi-fidelity analysis with identically defined input spaces and the contributions on multi-fidelity analysis with different input space definitions.

The list of contributions published as journal articles and book chapters and the list of communications during the thesis are presented in Appendix A. The standard Gaussian identities used for Gaussian processes are displayed in Appendix B. The analytical problems used for the numerical experiments are presented in Appendix C. The numerical setup used throughout this thesis is detailed in Appendix D.

Part I

Overview of Gaussian Processes and derived models, applications to single and multi-objective optimization, and multi-fidelity modeling

Chapter 2

From Linear models to Deep Gaussian Processes

"The Future of Statistics: A Bayesian 21st Century"
D. V. Lindley (1975)

Chapter goals

- Introduction of the concepts of Bayesian modeling and review of Bayesian inference approaches.
- Introduction of Gaussian processes and taxonomy of sparse adaptation of Gaussian processes.
- Introduction of deep Gaussian processes and review of deep Gaussian process inference approaches and applications.

$\mathcal{CH}_2$

This chapter of literature-review serves as an introduction to the methodological means that are used to solve the problems addressed in this thesis. A pedagogical path is followed, starting from a basic linear regression model until reaching the cornerstone of this thesis: deep Gaussian processes. For that, the Bayesian modeling perspective is motivated in Section 2.1, with an emphasis on the different inference approaches developed in the literature. This first section allows us to go from a frequentist linear regression model to a Bayesian one. Then, Gaussian processes are introduced in Section 2.2 as a non-parametric extension of Bayesian linear regression. This second section aims to describe the different concepts of a Gaussian process, its limits and also its relations with other machine learning models. From there, deep Gaussian processes, a layer-wise hierarchical generalization of Gaussian processes are presented in Section 2.3. This section is devoted to the definition of deep Gaussian processes, as well as to the different inference approaches developed for this model, and to its applications in the literature. This section sets the theoretical basis of the core elements of this thesis that are deep Gaussian processes. In fact, the methods developed in Chapter 4, Chapter 5, and Chapter 6 for the analysis and optimization of complex systems are all based on deep Gaussian processes. Moreover, these contributions are put into perspective within the different applications of this model in the literature.

2.1 Bayesian modeling

The task of a model in machine learning, in its essence, is to predict a response of interest given available data (called observations or training data). In contrast with a physical model, which is based on physical equations to give a response, a machine learning model (henceforth referred simply as model) executes the prediction task based on statistical patterns deduced (*inferred*) from the available data. Therefore, there is no guarantee that the response of the model for a new set of data would be accurate with respect to the exact response of interest. In that case, an important information for the user of the model is the degree of precision of this prediction. However, it is not an information that is intrinsic to classical machine learning models. Consider for instance a linear regression model (Fig. 2.1). The prediction obtained using this model does not match the exact function and the model does not give information about when its prediction is close to the exact function (over-confident prediction) and when it is far from it (under-confident prediction). However, a desirable output of the model would be a degree of belief associated to its prediction, which may depend on the spatial distribution of the training data in the input space. In fact, a prediction at a new data-point that is similar to a set of training data-point would have a high degree of belief (a low level of uncertainty). While, for a new data-point which is completely different than the training data-set, its prediction would have a low degree of belief associated to it (a high level of uncertainty). This type of uncertainty is due to our lack of knowledge (*episteme* in latin) about the latent (non-observed) function that we aim to approximate, and is therefore called epistemic uncertainty. In the same category of uncertainty, there is the model uncertainty *i.e.* uncertainty on its parameters and uncertainty on its structure to best explain the data. These uncertainties are reduced when a better knowledge of the latent function is acquired by gathering more training data. Another type of uncertainty is the one due to aleatoric sources as the error in measurements, this type of uncertainty induces noise in the training data.

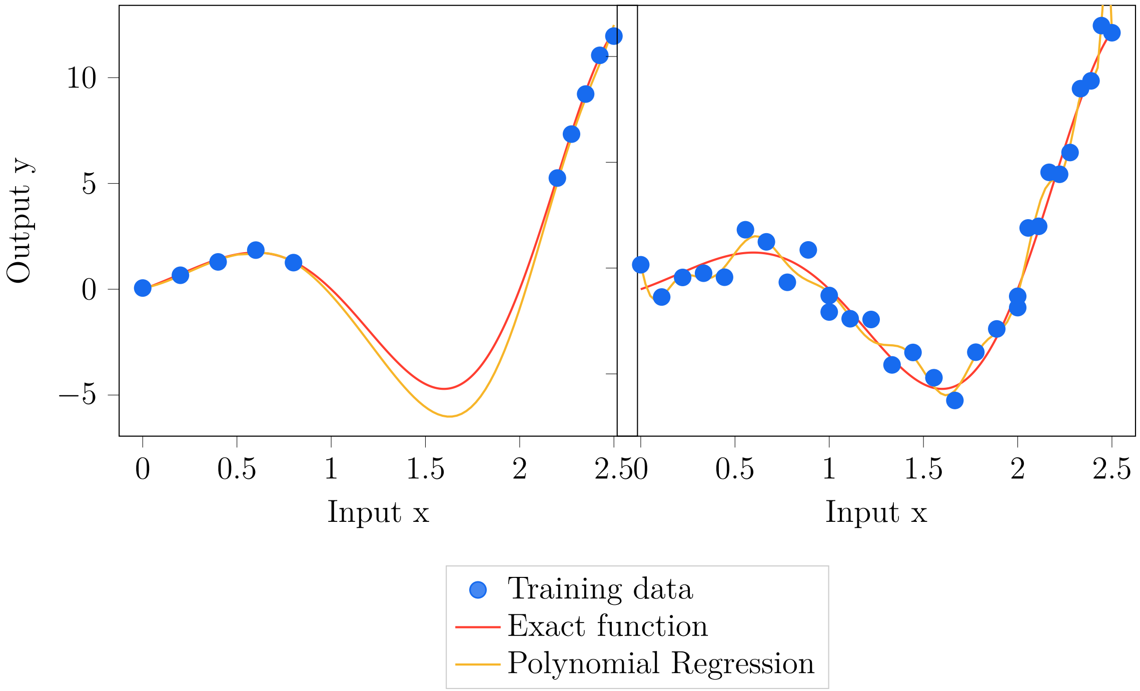


Fig. 2.1 A linear regression using the canonical polynomial basis function of degree 10 trained using the ordinary least square estimate (Eq. (2.6)). (left), There are no training data corresponding to the input range [1,2], however the model gives prediction without information about its confidence. (right) by introducing a noise in the observations, the prediction over-fit (a complex fit of the training data). This is due to the point estimate of the parameters that best explains the data.

One way to express all these forms of uncertainty is to rely on the tools of probability theory [Jaynes, 2003; Murphy, 2012; Ghahramani, 2015]. The basic idea in a nutshell, is that given a model $\mathcal{M}$, each source of uncertainty (*i.e.* the parameters $\mathbf{w}$, the structure, the noise) is expressed with a probability distribution. Then, based on the training data $(\mathbf{X}, \mathbf{y})$, these distributions are updated by applying Bayes rule. Finally, the distribution at unobserved locations $\mathbf{X}^*$ is predicted using simple sum and product rules of probability (Fig. 2.2). This approach to modeling with a probabilistic perspective is called Bayesian modeling.

2.1.1 An illustrative integration of Bayesian concepts into a model

In this section, in order to introduce some definitions and concepts with an illustrative example, the Bayesian concepts are used with a linear regression model.

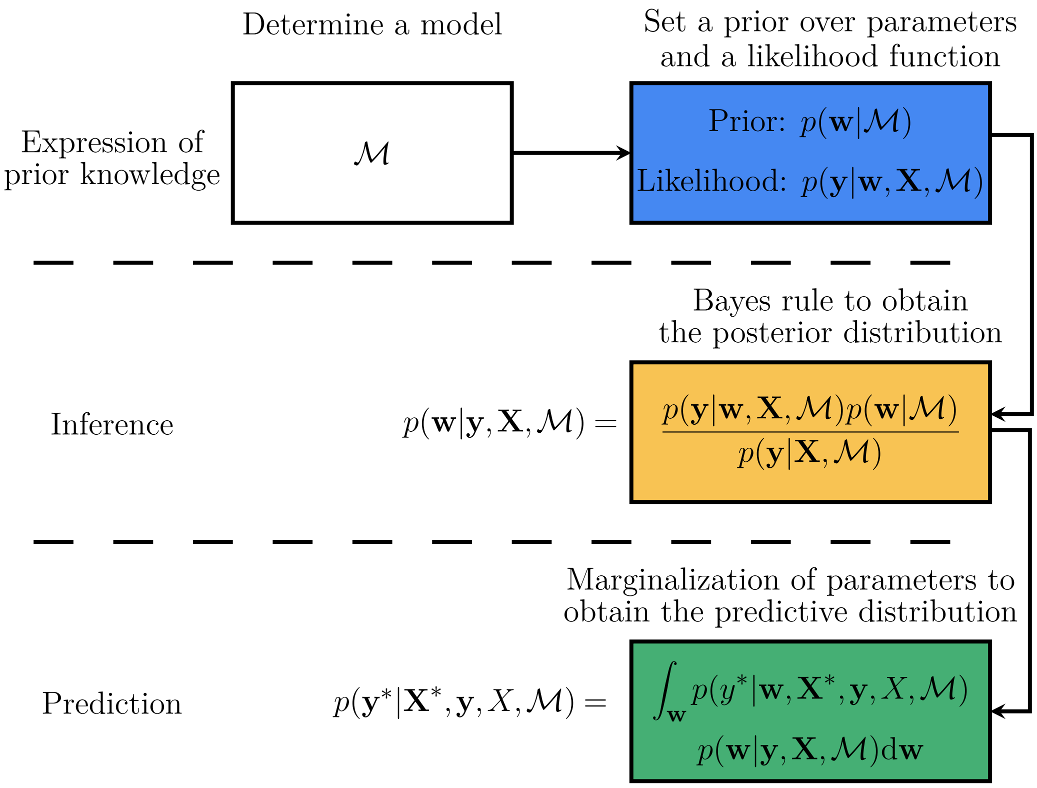

Determine a model
Set a prior over parameters and a likelihood function
Expression of prior knowledge
$\mathcal{M}$
Prior: $p(\mathbf{w}|\mathcal{M})$
Likelihood: $p(\mathbf{y}|\mathbf{w},\mathbf{X},\mathcal{M})$
Bayes rule to obtain the posterior distribution
Inference
$p(\mathbf{w}|\mathbf{y},\mathbf{X},\mathcal{M}) = \frac{p(\mathbf{y}|\mathbf{w},\mathbf{X},\mathcal{M})p(\mathbf{w}|\mathcal{M})}{p(\mathbf{y}|\mathbf{X},\mathcal{M})}$
Marginalization of parameters to obtain the predictive distribution
Prediction
$p(\mathbf{y}^*|\mathbf{X}^*,\mathbf{y},X,\mathcal{M}) = \int_{\mathbf{w}} p(y^*|\mathbf{w},\mathbf{X}^*,\mathbf{y},X,\mathcal{M})\, p(\mathbf{w}|\mathbf{y},\mathbf{X},\mathcal{M})\mathrm{d}\mathbf{w}$

Fig. 2.2 General framework of a Bayesian regression model.

Consider a regression problem $\mathcal{P}_{\text{reg}}$ defined by the couple of training inputs/outputs $(\mathbf{X},\mathbf{y})$, the size of the data-set (number of observations) n, and the dimension of the input data (number of features) d.

The maximum likelihood estimate procedure

A linear regression model $\mathcal{M}$ with a basis function expansion and a Gaussian noise is defined as:

$$y(\mathbf{x}) = \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}) + \epsilon \tag{2.1}$$

where $\mathbf{w}$ is the vector of parameters of size m, $\boldsymbol{\phi}(\mathbf{x})$ is the vector of basis functions of size m such as polynomial or multivariate Gaussian basis functions, and ϵ is a white Gaussian noise with variance σ^2 *i.e.* $\epsilon \sim \mathcal{N}(0,\sigma^2)$. For the sake of simplicity and for illustrative purposes σ^2 is assumed known. A likelihood function $p(\mathbf{y}|\mathbf{X},\mathbf{w},\sigma^2,\mathcal{M})$ is defined as the distribution of the observations conditioned on the parameters of the model. Assuming independent and identically distributed $(i.i.d)$ training data, the likelihood can be written as:

$$p(\mathbf{y}|\mathbf{X},\mathbf{w},\sigma^2) = \prod_{i=1}^{n} \mathcal{N}\left(y^{(i)}|\mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}^{(i)}), \sigma^2 \mathbf{I}_n\right) \tag{2.2}$$

where $\mathbf{I}_n$ is the identity matrix of size n and the mention of the dependence on the model $\mathcal{M}$ is dropped for notation simplicity. Maximizing this likelihood function with respect to the parameters of the model $\mathbf{w}$ yields to their estimation. This procedure is called a Maximum Likelihood Estimate (MLE):

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} \mathcal{N}\left(y^{(i)}|\mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}^{(i)}), \sigma^2 \mathbf{I}_n\right) \tag{2.3}$$

To simplify the expression of the Gaussian density, the likelihood is composed with the natural logarithm to obtain the log likelihood which conserves the maximum since the logarithm is a monotonically increasing function:

$$\begin{aligned} \hat{\mathbf{w}} &= \underset{\mathbf{w}}{\operatorname{argmax}} \ \log\left(p\left(\mathbf{y}|\mathbf{X},\mathbf{w},\sigma^2\right)\right) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} \ -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y^{(i)} - \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}^{(i)})\right)^2 \end{aligned} \tag{2.4}$$

Therefore, the maximization of the likelihood comes back to the minimization of the Residual Sum of Squares (RSS):

$$\mathrm{RSS} = \sum_{i=1}^{n}(y^{(i)} - \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}^{(i)}))^2 \tag{2.5}$$

The RSS in the case of linear regression is convex, therefore, the minimization can be performed by equalizing the gradients of the RSS to zero, which gives the ordinary least square estimate:

$$\mathbf{w} = (\boldsymbol{\Phi}^\intercal \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\intercal \mathbf{y} \tag{2.6}$$

where $\boldsymbol{\Phi}$ corresponds to the values of all basis functions for all the training inputs *i.e.* $\boldsymbol{\Phi}_{i,j} = \phi_j\left(x^{(i)}\right)$. Hence, a point estimate of the linear regression model parameters is obtained using a MLE procedure. This estimate has some interesting properties, such as consistency and analytical tractability [Wald, 1949]. However, being a frequentist approach, the MLE considers the parameters fixed and does not include an uncertainty quantification measure that provides information about where the model is confident and where it is not. Moreover, since only the parameter values that best explain the training data are chosen, this approach is prone to *over-fitting* *i.e.* a complex fit of

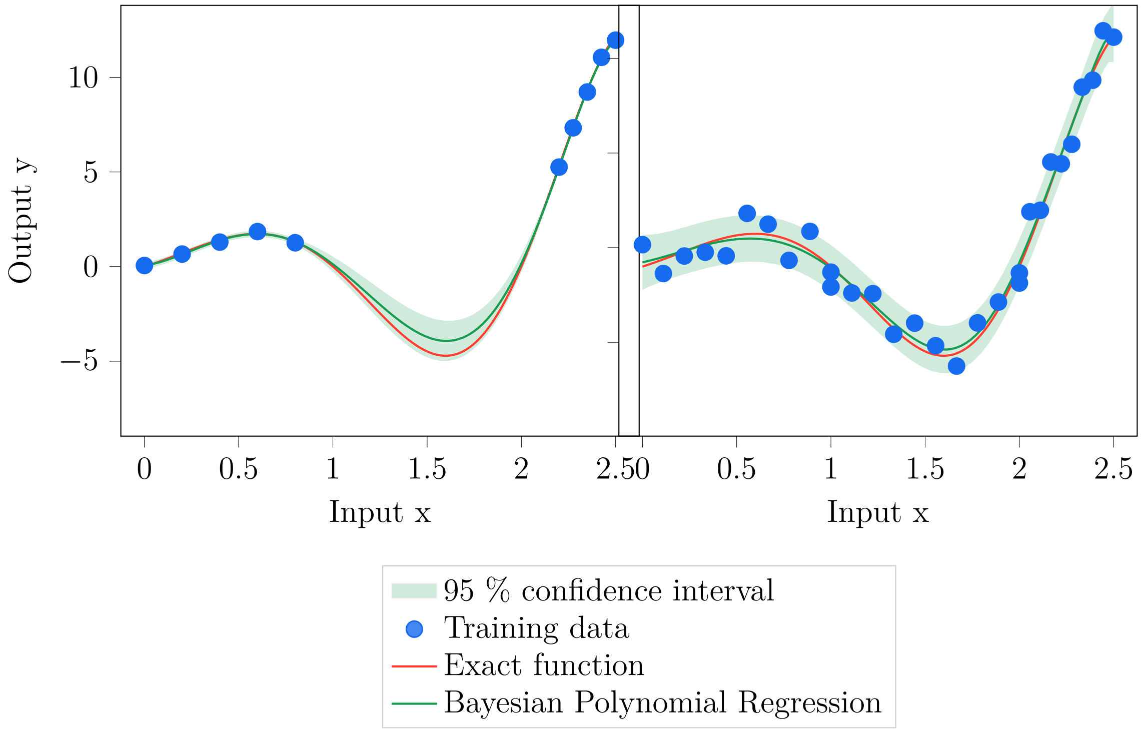


Fig. 2.3 A Bayesian linear regression using the canonical polynomial basis function of degree 10. The prediction is obtained by marginalizing out the weights following Eq. (2.8). (left) the prediction obtained by the model is associated with an uncertainty estimate. This uncertainty increases when the prediction is not confident, hence taking into account the lack of information about the input range [1,2]. (right) the Bayesian model avoids over-fitting by averaging over all possible parameter values.

the training data which breaches the trade-off training error (error on the fit of the training data) and generalization error (error on the fit at unobserved locations) in favor of the former (Fig. 2.1). Other pathological behaviors of frequentist estimators such as the confidence interval construction and the violation of the likelihood principle are intensively discussed in [Lindley, 1972, 1975; Berger et al., 1988; Jaynes, 2003].

The Bayesian perspective

In contrast with a frequentist approach, a Bayesian approach considers the model parameters as random variables rather than fixed real values. The steps of the Bayesian approach summarized in Fig. 2.2 are followed in this section. First, the determination of the prior knowledge has to be expressed through the ***likelihood*** and the ***prior*** distributions.

Likelihood

The **likelihood** encodes the uncertainty of the data that is not explained by the parameters of the model *e.g.*, the noise of the observations. A Gaussian distribution is often used:

$$p(\mathbf{y}|\mathbf{X},\mathbf{w}) = \mathcal{N}(\mathbf{y}|\mathbf{\Phi}\mathbf{w},\sigma^2\mathbf{I}_n)$$

The Gaussian form is practical for analytical tractability as it will be illustrated in the derivation of the posterior. However, for specific cases, other distributions are used. For instance, to gain robustness in the case of outliers in the data, heavy-tailed distributions are preferred such as t-student and to handle noise depending on the location of the data $\mathbf{x}$, *heteroscedasticity* is introduced *i.e.* a noise variance function $\sigma^2(\cdot)$ depending on the input location.

Prior

The prior distribution is what actually differentiates the Bayesian from the frequentist. The **prior** over the parameters $\mathbf{w}$ encodes our beliefs *a priori* to the observations. Even when there is no strong belief *a priori*, the prior has practical advantages. For instance, the prior usually expresses a preference for simpler models and therefore avoids over-fitting (Occam's razor effect [Jefferys and Berger, 1992; Wolpert et al., 1995; Rasmussen and Ghahramani, 2001; Ghahramani, 2005]). Nevertheless, certain strong assumptions might drive the model from the desirable fit of the data. Hence, flexible distributions with a high entropy (important variance) are usually preferred. In the following, a Gaussian prior distribution is used on the parameters $\mathbf{w}$:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0,\mathbf{\Sigma}_{\text{prior}})$$

More sophisticated non-informative priors such as Jeffreys priors ([Jaynes, 1968; Ibrahim and Laud, 1991; Tuyl et al., 2008]) can be used. Heavy tails distributions can be naturally preferred when there is important uncertainty on the prior. Also, to conceive more flexible priors, parameterized priors are used. Those parameters are called hyper-parameters and can be learned via a maximum likelihood estimate on the data (Empirical Bayes approach [Jamil and ter Braak, 2012]). In this case, the prior depends on the training data. A more Bayesian treatment considers a prior on the prior, *i.e.* a prior over the hyper-parameters (Hierarchical Bayes approach [Allenby et al., 2005]).

Inference

Bayes rule is the bread-and-butter of Bayesian statistics. In the inference step, the posterior is inferred using Bayes rule.

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{X},\mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \\
&= \frac{p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w}}
\end{aligned}
\tag{2.7}
$$

Inference of the parameters comes back to the transformation of the prior information (*prior*) by combining it with the likelihood of the data (*likelihood*) and taking into account all the possible outcomes of the parameters (*marginal likelihood* also called the *evidence*) into a posterior knowledge (*posterior*).

The marginal likelihood and the posterior are analytically tractable for a Gaussian likelihood and a Gaussian prior (see Appendix B):

$$
p(\mathbf{w}|\mathbf{X},\mathbf{y}) = \mathcal{N}\left(\frac{1}{\sigma^2}\boldsymbol{\Sigma}_{\text{posterior}}\boldsymbol{\Phi}\mathbf{y}, \boldsymbol{\Sigma}_{\text{posterior}}\right) \tag{2.8}
$$

with $\boldsymbol{\Sigma}_{\text{posterior}} = \left(\frac{1}{\sigma^2}\boldsymbol{\Phi}\boldsymbol{\Phi}^\intercal + \boldsymbol{\Sigma}_{\text{prior}}^{-1}\right)^{-1}$. In this case, the Gaussian prior is said to be conjugate for the Gaussian likelihood *i.e.* the posterior and the prior have the same form for the chosen likelihood. However, for more sophisticated priors this may not be the case, and in the non-conjugate case the analytic tractability is lost and approximation approaches are used (Section 2.1.2).

Prediction

For the prediction task at a location $\mathbf{x}^*$, a marginalization over all the possible values of the parameters $\mathbf{w}$ is done in order to obtain a posterior prediction $y^*|\mathbf{x}^*,\mathbf{X},\mathbf{y}$ with:

$$
\begin{aligned}
p(y^*|\mathbf{x}^*,\mathbf{X},\mathbf{y}) &= \int p(y^*|\mathbf{w},\mathbf{x}^*,\mathbf{X},\mathbf{y})p(\mathbf{w}|\mathbf{X},\mathbf{y})\mathrm{d}\mathbf{w} \\
&= \int \mathcal{N}\left(y^*|\mathbf{w}^\intercal\boldsymbol{\phi}(\mathbf{x}^*),\sigma^2\right)\mathcal{N}\left(\mathbf{w}\middle|\frac{1}{\sigma^2}\boldsymbol{\Sigma}_{\text{posterior}}\boldsymbol{\Phi}\mathbf{y},\boldsymbol{\Sigma}_{\text{posterior}}\right)\mathrm{d}\mathbf{w} \\
&= \mathcal{N}\left(y^*\middle|\frac{1}{\sigma^2}\boldsymbol{\phi}(\mathbf{x}^*)^\intercal\boldsymbol{\Sigma}_{\text{posterior}}\boldsymbol{\Phi}\mathbf{y},\sigma^2+\boldsymbol{\phi}(\mathbf{x}^*)^\intercal\boldsymbol{\Sigma}_{\text{posterior}}\boldsymbol{\phi}(\mathbf{x}^*)\right)
\end{aligned}
\tag{2.9}
$$

It is interesting to observe that the variance obtained on the prediction is constituted of two terms. The first term σ^2 is due to the likelihood and it takes into account the noise of the observations. The second term encodes a variance depending on how similar $\mathbf{x}^*$ is to our training data. That second term, is what makes Bayesian statistics interesting in applications where the information about the confidence of a prediction is important (*e.g.*, scarce training data, heterogeneous response). Moreover, by averaging over all possible parameter values and not taking the parameter values that best explain the training data, the Bayesian approach is more robust against over-fitting ([Rasmussen and Ghahramani, 2001; Neal, 2012]).

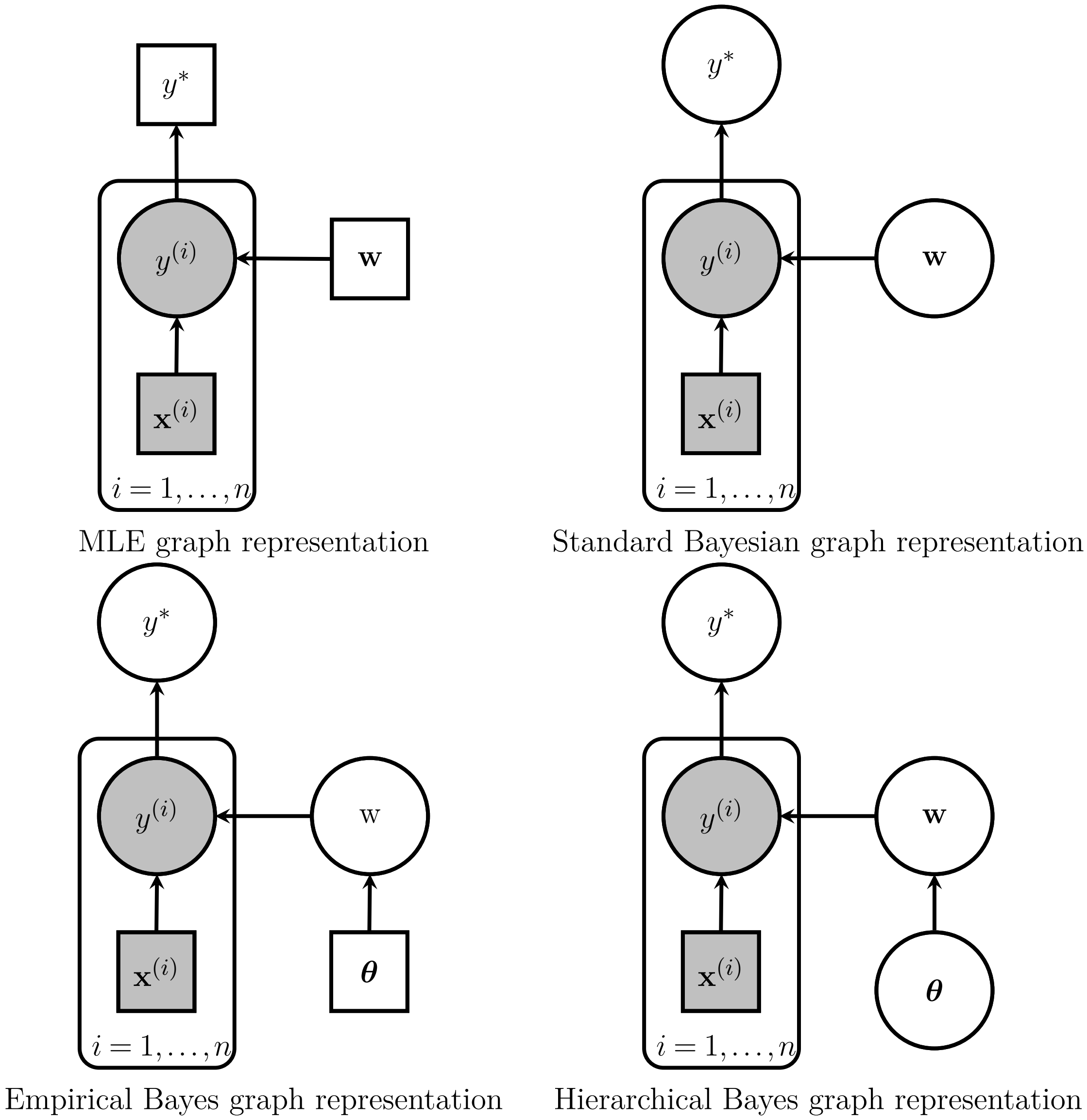
In addition, Bayesian statistics offer a natural way to compare between models based on the Occam's Razor principle: *"All things being equal, the simplest solution tends to be the best one"*. In fact, the marginal likelihood $p(\mathbf{y}|X,\mathcal{M}_i)$ for a model $\mathcal{M}_i$ is a measure that can be used to compare between models $\{\mathcal{M}_1,\ldots,\mathcal{M}_s\}$. It penalizes complex models since their probability density is largely spread over the support of the data, and simple models since their probability density is too narrow ([Jefferys and Berger, 1992; Wolpert et al., 1995; Rasmussen and Ghahramani, 2001; Ghahramani, 2005]).

Graphical representation

A directed graph $G = (\mathcal{V},\mathcal{E})$ can be used to represent the interactions between the different random variables involved in a model [Bishop, 2006]. $\mathcal{V}$ stands for a set of vertices, which correspond to the random variables involved in the model. Observed random variables such as the observations $\mathbf{y}$ are represented with shaded circular nodes, while unobserved (*latent*) random variables such as the parameters $\mathbf{w}$ in Bayesian regression are represented with unshaded circular nodes. Deterministic variables, such as hyper-parameters or observed inputs, are also represented as vertices, but using squared nodes. Dashed squared nodes corresponds to observed deterministic variables, while the unshaded ones corresponds to unobserved deterministic variables. $\mathcal{E}$ stands for the set of directed edges that connect between the vertices. An edge goes from A to B if B is conditioned on A. A missing edge represents conditional independence. Boxes called plates are used to represent *i.i.d.* data with a specified size. For instance, in Fig. 2.4, a graphical representation of the frequentist approach to regression, the classical Bayesian approach, the empirical Bayesian approach, and the Hierarchical Bayesian approach are represented. This graphical representation enables one to synthesize a machine learning model and will be of use when introducing the reviewed Bayesian models in this chapter and also the developed models in the contribution chapters (Chapter 4, Chapter 5, and Chapter 6).

2.1.2 Review on approximate inference techniques

In the illustrative example used previously, the prior was conjugate to the likelihood which is computationally convenient. However, for more sophisticated priors/likelihoods,



MLE graph representation

Standard Bayesian graph representation

Empirical Bayes graph representation

Hierarchical Bayes graph representation

Fig. 2.4 Graph representation of different models. (top left) MLE representation, the parameters $\mathbf{w}$ are point-estimated and so is the prediction y^* . (top right) A standard Bayesian graph representation, the parameters $\mathbf{w}$ are given a prior distribution, which yields to a posterior predictive distribution $p(y^*|\mathbf{y}, \mathbf{X})$. (below left) An empirical Bayes graph representation, the prior on the parameters $\mathbf{w}$ is parameterized and the hyper-parameters $\boldsymbol{\theta}$ as estimated by an MLE procedure. (below right) A hierarchical Bayes graph representation, the prior on the parameters $\mathbf{w}$ is parameterized and the hyper-parameters $\boldsymbol{\theta}$ are given a prior distribution which yields to a fully Bayesian treatment of the model.

it is usually not the case, which makes the integral computation of the marginal likelihood analytically not tractable. In that case, approximation approaches are used. In the next paragraphs, the main approximate inference methods are described.

Maximum a Posteriori

Due to the computational burden of the marginal likelihood, the Maximum A Posteriori (MAP) estimate considers only the mode of the posterior distribution. This is practical since the marginal likelihood does not depend on the parameters $\mathbf{w}$. Hence, the MAP computation comes back to a simple optimization problem:

$$\begin{aligned}\hat{\mathbf{w}}_{\text{MAP}} &= \underset{\mathbf{w}}{\text{argmax}}\ \frac{p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \\ &= \underset{\mathbf{w}}{\text{argmax}}\ p(\mathbf{y}|\mathbf{X},\mathbf{w})p(\mathbf{w})\end{aligned} \tag{2.10}$$

However, the computationally appealing aspect of the MAP is overtaken by its point estimate nature. Indeed, as the MLE, the MAP is a point estimate, and consequently does not provide a measure of uncertainty and might results in over-fitting. Another critic of the MAP is the use of the mode, in fact, the mode is variant to reparametrization and is not a representative statistic unlike the median or the mean ([Murphy, 2012]).

Laplace approximation

Instead of considering a point estimate corresponding to the mode of the posterior (MAP), Laplace approximation ([De Bruijn, 1981; Tierney and Kadane, 1986]) provides an intuitive way to approximate the posterior with a distribution around its mode. To do so, a second order Taylor series expansion is performed around the mode $\hat{\mathbf{w}}_{\text{MAP}}$ of the energy function of the parameters $e(\mathbf{w}) = -\log p(\mathbf{y},\mathbf{w}|\mathbf{X})$:

$$e(\mathbf{w}) \approx e(\hat{\mathbf{w}}) + (\mathbf{w}-\hat{\mathbf{w}}_{\text{MAP}})^{\intercal}\nabla e(\mathbf{w})|_{\hat{\mathbf{w}}_{\text{MAP}}} + \frac{1}{2}(\mathbf{w}-\hat{\mathbf{w}}_{\text{MAP}})^{\intercal}\frac{\partial^2 e(\mathbf{w})}{\partial\mathbf{w}\partial\mathbf{w}^{\intercal}}|_{\hat{\mathbf{w}}_{\text{MAP}}}(\mathbf{w}-\hat{\mathbf{w}}_{\text{MAP}}) \tag{2.11}$$

The first order gradient term is equal to zero when evaluated in the mode, therefore, the equation is simplified to:

$$e(\mathbf{w}) \approx e(\hat{\mathbf{w}}) + \frac{1}{2}(\mathbf{w}-\hat{\mathbf{w}}_{\text{MAP}})^{\intercal}A(\mathbf{w}-\hat{\mathbf{w}}_{\text{MAP}}) \tag{2.12}$$

where $\mathbf{A}$ corresponds to the Hessian matrix of the energy function. By combining the energy function with the exponential function, the posterior distribution can be rewritten as follows:

$$p(\mathbf{w}|\mathbf{y},X) \approx \frac{1}{p(\mathbf{y}|X)} \times p(\mathbf{y},\hat{\mathbf{w}}_{\text{MAP}}) \exp\left(-\frac{1}{2}(\mathbf{w}-\hat{\mathbf{w}}_{\text{MAP}})^{\intercal}\mathbf{A}(\mathbf{w}-\hat{\mathbf{w}}_{\text{MAP}})\right) \tag{2.13}$$

Notice that this expression corresponds to a non-normalized multivariate Gaussian distribution centered around the MAP estimate and with covariance matrix equal to the inverse of the Hessian $\mathbf{A}^{-1}$. Laplace approximation consists in approximating the marginal likelihood $p(\mathbf{y}|X)$ such as the approximated posterior is a normalized multivariate Gaussian. Thus, the following approximation is achieved:

$$p(\mathbf{y}|X) \rightarrow p(\mathbf{y},\hat{\mathbf{w}}_{\text{MAP}})2\pi^{\frac{m}{2}}|\mathbf{A}|^{-\frac{1}{2}}$$

where m is the number of parameters. Accordingly, the obtained posterior is Gaussian. This approximation is based on the fact that with a large amount of data compared to the number of parameters, the posterior of the parameters is approximately Gaussian around the MAP based on Bernstein-von Mises theorem [Freedman et al., 1999]. Consequently, with not enough data the Gaussian approximation around the MAP can be of poor quality. Moreover, as the MAP, it suffers from the use of the approximation around the mode, which may not be a suitable statistic of the distribution. Another problematic aspect of Laplace approximation is the computation of the Hessian that can be computationally expensive, especially in high-dimensional parameter space.

Variational Inference

One way to approximate the true posterior $\tilde{p}(\mathbf{w}) = p(\mathbf{w}|X,\mathbf{y})$, is to choose an approximation $q(\mathbf{w})$ from a family of distributions that is the most similar to the intractable true posterior *i.e.* that minimizes a distance measure between the two distributions. To measure a distance between probability distributions, the Kullback-Leibler (KL) [Kullback and Leibler, 1951] divergence is usually used [Hobson and Cheng, 1973]. KL divergence measures the dissimilarities between two distributions. It can be interpreted as the information lost using the approximation distribution instead of the true posterior. KL is not a symmetrical measure, in fact, it comes back to the computation of an expectation with respect to $\tilde{p}$, $\mathbb{KL}[\tilde{p}||q]$ (forward KL) or $q(\cdot)$, $\mathbb{KL}[q||\tilde{p}]$ (reverse KL). A popular set of approaches called Variational Inference (VI) [Jordan et al., 1999; Blei et al., 2017] consists in minimizing the KL divergence from $\tilde{p}$ to a parameterized distribution $q_{\boldsymbol{\theta}_q}$ (reverse KL). In this context, $q_{\boldsymbol{\theta}_q}$ is called variational distribution and the parameters $\boldsymbol{\theta}_q$ are called variational parameters. $q_{\boldsymbol{\theta}_q}$ is chosen from a family of parameterized distributions $\mathcal{Q}$ and the minimization of the KL divergence comes back to finding the parameters $\hat{\boldsymbol{\theta}}_q$ within this family of distributions that lead to the best matching between the posterior and the variational distributions:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_q &= \underset{\boldsymbol{\theta}_q}{\text{argmin}}\ \mathbb{KL}(q_{\boldsymbol{\theta}_q}||\tilde{p}) \\ &= \underset{\boldsymbol{\theta}_q}{\text{argmin}}\ \int_{\mathbf{w}} q_{\boldsymbol{\theta}_q}(\mathbf{w}) \log \frac{q_{\boldsymbol{\theta}_q}(\mathbf{w})}{\tilde{p}(\mathbf{w})} \mathrm{d}\mathbf{w}\end{aligned} \tag{2.14}$$

In VI, KL is computed from $\tilde{p}$ to q (for the sake of brevity q corresponds to $q_{\boldsymbol{\theta}_q}$) in order to avoid the expectation with respect to the intractable true posterior $\tilde{p}$. The family of distribution $\mathcal{Q}$ is based on a trade-off between expressiveness power and tractability. The exponential family or mixture of exponentials are usually used [Jaakkola and Jordan, 1999; Blei et al., 2017]. However, Eq. (2.14) still got the true posterior term. To completely remove $\tilde{p}$ from the expression, Bayes rule is used, and since the marginal likelihood is constant, it yields to:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_q &= \underset{\boldsymbol{\theta}_q}{\text{argmin}} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{y},\mathbf{w})} \mathrm{d}\mathbf{w} + \int_{\mathbf{w}} q(\mathbf{w}) \log p(\mathbf{y}|X) d\mathbf{w} \\ &= \underset{\boldsymbol{\theta}_q}{\text{argmin}} \int_{\mathbf{w}} -q(\mathbf{w}) \log p(\mathbf{y}|\mathbf{w}) \mathrm{d}\mathbf{w} + \mathbb{KL}[q||p] + C \\ &= \underset{\boldsymbol{\theta}_q}{\text{argmin}}\ \mathbb{E}_q\left[-\log p(\mathbf{y}|\mathbf{w})\right] + \mathbb{KL}[q||p]\end{aligned} \tag{2.15}$$

This minimization can also be interpreted as a maximization of a lower bound on the logarithm of the true marginal likelihood called Evidence Lower Bound (ELBO). In fact, by introducing the approximation q and using Jensen inequality, the following is obtained:

$$\begin{aligned}\log p(\mathbf{y}) &= \log \int_{\mathbf{w}} p(\mathbf{y}|\mathbf{w}) p(\mathbf{w}) \mathrm{d}\mathbf{w} \\ &= \log \int_{\mathbf{w}} \frac{q(\mathbf{w})}{q(\mathbf{w})} p(\mathbf{y}|\mathbf{w}) p(\mathbf{w}) \mathrm{d}\mathbf{w} \\ &\geq \mathcal{L} = \int_{\mathbf{w}} q(\mathbf{w}) \log \left(\frac{p(\mathbf{y}|\mathbf{w}) p(\mathbf{w})}{q(\mathbf{w})} \right) \\ \mathcal{L} &= \int_{\mathbf{w}} q(\mathbf{w}) \log(p(\mathbf{y}|\mathbf{w})) + q(\mathbf{w}) \log \frac{p(q(\mathbf{w})}{q(\mathbf{w})} \mathrm{d}\mathbf{w} \\ &= \mathbb{E}_q\left[\log p(\mathbf{y}|\mathbf{w})\right] - \mathbb{KL}[q||p]\end{aligned} \tag{2.16}$$

where $\mathcal{L}$ is the ELBO. The obtained expression is interesting to analyze. In fact, the maximization of the ELBO (minimization of the KL divergence) comes back to the maximization of a first term, that improves the fit of the data using $q(\mathbf{w})$ and the minimization of a second term, that avoids over-fitting by assuring that $q(\mathbf{w})$ is as similar as possible to the prior (regularization term). Therefore, an automatic Occam's razor effect is intrinsic to VI by penalizing distributions $q(\mathbf{w})$ that are too complex compared to our prior beliefs.

For the form of the approximation distributions, usually, a fully factorized distribution over the parameters is used $q(\mathbf{w}) = \prod_{i=1}^{m} q(w_i)$. This approach is called a mean field approximation. More sophisticated approaches can be used, as hierarchical mean field approximations. For instance, in deep structures, the correlation layer-wise of the parameters are kept in the approximation. However, sophisticated forms of the variational distributions often yield to an analytically non-tractable expectation term. To overcome this issue, different works are built on the idea of Monte-Carlo sampling to approximate the derivative of the expectation [Graves, 2011; Kingma and Welling, 2013; Titsias and Lázaro-Gredilla, 2014; Paisley et al., 2014; Schulman et al., 2015]. A variance analysis of these different estimators used in VI is presented in [Gal, 2016].

Stochastic optimization has been used to optimize the ELBO [Hoffman et al., 2013]. This optimization can be tricky since it is performed with respect to the variational parameters. Specifically, the variational distribution parameter space has a Riemannian structure defined by the Fisher information [Amari, 1998]. In fact, optimizing with respect to the parameters of a distribution makes the parameter space not euclidean, hence, the ordinary gradient is not a suitable direction to follow [Amari, 1998]. In this case, the natural gradient which comes back to the ordinary gradient rescaled by the inverse Fisher information matrix, is the steepest descent direction.

Recently, to overcome the limitation of an explicit form of the variational distribution, [Mescheder et al., 2017] proposed VI with implicit posteriors. More specifically, the variational posterior is defined by a parameterized black-box procedure. The implicit form leads to an intractable KL divergence between the prior and posterior distributions in Eq. (2.15). The idea then, is to express this intractable term as the optimization result of a discriminative network [Goodfellow et al., 2014]. Therefore, the VI optimization problem comes back to two nested optimization problems, which are formulated as a game theory problem [Gibbons, 1994] and where the Nash-equilibrium corresponds to the global optimum of the ELBO.

One of the drawbacks of VI is that it tends to underestimate the variance of the approximate posterior [Murphy, 2012; Blei et al., 2017]. This is due to the objective function of VI (reverse KL) where the expectation is computed under $q(\mathbf{w})$. This objective penalizes regions where $q(\mathbf{w})$ is high, while regions where $q(\mathbf{w})$ is very low have no consequences on the objective function.

Expectation Propagation

In contrast to VI approaches that consider the minimization of the reverse KL divergence, the Expectation Propagation (EP) [Minka, 2001, 2013] aims to minimize the forward KL divergence. Since the forward KL is intractable, EP approximates the joint distribution $p(\mathbf{y}, \mathbf{w})$.

Given the joint distribution $p(\mathbf{y}, \mathbf{w}) = \prod_{i=0}^{n} t_i(\mathbf{w})$ where $t_0(\mathbf{w}) = p(\mathbf{w})$ and $t_i(\mathbf{w}) = p(y^{(i)}|\mathbf{w})$ for $i = 1, \dots, n$. The idea of EM is to approximate each t_i with a $\hat{t}_i$, hence, defining an approximated distribution of the joint distribution that is $q(\mathbf{w}) = \prod_{i=0}^{n} \hat{t}_i(\mathbf{w})$. Each $\hat{t}_i$ for $i = 1, \dots, n$ can be interpreted as an approximate contribution term of data-point i to the likelihood. The family of distributions of $\hat{t}_i$ is the exponential family so that $\int_{\mathbf{w}} q(\mathbf{w}) d\mathbf{w}$ and $\frac{q(w)}{\int_{\mathbf{w}} q(\mathbf{w})\mathrm{d}\mathbf{w}}$ which respectively approximate the marginal likelihood and the posterior distribution are analytically tractable (exponential families are close under multiplication). The approximation is performed through a loop procedure:

1. Initialize $\hat{t}_i$ to constant densities.
2. Until convergence (defined by a threshold change in the parameters of each density), choose a $j \in 1, \dots, n$ then:

   $q \leftarrow \underset{q'}{\operatorname{argmin}} \; \mathbb{KL}[q_{-j} t_j || q']$
3. return q

where $q_{-j}(\mathbf{w}) = \prod_{i \neq j} \hat{t}_i(\mathbf{w})$ is called the cavity distribution. Hence, the update at each iteration allows to update a $\hat{t}_j$ so that $q(\mathbf{w})$ is as close as possible to the true distribution with respect to the term t_j *i.e.* $t_j(\mathbf{w}) \prod_{i \neq j} \hat{t}_i(\mathbf{w})$. EP may seem similar to the Assumed Density Filtering approach (ADF) [Ranganathan, 2004]. However, in ADF the joint distribution $p(\mathbf{y}, \mathbf{w})$ is kept exact and the evidence is approximated iteratively by considering at each iteration another t_i, hence creating an order dependence. EM is free from the order constraint and multiple loops can be performed until reaching convergence.

This loop procedure can be relaxed by considering each data-point contribution to the likelihood equal to the average of contributions $\bar{\hat{t}}$. Thus, the approximation comes back to $q(\mathbf{w}) = t_0 t_i^n$ and instead of optimizing n densities only one is considered. This tied factor constraint is used in Stochastic EP (SEP) [Li et al., 2015]. In addition to the steps described previously in the case of EP, SEP adds a final step consisting in a moment update of the average contribution: $\bar{\hat{t}} \leftarrow \bar{\hat{t}}^{1-\beta}\bar{\hat{t}}^{\beta}_{\text{updated}}$ where β is a chosen learning rate. However, by reducing the complexity of the EP, the tied factor constraint yields to less accurate posterior approximations.

Markov Chain Monte Carlo

The different approaches described above are deterministic approaches (VI can have a stochastic term if the log expectation term is approximated using Monte Carlo approaches). Another set of fully stochastic methods aims to sample from the true posterior using Markov Chain Monte Carlo (MCMC) [Neal, 1993]. MCMC techniques consist in approaching the true intractable distribution $\tilde{p}$ by sampling in the parameter space (Monte Carlo aspect) through a chain of distributions (Markov chain), so that the set of samples obtained at the end of the chain is representative of the true distribution ($\tilde{p}$ is said to be the stationary distribution of the Markov chain). The construction of this chain is based on the definition of a transition from one state of the parameters to the next (in a Markov chain the current state depends only on the previous one). So, the different MCMC approaches differ in the construction of the transition in this chain *i.e.* how to walk in the state space (hence the name random walk). In this construction, some desirable properties are expected from the transition so that from any initial state with enough samples one can approach the true distribution [Neal, 1993]. Gibbs sampling [Gelfand et al., 1990] is a popular sampling approach when the conditional density of each parameter alone can be sampled analytically. For a joint posterior distribution of different variables, it consists in sampling from the posterior distribution of one variable while conditioning on all the others and so on for all the variables using multiple passes through. A more general MCMC approach is the Metropolis Hastings (MH) algorithm [Hastings, 1970; Chib and Greenberg, 1995], the transition is defined by a distribution $\eta(\cdot)$ known as the proposal distribution (for instance, a random Gaussian walk $\eta(\mathbf{w}^{(i+1)}|\mathbf{w}^{(i)}) = \mathcal{N}(\mathbf{w}^{(i+1)}|\mathbf{w}^{(i)}, \boldsymbol{\Sigma})$). It is called proposal distribution because it proposes a move that is not necessarily accepted. In fact, the sample obtained with this jump is accepted with a probability $\frac{p(\mathbf{w}_{\text{new}}|\mathbf{y})}{p(\mathbf{w}_{\text{old}}|\mathbf{y})}$ and is rejected otherwise. Notice that the ratio uses the posterior intractable distribution, however, since it is a ratio, the marginal likelihood is eliminated and it comes back to the ratio of the joint distribution of the prior and the likelihood that is analytically tractable. This walk in the parameter space is interesting, samples are pushed in regions of higher posterior probability *i.e.* regions with important information about the probability density. This intelligent sampling reflects the interest of using MCMC in a high-dimensionnal parameter space where the volume within which lies the samples is much more important than the concentration volume of the target density. Instead of considering a random proposal distribution, Metropolis adjusted Langevin algorithm [Atchadé, 2006] relies on the information about the differential geometry of the target distribution (that are obtainable up to a constant) to make a move based on Langevin dynamics. However, using the information about the gradient directly to sample an approximate from the target distribution will lead to over-sampling around the mode which is sensitive to re-parametrization. In Hamiltonian Monte Carlo (HMC) [Duane et al., 1987; Neal et al., 2011] by an analogy with Hamiltonian mechanics, the walk follows the Hamilton's equations. To do so, the state space is augmented into a phase space, with the introduction of auxiliary momentum vector $\boldsymbol{\theta}$ conjugated to the parameter vector $\mathbf{w}$ that plays the role of the generalized coordinates. Hence, the gradient is used to update the momentum vector instead of the parameter vector. The conditional distribution over the momentum $\tilde{p}(\boldsymbol{\theta}|\mathbf{w})$ (corresponding to the kinetic energy of the system) has to be specified, and also the discretization times used in the Hamiltonian equations which can be tricky *i.e.* if badly chosen can result in poor approximation. Moreover, the gradient of the joint likelihood-prior distribution has to be computed over all the data set, which can be computationally expensive. In order to overcome the computational burden of the gradient computation in HMC, [Chen et al., 2014] used stochastic gradient with some adaptations of HMC to limit the undesirable effects of the noisy gradients.

The different approaches described above are summarized in table 2.1, with an emphasis on the advantages and the drawbacks of each family of methods.

2.2 Gaussian Processes (GPs)

Bayesian linear regression, described previously, uses a parameteric function $\boldsymbol{\phi}(\mathbf{x})^\intercal \mathbf{w}$ and the parameters $\mathbf{w}$ are inferred following Bayesian inference. Instead of considering a distribution over parameters to describe a parametric function, a distribution over function may be used. A distribution over function defines a stochastic process. In this section, Gaussian processes, one of the most popular stochastic processes are introduced.

Table 2.1 Non-exhaustive summary of some approximate Bayesian inference approaches with a brief description of their respective concept, advantages and drawbacks.

Approach	Concept	Advantages	Drawbacks
MAP estimate	Mode of the posterior	Easy to compute	Not Bayesian, pathologies of the mode
Laplace Approximation	Gaussian approximation around the MAP	for $n \to \infty$ posterior $\to$ Gaussian	Computation of the Hessian, scarce data case, pathologies of the mode
Variational Inference	Minimization of the reverse KL	Flexible, ELBO, different variants	Mean-field approximations, under-estimate the variance, log expectation term
Expectation Propagation	Minimization of the direct KL	Highly parallelizable, the exponential family, Fast to converge	Multi-modal posterior, scarce data case, high-dimensionnal problems
Monte-Carlo Markov-Chain	Sampling through a defined Markov chain	Easy to implement, Adaptation to problems	Computationally intensive, stopping criteria

2.2.1 Definitions

A Gaussian Process (GP) [Williams and Rasmussen, 2006] f is a stochastic process indexed by a set $\mathbb{X} \subseteq \mathbb{R}^d$: $\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{X}\}$ such as any finite number of random variables of the process has a joint Gaussian distribution:

$$\forall n' \in \mathbb{N}^*, \forall \mathbf{X}' = \left[\mathbf{x}'^{(1)}, \dots, \mathbf{x}'^{(n')}\right]^\top \in \mathbb{X}, f(\mathbf{X}') \sim \mathcal{N}\left(\mu\left(\mathbf{X}'\right), k\left(\mathbf{X}', \mathbf{X}'\right)\right) \tag{2.17}$$

with $f(\mathbf{X}') = \left[f\left(\mathbf{x}'^{(1)}\right), \dots, f\left(\mathbf{x}'^{(n')}\right)\right]^\top$. A GP is completely defined by its mean and covariance functions and is noted $f(\cdot) \sim \mathcal{GP}(\mu(\cdot), k(\cdot, \cdot))$, with $\mu(\cdot)$ the mean function and $k(\cdot, \cdot)$ the covariance function also called kernel.

GPs are a popular approach for regression and it is used in multiple scientific fields. Especially in the geostatistical community, where it is known as Kriging [Matheron, 1967; Oliver and Webster, 1990]. A graphical representation of a GP is presented in Fig. 2.5.

Given the regression problem $\mathcal{P}_{\text{reg}}$, a GP prior is considered $f(\cdot) \sim \mathcal{GP}\left(\mu(\cdot), k^{\boldsymbol{\theta}}(\cdot, \cdot)\right)$ to express the prior belief of the response. The prior mean function $\mu(\cdot)$ can take a

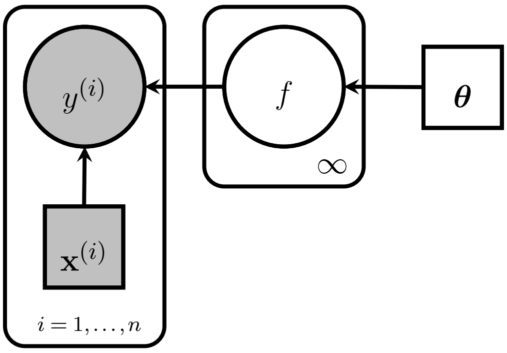


Fig. 2.5 Graphical representation of a Gaussian process with a parameterized prior with parameters $\boldsymbol{\theta}$ estimated using a MLE procedure. The GP prior is defined over an infinite number of inputs, hence, there are infinite GP nodes one for every possible input, and is called Gaussian random field.

form that describes the trend of the exact unknown function if information about the trend is available (universal Kriging) otherwise a constant mean function μ may be considered (ordinary Kriging). The prior covariance function $k^{\boldsymbol{\theta}}(\cdot,\cdot)$ parameterized with a parameter vector $\boldsymbol{\theta}$ represents the prior belief of the unknown function to be modeled (*e.g.*, smoothness, periodicity, stationarity, separability). Samples from two different covariance functions are illustrated in Fig. 2.6. A likelihood function is defined to take into account the noise in the observations such as the relationship between the latent (non-observed) function values $\mathbf{f} = f(\mathbf{X})$ and the observed response $\mathbf{y}$ is given by: $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f},\sigma^2\mathbf{I}_n)$. Using Bayes rule, the marginal likelihood is obtained based on multivariate Gaussian identities (Appendix B):

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}) &= \int_{\mathbf{f}} p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X})\mathrm{d}\mathbf{f} \\ &= \int_{\mathbf{f}} \mathcal{N}\left(\mathbf{y}|\mathbf{f},\sigma^2\mathbf{I}_n\right)\mathcal{N}\left(\mathbf{f}|\mu(\mathbf{X}),k^{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})\right)\mathrm{d}\mathbf{f} \\ &= \mathcal{N}\left(\mathbf{y}|\mu(\mathbf{X}),k^{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I}_n\right) \end{aligned} \tag{2.18}$$

To construct priors that are more adapted to the data, the hyper-parameters of the covariance function $\boldsymbol{\theta}$, the constant mean function μ (ordinary kriging), and the Gaussian noise variance σ^2 are estimated through a maximization of the marginal likelihood (empirical Bayes):

$$\hat{\boldsymbol{\theta}},\hat{\mu},\hat{\sigma} = \underset{\boldsymbol{\theta},\mu,\sigma}{\arg\max}\ \mathcal{N}\left(\mathbf{y}\,\middle|\,\mathbf{1}\mu,k^{\boldsymbol{\theta}}(\mathbf{X},\mathbf{X})+\sigma^2\mathbf{I}_n\right) \tag{2.19}$$

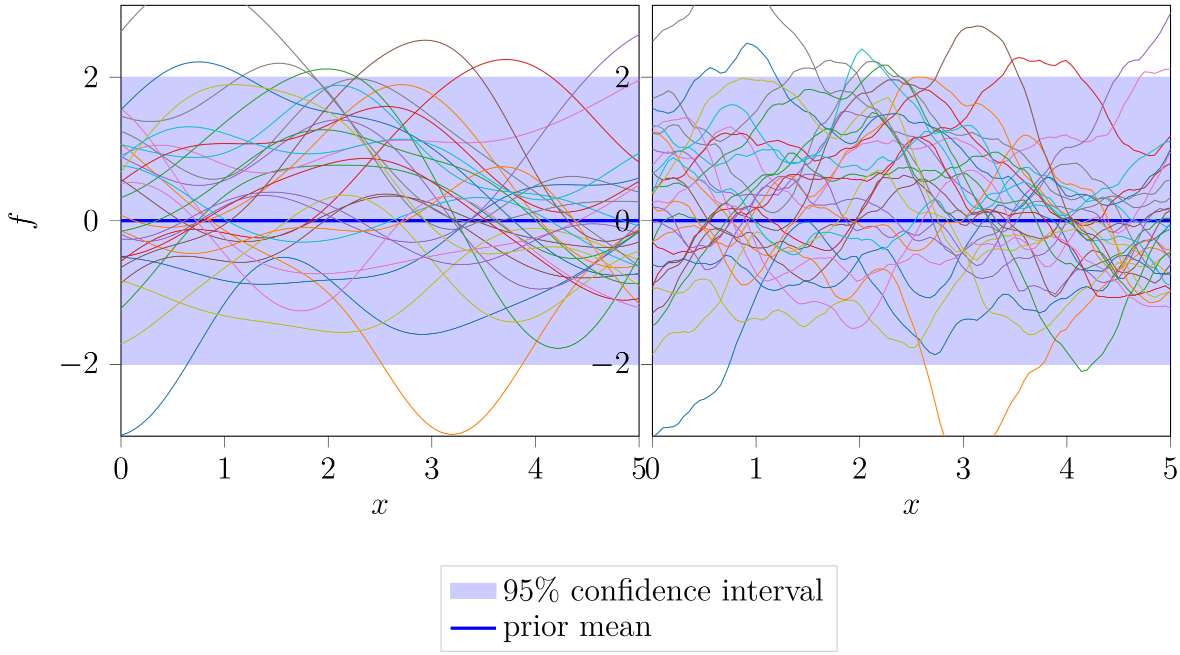


Fig. 2.6 The confidence interval and samples from a zero-mean GP (left) with a squared exponential kernel (also known as RBF) $k^{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x'}) = \exp\left(-\sum_{i=1}^{d} \theta_i.|x_i - x_i'|^2\right)$ (right) with a 3/2 Matérn kernel $k^{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x'}) = \left(1+\sqrt{3}\sum_{i=1}^{d}\theta_i.|x_i - x_i'|\right)\exp\left(-\sqrt{3}\sum_{i=1}^{d}\theta_i.|x_i - x_i'|\right)$

where $\mathbf{1}$ denotes an n-vector of ones. The posterior predictive distribution is obtained in new locations $\mathbf{X}^* = \left[\mathbf{x}^{*(1)},\dots,\mathbf{x}^{*(n^*)}\right]^\top$ through two steps. Firstly, using the property of a GP, Eq.(2.17), that is the joint distribution of the predicted outputs $\mathbf{f}^* = f(\mathbf{X}^*)$ and the observed outputs $\mathbf{y}$ is Gaussian:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{1}\hat{\mu}, \begin{matrix} k^{\hat{\boldsymbol{\theta}}}(\mathbf{X},\mathbf{X}) + \hat{\sigma}^2\mathbf{I}_n & ,k^{\hat{\boldsymbol{\theta}}}(\mathbf{X},\mathbf{X}^*) \\ k^{\hat{\boldsymbol{\theta}}}(\mathbf{X}^*,\mathbf{X}) & ,k^{\hat{\boldsymbol{\theta}}}(\mathbf{X}^*,\mathbf{X}^*) \end{matrix}\right) \tag{2.20}$$

Then, the posterior predictive distribution is obtained by conditioning the prior distribution on the observations, which comes back to the conditional distribution of a joint Gaussian distribution:

$$\mathbf{f}^*|\mathbf{X}^*,\mathbf{y},\mathbf{X} \sim \mathcal{N}\left(\hat{f}(\mathbf{X}^*),\hat{\boldsymbol{\Sigma}}(\mathbf{X}^*)\right) \tag{2.21}$$

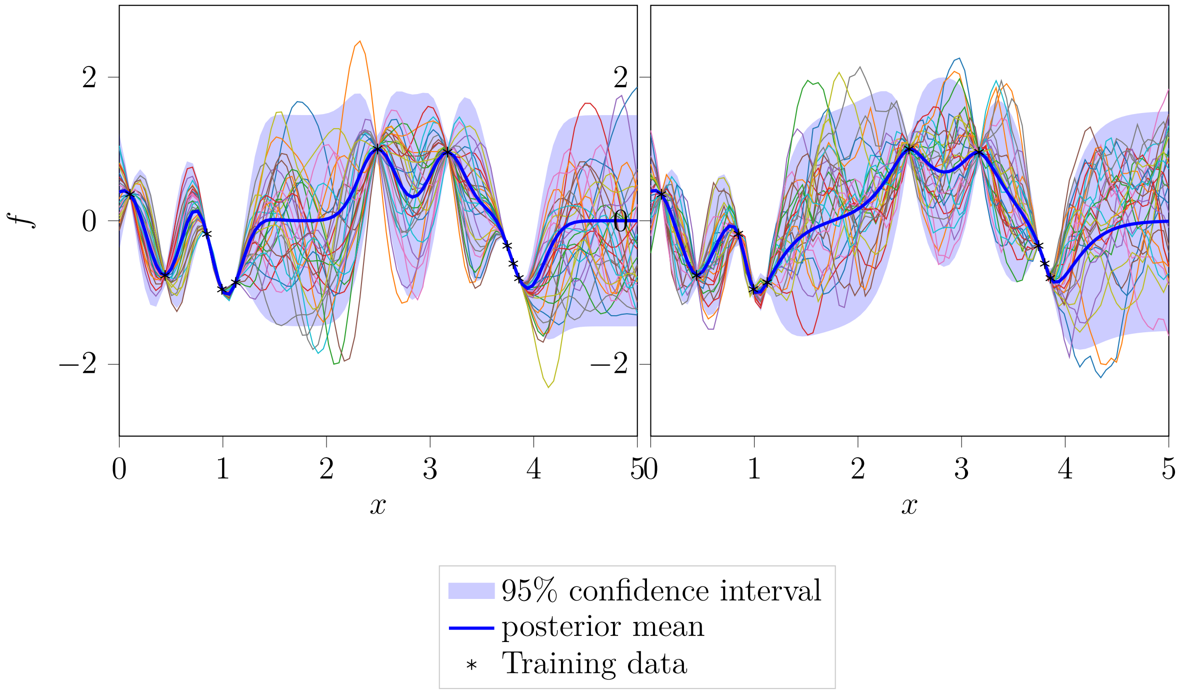


Fig. 2.7 The posterior mean, uncertainty measure and one-dimensional samples from the posterior of a GP (left) with a squared exponential kernel (right) with a 3/2 Matérn kernel (left).

where $\hat{f}(\mathbf{X}^*)$ and $\hat{\Sigma}(\mathbf{X}^*)$ are respectively the mean and the covariance of the posterior distribution and are defined as:

$$\hat{f}(\mathbf{X}^*) = \mathbf{1}\hat{\mu} + k^{\hat{\boldsymbol{\theta}}}(\mathbf{X}^*,\mathbf{X})\left(k^{\hat{\boldsymbol{\theta}}}(\mathbf{X},\mathbf{X}) + \hat{\sigma}^2\mathbf{I}_n\right)^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{2.22}$$

$$\hat{\boldsymbol{\Sigma}}(\mathbf{X}^*) = k^{\hat{\boldsymbol{\theta}}}(\mathbf{X}^*,\mathbf{X}^*) - k^{\hat{\boldsymbol{\theta}}}(\mathbf{X}^*,\mathbf{X})\left(k^{\hat{\boldsymbol{\theta}}}(\mathbf{X},\mathbf{X}) + \hat{\sigma}^2\mathbf{I}_n\right)^{-1} k^{\hat{\boldsymbol{\theta}}}(\mathbf{X},\mathbf{X}^*) \tag{2.23}$$

It is interesting to notice that the prediction depends on the inverse of $k^{\hat{\boldsymbol{\theta}}}(\mathbf{X},\mathbf{X})$ (called the Gram matrix) which grows with the amount of the training data, illustrating the non-parametric aspect of GPs. This gives GPs more flexibility, however, the inversion of the matrix can quickly become a computational burden for large data-set (see Section 2.2.2).

Estimating the hyper-parameters is called the learning step, while the conditioning on the observations to obtain the posterior distribution is the inference step. Instead of using an empirical Bayesian approach to deal with the hyper-parameters, a fully-Bayesian approach can be used. In fact, in [Snoek et al., 2012] the hyper-parameters of the kernel have been marginalized out and an MCMC approach was used to get the posterior distribution.

In the previous definition of GPs, the single-output regression framework has been used. However, GPs are also used for the approximation of vector-valued functions. In that case, it is called multi-output GPs [Alvarez et al., 2011]. The inference and learning steps for multi-output GPs follow the same equations introduced in this section.

In the presented GP regression model, a Gaussian likelihood function has been used. It is motivated by its conjugancy with the prior GP, yielding to an analytical form of the posterior as described previously. However, in some specific cases, one may prefer another form for the likelihood distribution. For instance, in [Vanhatalo et al., 2009] to deal with outliers in the training data a t-student likelihood was used which better handles the outliers than Gaussian likelihood due to its heavier tails. In [Goldberg et al., 1998], a heteroscedastic Gaussian likelihood has been proposed. It is practical to use for problems where the error of measurements depends on the observed values.

Henceforth, for notation simplifications, the dependence of the prior covariance function on $\boldsymbol{\theta}$ is dropped, and $k(\mathbf{X},\mathbf{X}')$ is written $\mathbf{K}_{\mathbf{X},\mathbf{X}'}$. Moreover, without loss of generality, the prior GP is considered with a zero constant mean function $\mu = 0$.

2.2.2 Sparse Gaussian Processes

The major drawback in GP concerns the handling of large data-sets. In fact, the training and prediction using GPs involves the inversion of the Gram matrix, that is the covariance matrix of the whole data-set $\mathbf{K}_{\mathbf{XX}} \in \mathbb{R}^{n\times n}$. This inversion has a cubic complexity $\mathcal{O}(n^3)$, which rapidly becomes computationally overwhelming. To overcome this limit of GPs, Sparse Gaussian Processes (SGPs) consisting of low rank approximation of the covariance matrix $\mathbf{K}_{\mathbf{XX}}$ have been developped. SGPs augment the latent space with a set of inputs/outputs called inducing input-output variables. Specifically, a set of $m << n$ inducing pair of input-output variables $\mathbf{Z} = \left\{\mathbf{z}^{(1)},\dots,\mathbf{z}^{(m)}\right\}$ and $\mathbf{u} = f(\mathbf{Z}) = \left\{u^{(1)},\dots,u^{(m)}\right\}$ are introduced in order to reduce the time complexity of GPs from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$. Different approaches that have been developed to determine this sparse approximation are described in the next paragraphs.

Prior approximation

One direction of approaches modifies the prior in order to get rid of $\mathbf{K_{XX}}$. For that, the induced variables are marginalized:

$$\begin{aligned} p(\mathbf{f}|\mathbf{X}) &= \int_{\mathbf{u}} p(\mathbf{f}|\mathbf{X},\mathbf{Z},\mathbf{u})p(\mathbf{u}|\mathbf{Z})\mathrm{d}\mathbf{u} \\ &= \int_{\mathbf{u}} \mathcal{N}\left(\mathbf{f}|\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}\right)\mathcal{N}(\mathbf{u}|0,\mathbf{K_{ZZ}})\,\mathrm{d}\mathbf{u} \\ &= \mathcal{N}(\mathbf{f}|0, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}} + \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) \\ &= \mathcal{N}(\mathbf{f}|0, \tilde{\mathrm{K}} + \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) \end{aligned} \tag{2.24}$$

where $\tilde{\mathrm{K}}$ corresponds to the difference between $\mathbf{K_{XX}}$ and the Nyström approximation $\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}$. In order to obtain an approximation which reduces the time complexity, an approximation $q(\mathbf{f}|\mathbf{X},\mathbf{Z},\mathbf{u}) = \mathcal{N}\left(\mathbf{f}|\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \tilde{Q}\right)$ is considered where $\tilde{Q}$ approximates $\tilde{\mathrm{K}}$ with a simpler form. Therefore, the prior itself is changed with this approximation as follows:

$$q(\mathbf{f}|\mathbf{X}) = \mathcal{N}\left(\mathbf{f}|\mathbf{0}, \tilde{Q} + \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}\right) \tag{2.25}$$

Notice that by marginalizing over $\mathbf{u}$, only the inducing inputs $\mathbf{Z}$ have to be determined. This is usually performed using a MLE procedure. Different choices of $\tilde{Q}$ have been proposed. The Projected Latent Variables (PLV) [Seeger et al., 2003] approximates $\mathbf{K_{XX}}$ with the Nyström approximation *i.e.* $\tilde{Q} = 0$. In [Snelson and Ghahramani, 2006], the Fully Independent Training Conditional (FITC) approach considers $\tilde{Q}$ as a diagonal matrix corresponding to the difference between the diagonals of the covariance matrix $\mathbf{K_{XX}}$ and its Nyström approximation *i.e.* $\tilde{Q} = \mathrm{diag}(\tilde{\mathrm{K}})$. Using the Woodbury matrix identity [Woodbury, 1950], the computation of the inverse of the obtained approximations comes back to that of a matrix of size $m \times m$. Notice that increasing the size of the inducing variables leads to the recovery of the exact GP model. However, when $m << n$ the prior of the GP is changed and there is no guarantee that the approximate marginal likelihood represents the true GP.

Variational Sparse GP

Another direction of approaches for Sparse Gaussian Processes considers a variational framework to infer the inducing variables. The variational formulation allows to keep the exact GP prior [Titsias, 2009]. For that, $\mathbf{u}$ is assumed to be a sufficient statistic for

$\mathbf{f}$ *i.e.* $p(\mathbf{f}|\mathbf{u},\mathbf{y}) = p(\mathbf{f}|\mathbf{u})$. Then, the following variational approximation is considered:

$$q(\mathbf{f},\mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$$

where $q(\mathbf{u})$ is a free variational Gaussian distribution over the inducing variables. By marginalizing over $\mathbf{u}$ and using the same trick as in Eq. (2.15) a first lower-bound on the marginal likelihood is obtained:

$$p(\mathbf{y}|\mathbf{X}) \geq \mathcal{L}_1 = \mathbb{E}_{q(\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{u},\mathbf{X},\mathbf{Z})\right) - \mathbb{KL}\left(q(\mathbf{u})||p(\mathbf{u})\right)$$

In this equation while the KL term is analytically computable for a Gaussian variational distribution, the log expectation term is not tractable. Therefore, in order to maximize the lower bound, MCMC approaches can be used to estimate the log expectation term [Hensman et al., 2015]. To avoid sampling, a loosen analytical lower-bound is achieved by considering a lower-bound on $\log\left(p(\mathbf{y}|\mathbf{u},\mathbf{X})\right)$ that is obtained by marginalizing over $\mathbf{f}$ and using the assumption of statistic sufficiency of $\mathbf{u}$ for $\mathbf{f}$:

$$\log p(\mathbf{y}|\mathbf{u},\mathbf{X},\mathbf{Z}) \geq \mathbb{E}_{p(\mathbf{f}|\mathbf{u},\mathbf{X},\mathbf{Z})}\left[p(\mathbf{y}|\mathbf{f})\right]$$

The advantage of this bound is that it has an analytical form and that it is fully factorizable over the observations [Hensman et al., 2013]. Moreover, taking the expectation of this achieved lower bound with respect to $q(\mathbf{u})$ maintains those desirable properties. Hence, the following fully factorizable analytical bound on the marginal likelihood is obtained:

$$p(\mathbf{y}|\mathbf{X}) \geq \mathcal{L}_2 = \sum_{i=1}^{n} \mathcal{G}^{(i)} - \mathbb{KL}\left(q(\mathbf{u})||p(\mathbf{u})\right) \tag{2.26}$$

where $\mathcal{G}^{(i)}$ are analytical terms obtained for each observation $i \in \{1,\ldots,n\}$ depending on $y^{(i)},\mathbf{x}^{(i)},\mathbf{Z}$, and $\boldsymbol{\theta}$. Notice here that the optimization is done according to the deterministic parameters that are the kernel parameters $\boldsymbol{\theta}$ and the induced inputs $\mathbf{Z}$ and also according the variational parameters $\boldsymbol{\theta}_{q(u)}$ of $q(\mathbf{u})$. This can lead to difficulty in the optimization of the lower-bound since $\boldsymbol{\theta}_{q(u)}$ are defined in a non-euclidean space [Salimbeni et al., 2018]. To tighten the bound and collapse $q(\mathbf{u})$, the bound can be maximized analytically according to $\boldsymbol{\theta}_{q(u)}$, and the optimal value of $\boldsymbol{\theta}_{q(u)}$ can then be injected into Eq. (2.26) to obtain an expression of a lower bound $\mathcal{L}_3$ depending only on the hyper-parameters of the kernel and the inducing inputs [Titsias, 2009]. However, the obtained lower-bound in that case is not factorizable over observations.

Inter-domain sparse GP

In the direct approaches or the variational ones, the inducing inputs $\mathbf{Z}$ share the same input space as the observed inputs $\mathbf{X}$. This formulation may lead to important inaccuracy in high-dimensionnal input spaces with $m << n$. This is due to the important distances between the limited number of inducing inputs in high-dimensionnal spaces that leads to negligible correlation between the inducing variables. To overcome this complication, [Lázaro-Gredilla and Figueiras-Vidal, 2009; Lázaro-Gredilla et al., 2010] proposed inter-domain GPs. The concept is to choose a domain where the function can be better described than the actual input space and overcome the limitation of local influence in the original input space. The chosen space can have a different dimension d' than the actual input space of dimension d. Thus, the inducing inputs are defined on $\mathbb{R}^{d'}$ and the expression of the inducing inputs and the inducing outputs are defined *via* the following transformation:

$$u(\mathbf{z}) = \int_{\mathbb{R}^d} f(\mathbf{x})\psi(\mathbf{x},\mathbf{z})d\mathbf{x}$$

where $\psi(\mathbf{x},\mathbf{z})$ is a chosen transformation. This yields to a different expression of the covariance matrix of $K_{\mathbf{ZZ}}$ and $K_{\mathbf{ZX}}$. [Lázaro-Gredilla and Figueiras-Vidal, 2009] used the FITC approach with a Fourier projection of the inducing variables. While in [Hensman et al., 2017], a Fourier transformation has been proposed for a variational sparse GP labeled Variational Fourier Features for Gaussian Processes. In [Dutordoir et al., 2020], a more big-data friendly variational sparse GP projection based on Spherical Harmonic features has been developed. Fig. 2.8 summarizes the different mentioned approaches. Wilson and Nickisch [2015] proposed a structured kernel interpolation that unifies and generalize the inducing inputs framework. This allowed the authors to introduce KISS-GP a highly scalable inducing points GP approach.

2.2.3 Gaussian Processes and other models

GPs can be derived from different machine learning models, which gives them different possible interpretations depending on the perspective taken. The introduction to GPs followed in this thesis started from a linear regression model. Then, the Bayesian concepts were introduced to obtain a Bayesian linear regression model. Finally, going non-parametric by describing a prior over functions instead of one over weights gave rise to a Gaussian process regression model. This is well described by the cube of

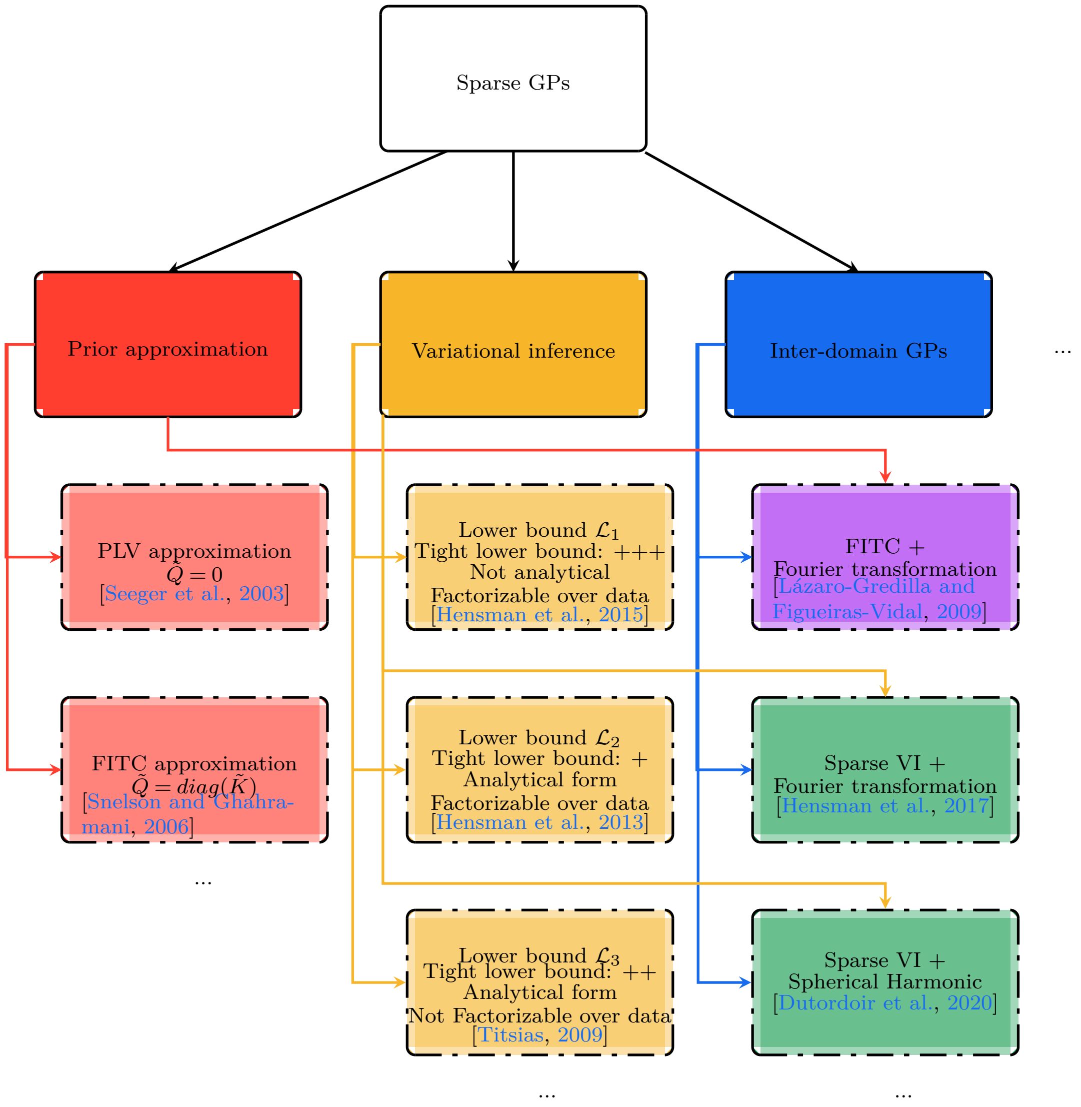


Fig. 2.8 Summary of the presented Sparse GP approaches.

Ghahramani (Fig. 2.9) (it also includes the classification part that is beyond the topic of this thesis).

From Bayesian linear regression to GPs

Let reconsider a Bayesian linear regression model. The prediction distribution given by this model is given in Eq. (2.9). This prediction can be "kernalized" *i.e.* expressed with a kernel, by defining the following kernel:

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\intercal \Sigma_{\text{posterior}} \boldsymbol{\phi}(\mathbf{x}')$$

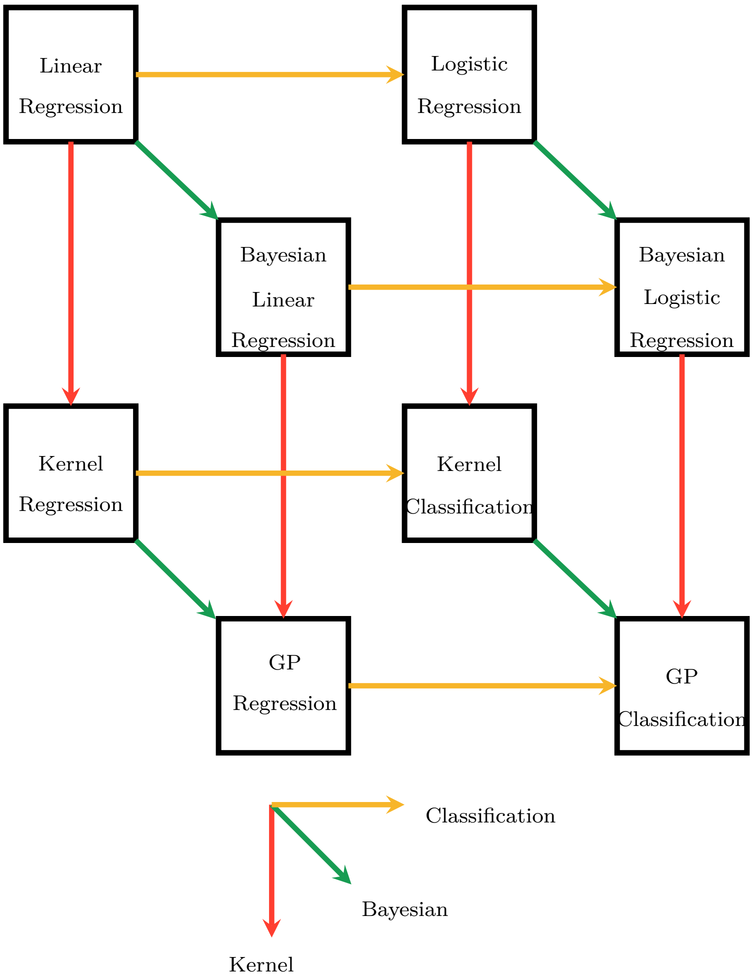


Fig. 2.9 The Ghahramani cube summarizes the different relations between linear regression, Bayesian linear regression, kernel methods, GP regression, and their classification equivalents. The transition from a regression approach to another goes through either kernalization, Bayesian approach, or the combination of the two.

In fact, by replacing this kernel definition in Eq. (2.9), the prediction comes back to the same expression as the one obtained by the posterior predictive distribution of a zero mean Gaussian Process in Eq. (2.22). However, the interesting observation is that this kernel defines a degenerate covariance matrix $\mathbf{K}_{\mathbf{XX}}$ with a number of eigenvalues equal at most to the number of parameters. This highlights the limited flexibility of a parametric model compared to a GP.

From Reproducing Kernel Hilbert Space to GP

Consider an input space $\mathbb{X}$ and a positive definite kernel $k(\cdot,\cdot)$ on $\mathbb{X}$, a Hilbert space $\mathcal{H}_k$ of functions over $\mathbb{X}$, with a defined inner-product $<\cdot,\cdot>_{\mathcal{H}_k}$ is said to be a Reproducing

Kernel Hilbert Space (RKHS) with reproducing kernel $k(\cdot,\cdot)$ if these two conditions are satisfied:

- $\forall \mathbf{x} \in \mathbb{X}, k(\cdot,\mathbf{x}) \in \mathcal{H}_k$
- $\forall (\mathbf{x}, f) \in \mathbb{X} \times \mathcal{H}_k, f(\mathbf{x}) =< f, k(\cdot,\mathbf{x}) >_{\mathcal{H}_k}$

RKHS have some interesting properties *e.g.*, a bijection exists between the set of positive definite kernels $k(\cdot,\cdot)$ and the set of RKHS for wich $k(\cdot,\cdot)$ is the reproducing kernel, the functions of the RKHS for which $k(\cdot,\cdot)$ is the reproducing kernel share the same properties as $k(\cdot,\cdot)$ (differentiability, smoothness, etc.) [Christmann and Steinwart, 2008]. Different works have been developed around the connection between RKHS and GPs, yielding to interesting results in both directions [Hofmann et al., 2008; Alvarez et al., 2011; Anjyo and Lewis, 2011]. To illustrate a brief connection between the two approaches, a kernel ridge regression over a RKHS is considered. It consists in minimizing a square loss function plus a regularization term σ^2 over a RKHS:

$$f = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(f(\mathbf{x}^{(i)}) - y^{(i)})^2 \right) + \sigma^2 ||f||^2_{\mathcal{H}_k} \tag{2.27}$$

where $||f||_{\mathcal{H}_k}$ is the norm of a function with respect to the defined inner-product over $\mathcal{H}_k$. The norm plays the role of a regularization term by penalizing complex functions with respect to the kernel. Using the representer theorem [Dinuzzo and Schölkopf, 2012], it can be shown that only one solution exists that is:

$$\hat{f}(\mathbf{x}) = k_{\mathbf{x}\mathbf{X}}(K_{\mathbf{X}\mathbf{X}} + \sigma^2 \mathbf{I}_n)^{-1}\mathbf{y} \tag{2.28}$$

Notice that the obtained solution corresponds to the mean prediction of a GP in Eq. (2.22) (MAP estimate of a GP) where σ^2 can be interpreted as the variance of a Gaussian noise. The interesting conclusion from this connection is that the mean prediction (MAP estimate) of a GP belongs to the RKHS of the prior kernel used. However, sampling from the GP posterior does not guarantee samples from the RKHS.

From Neural Networks to GP

Another model from which Gaussian processes can be derived are Artificial Neural Networks (ANN). ANNs have been widely used in the machine learning community for a broad spectrum of applications [Basheer and Hajmeer, 2000; Bishop, 2006]. To illustrate the connections of ANNs to GPs, consider a multi-layer perceptron with one hidden layer. The prediction is obtained with the following:

$$f(\mathbf{x}) = b + \sum_{i=1}^{s} w_i \tau(\mathbf{x}, \mathbf{h}_i)$$

where b corresponds to a bias term, s the number of hidden units, w_i to the weight of the output of the hidden unit i, $\mathbf{h}_i$ to the input/feature weights of unit i, and $\tau(\cdot)$ is the activation function. As it is now settled from the previous sections, to get a GP one has to think Bayesian and non-parametric, in other words, includes priors and get rid of weights.

For that purpose, a factorizable Gaussian prior over the bias and the output weights with zero mean and variance σ^2 and an unspecified prior on the inputs weights are considered, yielding to a Bayesian Neural Network (BNN) [Neal, 2012]. Based on the central limit theorem for a large enough sum *i.e.* a large number of hidden units (a very wide hidden layer), there is a joint Gaussian distribution for any set of function evaluations of the ANN hence defined. The infinite sum gets rid of the weight parameters w_i and the bias b. Therefore, a non-parametric model with a joint Gaussian distribution for any set of outputs is defined, thus the equivalence with a GP. This connection between the two approaches was first presented in [Neal, 1995]. This connection certainly illustrates the power of representation of a GP since the power of representation of an ANN increases with its number of units.

The other direction, going from a GP to an ANN is also possible with a slight detour to kernel approaches. In fact, Mercer theorem [Carmeli et al., 2005] postulates that any positive-definite kernel can be represented by the inner product of features $k(\mathbf{x}, \mathbf{x}') =< \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') >$. Therefore, to obtain an ANN from a GP, one has to get the features from the kernel and use them as activation functions.

Other works investigate these connections between GPs and ANNs and its hierarchical generalization called Deep Neural Networks (DNNs) [Lee et al., 2017; Matthews et al., 2018; Yang, 2019; Agrawal et al., 2020]. DNNs have gained a lot of popularity these last years. Their prowess in resolving some problems this last decade that were not possible for other models has been overwhelming [Goodfellow et al., 2016]. One can also think of the same hierarchical generalization to GPs due to their connection with ANNs illustrated previously. This deep generalization has been introduced in [Damianou and Lawrence, 2013] and it is called Deep Gaussian processes. Inspired by the cube of Ghahramani, a spectrum illustrating the connections between the ANN, BNN, GPs, and their hierarchical generalization DNN, Deep BNN (DBNN), and

Deep Gaussian Processes (DGPs) is presented in Fig. 2.10. In the next section, this hierarchical generalization of GP is introduced.

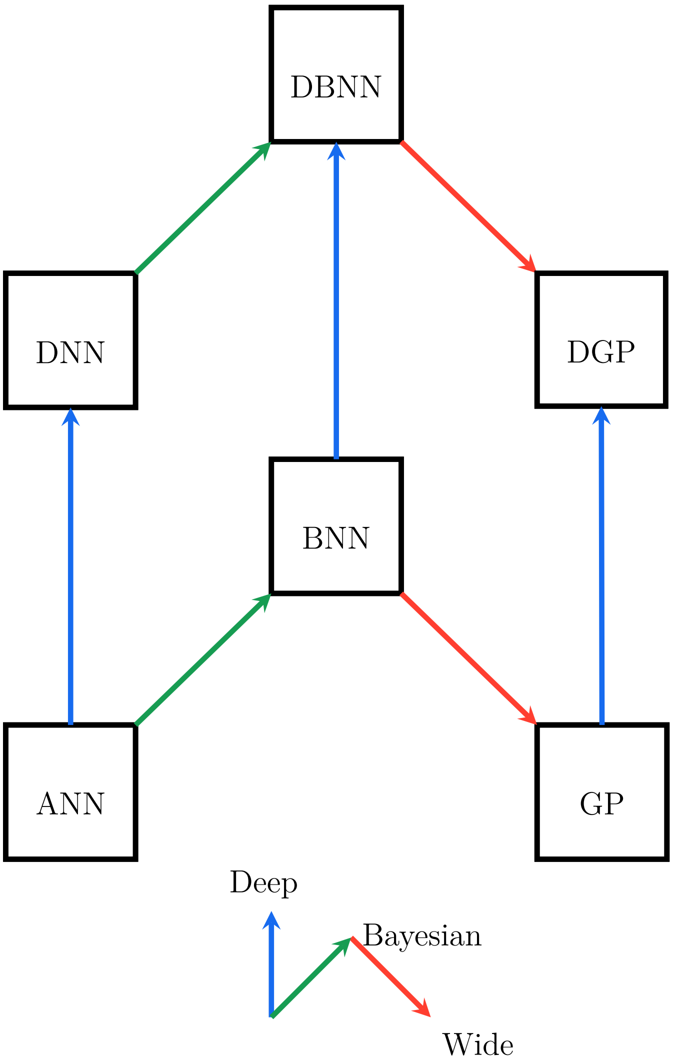


Fig. 2.10 The spectrum expresses the relationship between shallow artificial neural networks (ANN), Bayesian neural networks (BNN), Gaussian processes (GPs), and their hierarchical generalization. The transition from one approach to another goes through either Bayesian approach, wider architecture, deeper architecture or the combination of the three.

2.3 Deep Gaussian Processes (DGPs)

This section serves as the introduction of the central machine learning model on which the contributions of this thesis are based. First, Deep Gaussian Processes (DGPs) are introduced and defined in Section 2.3.1. Then, a unified view of DGP inference is presented in Section 2.3.2. Finally, an overview of applications of DGPs in the literature is exposed in Section 2.3.3 with a perspective view of the contributions of this thesis within these applications.

2.3.1 Definitions

A Deep Gaussian Process (DGP) is a nested structure of GPs considering the relationship between the inputs and the final output as a functional composition of GPs (Fig. 2.11)

$$y = f_{[l-1]}(\dots \mathbf{f}_{[i]}(\dots (\mathbf{f}_{[1]}(\mathbf{f}_{[0]}(\mathbf{x}) + \boldsymbol{\epsilon}_{[0]}) + \boldsymbol{\epsilon}_{[1]}) + \boldsymbol{\epsilon}_{[i]}) + \epsilon_{[l-1]}) \tag{2.29}$$

where l is the number of layers and $\mathbf{f}_{[i]}(\cdot)$ is an intermediate GP. Each layer i is composed of an input node $\mathbf{H}_{[i]}$ of dimension $d_{[i]}$, an output node $\mathbf{H}_{[i+1]}$ of dimension $d_{[i+1]}$ and a multi-output GP $\mathbf{f}_{[i]}(\cdot)$ mapping between the two nodes, getting the recursive equation: $\mathbf{H}_{[i+1]} = \mathbf{f}_{[i]}\left(\mathbf{H}_{[i]}\right)$. A Gaussian noise $\epsilon_{[i]} \sim \mathcal{N}(0, \sigma^2_{[i]})$ is introduced such as $\mathbf{H}_{[i+1]} = \mathbf{f}_{[i]}(\mathbf{H}_{[i]}) + \boldsymbol{\epsilon}_{[i]}$. The one column matrix $\mathbf{H}_{[l]} = f_{[l-1]}\left(\mathbf{H}_{[l-1]}\right)$ refers to an unobserved noiseless version of $\mathbf{y}$. An exploded view showing the multidimensional aspect of DGPs is illustrated in Fig. 2.12.

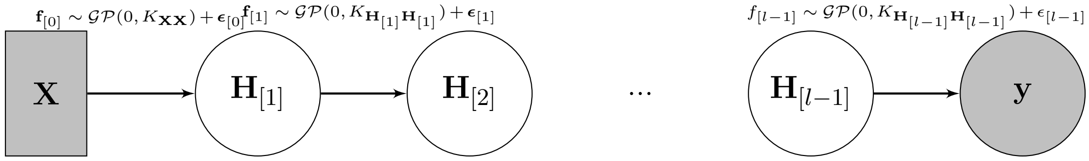

$\mathbf{f}_{[0]} \sim \mathcal{GP}(0, K_{\mathbf{XX}}) + \boldsymbol{\epsilon}_{[0]}$
$\mathbf{f}_{[1]} \sim \mathcal{GP}(0, K_{\mathbf{H}_{[1]}\mathbf{H}_{[1]}}) + \boldsymbol{\epsilon}_{[1]}$
$f_{[l-1]} \sim \mathcal{GP}(0, K_{\mathbf{H}_{[l-1]}\mathbf{H}_{[l-1]}}) + \epsilon_{[l-1]}$
$\mathbf{X}$
$\mathbf{H}_{[1]}$
$\mathbf{H}_{[2]}$
$\cdots$
$\mathbf{H}_{[l-1]}$
$\mathbf{y}$

Fig. 2.11 A representation of the structure of a DGP

This hierarchical composition of GPs presents better results than regular GPs in the approximation of complex functions [Damianou and Lawrence, 2013; Dai et al., 2015; Salimbeni and Deisenroth, 2017]. In fact, DGPs allow a flexible way of kernel construction through input warping and dimensionality expansion to better fit data (see Chapter 4 for more details).

In GP regression models, the hyper-parameters involved are the kernel parameters, the mean function parameters and the likelihood parameters. The optimization of these hyper-parameters in the training of GPs is analytically tractable for a Gaussian likelihood function. In DGPs, in addition to the hyper-parameters considered for each layer, non-observable variables $\mathbf{H}_{[1]}, \dots, \mathbf{H}_{[i]}, \dots, \mathbf{H}_{[l]}$ are involved. Hence, the marginal likelihood for DGP can be written as:

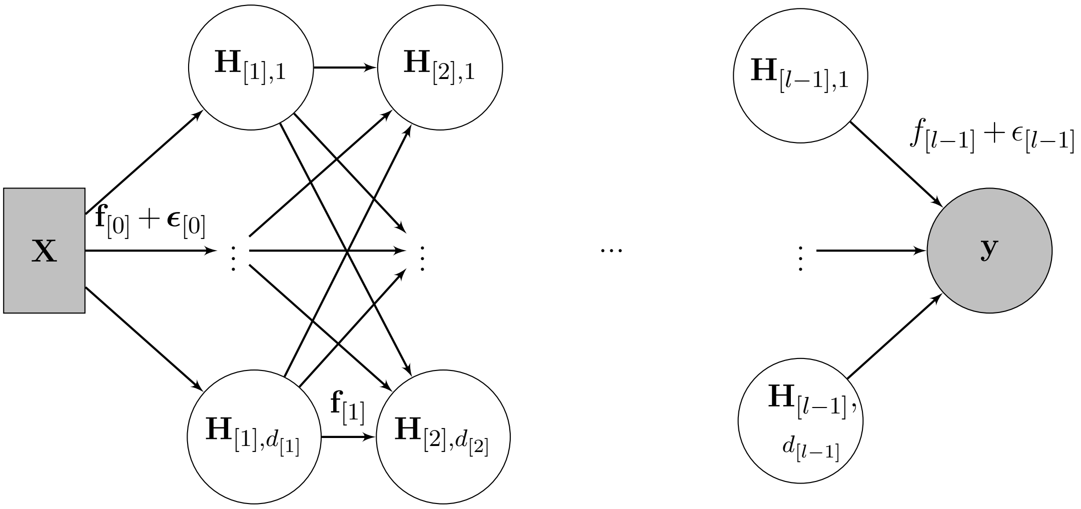


Fig. 2.12 An exploded view of the structure of a DGP

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{X}) &= \int_{\mathbf{H}_{[1]}} \cdots \int_{\mathbf{H}_{[l]}} \cdots \int_{\mathbf{H}_{[l]}} p\left(\mathbf{y}, \mathbf{H}_{[1]}, \ldots, \mathbf{H}_{[i]}, \ldots, \mathbf{H}_{[l]} | \mathbf{X}\right) \mathrm{d}\mathbf{H}_{[1]} \ldots \mathrm{d}\mathbf{H}_{[i]} \ldots \mathrm{d}\mathbf{H}_{[l]} \\
&= \int_{\{\mathbf{H}_{[i]}\}_1^l} p\left(\mathbf{y}, \{\mathbf{H}_{[i]}\}_1^l | \mathbf{X}\right) \mathrm{d}\{\mathbf{H}_{[i]}\}_1^l \\
&= \int_{\{H_{[l]}\}_1^l} p(\mathbf{y}|\mathbf{H}_{[l]}) p(\mathbf{H}_{[l]}|\mathbf{H}_{[l-1]}) \ldots p(\mathbf{H}_{[1]}|\mathbf{X}) \mathrm{d}\{\mathbf{H}_{[i]}\}_1^l \qquad (2.30)
\end{aligned}
$$

where $\{\mathbf{H}_{[i]}\}_1^l$ is the set of non-observable (latent) variables $\{\mathbf{H}_{[1]}, \ldots, \mathbf{H}_{[l]}\}$.

The computation of this marginal likelihood is not analytically tractable. Indeed, $p(\mathbf{H}_{[i+1]}|\mathbf{H}_{[i]})$ non-linearly involves the inverse of the covariance matrix $\mathbf{K}_{\mathbf{H}_{[i]}\mathbf{H}_{[i]}}$, which makes the integration of the conditional probability $p(\mathbf{H}_{[i+1]}|\mathbf{H}_{[i]})$ with respect to $\mathbf{H}_{[i]}$ analytically not tractable.

2.3.2 Advances in Deep Gaussian Processes inference

To overcome this issue, the marginal likelihood is approached using approximate inference techniques. Several approaches based on variational inference, expectation propagation, Markov chain Monte-Carlo have been developed and are discussed in this section.

Direct variational inference approach

In [Damianou and Lawrence, 2013], a variational approach is followed to obtain a lower bound on the marginal likelihood. For that, a variational distribution on the latent variables $q\left(\left\{\mathbf{H}_{[i]}\right\}_1^l\right)$ is introduced, and by applying the results of variational inference (see Section 2.1.2) the following result is obtained:

$$\begin{aligned}
\log p(\mathbf{y}|\mathbf{X}) &\geq \mathbb{E}_q\left[\log\left(\frac{p(\mathbf{y},\left\{\mathbf{H}_{[i]}\right\}_1^l|\mathbf{X})}{q\left(\left\{\mathbf{H}_{[i]}\right\}_1^l\right)}\right)\right] \\
&\geq \mathbb{E}_q\left[\log\left(p\left(\mathbf{y}|\left\{\mathbf{H}_{[i]}\right\}_1^l,\mathbf{X}\right)\right)\right]+\mathbb{E}_q\left[\log\left(\frac{p\left(\left\{\mathbf{H}_{[i]}\right\}_1^l|\mathbf{X}\right)}{q\left(\left\{\mathbf{H}_{[i]}\right\}_1^l\right)}\right)\right] \qquad (2.31) \\
&\geq \mathbb{E}_q\left[\log\left(p\left(\mathbf{y}|\left\{\mathbf{H}_{[i]}\right\}_1^l,\mathbf{X}\right)\right)\right]-\mathbb{KL}\left(q\left(\left\{\mathbf{H}_{[i]}\right\}_1^l\right)||p\left(\left\{\mathbf{H}_{[i]}\right\}_1^l|\mathbf{X}\right)\right)
\end{aligned}$$

The second term in Eq.(2.32) is the KL divergence between the variational distribution and the prior distribution of the latent variables. The KL divergence is analytically tractable if the prior and the variational distributions on the latent variables are restrained to Gaussian distributions. However, the first term is still analytically intractable since it involves the integration of the inverse of the covariance matrices with respect to the latent variables. To overcome this issue, [Damianou and Lawrence, 2013] followed the work of [Titsias and Lawrence, 2010] in the context of Bayesian Gaussian process latent variable model by introducing a set of inducing variables to obtain an analytical tractable lower bound based on the sparse variational GP described previously (Section 2.2.2). Specifically, in each layer of a DGP, a set of inducing variables is introduced $\mathbf{Z}_{[i]}=\left[\mathbf{z}_{[i]}^{(1)},\ldots,\mathbf{z}_{[i]}^{(m_{[i]})}\right]^\top$, $\mathbf{z}_{[i]}^{(j)}\in\mathbb{R}^{d_{[i]}},\forall j\in\{1,\ldots,m_{[i]}\}$ and $\mathbf{U}_{[i]}=\mathbf{f}_{[i-1]}\left(\mathbf{Z}_{[i]}\right)$ (Fig. 2.13) (notice here that since the intermediate layers are multi-output GPs, $\mathbf{U}_{[i]}$ are matrices $\in\mathbb{R}^{m_{[i]}\times d_{[i]}}$ and not vectors, except in the last layer where $\mathbf{U}_{[l]}$ corresponds to a one column matrix). Henceforth, for notation simplicity, the number of induced inputs in each layer is considered equal to m.

Now that the latent space has been augmented with the inducing variables, the posterior of the joint distribution of the latent variables $p\left(\left\{\mathbf{H}_{[i]},\mathbf{U}_{[i]}\right\}_1^l|\mathbf{y},\mathbf{X}\right)$ is approximated by a variational distribution $q\left(\left\{\mathbf{H}_{[i]},\mathbf{U}_{[i]}\right\}_1^l\right)$ with the assumption of

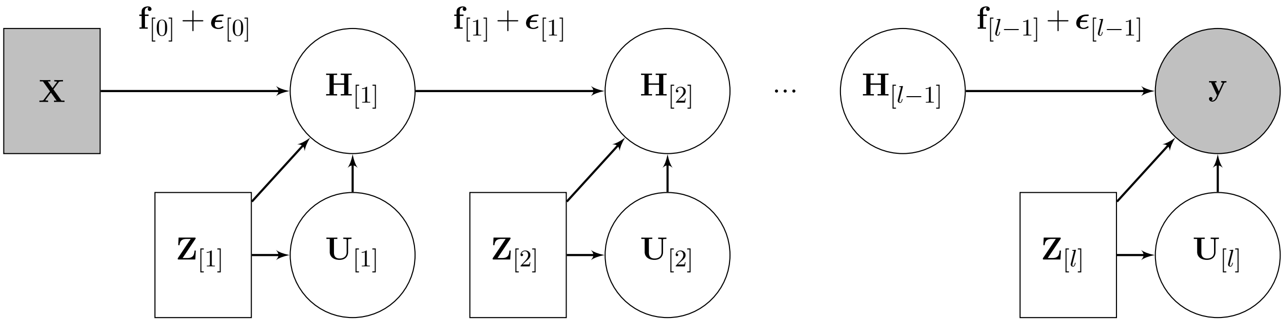


Fig. 2.13 Representation of the introduction of the inducing variables in DGPs

independency between layers:

$$q\left(\left\{\mathbf{H}_{[i]},\mathbf{U}_{[i]}\right\}_1^l\right)=\prod_{i=1}^{l} q\left(\mathbf{H}_{[i]}\right)q\left(\mathbf{U}_{[i]}\right)=\prod_{i=1}^{l}\left(q\left(\mathbf{H}_{[i]}\right)\prod_{j=1}^{m} q\left(\mathbf{u}_{[i]}^{(j)}\right)\right) \tag{2.32}$$

Moreover, for the sake of analytical tractability, the variational distributions are restrained to the Gaussian family. Then, by following the same derivation as in Eq.(2.32) it holds:

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{X}) &\geq \mathbb{E}_{q(\{\mathbf{H}_{[i]}\}_1^l,\{\mathbf{U}_{[i]}\}_1^l)}\left[\log\frac{p\left(\mathbf{y},\{\mathbf{H}_{[i]}\}_1^l,\{\mathbf{U}_{[i]}\}_1^l|\mathbf{X},\{\mathbf{Z}_{[i]}\}_1^l\right)}{q(\{\mathbf{H}_{[i]}\}_1^l,\{\mathbf{U}_{[i]}\}_1^l)}\right]\\ &\geq \mathbb{E}_{q(\{\mathbf{H}_{[i]}\}_1^l,\{\mathbf{U}_{[i]}\}_1^l)}\left[p\left(\mathbf{y},\{\mathbf{H}_{[i]}\}_1^l|\{\mathbf{U}_{[i]}\}_1^l|\right)\right]-\sum_{i=2}^{l}\mathbb{KL}\left[q(\mathbf{H}_{[i]})||p(\mathbf{U}_{[i]})\right]+\sum_{i=1}^{l}\frac{1}{q(\mathbf{H}_{[l]})}\end{aligned} \tag{2.33}$$

The KL term and the last term that corresponds to the entropy are analytically tractable for Gaussian distributions. The expectation term can be further bounded by an analytical expression for kernels that are feasibly convoluted with the Gaussian density such as the Automatic Relevance Determination (ARD) squared exponential kernel. Therefore, a fully analytical lower bound on the marginal likelihood is obtained. The maximization of the lower bound depends on the model parameters $\{\boldsymbol{\theta}_{[i]}\}_1^l,\{\sigma_{[i]}\}_1^l$, the induced inputs $\{\mathbf{Z}_{[i]}\}_1^l$, and the variational parameters $\{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})},\boldsymbol{\theta}_{q(\mathbf{H}_{[i]})}\}_1^l$. In the case of a mean and covariance parametrization of the variational distributions, the variational parameters can be expressed as: $\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}=\left[\bar{\mathbf{U}}_{[i]},\boldsymbol{\Gamma}_{[i]}\right]^\intercal$ and $\boldsymbol{\theta}_{q(\mathbf{H}_{[i]})}=\left[\bar{\mathbf{H}}_{[i]},\boldsymbol{\Lambda}_{[i]}\right]^\intercal$ where $q(\mathbf{U}_{[i]})=\mathcal{N}\left(\mathbf{U}_{[i]}|\bar{\mathbf{U}}_{[i]},\boldsymbol{\Gamma}_{[i]}\right)$ and $q(\mathbf{H}_{[i]})=\mathcal{N}\left(\mathbf{H}_{[i]}|\bar{\mathbf{H}}_{[i]},\boldsymbol{\Lambda}_{[i]}\right)$.

By equalizing the derivative of the ELBO in Eq. (2.33) with respect to $\{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}\}_1^l$ to zero, an optimal analytical form of $\{q(\mathbf{U}_{[i]})\}_1^l$ is obtained. By injecting this expression into the expression of the ELBO, $\{q(\mathbf{U}_{[l]})\}_1^l$ is collapsed, and a lower bound depending only on the model parameters $\{\boldsymbol{\theta}_{[i]}\}_1^l, \{\sigma_{[i]}\}_1^l$, the induced inputs $\{\mathbf{Z}_{[i]}\}_1^l$, and the variational parameters of the hidden layers introduced $\{\boldsymbol{\theta}_{q(\mathbf{H}_{[i]})}\}_1^l$. This lower bound is tighter than the full lower bound, however, as in the case of sparse variational GPs, the factorization over the observations is lost.

The major drawback of this approach is the assumption of independency between layers in the formulation of the variational distribution of the latent variables $q(\{\mathbf{H}_{[i]})\}_1^l)$. This assumption prevents efficient exploitation of the deep architecture of DGPs.

Variatianal Auto-encoded Deep GP

Instead of considering the variational posteriors $\{q(\mathbf{H}_{[i]})\}_1^l$ as independent, [Dai et al., 2015] considered a chain of transformations linking the observed variables $\mathbf{y}$ to the variational parameters. More specifically, the mean and covariance parametrization is considered *i.e.* $\boldsymbol{\theta}_{q(\mathbf{H}_{[i]})} = \left[\bar{\mathbf{H}}_{[i]}, \boldsymbol{\Lambda}_{[i]}\right]^\intercal$ where $q(\mathbf{H}_{[i]}) = \mathcal{N}\left(\mathbf{H}_{[i]}|\bar{\mathbf{H}}_{[i]}, \boldsymbol{\Lambda}_{[i]}\right)$, then the mean $\bar{\mathbf{H}}_{[i]}$ is considered as a transformation of the mean of the next layer variational posterior $\bar{\mathbf{H}}_{[i]}$ by a parameterized function $\boldsymbol{\psi}_{[i]}(\cdot)$, and so on until reaching the final layer that is a transformation of the observed values $\mathbf{y}$ by $\boldsymbol{\psi}_{[l]}(\cdot)$:

$$\begin{aligned} \bar{\mathbf{H}}_{[i]} &= \boldsymbol{\psi}_{[i]}(\bar{\mathbf{H}}_{[i+1]}), \forall i = 1, \ldots, l-1 \\ \bar{\mathbf{H}}_{[i]} &= \boldsymbol{\psi}_{[l]}(\mathbf{y}) \end{aligned} \tag{2.34}$$

Therefore, instead of optimizing with respect to the mean of the variational distributions, the optimization is performed with respect to the parameters of the functions $\{\boldsymbol{\psi}_{[i]}(\cdot)\}_1^l$. This is interesting when dealing with large sized data, since the parameters of the transformation are independent to the number of observations, hence, reducing the complexity of the optimization problem. Moreover, it creates a relationship between the variational means which is a desirable feat for the exploitation of the DGP architecture. In [Dai et al., 2015], the transformation functions are chosen to be parameterized by a Multi-Layer Perceptron (MLP) with tangent activation functions. This choice enables to take advantage of the different approaches for the initialization of the parameters of MLP and avoids to initialize directly the means of the variational distributions. However, while creating a chain relationship between the mean of the variational distributions, the covariance matrices are still considered independent which may not be adapted due to the deep structure of DGPs.

Random feature expansions for deep Gaussian processes

In [Cutajar et al., 2017], an inference approach for DGPs that couples between random feature expansion for GPs [Rahimi and Recht, 2008] and variational inference is developed. Similarly to the inter-domain sparse GP presented in Section 2.2.2, the random feature expansion yields to a low-dimensional representation of a covariance function feature map. This allows to approximate a GP by a two-layer weight-space approximation involving the random features obtained and a Gaussian prior over the weights. Using this approximation in each layer of a DGP yields to a Bayesian deep neural network approximation with Gaussian priors over the weights. This highlights once more the relationship between DGPs and DNNs. In fact, a Bayesian DNN can be seen as a parametric approximation of a DGP. Once the Bayesian DNN approximation obtained, a stochastic variational inference is used. For that, a Gaussian variational distribution that factorizes over layers and weights is considered, then, the ELBO is obtained as described in Section 2.1.2. This approach overcomes the issue of the variational distribution of the latent GPs that factorizes across layers present in [Damianou, 2015], by using a DNN approximation and a variational inference on the weights that are naturally independent. However, in addition to the approximation induced by the variational inference there is also the approximation induced by the DNN representation. Moreover, the covariance functions must have an analytical spectral density.

The Doubly Stochastic approach

The doubly stochastic approach proposed by [Salimbeni and Deisenroth, 2017] drops the assumption of independence between layers and the special form of kernels. Indeed, the posterior approximation maintains the exact model conditioned on $\mathbf{U}_{[i]}$:

$$q\left(\{\mathbf{H}_{[i]}, \mathbf{U}_{[i]}\}_1^l\right) = \prod_{i=1}^{l} p(\mathbf{H}_{[i]}|\mathbf{H}_{[i-1]}, \mathbf{U}_{[i]}) q(\mathbf{U}_{[i]}) \qquad (2.35)$$

However, the analytical tractability of the lower bound obtained in the direct variational inference approach is not maintained. The variational lower bound is then rewritten as follows (the mention of the dependence on $\mathbf{X}$ and $\mathbf{Z}_{[i]}$ is omitted for the sake of simplicity):

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q(\{\mathbf{H}_{[i]},\mathbf{U}_{[i]}\}_1^l)}\left[\log \frac{p\left(\mathbf{y},\{\mathbf{H}_{[i]}\}_1^l,\{\mathbf{U}_{[i]}\}_1^l\right)}{q(\{\mathbf{H}_{[i]}\}_1^l,\{\mathbf{U}_{[i]}\}_1^l)}\right] \\
&= \mathbb{E}_{q(\{\mathbf{H}_{[i]},\mathbf{U}_{[i]}\}_1^l)}\left[\log \frac{p\left(\mathbf{y}|\{\mathbf{H}_{[i]}\}_1^l,\{\mathbf{U}_{[i]}\}_1^l\right)\prod_{i=1}^l p(\mathbf{H}_{[i]}|\mathbf{H}_{[i-1]},\mathbf{U}_{[i]})p(\mathbf{U}_{[i]})}{\prod_{i=1}^l p(\mathbf{H}_{[i]}|\mathbf{H}_{[i-1]},\mathbf{U}_{[i]})q(\mathbf{U}_{[i]})}\right] \\
&= \mathbb{E}_{q\left(\{\mathbf{H}_{[i]},\mathbf{U}_{[i]}\}_1^l\right)}\left[\log \frac{\prod_{j=1}^n p\left(y^{(j)}|h_{[l]}^{(j)}\right)\prod_{i=1}^l p(\mathbf{U}_{[i]})}{\prod_{i=1}^l q(\mathbf{U}_{[i]})}\right] \\
\mathcal{L} &= \sum_{j=1}^n \mathbb{E}_{q\left(h_{[l]}^{(j)}\right)}\left[\log p\left(y^{(j)}|h_{[l]}^{(j)}\right)\right] - \sum_{i=1}^l \mathbb{KL}\left[q(\mathbf{U}_{[i]})||p(\mathbf{U}_{[i]})\right] \qquad (2.36)
\end{aligned}
$$

Keeping $\{\mathbf{U}_{[i]}\}_1^l$ in this formulation of the ELBO instead of collapsing them allows factorization over the data $\mathbf{y}$ which enables parallelization. The computation of this bound is done by approximating the expectation with Monte Carlo sampling, which is straightforward using the propagation of each data-point $\mathbf{x}^{(j)}$ through all the GPs:

$$
q\left(h_{[l]}^{(j)}\right) = \int \prod_{i=1}^{l-1} q\left(\mathbf{h}_{[i]}^{(j)}|\mathbf{U}_{[i]},\mathbf{h}_{[i-1]}^{(j)},\mathbf{Z}_{[i-1}\right) \mathrm{d}\mathbf{h}_{[i]}^{(j)} \qquad (2.37)
$$

with $\mathbf{h}_{[0]}^{(i)} = \mathbf{x}^{(i)}$. The optimization of this formulation of the bound is done according to the kernel parameters $\{\boldsymbol{\theta}_{[i]}\}_1^l$, the inducing inputs $\{\mathbf{Z}_{[i]}\}_1^l$, and the variational parameters of the inducing variables $\left\{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}\right\}_1^l$.

Expectation Propagation

An expectation propagation inference approach (Section 2.1.2) for DGPs was proposed in [Bui et al., 2016]. The proposed inference goes through three steps. First, the FITC method described previously (Section 2.2.2) is used to introduce the latent induced variables in each layer of the DGP:

$$
\begin{aligned}
p(\mathbf{H}_{[i]}|\mathbf{U}_{[i]},\mathbf{H}_{[i-1]}) = \prod_{j=1}^n \prod_{r=1}^{d_{[i]}} \mathcal{N}(h_{[i],r}^{(j)}|&\mathbf{k}_{\mathbf{h}_{[i-1]}^{(j)},\mathbf{Z}_{[i-1]}}\mathbf{K}_{\mathbf{Z}_{[i-1]},\mathbf{Z}_{[i-1]}}^{-1}u_{[i],r} \\
&k_{\mathbf{h}_{[i-1]}^{(j)},\mathbf{h}_{[i-1]}^{(j)}} - \mathbf{k}_{\mathbf{h}_{[i-1]}^{(j)},\mathbf{Z}_{[i-1]}}\mathbf{K}_{\mathbf{Z}_{[i-1]},\mathbf{Z}_{[i-1]}}^{-1}\mathbf{k}_{\mathbf{Z}_{[i-1]},\mathbf{h}_{[i-1]}^{(j)}} + \sigma_{[i-1]}^2)
\end{aligned}
\qquad (2.38)
$$

Then, a stochastic EP approach is followed in order to approximate the marginal likelihood:

$$q\left(\{\mathbf{U}_{[i]}\}_1^l\right) = p\left(\{\mathbf{U}_{[i]}\}_1^l\right)\bar{\hat{\mathbf{t}}}^n \tag{2.39}$$

where $\bar{\hat{\mathbf{t}}}$ is the approximated average contribution in the EP. The approximated marginal likelihood obtained following this approach involves the term:

$$\int_{\{\mathbf{U}_{[i]}\}} q_{-j}(\{\mathbf{U}_{[i]}\}_1^l)p(y^{(j)}|\{\mathbf{U}_{[i]}\}_1^l)\mathrm{d}\{\mathbf{U}_{[i]}\}_1^l \tag{2.40}$$

which is not tractable for $l > 1$. In fact, the computation of this term yields to marginalization over the latent variables $\{\mathbf{H}_{[i]}\}_1^l$ which propagated through the DGP leads to complex distributions in the integrand.

The third step of the proposed approach is to approximate the complex distributions by Gaussian distributions using matching moments. Therefore, a sequential approximation procedure is used where at each layer the distribution of the latent variables is approximated by a Gaussian and is propagated to the next layer and so on [Hernández-Lobato and Adams, 2015]. This procedure requires a special form of the kernel function in order to compute the moments of these distributions (the kernel must have analytic expectations under a Gaussian *e.g.*, exponential quadratic, linear).

Hamiltonian Monte-Carlo

In the previous approaches the posterior distribution $p(\{\mathbf{U}_{[i]}\}_1^l|\mathbf{y})$ is approximated by a Gaussian distribution. The analysis of the posterior distribution $p(\{\mathbf{U}_{[i]}\}_1^l|\mathbf{y})$ in [Havasi et al., 2018], however, demonstrates non-Gaussian and multi-modal behavior. Therefore, [Havasi et al., 2018] propose an MCMC approach to deal with the inference in DGP based on Stochastic Gradient Hamiltonian Monte-Carlo (SGHMC). Moreover, a Markov Chain Expectation Maximization algorithm is developed for the optimization of the hyper-parameters.

Implicit Posterior Variational Inference

Following the analysis of the posterior in [Havasi et al., 2018], an implicit posterior variational inference (IPVL) approach for DGP inference is developed in [Haibin et al., 2019]. In fact, the proposed variational inference approach does not assume a Gaussian variational posterior for the induced variables. Moreover, the assumption of a factorized form over the layers of the induced variables is also relaxed. The proposed methodology draws posterior samples using a black-box generator parameterized by a parameter vector $\boldsymbol{\theta}_{B-B}$ and depending on the corresponding induced inputs. However, in the ELBO, the KL divergence between the prior and the approximated posterior can not be computed due to the implicit definition of the latter. To overcome this limitation, it is shown that the KL term can be expressed as an optimization result with respect to the parameters of a considered artificial neural network $\boldsymbol{\theta}_{ANN}$. This yields to two optimization problems, the optimization of the ELBO with respect to the parameters of the DGP and $\boldsymbol{\theta}_{B-B}$ and the optimization of the ANN parameters $\boldsymbol{\theta}_{ANN}$. Afterwards, the problem is casted into a game-theory problem where two players are considered in which each optimization problem corresponds to a strategy. It is shown then that the Nash-equilibrium of this game is a global optimizer of the ELBO and the optimal value $\boldsymbol{\theta}^*_{B-B}$ yields to the true posterior. Therefore, a best-response dynamics algorithm [Roughgarden, 2010] is used where each player improves its strategy using a stochastic gradient ascent update to obtain a Nash-equilibrium.

Sequential Inference

In [Wang et al., 2016], a sequential inference approach for DGP based on sampling is proposed. More specifically, an online setting is considered where the training data is taken sequentially. For each data input/output pair $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$, an estimation of the latent variables $\{\mathbf{h}^{(i)}_{[l]}\}^{L}_{l=1}$ is performed using sequential Monte Carlo sampling and each sample is weighted by the corresponding likelihood (state estimation step). Notice that it is only a point estimate of the latent variables, which does not propagate uncertainty. Once the latent variables are estimated, the posterior mean and covariance of each GP is updated based on a sparse online approximation.

Fig. 2.14 summarizes the different inference approaches of DGP discussed, with an emphasis on the particularity of each one. Table. 2.2 shows for each approach which model approximation and which inference approach are used.

Table 2.2 Model approximations and inference approaches for DGP training methods

Approach	**Inference approach**	**Approximation approach**
Damianou and Lawrence [2013]	Variational inference	Variational sparse GPs
Dai et al. [2015]	Variational inference	Variational sparse GPs
Salimbeni and Deisenroth [2017]	Variational inference	Variational sparse GPs
Haibin et al. [2019]	Variational inference	Variational sparse GPs
Cutajar et al. [2017]	Variational inference	Random feature-based GP
Bui et al. [2016]	Expectation propagation	Fully independent training conditional GPs
Havasi et al. [2018]	Markov-Chain Monte-Carlo	Variational sparse GPs
Rossi et al. [2020]	Markov-Chain Monte-Carlo	Variational sparse GPs

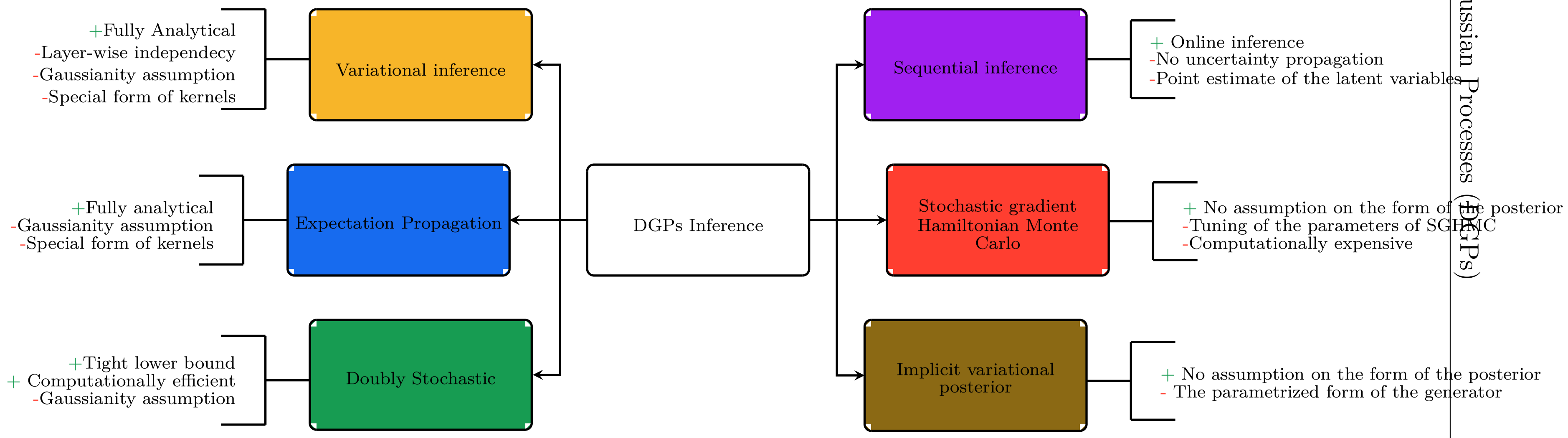

DGPs Inference
Variational inference
+Fully Analytical
-Layer-wise independecy
-Gaussianity assumption
-Special form of kernels
Expectation Propagation
+Fully analytical
-Gaussianity assumption
-Special form of kernels
Doubly Stochastic
+Tight lower bound
+ Computationally efficient
-Gaussianity assumption
Sequential inference
+ Online inference
-No uncertainty propagation
-Point estimate of the latent variables
Stochastic gradient Hamiltonian Monte Carlo
+ No assumption on the form of the posterior
-Tuning of the parameters of SGHMC
-Computationally expensive
Implicit variational posterior
+ No assumption on the form of the posterior
- The parametrized form of the generator

Fig. 2.14 The different DGPs inference approaches in the literature with an emphasis on the particularity of each approach.

2.3.3 Applications of Deep Gaussian Processes

Deep Gaussian processes emerged from the machine learning community, therefore, the first applications of DGPs concerned different problems in this field of research. For instance, DGPs showed a great potential in computer vision applications such as object detection and image classification [Damianou, 2015; Kumar et al., 2018; Blomqvist et al., 2019]. Adaptation of DGPs to computer vision yielded to convolutional kernels for DGPs [Kumar et al., 2018] and DGPs with convolutional structure [Blomqvist et al., 2019]. In [Kandemir, 2015], DGPs with a two-layer structure were used as transfer learning models, resulting in an asymmetric transfer strategy which outperforms state-of-the-art transfer learning models. DGPs were also adapted to speech synthesis [Koriyama and Kobayashi, 2019], where a DGP was incroporated into a statistical parametric speech synthesis model. The experimentations showed a better efficiency and robustness of DGPs compared to feed-forward DNNs. An autoencoder DGP model for novelty detection was proposed in [Domingues et al., 2018], it achieves competitive results to deep learning approaches. Inverse reinforcement learning has also witnessed the efficiency of a DGP [Jin et al., 2015], where it was used to learn latent rewards from limited data with complex feature representations.

DGPs are gaining popularity across other research fields. Since its potential in different machine learning problems, DGPs were naturally used in medicine for disease identification and diagnosis [Kandemir, 2015; Alaa and van der Schaar, 2017; Feng et al., 2018]. In [Kandemir, 2015], a DGP transfer learning model was used for cross-tissue tumor detection. In [Alaa and van der Schaar, 2017], a multi-task deep GP was developed for survival analysis with competing risks, where each task corresponds to a cause specific survival time. A supervised DGP were proposed in [Feng et al., 2018] for the classification of fetal heart rate tracings based on the pH values of the fetuses. Moreover, an unsupervised DGP was also developed for dimensional reduction of fetal heart rate signals. Another field of application of DGPs is ecological studies. For instance, in [Jančič et al., 2018], DGPs have been applied to atmospheric data with radionuclides in order to be used as part of a an assessment modeling system for nuclear plants.

Recently, an interest of DGP has arisen in engineering applications. In fact, due to their flexibility and power of representation in addition to the uncertainty quantification, DGPs have desirable feats as surrogate-models. Moreover, unlike the previous applications, engineering problems often involve computationally expensive simulation yielding to small-sized data-sets. DGPs prove to be efficient in this configuration [Damianou, 2015; Bui et al., 2016; Salimbeni and Deisenroth, 2017]. [Dutordoir et al., 2017] used

Deep Gaussian Processes in regression for non-stationary data where standard GPs are not suitable due to stationary covariance functions (see Section 3.1). It is used to approximate the lift of a Langley glide-back booster given the speed at re-entry and the angle of attack. Due to the transition from subsonic to supersonic non-stationarity is involved and DGPs show better performance than GP. In [Radaideh and Kozlowski, 2020], DGPs were used for modeling nuclear reactor simulation codes such as reactor fuel performance and reactor kinetic parameters and also for uncertainty quantification tasks such as uncertainty propagation and variance decomposition. A contaminant source localization application of DGP have been investigated in [Park et al., 2018], where a multi-output DGP is proposed to approximate a multi-zone computational fluid dynamics model. [Zhao et al., 2019] used DGP regression to predict the eddy current losses of a large turbine generator with a 6-dimensional input space formulation. The results obtained outperform GPs with different kernels and different machine learning models used for comparison such as support vector regression and AdaBoost. Moreover, the prediction obtained proves to be significantly close to the finite element simulation with an important saving in computational time. To deal with different levels of code fidelity, [Cutajar et al., 2019] proposed a multi-fidelity DGP where each layer corresponds to a fidelity level, resulting in a fusion of information across the different fidelities. DGPs were used for modeling aerodynamic flow quantities of interest such as the lift and drag coefficients given the flight conditions and the aerodynamic shape of the aircraft [Rajaram et al., 2020]. The results obtained are compared to the ones obtained by standard GPs, and proves the better accuracy of DGPs, however, the DGPs comes with a computational burden due to the approximate inference approaches used for training (Section 2.3.2) compared to GPs.

The contributions of this thesis fall within this continuity of work on DGPs in engineering applications and more specifically in the analysis and optimization of complex systems. The use of DGPs for non-stationarity elucidated in [Dutordoir et al., 2017] is investigated deeper in the context of Bayesian Optimization (BO) in Chapter 4 and adaptations are proposed to couple between DGPs and BO. In Chapter 5, the context of multiple objectives is considered. To take into account the correlation between the objectives, a multi-objective DGP model (MO-DGP) is developed where the layers of the DGP correspond to the objectives and are codependent. In Chapter 6, the multi-fidelity model proposed by [Cutajar et al., 2019] is improved by proposing another optimization framework for the model. Moreover, a new multi-fidelity model is developed for multi-fidelity problems characterized by different input space parametrizations. The developed model embeds a non-parametric Bayesian mapping from one fidelity input space to another, hence the name multi-fidelity embedded mapping model (MF-DGP-EM).

As mentioned in the introduction of this chapter, approaches based on DGPs are developed in this thesis to address different axes of the analysis and optimization of complex systems. While this chapter has served as an introduction to DGPs and their rich background from Bayesian modeling to Gaussian Processes, the next chapter introduces these different analyses and optimization problems for which the DGP based approaches are developed in the contribution chapters. More specifically, a review on non-stationary approaches for GPs as well as on single and multi-objective Bayesian optimization, and on multi-fidelity GP approaches is presented in the next chapter.

Chapter 3

Gaussian Process applications to the analysis and optimization of complex systems

"One of the characteristic features of mathematical models is that the same model, in a sense to be explained, can occur in, and be successfully employed in, fields with quite different subject matters."

P. Humphreys (2002)

Chapter goals

- Review and classification of the different Gaussian processes adaptations to non-stationary problems.
- Description of single-objective Bayesian optimization.
- Review of multi-objective Bayesian optimization for independent and correlated objective models.
- Review of Gaussian process-based multi-fidelity approaches with an emphasis on the case where model input variables are defined on different spaces.

$\mathcal{CH}_3$

The design analysis and optimization of complex systems often require computationally intensive simulation codes that involve black-box functions. For instance, within the context of multidisciplinary design optimization problems, disciplinary codes are often modeled as black-box functions and an evaluation requires an iterative loop between these disciplines (*e.g.*, structure using finite element analysis, aerodynamics using computational fluid dynamics for aerospace systems), inducing a computational burden [Balesdent et al., 2012b]. The analysis and optimization of such problems relying only on the simulation codes are difficult since only a few evaluations are available due to limited duration and computational budget. To avoid running excessively a computationally intensive function $f(\cdot)$, a limited number of evaluations n is used as training data (Design of Experiment DoE):

$$\begin{cases} \mathbf{X} = \left[\mathbf{x}^{(1)},\dots,\mathbf{x}^{(n)}\right]^{\top} & , \quad \mathbf{x}^{(i)} \in \mathbb{X} \subset \mathbb{R}^d, \forall i \in \{1,\dots,n\} \\ \mathbf{y} = \left[y^{(1)} = f\left(\mathbf{x}^{(1)}\right),\dots,y^{(n)} = f\left(\mathbf{x}^{(n)}\right)\right]^{\top} & , \quad y^{(i)} \in \mathbb{Y} \subset \mathbb{R}, \forall i \in \{1,\dots,n\} \end{cases} \tag{3.1}$$

Then, a regression model (called surrogate model, response surface model or metamodel) is used. Different surrogate models can be used in the analysis and optimization of complex systems (Chapter 1). Due to their interesting properties, Gaussian Processes (GPs) (Chapter 2) are a popular approach to address different problems in the analysis and optimization of complex systems [Wang et al., 2005; Wang and Shan, 2007; Forrester et al., 2008; Archetti and Candelieri, 2019]. This chapter presents a review of GPs in engineering design that covers the problems which are addressed in the contribution chapters of this thesis. The objective of this chapter is to set a unified view of the existing GP adaptations to each application before presenting the novel approaches proposed in the next chapters. Specifically, Section 3.1 reviews GP adaptations to non-stationary problems and Section 3.2 presents the Bayesian optimization framework, preparing the ground for Chapter 4. Next, Section 3.3 introduces multi-objective Bayesian optimization and its generalization to models taking into account the correlation between objectives, serving as a background for Chapter 5. Finally, Section 3.4 reviews the related existing works to the contributions of Chapter 6, covering GP literature for multi-fidelity analysis with an emphasis on the case where each fidelity is defined on its own input-space.

3.1 Non-stationary Gaussian Processes

The question of non-stationarity is discussed in different fields of research. In climate science due to dramatic changes in precipitation, the stationarity assumption is dropped for modeling climate phenomena [Cordery and L.Yao, 1993; Milly et al., 2008; Garg et al., 2012]. In signal processing and finance among other fields, non-stationary models are often used to fit time series over a long period of time [Konda, 2006].

Also in geostatistics, non-stationarity occurs when dealing with a region with different landscapes and topographic features [Atkinson and Lloyd, 2007].

In engineering design, due to the abrupt change of a physical property in one of the disciplines involved in the design process, the response of interest may vary with different degrees of smoothness from one region of the design space to another. Specifically, aerospace design engineering involves different disciplines that can induce non-stationary phenomena. For example in aerodynamics, Computational Fluid Dynamics (CFD) problems often have different specific flow regimes due to separation zones, circulating flows, vortex bursts, transitions from subsonic to transonic, supersonic, and hypersonic flow regimes. In propulsion, the combustion involves irreversible thermodynamics transformations that are characterized by sudden and rapid changes (*e.g.*, sudden state change of the matter, spontaneous chemical reactions, spontaneous mixing of matter of different states). There can also be non-stationarities in the structure discipline, for example in the stress-strain curve of a material, the elastic region, the strain hardening region, and the necking region present different behaviors.

Standard GP regression is based on the *a priori* that the variation of the output depends only on the variation of the corresponding inputs and not in the considered region. This is induced by the use of stationary covariance functions that depends only on a distance in the input space:

$$\forall \mathbf{x}, \mathbf{x}', \boldsymbol{\lambda} \in \mathbb{R}^d, k(\mathbf{x}+\boldsymbol{\lambda}, \mathbf{x}\text{'}+\boldsymbol{\lambda}) = k(\mathbf{x}, \mathbf{x}\text{'}) = k_* \left(\Delta_{\text{Mahalanobis}}(\mathbf{x}, \mathbf{x}')\right) \tag{3.2}$$

with:

$$\Delta_{\text{Mahalanobis}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x}-\mathbf{x}')^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\mathbf{x}')} \tag{3.3}$$

where $k_*(\cdot)$ is a scalar function defined on $\mathbb{R}$, $\Delta_{\text{Mahalanobis}}(\cdot,\cdot)$ refers to the Mahalanobis distance, and $\boldsymbol{\Sigma}$ is a $d \times d$ positive definite matrix. This *a priori* is generally valid for functions where there is no change in the smoothness of the function considered along the design space. However, this is not suitable for functions with abrupt and local variations. In fact, stationary covariance functions have a constant parameter called length-scale encoded in $\boldsymbol{\Sigma}$ that controls the variations of the response. For instance, for Automatic Relevance Determination (ARD) kernels [Williams and Rasmussen, 2006], the length-scale of each dimension θ_{ls_i} is encoded in a diagonal matrix $\boldsymbol{\Sigma} = diag\left(\left[\frac{1}{\theta^2_{ls_1}}, \dots, \frac{1}{\theta^2_{ls_d}}\right]\right)$. The length-scale θ_{ls_i} will take high values for strong correlations *i.e.* regions with low variations and will take low values for weak correlations *i.e.* regions with high variations.

As a representative example, the modified Xiong function [Xiong et al., 2007] (*cf.* Eq.(C.1) in Appendix C, Fig. 3.1), has two regions with different levels of variation. It presents one region where the function varies with a high frequency $x \in [0, 0.3]$ and the other where the function varies slowly $x \in [0.3, 1]$. This makes the classical GP regression not suitable for this function (Fig. 3.1). As it can be seen in this case, the learning process results in a length-scale value that is consistent in the high frequency region but not in the low frequency region. This yields to a GP model that continues to oscillate and can not capture the two trends of this function. To overcome this issue, different GP adaptations to non-stationarity have been proposed. These adaptations can be classified into three main classes: direct formulation of a non-stationary covariance function, local stationary covariance functions, and input-space warping approaches.

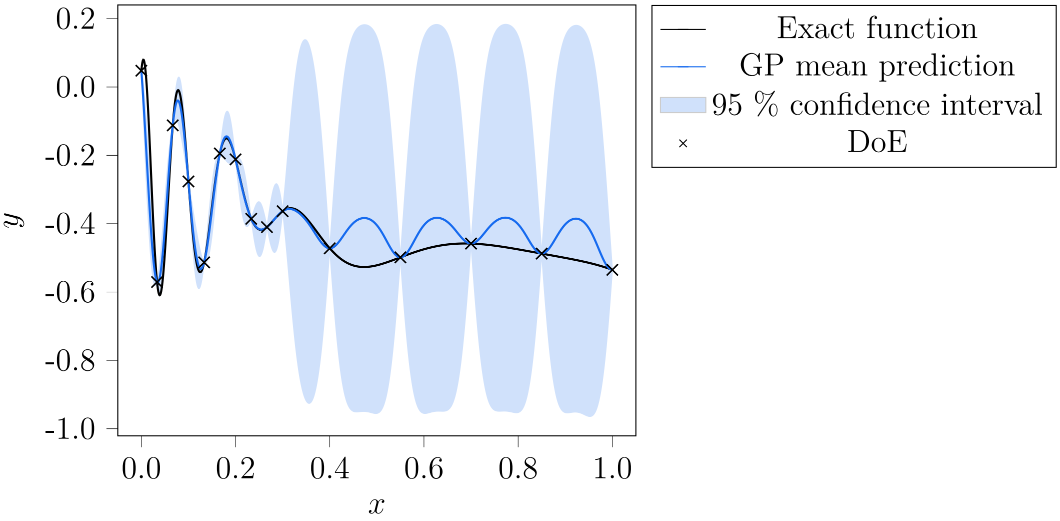


Fig. 3.1 Approximation of the modified Xiong-function, a non-stationary 1-dimensional function, by a standard GP model. GP can not capture the stability of the region $[0.4, 1]$ and continues to oscillate.

3.1.1 Direct formulation of non-stationary kernels

Most of the methods in the literature that use a direct formulation of a non-stationary covariance function follow the work of [Higdon et al., 1999]. The main idea is to use a convolution product of a spatially-varying kernel function to define a class of non-stationary kernels:

$$k^{NS}(\mathbf{x}, \mathbf{x}') = \int_{\mathbb{R}^d} k^S(\mathbf{x}, \mathbf{v}) k^S(\mathbf{x}', \mathbf{v}) \mathrm{d}\mathbf{v} \tag{3.4}$$

where $k^S(\mathbf{x},\mathbf{x}') = k_*^S(\Delta_{\text{Mahalanobis}}(\mathbf{x},\mathbf{x}'))$ is a stationary kernel function and $\mathbf{x},\mathbf{x}'$ are locations in $\mathbb{R}^d$. The analytical form of the non-stationary covariance resulting from the convolution of Gaussian kernels is derived in [Higdon et al., 1999]. This approach has been extended in [Paciorek and Schervish, 2006] where the analytical form of the non-stationary covariance function resulting from the convolution of any stationary kernel is given:

$$k^{NS}(\mathbf{x},\mathbf{x}') = |\boldsymbol{\Sigma}_{\mathbf{x}}|^{\frac{1}{4}} |\boldsymbol{\Sigma}_{\mathbf{x}'}|^{\frac{1}{4}} \left| \frac{\boldsymbol{\Sigma}_{\mathbf{x}} + \boldsymbol{\Sigma}_{\mathbf{x}'}}{2} \right|^{-\frac{1}{2}} k_*^S(\Delta_{\text{NS}}(\mathbf{x},\mathbf{x}')) \tag{3.5}$$

where:

$$\Delta_{\text{NS}}(\mathbf{x},\mathbf{x}') = \sqrt{(\mathbf{x}-\mathbf{x}')^T \left(\frac{\boldsymbol{\Sigma}_{\mathbf{x}} + \boldsymbol{\Sigma}_{\mathbf{x}'}}{2}\right)^{-1} (\mathbf{x}-\mathbf{x}')} \tag{3.6}$$

and $\boldsymbol{\Sigma}_{\mathbf{x}} = \boldsymbol{\Sigma}(\mathbf{x})$ is a $d \times d$ matrix-valued function which is positive definite for all $\mathbf{x}$ in $\mathbb{R}^d$. In the stationary case, $\boldsymbol{\Sigma}(\cdot)$ is a constant arbitrary matrix. The interesting observation is that in the resulting non-stationary covariance function $k^{NS}(\cdot,\cdot)$, the Mahalanobis distance $\Delta_{\text{Mahalanobis}}(\cdot,\cdot)$ is not used within the stationary covariance function $k_*^S(\cdot)$. Instead, a distance measure $\Delta_{\text{NS}}(\mathbf{x},\mathbf{x}')$ with the average of the kernel matrices $\boldsymbol{\Sigma}(\cdot)$ in the two locations $\mathbf{x}$ and $\mathbf{x}'$ is used. Therefore, the local characteristics of the two locations encoded in their respective kernel matrices influence the covariance value yielding to a non-stationary model. The special case of a non-stationary Matérn covariance function is derived in [Paciorek and Schervish, 2006] using Eq. (3.5). The construction of the kernel matrix $\boldsymbol{\Sigma}(\cdot)$ for each $\mathbf{x}$ in the domain is performed *via* an eigendecomposition, which can be difficult when increasing the input-space dimension. [Gibbs, 1998] proposed a simpler parameterization by choosing the matrix $\boldsymbol{\Sigma}(\mathbf{x})$ as a diagonal matrix of length-scales parameterized using a set of basis functions. Hence, length-scales depending on the location of $\mathbf{x}$ are obtained. In [Plagemann et al., 2008], a Gaussian Process Local Length-scales (GP-LL) model is developed. It consists in augmenting the latent space with a set of locations $\mathbf{X}_{\boldsymbol{\theta}_{\text{ls}}} = \left[\mathbf{x}_{\boldsymbol{\theta}_{\text{ls}}}^{(1)}, \ldots, \mathbf{x}_{\boldsymbol{\theta}_{\text{ls}}}^{(m_{\text{ls}})}\right]$ and their corresponding length-scales $\boldsymbol{\Theta}_{\text{ls}} = \left[\boldsymbol{\theta}_{\text{ls}}^{(1)}, \ldots \boldsymbol{\theta}_{\text{ls}}^{(m_{\text{ls}})}\right]$ where m_{ls} is a determined number of length-scale locations. A GP prior $f_{\boldsymbol{\Theta}_{\text{ls}}}(\mathbf{X}_{\boldsymbol{\theta}_{\text{ls}}}) = \boldsymbol{\Theta}_{\text{ls}}$ is then placed over $\boldsymbol{\Theta}_{\text{ls}}$. The learning of the hyper-parameters as well as the latent variables $\boldsymbol{\Theta}_{\text{ls}}$ and $\mathbf{X}_{\boldsymbol{\theta}_{\text{ls}}}$ is performed by a Maximum A Posteriori (MAP) procedure of the latent length-scales $p\left(\boldsymbol{\Theta}_{\text{ls}}|\mathbf{y},\mathbf{X},\mathbf{X}_{\boldsymbol{\theta}_{\text{ls}}}\right)$. Once this optimization is performed, the inference of the GP $f(\cdot)$ is carried out classically to obtain a predictive mean response and associated variance. In Fig. 3.2, three length-scale locations are used to approximate the modified Xiong-function. In the learning process, the length-scale values adapt to the variations of the

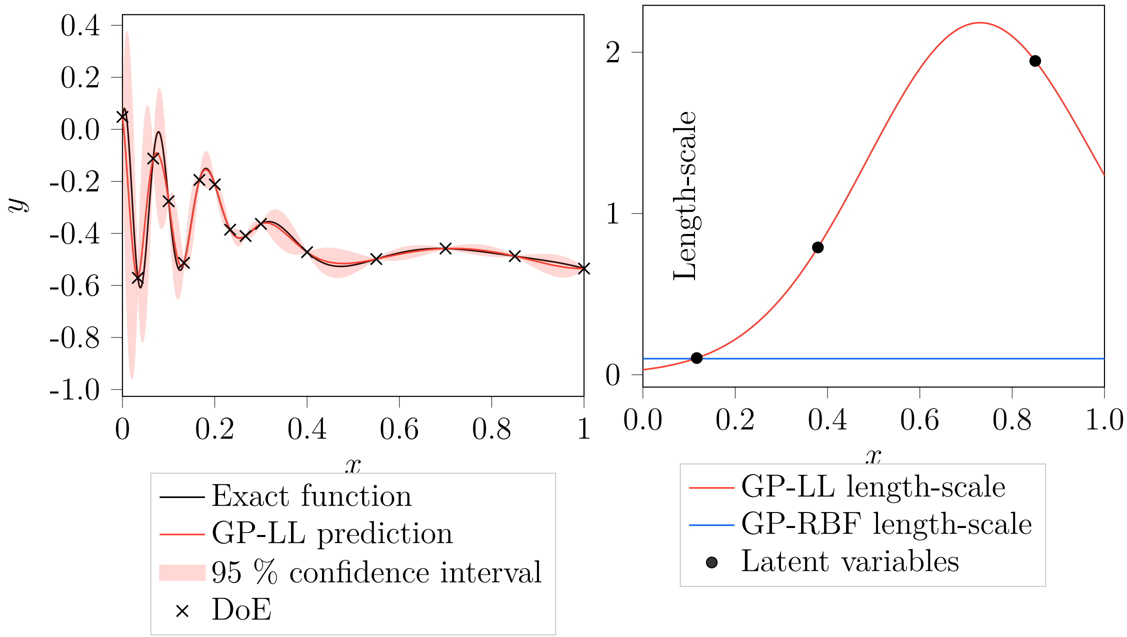


Fig. 3.2 Approximation of the modified Xiong-function by a Gaussian Process Local Length-scales (GP-LL) model. (left) The prediction of the GP-LL model captures the non-stationarity behavior by using an input dependent length-scale. (right) The input dependent length-scale is learned using a GP with 3 training locations that are also optimized. The length-scale decreases in regions of high-variations and increases in regions of low-variations.

target function yielding to high-values in regions with low-variations and low-values in regions with high variations, in contrast with a stationary kernel where the value of the length-scale is constant. However, using a MAP estimate does not take into account the uncertainty of the latent length-scales and may under-estimate the overall predictive uncertainty. To overcome this issue, [Heinonen et al., 2016] proposed sampling the exact posterior using Hamiltonian Monte Carlo (HMC) (see Chapter 2, Section 2.1.2) instead of using a MAP estimate of the latent length-scales. However, this class of approaches may not be suitable for high-dimensional problems due to the high number of parameters required [Paciorek and Schervish, 2006; Plagemann et al., 2008].

3.1.2 Local stationary covariance functions

The local stationary approaches are based on the idea that non-stationary functions have a local stationary behavior. In [Haas, 1990] a moving window approach is proposed where the training and prediction regions move along the input space using a stationary covariance function. This window has to be restrained enough so that the function is stationary within it. Other methods consist in dividing the input space into various subsets and using a different model for each subset [Tresp, 2001; Rasmussen and Ghahramani, 2002; Gramacy and Apley, 2015; Krityakierne and Ginsbourger, 2015], this is also known as mixture of experts. Specifically, in a GP mixture of expert (MoE) m_{clusters} independent GPs are considered (experts) and the likelihood is modeled as a probabilistic mixture over all possible assignments of the data to the experts:

$$p(\mathbf{y}|\mathbf{X}, \{\boldsymbol{\theta}^{(j)}\}_{j=1}^{m_{\text{clusters}}}) = \prod_{i=1}^{n} \sum_{j=1}^{m_{\text{clusters}}} p(y^{(i)}|\mathcal{C} = \mathcal{C}_j, \mathbf{x}^{(i)}, \boldsymbol{\theta}^{(j)}) p(\mathcal{C} = \mathcal{C}_j|\mathbf{x}^{(i)}, \boldsymbol{\theta}_{\mathcal{C}}) \tag{3.7}$$

where $\boldsymbol{\theta}^{(j)}$ are the parameters of expert j, $\mathcal{C}$ determines which cluster is active, and $\boldsymbol{\theta}_{\mathcal{C}}$ is the vector of parameters of the gating network that is the function that assigns a probability to an expert given the input. Therefore, learning in GPs MoE comes back to learning the hyper-parameters of each GP as well as the parameters of the gating network. Since the models are considered independent, the posterior predictive variable $y^*|\mathbf{y}, \mathbf{X}$ of a MoE GP in a location $\mathbf{x}^*$ is a linear combination of each expert posterior predictive variable, thus, maintaining the Gaussian distribution form of the predictive posterior. In [Tresp, 2001] where the GPs MoE were first introduced, in addition to the GP experts, the gating functions are considered as GPs in order to define a soft-max gating network and the learning is performed using an Expectation-Maximization algorithm [Moon, 1996]. A gating network based on Dirichlet Process and a kernel classifier is developed in [Rasmussen and Ghahramani, 2002], this allows a flexible number of experts that depends on the size of the data and also that each expert uses its own set of data. The learning is performed using MCMC approaches (Gibbs sampling for the conditional distribution of the state variable $\mathcal{C}$ and HMC for the GP hyper-parameters). A generative formulation of this approach (considering the joint distribution inputs, outputs) has been proposed in [Meeds and Osindero, 2006] and then improved in [Gadd et al., 2020] by using the enriched Dirichlet process. [Gramacy and Lee, 2008] proposed a tree-based approach where the input space is divided into rectangular sub-spaces yielding to a hierarchical tree structure where multiple splits of the input space occur at each level of the tree until reaching the leaves which correspond to the different GPs. The tree is constructed using well-established techniques on Bayesian classification and regression trees to split, grow, or prune the tree, and the inference is performed using MCMC approaches. In [Bettebghor et al., 2011], a mixture of a portfolio of models including GPs and other regression models, such as, artificial neural network and moving least squares, is developed. As in [Tresp, 2001], an Expectation-Maximization algorithm for Gaussian mixture models is used to subdivide the input space. The parameters obtained by the algorithm are used to combine the different models. The interest of the developed framework is that it uses different classes of models. However, the choice of which expert to use is based on cross-validation which is not usually practical when dealing with computationally intensive problems.

Recently, approaches based on Sum-Product Networks (SPN) have been adapted to GPs [Trapp et al., 2020]. An SPN-GP corresponds to an acyclic directed graph containing different types of nodes: sum nodes, product nodes, split nodes, and GPs leave nodes. A sum node computes a weighted sum over its children (mixture), the product node computes the Cartesian product of the outputs in the case of multi-output GPs (output independence), and the split nodes divide the input space (input independence). The response of an SPN-GP is obtained by propagating the input through the leaves until reaching the root. The SPN-GP, unlike the previously described MoE, allows exact inference of the posterior distribution. In Fig. 3.3, an SPN-GP model is used to approximate the modified Xiong-function. The formulation involves a sum of three nodes with their respective weights and each of these nodes subdivides the input space into two regions governed each by its own GP.

These approaches present some limitations. Indeed, in computationally expensive problems, data are sparse and using a local surrogate model with sparser data may be problematic.

3.1.3 Warped GPs

These approaches first introduced by [Sampson and Guttorp, 1992], also called non-linear mapping, consist in deforming the input space to express the non-stationarity using a stationary covariance function. Specifically, a stationary covariance function $k^S(\cdot,\cdot)$, and a function $\boldsymbol{\psi}(\cdot):\mathbb{R}^d\rightarrow\mathbb{R}^d$ are considered, then, the non-stationary covariance function is obtained by simply combining $\boldsymbol{\psi}(\cdot)$ and $k^S(\cdot,\cdot)$:

$$k^{NS}(\mathbf{x},\mathbf{x}')=k^S(\boldsymbol{\psi}(\mathbf{x}),\boldsymbol{\psi}(\mathbf{x}')) \tag{3.8}$$

The difficult task in this class of approaches is the determination of $\boldsymbol{\psi}(\cdot)$. Gibbs approach that was described in the direct formulation methods [Gibbs, 1998] can also be obtained *via* input warping. It consists in considering a mapping $\boldsymbol{\psi}^{\text{Gibbs}}(\cdot)$ as a multidimensional integral of non-negative density functions $\{\eta_i^{\text{Gibbs}}(\cdot)\}_{i=1}^{d}$. These density functions are defined as a weighted sum of m_{rbf} positive Gaussian radial basis

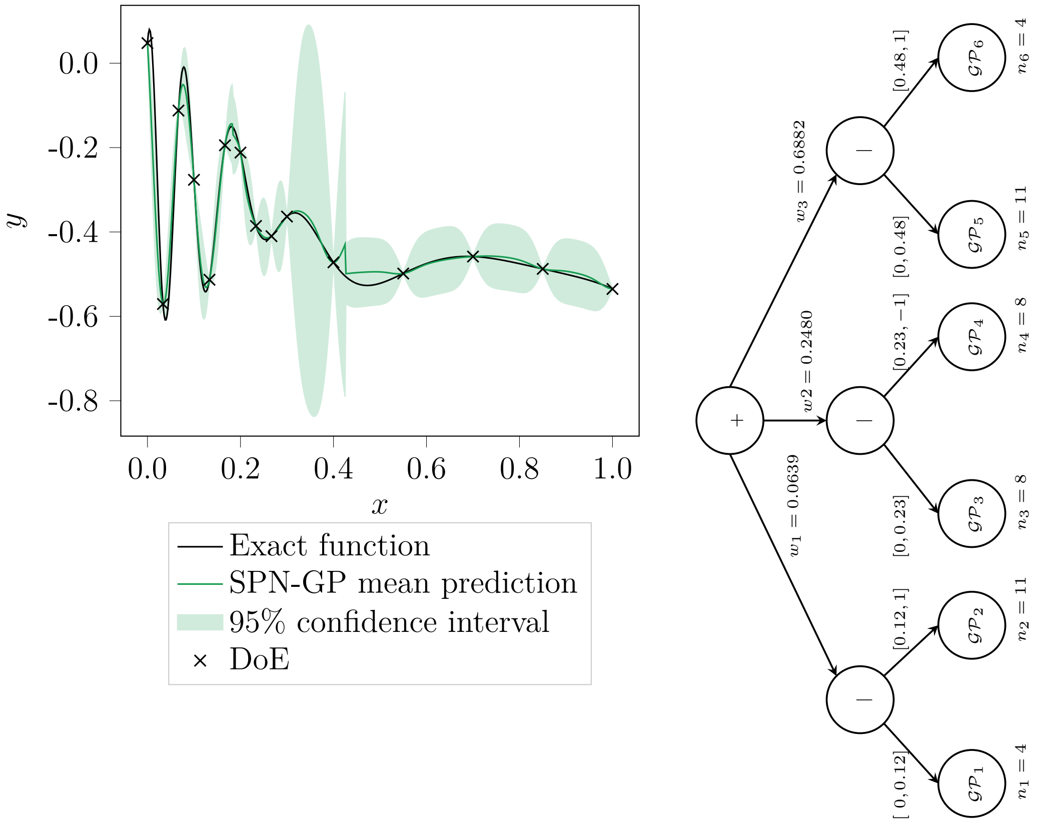


Fig. 3.3 Approximation of the modified Xiong-function by a Sum-Product Network GP (SPN-GP) model. (left) The prediction of the SPN-GP model captures the non-stationarity behavior by using a mixture of GPs each with its own length-scale. (right) Illustration of the structure of the SPN with the sum node represented by + with weights $\mathbf{w}$ over its children and the split nodes represented by |.

functions.

$$\psi_i^{\text{Gibbs}}(\mathbf{x}) = \int_0^{x_1} \cdots \int_0^{x_d} \eta_i^{\text{Gibbs}}(\mathbf{v}) \mathrm{d}v_1 \ldots \mathrm{d}v_d \tag{3.9}$$

$$\eta_i^{\text{Gibbs}}(\mathbf{v}) = \sum_{j=1}^{m_{\text{rbf}}} w_i^{(j)} \phi_{\text{rbf}}^{(j)}(\mathbf{v}), i = 1, \ldots, d \tag{3.10}$$

where $\boldsymbol{\psi}^{\text{Gibbs}}(\cdot) = [\psi_1^{\text{Gibbs}}(\cdot), \ldots, \psi_d^{\text{Gibbs}}(\cdot)]$, $\mathbf{v} = [v_1, \ldots, v_d]$, $\eta_i^{\text{Gibbs}}(\cdot)$ is the i^{th} coordinate of the density function, $\phi_{\text{rbf}}^{(j)}(\cdot)$ is a fixed Gaussian radial basis function, and $w_i^{(j)}$ is the weight of the j^{th} basis function in the i^{th} coordinate of the density function. The drawback of this approach is that the number of radial basis functions m_{rbf} needed to capture the non-stationarity increases with the dimension of the space d, inducing an over-parameterized structure of the covariance function in high-dimensional situations [Xiong et al., 2007]. To overcome this issue, the non-linear mapping approach proposed by [Xiong et al., 2007] assumes independency between each dimension of the mapping, this reduces the multivariate density function in Eq. (3.9) to a univariate one.

$$\psi_i^{\text{Xiong}}(\mathbf{x}) = \int_0^{x_i} \eta_i^{\text{Xiong}}(v_i)\mathrm{d}v_i \tag{3.11}$$

Furthermore, the density function is defined as a sum of $m_{h_{\text{linear}}}$ piece-wise linear functions $h_{\text{linear}}(\cdot)$:

$$\eta_i^{\text{Xiong}}(v_i) = \sum_{j=1}^{m_{h_{\text{linear}}}} h_{\text{linear},i}^{(j)}(v_i),\ i = 1,\dots,d \tag{3.12}$$

where:

$$h_{\text{linear},i}^{(j)}(x_i) = a_i^{(j)}x_i + b_i^{(j)}, x_i \in [\delta^{(j-1)}, \delta^{(j)}] \tag{3.13}$$

$a_i^{(j)}, b_i^{(j)}$ are respectively the slope and the intercept of the linear function $h^{(j)}$ at the i^{th} coordinate and $(\delta^{(0)},\dots,\delta^{(m)})$ are a series of knots evenly placed along each dimension. This approach allows a better scalability to high dimensional design spaces. However, the deformation is done only along canonical axes which may not be adapted to handle non-stationarity behavior following non-canonical axes. [Marmin et al., 2018] addressed this issue by introducing a parameterized matrix $\mathbf{A}$. This allows a linear mapping of the input space before undergoing the non-linear mapping of $\boldsymbol{\psi}(\cdot)$.

$$k^{NS}(\mathbf{x},\mathbf{x}') = k^S(\boldsymbol{\psi}(\mathbf{A}.\mathbf{x}), \boldsymbol{\psi}(\mathbf{A}.\mathbf{x}')) \tag{3.14}$$

[Snoek et al., 2014] proposed an input warping using a Beta cumulative distribution function as a mapping.

$$\begin{aligned}\psi_i^{\text{Snoek}}(\mathbf{x}) &= \Phi_{\text{Beta}}(x_i; \alpha_i, \beta_i) \\ &\propto \int_0^{x_i} v^{\alpha_i - 1}(1-v)^{\beta_i - 1}\mathrm{d}v\end{aligned} \tag{3.15}$$

where Φ_{Beta} is the Beta cumulative distribution function and α_i and β_i are the parameters of the Beta distribution. The interesting characteristic of this approach is its low parameterization form. In fact, the Beta distribution is defined only by two parameters and its cumulative distribution can express a wide variety of monotonic functions. A log-normal prior is also placed on the parameters α_i and β_i and an MCMC approach is followed for inference. In Fig. 3.4, this approach is used to approximate the modified Xiong-function. The function captures well the non-stationary behavior by stretching the input space region with high-variations.

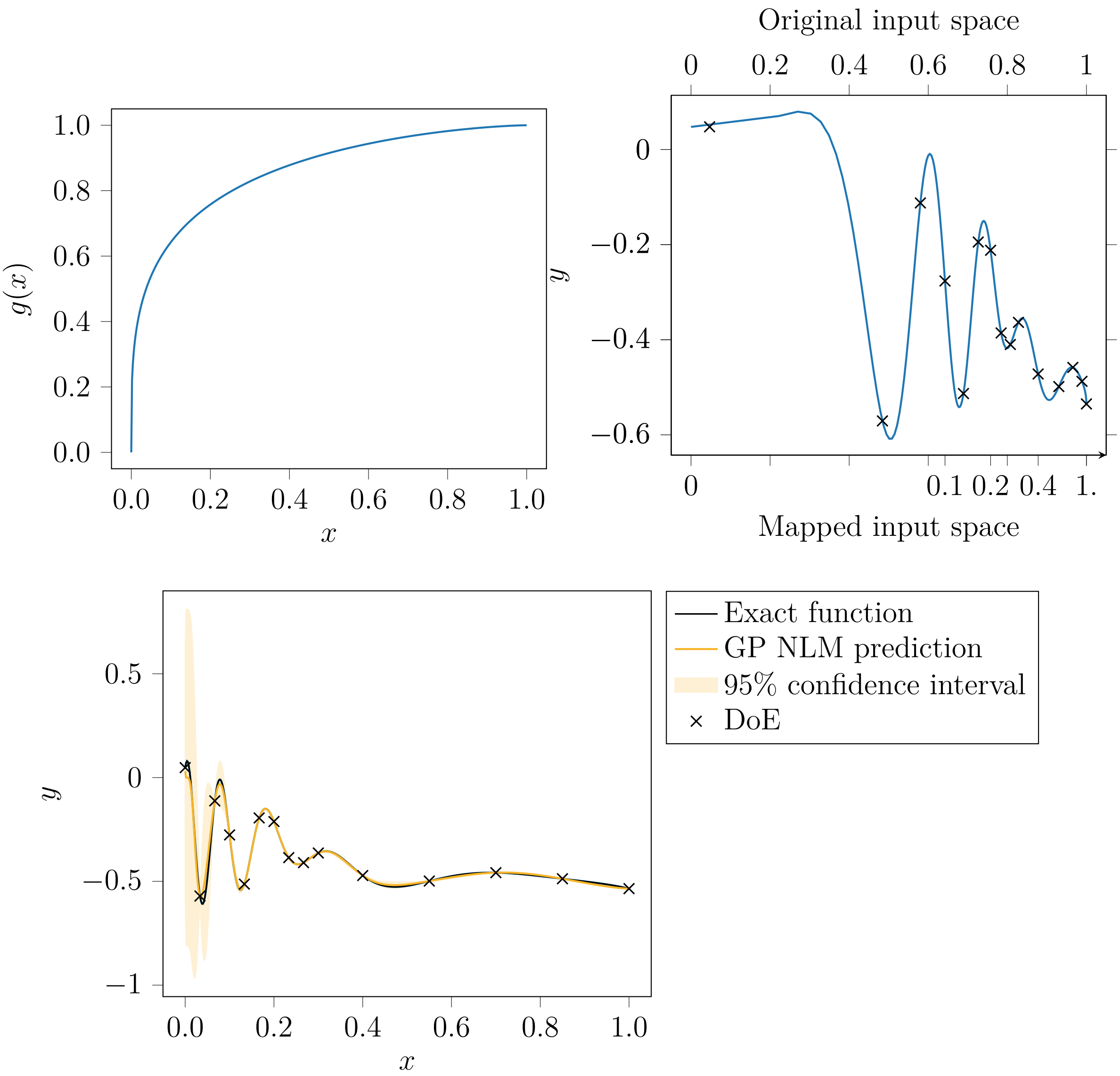


Fig. 3.4 Approximation of the modified Xiong-function by a Non-Linear Mapping GP(NLM-GP) model. (top left) The warping function of the NLM using the cumulative distribution of the Beta distribution. (top right) The region of high-variation of the modified Xiong-function is stretched yielding to relatively stabilized variations along the mapped input space. (bottom) The prediction obtained by the NLM-GP captures the non-stationarity of the modified Xiong-function.

The non-linear mapping approaches use a parameterized function to map the original input space to a mapped space with non-stationary behavior. However, the choice of a parameterized function is not an easy task and can be problem-dependent [Xiong et al., 2007], moreover, it does not include uncertainty information. In Chapter 4, Deep Gaussian Processes (DGPs, Chapter 2, Section 2.3) are proposed to overcome these limitations of the non-linear mapping for handling non-stationarity. Deep Gaussian Processes were first used to handle non-stationarity in [Dutordoir et al., 2017]. In fact, DGPs can be seen as an unparameterized version of input-warping where the first layers of DGPs stretch the input-space to allow better representation of the non-stationarity. Moreover, being fully Bayesian models, DGPs allow the uncertainty to be propagated through these input-warping layers.

To summarize these different approaches, the three classes of non-stationary GPs are depicted in Fig. 3.5.

Modeling expensive black-box functions given a DoE with these approaches allows one to exhibit possible non-stationary behaviors and to analyze the regions of the design space that show important variations relatively to other regions. However, in some applications, in addition to the analysis of the expensive black-box functions given a DoE, the final objective is to obtain the optimum of that function with a minimum number of evaluations. For that, Bayesian optimization is a popular approach. The next section introduces this approach of optimization in the context of expensive black-box function.

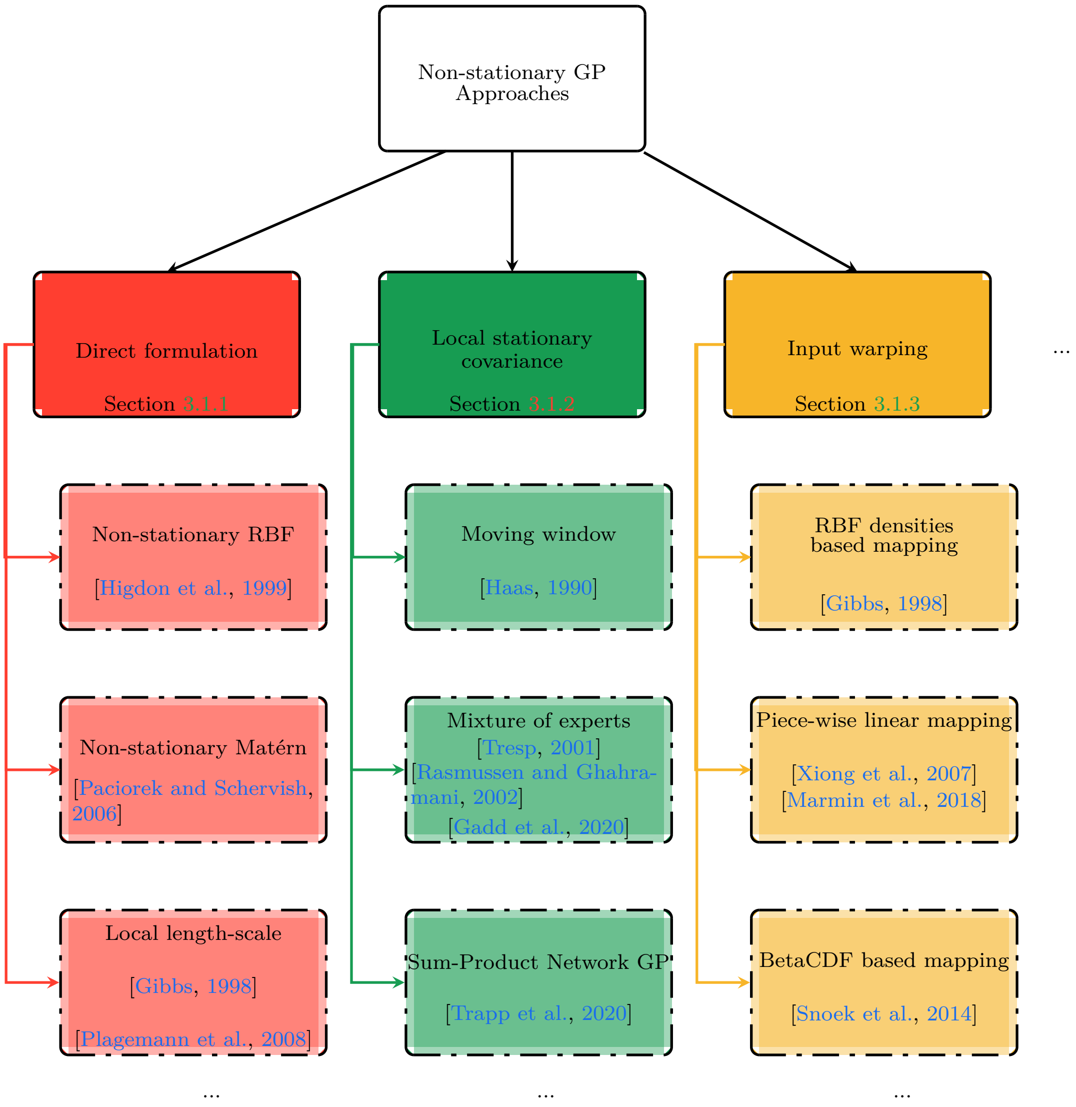


Fig. 3.5 Summary of the presented non-stationary GP approaches.

3.2 Bayesian Optimization (BO)

Given computationally intensive and black-box functions as objective $f : \mathbb{X} \subseteq \mathbb{R}^d \to \mathbb{R}$ and n_c constraints $g_j : \mathbb{X} \subseteq \mathbb{R}^d \to \mathbb{R}, j \in \{1, \ldots, n_c\}$, the following minimization problem $(\mathcal{P}_{\min})$ is defined (minimization is considered without loss of generality):

$$(\mathcal{P}_{\min}) \left| \begin{array}{lll} \underset{\mathbf{x}}{\text{Minimize}} & y & = f(\mathbf{x}) \\ \text{subject to} & g_j(\mathbf{x}) & \leq 0, \ \forall j \in \{1, \ldots, n_c\} \end{array} \right. \tag{3.16}$$

When dealing with expensive and black-box functions relying on legacy codes that do not provide analytical forms of the functions or the gradients (*e.g.*, coupled multi-disciplinary analysis [Balesdent et al., 2012b]), the use of exact optimization approaches is often not tractable [Wang and Shan, 2007; Forrester et al., 2008; Archetti and Candelieri, 2019]. Furthermore, the high computational cost makes the use of algorithms that require a large number of evaluations (gradient approximation, evolutionary algorithms, *etc.*) not suitable. Moreover, the objective and constraint functions involved often have non-linear landscapes with multiple local optima, hence, making the optimization problem more complex to solve.

One popular way to deal with expensive black-box function optimization is Bayesian Optimization (BO) [Močkus, 1975]. BO algorithms are iterative sampling procedures based on Bayesian models. To avoid running excessively the computationally intensive functions, Bayesian models emulate the statistical relationships between the design variables and the responses (objective function and constraints) given the DoE:

$$(DoE)\left\{\begin{array}{lcllr} \mathbf{X} & = & \left[\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}\right]^\top & , & \mathbf{x}^{(i)}\in\mathbb{X}\subset\mathbb{R}^d,\forall i\in\{1,\ldots,n\} \\ \mathbf{y} & = & \left[y^{(1)}=f\left(\mathbf{x}^{(1)}\right),\ldots,y^{(n)}=f\left(\mathbf{x}^{(n)}\right)\right]^\top & , & y^{(i)}\in\mathbb{Y}\subset\mathbb{R},\forall i\in\{1,\ldots,n\} \\ \mathbf{c}_j & = & \left[c_j^{(1)}=g_j\left(\mathbf{x}^{(1)}\right),\ldots,c_j^{(n)}=g_j\left(\mathbf{x}^{(n)}\right)\right]^\top & , & \forall j\in\{1,\ldots,n_c\} \end{array}\right. \tag{3.17}$$

Different surrogate models can be used in design optimization [Wang and Shan, 2007; Forrester et al., 2008; Viana et al., 2013]. The most popular BO algorithms are based on GP regression [Jones et al., 1998; Shahriari et al., 2015; Bouhlel et al., 2018; Frazier, 2018; Archetti and Candelieri, 2019]. The main advantage of a GP is that in addition to the prediction, it provides an uncertainty estimation of the surrogate model response that is obtained analytically and can be used for optimization purposes (Chapter 2, Section 2.2).

3.2.1 Bayesian Optimization Framework

BO algorithms are sequential design algorithms. The design space is filled sequentially with new candidates to improve the current minimum in the DoE:

$$y_{\min} = \min\left\{f\left(\mathbf{x}^{(i)}\right)|i\in\{1,\ldots,n\} \text{ and } \forall j\in\{1,\ldots,n_c\}, g_j\left(\mathbf{x}^{(i)}\right)\leq 0\right\} \tag{3.18}$$

This sequential aspect of BO algorithms consists of two iterative operations. The first one is the modeling of the expensive black-box functions $(f(\cdot), g_1(\cdot),\ldots,g_{n_c}(\cdot))$ involved in the optimization problem based on the DoE $\mathbf{X}$ and the corresponding exact evaluations $\mathbf{y},\mathbf{c}_1,\ldots,\mathbf{c}_{n_c}$ using GPs to obtain posterior mean prediction and associated variance $\left(\left(\hat{f}(\cdot), \hat{s}_f^2(\cdot)\right), \left(\hat{g}_1(\cdot), \hat{s}_{g_1}^2(\cdot)\right) \ldots, \left(\hat{g}_{n_c}(\cdot), \hat{s}_{g_{n_c}}^2(\cdot)\right)\right)$. These latter are cheaper to evaluate, which enables a larger number of evaluations than the exact functions.

The second operation consists in determining the most promising candidate to add to the current DoE in order to improve the current minimum y_{min} using the information given by the GPs. This is achieved by optimizing an acquisition function (also called infill sampling criterion) on the design space, which is a performance measure of the potential of a candidate from a minimization point of view. Once the most promising point is added to the data-set, it is evaluated on the exact expensive functions and the surrogate models are updated, and so on until a stopping criterion is reached (Fig. 3.6). Hence, the two key aspects in BO algorithms are the surrogate model and the infill sampling criterion.

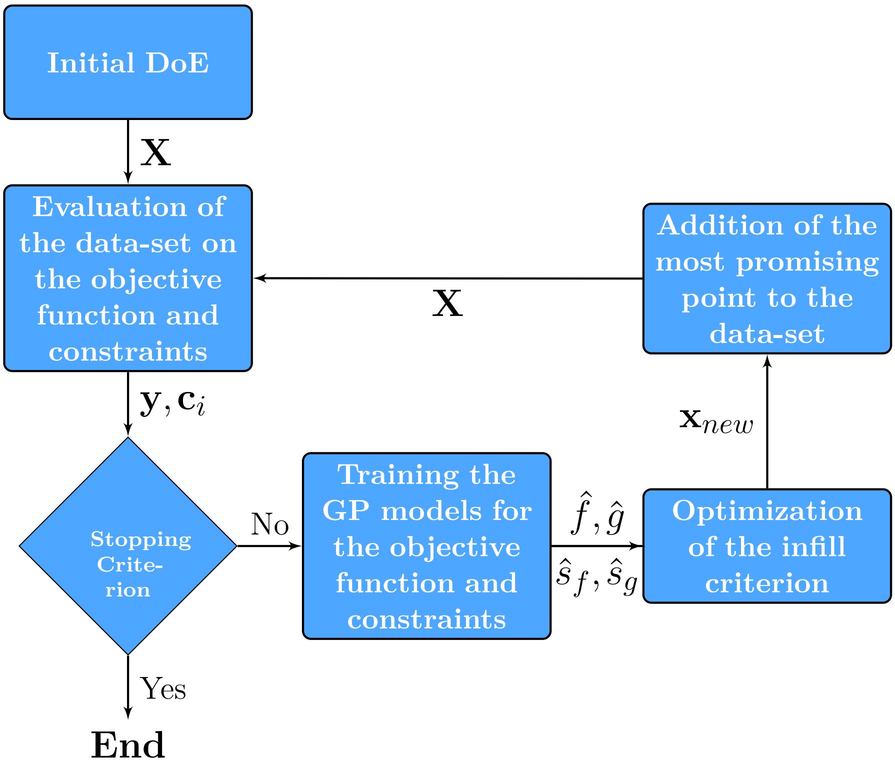

Initial DoE
$\mathbf{X}$
Evaluation of the data-set on the objective function and constraints
$\mathbf{X}$
Addition of the most promising point to the data-set
$\mathbf{y}, \mathbf{c}_i$
$\mathbf{x}_{new}$
Stopping Crite-rion
No
Training the GP models for the objective function and constraints
$\hat{f}, \hat{g}$
$\hat{s}_f, \hat{s}_g$
Optimization of the infill criterion
Yes
End

Fig. 3.6 Framework of Bayesian optimization with Gaussian process. It consists of two iterative procedures, 1) training Gaussian process models 2) optimization of an infill criterion to add the most promising candidate to the data-set. The stopping criterion is often chosen to be the number of evaluations of the expensive functions.

3.2.2 Infill criteria

For selecting infill sample candidates, a variety of criteria has been developed [Picheny et al., 2013]. Each criterion performs a trade-off between exploration *i.e.* investigating regions where the variance of the GP model is large and exploitation *i.e.* investigating regions where the GP prediction is minimal. One of the most used criteria is the Expected Improvement (EI) [Schonlau et al., 1996; Jones et al., 1998]. It takes into account the improvement induced by a candidate $\mathbf{x}$ that is defined as: $\mathcal{I}(\mathbf{x}) = \max\{0, y_{\min} - f(\mathbf{x})\}$. EI is then computed as the expectation of the improvement with respect to the posterior distribution:

$$\begin{aligned} EI(\mathbf{x}) =& \mathbb{E}_{p(f(\mathbf{x})|\mathbf{y},\mathbf{x},\mathbf{X})}\left[\mathcal{I}(\mathbf{x})\right] \\ & \int_{\mathbb{R}} \max\{0, y_{\min} - f(\mathbf{x})\} p(f(\mathbf{x})|\mathbf{y},\mathbf{x},\mathbf{X}) \mathrm{d}f(\mathbf{x}) \end{aligned} \tag{3.19}$$

For Gaussian posterior distributions the EI has a fully analytical form:

$$EI(\mathbf{x}) = (y_{\min} - \hat{f}(\mathbf{x}))\Phi_{\mathcal{N}(0,1)}\left(\frac{y_{\min} - \hat{f}(\mathbf{x})}{\hat{s}_f(\mathbf{x})}\right) + \hat{s}_f(\mathbf{x})\phi_{\mathcal{N}(0,1)}\left(\frac{y_{\min} - \hat{f}(\mathbf{x})}{\hat{s}_f(\mathbf{x})}\right) \tag{3.20}$$

where $\phi_{\mathcal{N}(0,1)}(\cdot)$ and $\Phi_{\mathcal{N}(0,1)}(\cdot)$ are respectively the Probability Density Function (PDF) and the Cumulative Distribution Function (CDF) of the standard univariate Gaussian probability distribution. Two important terms constitute the EI formula. The first part of the sum is the probability of improvement $\mathbb{P}(\mathcal{I}(\mathbf{x}) \geq 0)$ multiplied by a factor $(y_{\min} - \hat{f}(\mathbf{x}))$ that scales the EI value on the supposed improvement value. The second part of the sum takes into account the uncertainty. It tends to be large when the uncertainty on the prediction is high. So, the EI is large for regions of improvement (exploitation) and also for regions of high uncertainty (exploration) as illustrated in Fig. 3.7. The maximization of the EI can be performed using multi-start gradient-based optimization algorithms, Monte-Carlo simulations or evolutionary algorithms [Frazier, 2018]. Fig. 3.8 shows the different added points to the data-set along the iterations of BO using the EI. Other infill criteria have been developped such as the Watson and Barnes 2nd (WB2) [Watson and Barnes, 1995] which shifts the EI with the GP mean prediction, hence, avoiding the large areas of the design space where the EI is null. The scaled WB2 (WB2S) [Bartoli et al., 2019] scales the EI with a factor to better handle the influence between the EI and the GP mean prediction in the infill. The EI has also been adapted to handle multiple points through the q-EI criterion [Ginsbourger et al., 2010; Chevalier et al., 2014], it allows parallel evaluation and determination of multiple added points at each iteration. Thompson sampling has been used as an acquisition function [Basu and Ghosh, 2017]. It consists in drawing a sample from the posterior distribution and choosing the index of the minimum of this sample as an infill candidate. Other methods can also be mentioned as confidence bound criteria [Cox and John, 1997] or information theory based infill criteria [Hernández-Lobato et al., 2014]. Recently, portfolio methods combining these different infill criteria have been developed [Hoffman et al., 2011; Shahriari et al., 2014]. This large variety of methods shows that there is no single infill criterion that performs better over all problem instances [Picheny et al., 2013].

To handle constraints in BO, different techniques have been proposed [Sasena, 2002; Parr et al., 2012]. The direct method [Sasena et al., 2001] consists in optimizing the infill criterion under the posterior mean prediction of the constraints:

$$\left|\begin{array}{lll} \underset{\mathbf{x}}{\text{Maximize}} & EI(\mathbf{x}) & \\ \text{subject to} & \hat{g}_i(\mathbf{x}) & \leq 0,\ i=1,\ldots,n_c \end{array}\right. \tag{3.21}$$

The drawback of this approach is that it does not take into account the uncertainty of the constraint models. The Expected Violation (EV) strategy [Audet et al., 2000] considers the optimization of the infill criterion under the constraint of an expected violation inferior to a threshold:

$$\left|\begin{array}{lll} \underset{\mathbf{x}}{\text{Maximize}} & EI(\mathbf{x}) & \\ \text{subject to} & EV_i(\mathbf{x}) & \leq t_i, i=1,\ldots,n_c \end{array}\right. \tag{3.22}$$

where $EV_i(\cdot)$ is the expected value of the violation of constraint i and t_i is a given threshold. In the Gaussian prediction case $EV_i(\cdot)$ takes a similar form to that of the EI (can be seen as the EI of $-g_i(\cdot)$ for a min $=0$):

$$EV_i(\mathbf{x}) = (0-\hat{g}_i(\mathbf{x}))\Phi_{\mathcal{N}(0,1)}\left(\frac{0-\hat{g}_i(\mathbf{x})}{\hat{s}_{g_i}(\mathbf{x})}\right) + \hat{s}_{g_i}(\mathbf{x})\phi_{\mathcal{N}(0,1)}\left(\frac{0-\hat{g}_i(\mathbf{x})}{\hat{s}_{g_i}(\mathbf{x})}\right) \tag{3.23}$$

The optimization of infill criteria under constraints restrains the choice of optimizers. The Probability of Feasibility (PoF) method [Schonlau et al., 1996] instead of considering the optimization of an infill criterion subject to constraints, optimizes freely the product of the infill criterion with the probability of feasibility of the constraints:

$$\underset{\mathbf{x}}{\text{Maximize}}\ EI(\mathbf{x}) \times PoF(\mathbf{x}) \tag{3.24}$$

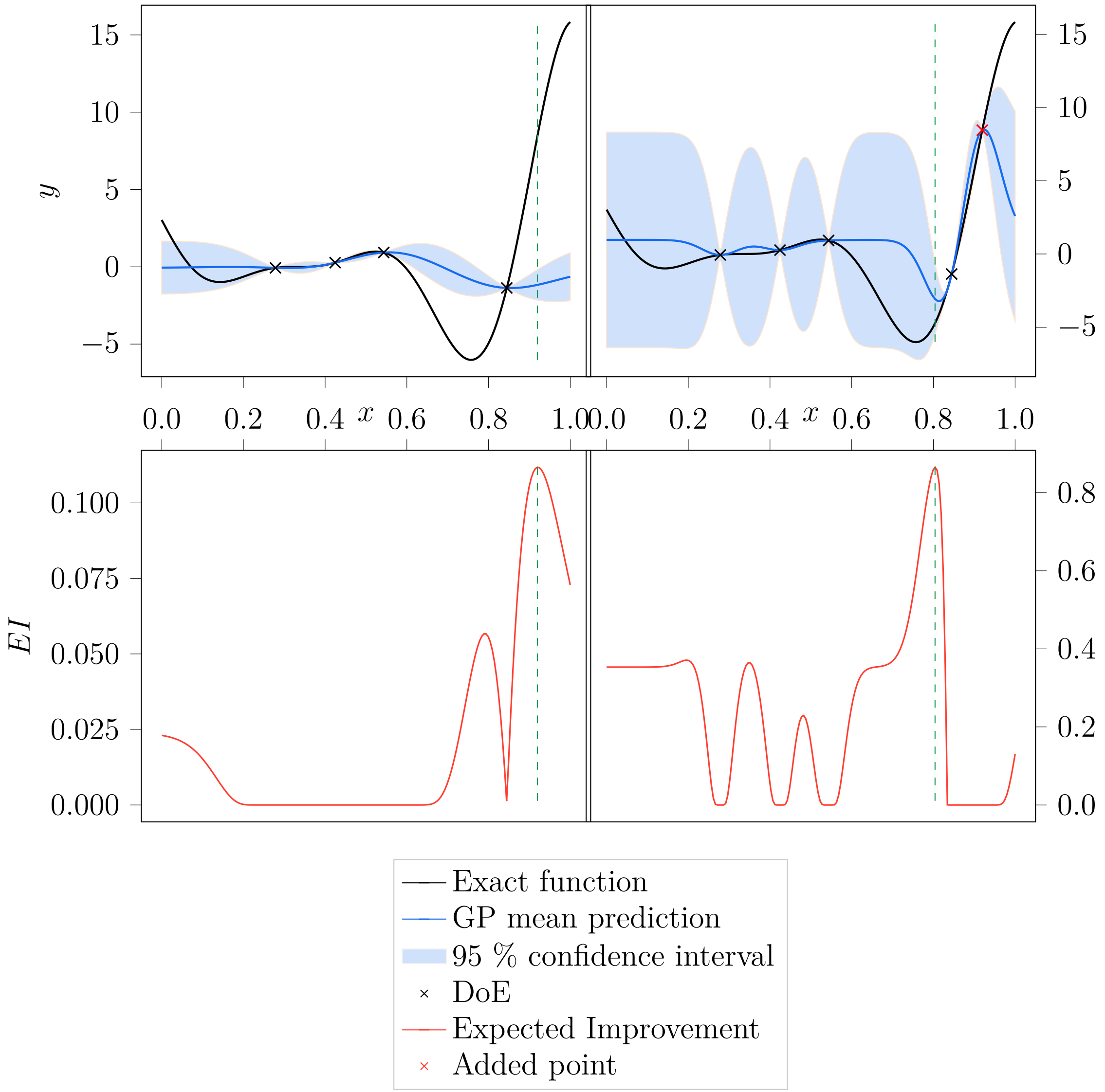


Fig. 3.7 (left) First iteration of BO for an initial DoE of 4 points. A GP model is fitted and the expected improvement criterion is maximized. (right) Second iteration of BO after adding the point that maximizes the expected improvement at the previous iteration to the DoE.

with

$$PoF(\mathbf{x}) = \prod_{i=1}^{n_c} \mathbb{P}\left(g_i(\mathbf{x}) < 0\right) \tag{3.25}$$

For multiple constraints the PoF can quickly collapse to zero making the optimization task difficult. The augmented Lagrangian approach has also been proposed in the BO context to transform the constrained problem to an unconstrained one [Picheny et al.,

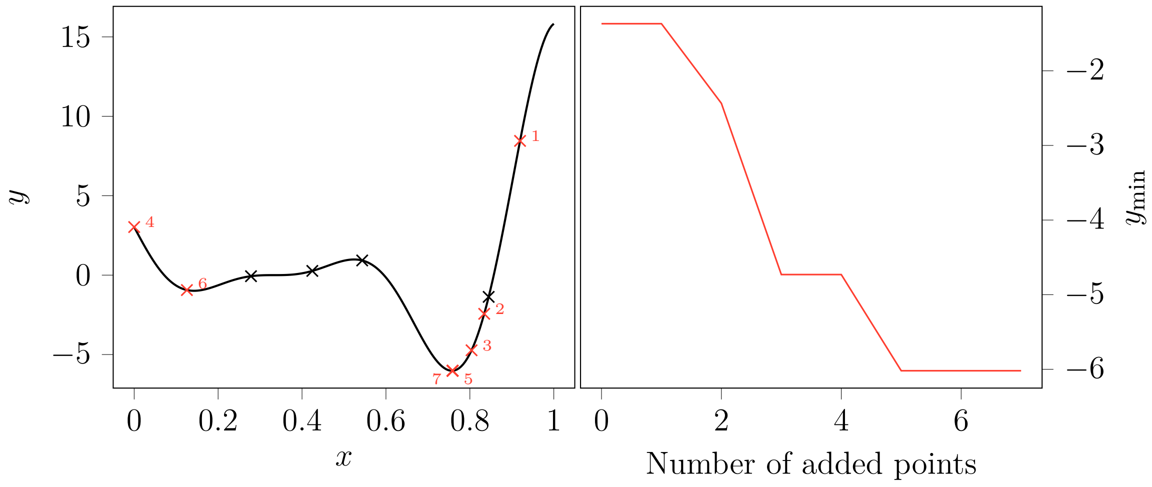


Fig. 3.8 (left) The different points added during BO until convergence. (right) The convergence graph showing the evolution of $y_{\min}$ with respect to the number of added points during BO iterations.

2016]. Other approaches based on feasibility probabilities and upper trust bounds can be used for constrained BO [Priem et al., 2019].

3.2.3 Bayesian Optimization with non-stationary GPs

To handle non-stationarity in BO, the approaches presented in Section 3.1 can be used within the BO framework. However, the direct formulation of non-stationary kernels is challenging to use in high-dimensional spaces as described previously. For the local stationary covariance approaches, [Bartoli et al., 2017] used a similar mixture of experts approach to the one developed in [Bettebghor et al., 2011]. Gaussian processes with partial least squares method [Bouhlel et al., 2016] are used as experts to allow a better modeling in high-dimensional spaces. However, the mixture of experts may not be adapted to the scarce data context due to the use of a subset of data for each expert.

The non-linear mapping approaches for non-stationary GPs (Section 3.1.3) are well adapted to scarce data and high-dimensional problems [Toal and Keane, 2012; Snoek et al., 2014] which make them interesting to use for BO instead of regular GPs in the case of non-stationary problems. This coupling has been studied by [Toal and Keane, 2012] using Xiong non-linear mapping [Xiong et al., 2007]. This allowed the authors to set up a new approach mixing regular GP with non-linear mapping when dealing with BO called Adaptive Partial Non-Stationary (APNS). [Snoek et al., 2014] uses the cumulative distribution of the Beta distribution as a mapping for input warping GPs in a BO framework and shows improved results compared to BO with regular GPs.

These approaches are used as a reference in the experimentations for the proposed BO framework with deep Gaussian processes in Chapter 4, which can be seen as a non-parameterized Bayesian generalization to the non-linear mapping. For now, only one objective has been considered, however, in different optimization problems, multiple antagonistic objectives can be formulated. The specificities of multi-objective optimization are introduced in the next section as well as multi-objective Bayesian optimization.

3.3 Multi-objective Bayesian optimization

Engineering design optimization problems can ideally be modeled as multi-objective and multi-disciplinary optimization problems. For instance, different conflicting objectives need to be considered for aerospace vehicle design such as the payload mass, the gross lift-off weight, the availability, or the production cost. In [Castellini et al., 2011; Arias-Montano et al., 2012], a rich taxonomy of the applications of multi-objective optimization in aerospace engineering is presented. These multi-objective problems are characterized by n_o objectives that are optimized under n_c constraints in a d-dimensional design space (minimization is considered without loss of generality):

$$(\mathcal{P}_{CMO})\left|\begin{array}{lcl} \underset{\mathbf{x}}{\text{Minimize}} & \mathbf{y} & = \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_{n_o}(\mathbf{x})] \\ \text{subject to} & g_i(\mathbf{x}) & \leq 0, \; i = 1, \dots, n_c \end{array}\right. \tag{3.26}$$

where $\mathcal{P}_{CMO}$ stands for Constrained Multi-Objective problem, $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{X} \subseteq \mathbb{R}^d$, and $\mathbf{y} = (y_1, \dots, y_{n_o}) \in \mathbb{Y} \subseteq \mathbb{R}^{n_o}$. $\mathbf{y}$ is called the objective vector and $\mathbb{Y}$ the objective space.

In the case of multiple objectives, since the objective evaluation of each input data point is a vector, the DoE objective evaluations are denoted by a matrix $\mathbf{Y}$. The DoE is rewritten in the multi-objective case as follows:

$$(DoE)\left\{\begin{array}{lcll} \mathbf{X} & = & \left[\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\right]^\top & , \quad \mathbf{x}^{(i)} \in \mathbb{X} \subset \mathbb{R}^d, \forall i \in \{1, \dots, n\} \\ \mathbf{Y} & = & \left[\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}\right]^\top & , \quad \mathbf{y}^{(i)} \in \mathbb{Y} \subset \mathbb{R}^{n_o}, \forall i \in \{1, \dots, n\} \\ \mathbf{c}_j & = & \left[c_j^{(1)} = g_j\left(\mathbf{x}^{(1)}\right), \dots, c_j^{(n)} = g_j\left(\mathbf{x}^{(n)}\right)\right]^\top & , \quad \forall j \in \{1, \dots, n_c\} \end{array}\right.$$

One of the most used approaches to solve these problems are Multi-Objective Evolutionary Algorithms (MOEAs) [Deb, 2001]. Among the most popular MOEAs, NSGA-II (Non-dominated Sorting Genetic Algorithm II) [Deb et al., 2000] or SMPSO (Speed-constrained Multi-objective Particle Swarm Optimization) [Nebro et al., 2009] can be cited. The advantage of these algorithms is that the use of a population-based search and diversity mechanisms makes them less prone to be trapped in local minima. Moreover, the use of simple operators for crossover and mutation allows the handling of highly non-linear or non-differentiable functions [Talbi, 2009; Talbi et al., 2012]. However, MOEAs tend to need a consequent number of evaluations to converge to the exact Pareto front. This may make MOEAs not suitable for computationally expensive functions, where the concern is to minimize the number of evaluations. To overcome this issue, Bayesian optimization has been adapted to multi-objective optimization [Beume et al., 2007] by using new infill sampling criteria based on the concept of Pareto-dominance as the Expected hyper-volume Improvement (EHVI) [Emmerich et al., 2006]. In this section, first, some notions about multi-objective optimization are introduced, then, the multi-objective BO based on independent models is described, finally, the case of correlated-objective modeling in multi-objective BO is considered.

3.3.1 Definitions

Pareto dominance

The Pareto dominance is a binary relation between two input design vectors. An input design vector $\mathbf{x}$ is Pareto dominant with respect to another input design vector $\mathbf{x}'$ (noted $\mathbf{x} \prec \mathbf{x}'$) if and only if :

$$\begin{aligned} &\forall i \in \{1,...,n_o\}; f_i(\mathbf{x}) \leq f_i(\mathbf{x}'), \\ &\exists j \in \{1,...,n_o\}; f_j(\mathbf{x}) < f_j(\mathbf{x}') \end{aligned} \tag{3.27}$$

For notation simplicity, this notation is generalized to objective vectors $\mathbf{y}$ and $\mathbf{y}'$ to express the Pareto dominance.

Pareto optimal set

The Pareto optimal set P also called the Pareto front, is defined as the set of the non-dominated objective vectors:

$$P = \{\mathbf{y} \in \mathbb{Y} | \nexists \mathbf{y}\text{'} \in \mathbb{Y} : \mathbf{y}\text{'} \prec \mathbf{y}\} \tag{3.28}$$

The approximation of the Pareto front P' for a set of solutions S is defined as:

$$P' = \{\mathbf{y} \in S | \nexists \mathbf{y'} \in S : \mathbf{y'} \prec \mathbf{y}\} \tag{3.29}$$

Fig 3.9 illustrates the exact Pareto front in a two-objective case as well as the approximated Pareto front for a set of solutions S.

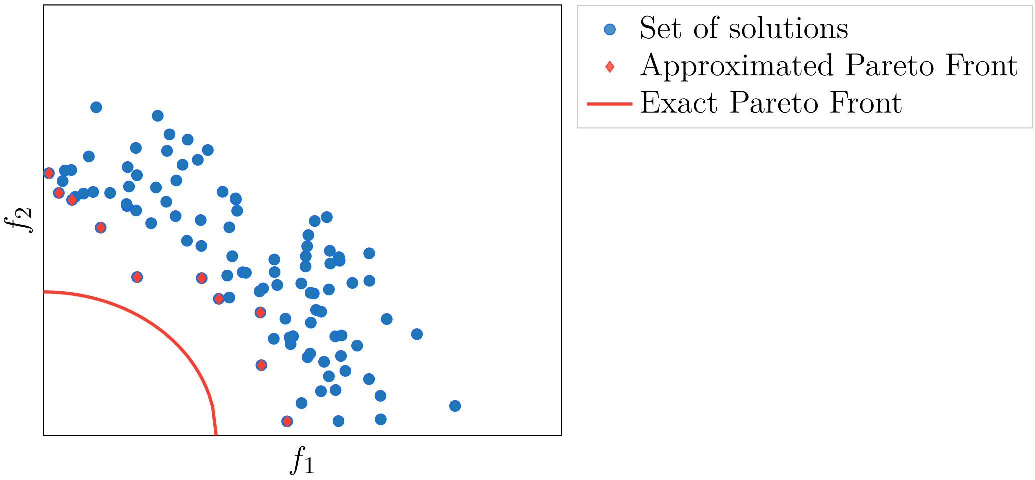


Fig. 3.9 Illustration of the Pareto front and the approximated Pareto set in a two-objective space.

Hyper-volume

Consider an unconstrained multi-objective problem and B a finite hyper-volume of the objective space where all possible solutions lie. $B = \left\{\mathbf{y} \in \mathbb{R}^{n_o}; \mathbf{y}^{\text{lb}} \leq \mathbf{y} \leq \mathbf{y}^{\text{up}}\right\}$ where $\mathbf{y}^{\text{lb}}$ and $\mathbf{y}^{\text{up}}$ are chosen lower and upper bounds respectively (*e.g.*, the ideal and nadir points). The dominated hyper-volume of the DoE is defined as follows:

$$H_{\mathbf{Y}} = \left\{\mathbf{y} \in B; \exists i \in \{1,\ldots,n\}, \mathbf{y}^{(i)} \prec \mathbf{y}\right\} \tag{3.30}$$

So $H_{\mathbf{Y}}$ is the subset of B whose points are dominated by the DoE (Fig. 3.10).

Let $\mathbf{x}^{(n+1)}$ be a new data-point added to the DoE and $\mathbf{Y}^{\text{new}} = \left[\mathbf{Y}, \mathbf{y}^{(n+1)}\right]^\top$ the DoE evaluation matrix plus the objective evaluation of the new data-point. Since $H_{\mathbf{Y}} \subset H_{\mathbf{Y}^{\text{new}}}$, the hyper-volume improvement by adding $\mathbf{x}^{(n+1)}$ to the DoE is given by: $\mathcal{I}_{\mathbf{Y}}(\mathbf{x}_{n+1}) = |H_{\mathbf{Y}^{\text{new}}} \setminus H_{\mathbf{Y}}|$ where $|\cdot|$ is the standard Lebesgue measure. The hyper-volume indicator is widely used as a quantitative measure of the quality of an approximated Pareto front [Zitzler et al., 2002; Bradstreet, 2011; Auger et al., 2012].

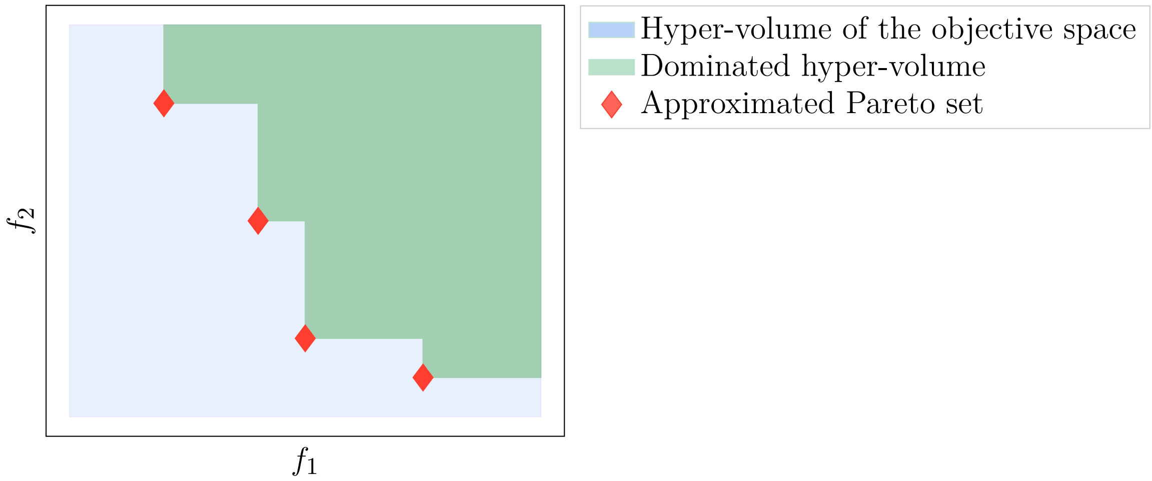


Fig. 3.10 Illustration of the dominated hyper-volume by an approximated Pareto front

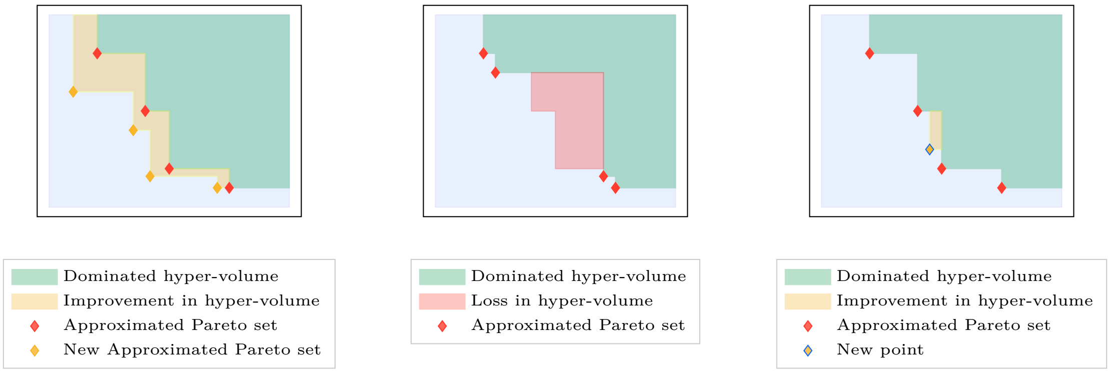
In fact, the hyper-volume takes into account the three characteristics that express the quality of an approximated Pareto front [Zitzler et al., 2002]:

- The distance from the exact Pareto front, the nearer it is, the better the solutions are.
- The diversity of the solutions in the front. The solutions must cover a large zone in the objective space, and not be located in some restricted area.
- The number of solutions. More solutions give more trade-offs, thus, more liberty for the decision maker.

The closer the approximated Pareto set is to the exact one, the larger is the improvement in hyper-volume. Moreover, the more diverse the population is, or the more points in the approximated Pareto set are, the larger the improvement is (Fig. 3.11). This shows that the hyper-volume indicator is suited to compare between approximated Pareto fronts.

3.3.2 Multi-Objective Bayesian Optimization with independent models

Bayesian optimization has been extended to solve multi-objective problems [Emmerich et al., 2006]. A variety of approaches has been proposed for Multi-Objective Bayesian Optimization (MO-BO) which can be classified into the aggregation-based methods (using BO on a weighted sum of objective functions) [Knowles, 2006; Zhang et al., 2010]



🟩 Dominated hyper-volume
🟨 Improvement in hyper-volume
♦ Approximated Pareto set
♦ New Approximated Pareto set
🟩 Dominated hyper-volume
🟥 Loss in hyper-volume
♦ Approximated Pareto set
🟩 Dominated hyper-volume
🟨 Improvement in hyper-volume
♦ Approximated Pareto set
♦ New point

Fig. 3.11 The hyper-volume indicator expresses the three quality characteristics of an approximated Pareto front. (left) Improvement of the hyper-volume by approaching the exact Pareto front. (middle) Loss in hyper-volume by using a non-diversified approximated Pareto front. (right) Improvement of the hyper-volume by adding a point to the approximated Pareto front.

and the dominance-based approaches (using new infill sampling criteria based on the concept of Pareto-dominance) [Emmerich et al., 2006; Svenson and Santner, 2016]. In this section, the second class of approaches is presented. It follows the same structure as single-objective BO, with the difference that for each objective an independent surrogate model is created and an infill sampling criterion based on the concept of Pareto-dominance such as the expected hyper-volume improvement [Emmerich et al., 2006] is used.

Definition of the Expected Hyper-Volume Improvement

The Expected Hyper-Volume Improvement (EHVI) was first introduced by [Emmerich et al., 2006]. Instead of using exact objective evaluations to assess the improvement in hyper-volume of a candidate, which is computationally expensive, the GP posterior of the objectives are used. Since GP predictions are random variables, an expectation value of the hyper-volume improvement is computed. Specifically, for an input data-point, $\mathbf{x}$, the EHVI is the expected value of hyper-volume improvement by adding this point to the data-set:

$$
\begin{aligned}
EHVI(\mathbf{x}) &= \mathbb{E}_{p(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X})}\left[\mathcal{I}_{\mathbf{Y}}(\mathbf{x})\right] \\
&= \mathbb{E}_{p(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X})}\left[\left|H_{[\mathbf{Y},\mathbf{f}(\mathbf{x})]^\top} \setminus H_{\mathbf{Y}}\right|\right] \\
&= \int_{B \setminus H_{\mathbf{Y}}} \left[\left|H_{[\mathbf{Y},\mathbf{v}]^\top} \setminus H_{\mathbf{Y}}\right|\right] p(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X})\mathrm{d}\mathbf{f}(\mathbf{x})
\end{aligned}
\tag{3.31}
$$

The final equation comes from the fact that the integrand is non-null only in the non-dominated region by the DoE *i.e.* $B \setminus H_{\mathbf{Y}}$. The EHVI is optimized to find the most promising candidate to improve the actual dominated hyper-volume. In the constrained case, the EHVI has to be coupled to constrained infill criteria such as the Probability of Feasibility or the Expected Violation in the same way as the Expected Improvement (Section 3.2.2).

Computation of the EHVI

Several methods [Emmerich and Klinkenberg, 2008; Bader and Zitzler, 2011; Yang et al., 2019] have been proposed to compute the EHVI, however, the computational complexity increases exponentially with the number of objectives. In this thesis, without loss of generality, the number of objectives is restrained to two objectives.

To compute the EHVI in the two-objective case, the approach developed in [Emmerich et al., 2016] is followed. For that purpose, the DoE approximated Pareto front is considered as sorted in a decreasing order of the first objective $\mathbf{y}_1$. The DoE objective evaluations of the approximated Pareto front are augmented by $\mathbf{y}^{(0)} = (y_1^{\text{up}}, y_2^{\text{lb}})$ and $\mathbf{y}^{(n_p+1)} = (y_1^{\text{lb}}, y_2^{\text{up}})$, where n_p is the number of approximated Pareto front solutions. Then, the non-dominated space by the DoE $B \setminus H_{\mathbf{Y}}$ is divided into n_p+1 disjoint rectangles $R^{(i)} = \left[\left(y_1^{(i)}, y_2^{\text{lb}}\right), \left(y_1^{(i-1)}, y_2^{(i)}\right)\right]$, $i \in 1, \ldots, n_p+1$ (Fig. 3.12). Therefore, the improvement can be expressed as follows:

$$
\begin{aligned}
\mathcal{I}(\mathbf{x}^*) &= \left| H_{[\mathbf{Y},\mathbf{f}(\mathbf{x})]^\top} \setminus H_{\mathbf{Y}} \right| \\
&= \left| H_{[\mathbf{Y},\mathbf{f}(\mathbf{x})]^\top} \bigcap \left(\bigcup_{i=1}^{n_p+1} R^{(i)} \right) \right| \\
&= \sum_{i=1}^{n_p+1} \left| H_{[\mathbf{Y},\mathbf{f}(\mathbf{x})]^\top} \bigcap R^{(i)} \right|
\end{aligned}
\tag{3.32}
$$

Then, injecting this improvement expression into Eq. (3.31) yields to:

$$
\begin{aligned}
EHVI(\mathbf{x}) &= \int_{B \setminus H_{\mathbf{Y}}} \sum_{i=1}^{n_p+1} \left| H_{[\mathbf{Y},\mathbf{f}(\mathbf{x})]^\top} \bigcap R^{(i)} \right| p(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}) \mathrm{d}\mathbf{f}(\mathbf{x}) \\
&= \sum_{i=1}^{n_p+1} \int_{y^{\text{lb}_1}}^{y_1^{(i-1)}} \int_{y^{\text{lb}_2}}^{y_2^{(i)}} \left| H_{[\mathbf{Y},\mathbf{f}(\mathbf{x})]^\top} \bigcap R^{(i)} \right| p(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}) \mathrm{d}\mathbf{f}(\mathbf{x})
\end{aligned}
\tag{3.33}
$$

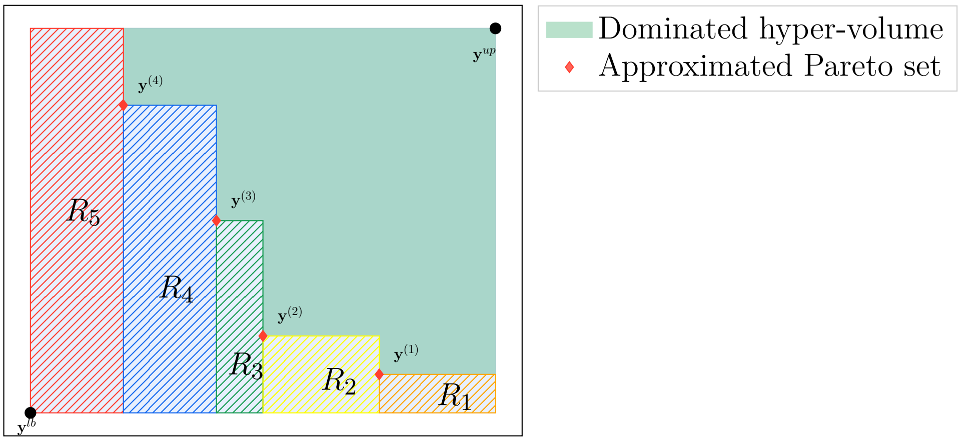


Fig. 3.12 Illustration of the partition of the non-dominated hyper-volume into disjoint rectangles.

where the second equality comes from the fact that $H_{[\mathbf{Y},\mathbf{f}(\mathbf{x})]^\top} \bigcap R^{(i)}$ is non-empty only if $\mathbf{f}(\mathbf{x})$ dominates the upper right corner of $R^{(i)}$ that is $\left(y_1^{(i-1)}, y_2^{(i)}\right)$. In the presented MO-BO framework, an independent GP is used for each objective function, hence, $p(\mathbf{f}(\mathbf{x})|\mathbf{x},\mathbf{y},\mathbf{X}) = p(f_1(\mathbf{x})|\mathbf{y}_1,\mathbf{x},\mathbf{X})p(f_2(\mathbf{x})|\mathbf{y}_2,\mathbf{x},\mathbf{X})$. Based on this independency assumption, the following is obtained:

$$
\begin{aligned}
EHVI(\mathbf{x}) = & \sum_{i=1}^{n_p+1} \int_{y^{\mathrm{lb}_1}}^{y_1^{(i)}} \int_{y_2^{\mathrm{lb}}}^{y_2^{(i)}} \left(y_1^{(i-1)} - y_1^{(i)}\right) p(f_1(\mathbf{x})|\mathbf{y}_1,\mathbf{x},\mathbf{X}) \\
& \left(y_2^{(i)} - f_2(\mathbf{x})\right) p(f_2(\mathbf{x})|\mathbf{y}_2,\mathbf{x},\mathbf{X}) \mathrm{d}f_1(\mathbf{x}) \mathrm{d}f_2(\mathbf{x}) + \\
& \sum_{i=1}^{n_p+1} \int_{y_1^{(i)}}^{y_1^{(i-1)}} \int_{y_2^{\mathrm{lb}}}^{y_2^{(i)}} \left(y_1^{(i-1)} - f_1(\mathbf{x})\right) p(f_1(\mathbf{x})|\mathbf{y}_1,\mathbf{x},\mathbf{X}) \\
& \left(y_2^{(i)} - f_2(\mathbf{x})\right) p(f_2(\mathbf{x})|\mathbf{y}_2,\mathbf{x},\mathbf{X}) \mathrm{d}f_1(\mathbf{x}) \mathrm{d}f_2(\mathbf{x}) \\
= & \sum_{i=1}^{n_p+1} \left(y_1^{(i-1)} - y_1^{(i)}\right) \left(\Phi_{\mathcal{N}(0,1)} \left(\frac{y_1^{(i)} - \hat{f}_1(\mathbf{x})}{\hat{s}_{f_1}(\mathbf{x})}\right) - \Phi_{\mathcal{N}(0,1)} \left(\frac{y_1^{\mathrm{lb}} - \hat{f}_1(\mathbf{x})}{\hat{s}_{f_1}(\mathbf{x})}\right)\right) \\
& \left(\xi(y_2^{(i)}, y_2^{(i)}, \hat{f}_2(\mathbf{x}), \hat{s}_{f_2}(\mathbf{x})) - \xi(y_2^{(i)}, y_2^{\mathrm{lb}}, \hat{f}_2(\mathbf{x}), \hat{s}_{f_2}(\mathbf{x}))\right) \\
& + \sum_{i=1}^{n_p+1} \left(\xi(y_1^{(i-1)}, y_1^{(i-1)}, \hat{f}_2(\mathbf{x}), \hat{s}_{f_2}(\mathbf{x})) - \xi(y_1^{(i-1)}, y_1^{(i)}, \hat{f}_1(\mathbf{x}), \hat{s}_{f_1}(\mathbf{x}))\right) \\
& \left(\xi(y_2^{(i)}, y_2^{(i)}, \hat{f}_2(\mathbf{x}), \hat{s}_{f_2}(\mathbf{x})) - \xi(y_2^{(i)}, y_2^{\mathrm{lb}}, \hat{f}_2(\mathbf{x}), \hat{s}_{f_2}(\mathbf{x}))\right)
\end{aligned}
\tag{3.34}
$$

where the first equality comes from the property of additivity of integration on intervals: $\int_{y^{\mathrm{lb}_1}}^{y_1^{(i-1)}} = \int_{y^{\mathrm{lb}_1}}^{y_1^{(i)}} + \int_{y_1^{(i)}}^{y_1^{(i-1)}}$ and the second equality comes from the computation of the following integral:

$$\begin{aligned} \xi(a,b,\mu,\sigma) &= \int_{-\infty}^{b} (a - f_i(\mathbf{x}))\frac{1}{\sigma}\phi_{\mathcal{N}(0,1)}\left(\frac{f_i(\mathbf{x})-\mu}{\sigma}\right)\mathrm{d}f_i(\mathbf{x}) \\ &= \sigma\phi_{\mathcal{N}(0,1)}\left(\frac{b-\mu}{\sigma}\right) + (a-\mu)\Phi_{\mathcal{N}(0,1)}\left(\frac{b-\mu}{\sigma}\right) \end{aligned} \tag{3.35}$$

Therefore, the EHVI is fully analytical in the case of two objectives. This derivation of the EHVI is based on the assumption of independent models for each objective. However, in multi-objective problems, the objectives are often negatively correlated. Considering each objective independently may be sub-optimal [Shah et al., 2015].

3.3.3 Multi-objective Bayesian Optimization taking into account correlation between objectives

Instead of modeling each objective using independent GPs, [Shah and Ghahramani, 2016] proposed to use a correlated GP for the different objectives. For that, a linear model of coregionalization is considered [Alvarez et al., 2011]. Specifically, a multi-output kernel function $\mathbf{K}(\cdot,\cdot)$ is defined as the following combination of m_{lmc} kernels $\{k_i(\cdot,\cdot)\}_{i=1}^{m_{\mathrm{lmc}}}$:

$$\mathbf{K}(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{m_{\mathrm{lmc}}} \mathbf{B}_i k_i(\mathbf{x},\mathbf{x}') \tag{3.36}$$

$\{\mathbf{B}_i\}_{i=1}^{m_{\mathrm{lmc}}}$ are $\mathbb{R}^{n_o\times n_o}$ matrices called coregionalization matrices. The coregionalization matrices encode the correlation between the outputs such as $cov(f_i(\mathbf{x}), f_j(\mathbf{x}')) = \mathbf{K}(\mathbf{x},\mathbf{x}')_{i,j}$ while the kernels express the correlation in the input space. [Shah and Ghahramani, 2016] use $m_{\mathrm{lmc}} = n_o$ for the number of kernels and coregionalization matrices. This allows a model of the objective functions which takes into account the correlations between the different objectives. More details on the linear model of coregionalization are presented in Section 3.4.1.

However, with a correlated objective model, the assumption of independency used to compute the EHVI in Eq. (3.34) does not hold. This is due to the analytical intractability of the following integrals:

$$\begin{aligned}
\varpi_1(\mathbf{x}) &= \int_{y_1^{\mathrm{lb}}}^{y_1^{(i-1)}} \int_{y_2^{\mathrm{lb}}}^{y_2^{(i)}} \left(y_1^{(i-1)} - y_1^{(i)}\right) \left(y_2^{(i)} - f_2(\mathbf{x})\right) p\left(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) \\
\varpi_2(\mathbf{x}) &= \int_{y^{(i)}}^{y_1^{(i-1)}} \int_{y_2=y^{\mathrm{lb}_2}}^{y_2^{(i)}} \left(y_1^{(i)} - f_1(\mathbf{x})\right) \left(y_2^{(i)} - f_2(\mathbf{x})\right) p\left(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}\right) \mathrm{d}\mathbf{f}(\mathbf{x})
\end{aligned} \tag{3.37}$$

To overcome this issue, in [Shah and Ghahramani, 2016], first, the bounds of the integrals are transformed to $\mathbb{R}^2$ by introducing the indicator function $\mathbb{I}[\cdot]$:

$$\begin{aligned}
\varpi_1(x) = &\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(y_1^{(i-1)} - y_1^{(i)}\right) \mathbb{I}\left[y_1^{(i-1)} \leq y_1^{(i)}\right] \left(y_2^{(i)} - f_2(\mathbf{x})\right) \\
&\mathbb{I}\left[y_2^{\mathrm{lb}} \leq f_2(\mathbf{x}) \leq y_2^{(i)}\right] p\left(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) \\
\varpi_2(x) = &\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(y_1^{(i)} - f_1(\mathbf{x})\right) \mathbb{I}\left[y^{(i)} \leq f_1(\mathbf{x}) \leq y_1^{(i-1)}\right] \left(y_2^{(i)} - f_2(\mathbf{x})\right) \\
&\mathbb{I}\left[y_2^{\mathrm{lb}} \leq f_2(\mathbf{x}) \leq y_2^{(i)}\right] p\left(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}\right) \mathrm{d}\mathbf{f}(\mathbf{x})
\end{aligned} \tag{3.38}$$

Then, the piece-wise linear functions $(y_1^{(i)} - f_1(\mathbf{x}))\mathbb{I}\left[y^{(i)} \leq f_1(\mathbf{x}) \leq y_1^{(i-1)}\right]$, $(y_2^{(i)} - f_2(\mathbf{x}))\mathbb{I}\left[y_2^{\mathrm{lb}} \leq f_2(\mathbf{x}) \leq y_2^{(i)}\right]$ and the constant piece-wise function $(y_1^{(i-1)} - y_1^{(i)})\mathbb{I}\left[y_1^{(i-1)} \leq y_1^{(i)}\right]$ are approximated by scaled Gaussian densities using moment matching. The result from this approximation is that the integrands in Eq.(3.37) come back to a product of multivariate Gaussians since the prediction of the model is Gaussian. Therefore, the integrand is approximated by a scaled multivariate Gaussian (See Eq. (B.6), Appendix B). Hence, the integral over $\mathbb{R}^2$ using this approximation is equal to the product of the scaling constants.

In Chapter 5, the quality of this approximation of the correlated EHVI is numerically studied on a benchmark of representative functions. Moreover, instead of using a linear coregionalization model which can exhibit only linear correlations between the objectives, a novel multi-objective model is proposed based on deep Gaussian processes.

3.4 Multi-Fidelity with Gaussian Processes

In design engineering problems, as described in the previous sections of this chapter, the exact evaluation of a quantity of interest, also called High-Fidelity (HF) evaluation, relies on computationally intensive simulation codes which limit the size of the available data-set. The analysis of complex systems using uncertainty propagation, sensitivity analysis or optimization requires repeated model evaluations at different locations in the design space which typically cannot be afforded with HF physical models. Moreover, using a surrogate model based only on the high-fidelity data can result on a poor prediction of the model because of the few available evaluations. Multi-fidelity approaches [Fernández-Godino et al., 2016; Peherstorfer et al., 2018] are used to overcome this issue by enhancing the high-fidelity data with Low-Fidelity (LF) physical model evaluations that are computationally cheaper to obtain but are less accurate. In fact, unlike in the previous sections of this chapter where only a HF physical model is considered, in multi-fidelity, there are different sources of information about the same response of interest but with different degrees of accuracy and computational cost (fidelities). Multi-fidelity approaches consist in managing these different levels of fidelity in order to achieve a trade-off between computational cost and prediction accuracy. Building other sources of information less costly of the HF physical model can be accomplished by three main modeling approaches [Fernández-Godino et al., 2016]:

- numerical relaxation, for instance, in a simulation code that requires an optimization sub-problem to be solved, a low number of iterations in the optimization process is chosen for the low-fidelity model. In [Jonsson et al., 2015], for the shape optimization of trawl-doors a low-fidelity CFD model similar to the high-fidelity CFD model is used but with a relaxed flow solver convergence criteria, this results in a LF model 78 times faster than the HF model.

- different assumptions about the physical model by neglecting some physical effects. For instance, in [Iyappan and Ganguli, 2020] a Euler-Bernoulli beam finite element model [Reddy, 1993] is considered as the low-fidelity model to compute the load-carrying and deflection characteristics of a short beam. The effects of rotary inertia and shear deformation are neglected in this model and the cross-section remains perpendicular to the bending axis. While in the high-fidelity, the Timoshenko beam theory [Reddy, 1993] is used, which takes into account the effects of rotary inertia and shear deformation and the cross-section has no longer to be perpendicular to the bending axis for short and small beams.

- different levels of space or time discretization. For example, in [Brooks et al., 2011], for the aerodynamic shape optimization of a transonic compressor rotor, in the low-fidelity model a coarse mesh refinement is used to solve the Reynolds-

Averaged steady Navier-Stokes (240000 nodes), while in the high-fidelity model, a fine mesh grid is used (740000 nodes).

Multi-fidelity modeling is a popular research topic both in the engineering and machine learning communities. In fact, different models have been developed based on Gaussian processes [Kennedy and O'Hagan, 2001; Le Gratiet and Garnier, 2014; Raissi and Karniadakis, 2016; Perdikaris et al., 2017; Cutajar et al., 2019], artificial neural networks [Kim et al., 2007; Minisci and Vasile, 2013] or support vector machines [Shi et al., 2019] and applied to a broad spectrum of engineering applications including aerodynamics [Kuya et al., 2011; Shah et al., 2015], electronics [Bekasiewicz and Koziel, 2015], thermodynamics [Reeve and Strachan, 2017], or mechanics [Vitali et al., 2002]. However, a rarely investigated case is when the input space definition is different in each fidelity. In fact, in practice for the sake of simplicity, the LF model may not consider some input variables or use a different modeling parameterization than the HF model. In this section, a review of literature on the different multi-fidelity models based on Gaussian processes is provided (Section 3.4.1) as well as a review on the methods used to handle the case where different input space parameterizations are considered for each fidelity (Section 3.4.2).

3.4.1 Multi-fidelity with identical input spaces

Due to their attractive features, GPs have been extended to multi-fidelity modeling which resulted on popular multi-fidelity models based on GPs. In engineering design field, linear models such as the Linear Model of Coregionalization (LMC) [Alvarez et al., 2011] or the Auto-Regressive (AR1) model [Kennedy and O'Hagan, 2000] are usually used [Laurenceau and Sagaut, 2008; Kuya et al., 2011; Toal and Keane, 2011; Keane, 2012; Toal et al., 2014; Fernández-Godino et al., 2016; Bailly and Bailly, 2019]. These approaches are presented as well as other approaches developed in the machine learning field [Kennedy and O'Hagan, 2000; Le Gratiet and Garnier, 2014; Perdikaris et al., 2017; Cutajar et al., 2019] to account for more complex dependencies between the available fidelities.

Let $(\mathbf{X}_t, \mathbf{y}_t)$ be the couple of inputs/outputs of each fidelity $t \in \{1, \dots, n_{\text{fi}}\}$, where n_{fi} is the number of fidelities sorted in an increasing order of fidelities *i.e.* $(\mathbf{X}_1, \mathbf{y}_1)$ corresponds to the lowest fidelity data-set and $(\mathbf{X}_{n_{\text{fi}}}, \mathbf{y}_{n_{\text{fi}}})$ to the highest fidelity data-set. Let d and n_t be respectively the dimension of the input data and the size of the training data at fidelity t.

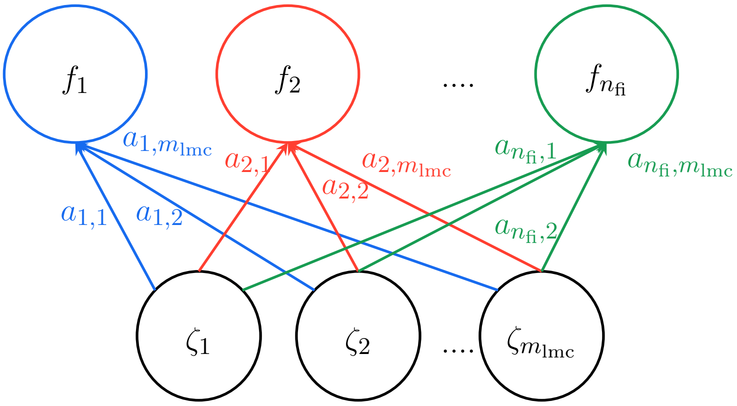


Fig. 3.13 Linear Model of Coregionalization schematic view

Linear Model of Coregionalization (LMC)

Instead of considering a set of n_{fi} independent GPs, one may consider a single multi-output GP of n_{fi} outputs [Alvarez et al., 2011]. Within the context of multi-fidelity, each output n_{fi} of this vector-valued GP corresponds to a fidelity.

For multi-output GPs, the covariance function takes its values in $\mathbb{R}^{n_{\mathrm{fi}} \times n_{\mathrm{fi}}}$ and can be expressed as:

$$\mathbf{K}(\mathbf{x},\mathbf{x}') = \begin{bmatrix} k_{1,1}(\mathbf{x},\mathbf{x}') & \dots & k_{1,n_{\mathrm{fi}}}(\mathbf{x},\mathbf{x}') \\ k_{2,1}(\mathbf{x},\mathbf{x}') & \dots & k_{2,n_{\mathrm{fi}}}(\mathbf{x},\mathbf{x}') \\ \dots\dots & \dots\dots & \dots\dots \\ k_{n_{\mathrm{fi}},1}(\mathbf{x},\mathbf{x}') & \dots & k_{n_{\mathrm{fi}},n_{\mathrm{fi}}}(\mathbf{x},\mathbf{x}') \end{bmatrix} \tag{3.39}$$

where $k_{i,j}(\cdot,\cdot)$ corresponds to the covariance function between the outputs i and j. In LMC, the outputs are expressed as linear combinations of m_{lmc} independent GPs (Figure 3.13):

$$f_t(\mathbf{x}) = \sum_{i=1}^{m_{\mathrm{lmc}}} a_{t,i} \times \zeta_i(\mathbf{x}) \tag{3.40}$$

with $\zeta_i(\cdot)$ a GP of mean zero and covariance matrix $cov(\zeta_i(\mathbf{x}),\zeta_i(\mathbf{x}')) = k_i(\mathbf{x},\mathbf{x}')$. Moreover, $\zeta_i(\cdot)$ and $\zeta_j(\cdot)$ may share the same covariance function $k_i(\mathbf{x},\mathbf{x}')$. It is therefore possible to rewrite Eq.(3.40) by regrouping the GPs with the same covariance function:

$$f_t(\mathbf{x}) = \sum_{i=1}^{m_{\mathrm{lmc}}} \sum_{j=1}^{r_i} a_{t,i}^j \times \zeta_i^j(\mathbf{x}) \tag{3.41}$$

The GP of fidelity t is expressed as a sum of m_{lmc} groups i of r_i independent GPs $\zeta_i^j(\cdot)$ that share the same covariance function $k_i(\mathbf{x},\mathbf{x}')$. Therefore, due to the independence of the GPs $\zeta_i^j(\cdot)$ it is possible to express the covariance function between two outputs $cov\left(f_t(\mathbf{x}), f_{t'}(\mathbf{x}')\right) = k_{t,t'}(\mathbf{x},\mathbf{x}')$ as:

$$k_{t,t'}(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{m_{\text{lmc}}} \sum_{j=1}^{r_i} a_{t,i}^j a_{t',i}^j \times k_i(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{m_{\text{lmc}}} b_{t,t'}^i \times k_i(\mathbf{x},\mathbf{x}') \tag{3.42}$$

with $b_{t,t'}^i = \sum_{j=1}^{r_i} a_{t,i}^j a_{t',i}^j$.

Eventually, the kernel matrix $\mathbf{K}(\mathbf{x},\mathbf{x}')$ may be written:

$$\mathbf{K}(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{m_{\text{lmc}}} \mathbf{B}_i k_i(\mathbf{x},\mathbf{x}') \tag{3.43}$$

with $\mathbf{B}_i$ a coregionalization matrix and its components $b_{t,t'}^i$. The rank of the matrix $\mathbf{B}_i$ is defined by r_i corresponding to the number of independent latent functions that share the same covariance function $k_i(\mathbf{x},\mathbf{x}')$. The inference in LMC follows the same procedure as described in Chapter 2, Section 2.2, with the supplementary consideration of the coregionalization matrix in the expression of the Gram matrix.

A limitation of LMC for multi-fidelity applications is that it considers all the outputs with the same weight, meaning that they provide the same level of information, it is referred to as a symmetrical approach. By treating the outputs equally, symmetric covariance functions are implemented in order to capture the output correlations through the share of useful information across the outputs as much as possible. However, in the multi-fidelity framework, asymmetrical information are available. Indeed, to improve the predictions of the expensive high-fidelity output $f_{n_{\text{fi}}}(\cdot)$ information is transferred from the inexpensive lower fidelity outputs. The multi-fidelity modeling utilizes the correlated inexpensive lower-fidelity information to enhance the expensive high-fidelity modeling. The GP-based approaches presented next account for this asymmetrical information.

Auto-Regressive model (AR1)

The Auto-Regressive (AR1) method is one of the most used approaches for multi-fidelity modeling in engineering design problems [Laurenceau and Sagaut, 2008; Kuya et al., 2011; Toal and Keane, 2011; Keane, 2012; Toal et al., 2014; Fernández-Godino et al., 2016; Bailly and Bailly, 2019]. It relies on a linear autoregressive information fusion scheme introduced by [Kennedy and O'Hagan, 2000], assuming a linear dependency between the different model fidelities.

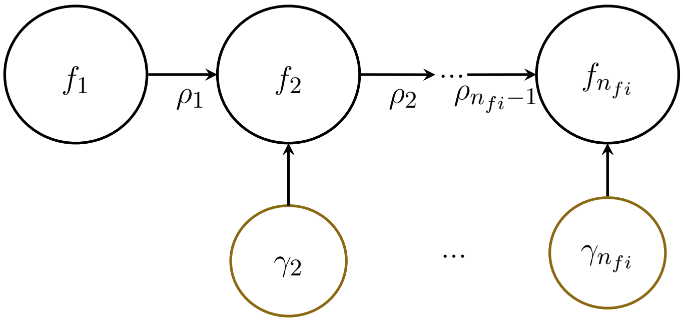


Fig. 3.14 AR1 schematic view

The auto-regressive denomination of this approach comes back from the fact that at each level, the GP $f_t(\cdot)$ is completely determined given the previous fidelity GP $f_{t-1}(\cdot)$. Specifically, AR1 assigns a GP prior to each fidelity model t where the higher-fidelity model prior $f_t(\cdot)$ is equal to the lower-fidelity prior $f_{t-1}(\cdot)$ multiplied by a scaling factor $\rho(\mathbf{x})$ plus an additive bias function $\gamma_t(\cdot)$ (Fig. 3.14):

$$f_t(\mathbf{x}) = \rho_{t-1}(\mathbf{x}) f_{t-1}(\mathbf{x}) + \gamma_t(\mathbf{x}) \tag{3.44}$$

$\rho_{t-1}(\mathbf{x})$ is a scale factor and quantifies the correlation between the fidelities y_t and y_{t-1}, and $\gamma_t(\cdot)$ is a GP with mean μ_{γ_t} and covariance function $k_{\gamma_t}(\cdot,\cdot)$. $\rho_{t-1}(\cdot)$ is often assumed as a constant function [Fernández-Godino et al., 2016], meaning that:

$$f_t(\mathbf{x}) = \rho_{t-1} f_{t-1}(\mathbf{x}) + \gamma_t(\mathbf{x}) \tag{3.45}$$

The relationship of the AR1 model in Eq. (3.44) is derived from the assumption that $cov\left(f_t(\mathbf{x}), f_{t-1}(\mathbf{x}')|f_{t-1}(\mathbf{x})\right) = 0,\ \forall \mathbf{x} \neq \mathbf{x}'$. It means that if $f_{t-1}(\mathbf{x})$ is known, nothing more can be learned for $f_t(\cdot)$ from any simulation of the cheaper code $f_{t-1}(\mathbf{x}')$ for $\forall \mathbf{x}' \neq \mathbf{x}$. Two main alternative numerical schemes exist for AR1 GPs inference: a fully coupled one proposed by [Kennedy and O'Hagan, 2000] and a recursive inference introduced by [Le Gratiet and Garnier, 2014].

[Kennedy and O'Hagan, 2000] derived the posterior distribution of the highest fidelity $f_{n_{\text{fi}}}(\cdot)$ by marginalizing out all the fidelity observations. Based on standard Gaussian identities, a Gaussian posterior distribution is obtained involving the inversion of a covariance matrix of size $\sum_{t=1}^{n_{\text{fi}}} n_t \times \sum_{t=1}^{n_{\text{fi}}} n_t$, hence inducing a computational complexity of $\mathcal{O}\left(\sum_{t=1}^{n_{\mathrm{fi}}} n_t\right)^3$. To reduce this computational complexity, [Le Gratiet and Garnier, 2014] instead of directly conditioning the highest fidelity on all the other fidelities, followed a recursive approach. Specifically, the GP prior $f_{t-1}(\cdot)$ in Eq.(3.44) is replaced by the GP posterior $\tilde{f}_{t-1} = f_{t-1}|\mathbf{y}_{t-1}, \mathbf{X}_{t-1}$ of the previous inference level. This results in n_{fi} standard GP regressions and offers a decoupled inference approach, reducing the training complexity of the model from $\mathcal{O}\left(\sum_{t=1}^{n_{\mathrm{fi}}} n_t\right)^3$ to $\mathcal{O}\left(\sum_{t=1}^{n_{\mathrm{fi}}} (n_t)^3\right)$. Under the assumption of nested DoE structure, meaning that the DoE of higher fidelity is a subset of the DoE of lower fidelity, this inference scheme is equivalent to the fully coupled one proposed by [Kennedy and O'Hagan, 2000].

By doing so, the multi-fidelity GP posterior predictive distribution $p(f_t|\mathbf{y}_t, \mathbf{X}_t, \tilde{f}_{t-1})$ for $t = 1, \dots, n_{\mathrm{fi}}$ for each level t is defined by the standard GP prediction equations given the previous level $t-1$:

$$\hat{f}_t(\mathbf{x}) = \rho_{t-1}\hat{f}_{t-1}(\mathbf{x}) + \mu_{\gamma_t} + \mathbf{k}_{\gamma_t}(\mathbf{x}, \mathbf{X}_t)\mathbf{K}_{\gamma_t}^{-1}(\mathbf{X}_t, \mathbf{X}_t)\left(\mathbf{y}_t - \rho_{t-1}\hat{f}_{t-1}(\mathbf{x}) - \mu_{\gamma_t}\right) \tag{3.46}$$

$$\hat{s}_t^2(\mathbf{x}) = \rho_{t-1}\hat{s}_{t-1}^2(\mathbf{x}) + k_{\gamma_t}(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\gamma_t}(\mathbf{x}, \mathbf{X}_t)\mathbf{K}_{\gamma_t}^{-1}(\mathbf{X}_t, \mathbf{X}_t)\mathbf{k}_{\gamma_t}(\mathbf{X}_t, \mathbf{x}) \tag{3.47}$$

where $\tilde{f}_t(\mathbf{x}) \sim \mathcal{N}(\hat{f}_t(\mathbf{x}), \hat{s}_t^2(\mathbf{x}))$. AR1 has been extended for scalability purpose to account for high dimensional problems (for instance with Proper orthogonal decomposition [Xiao et al., 2018] or Nystrom approximation of sample covariance matrices [Zaytsev and Burnaev, 2017]).

As it can be seen in Eq.(3.44), AR1 only assumes a certain linear relationship between the fidelities. Moreover, AR1 may be seen as a particular case of co-kriging using LMC for a particular value of the coregionalization matrix. This linear mapping between the fidelities may be a limitation for some engineering design problems where this dependence structure is not appropriate. Other approaches have been developed to account for non-linear dependencies between the fidelities.

Non-linear Auto-Regressive multi-fidelity Gaussian Process (NARGP)

In order to generalize the AR1 approach, [Perdikaris et al., 2017] proposed a non-linear mapping between the fidelities called Non-linear Auto-Regressive (NARGP):

$$f_t(\mathbf{x}) = \varrho_{t-1}\left(f_{t-1}(\mathbf{x})\right) + \gamma_t(\mathbf{x}) \tag{3.48}$$

with $\varrho_{t-1}(\cdot)$ a mapping function between two successive fidelity models with an assigned GP prior. As $f_{t-1}(\cdot)$ is a GP, $\varrho_{t-1}\left(f_{t-1}(\cdot)\right)$ is a composition of two GPs which comes back to a deep Gaussian process. NARGP avoids a DGP formulation due to the non-tractability of its exact inference (Chapter 2, Section 2.3). Instead, NARGP follows the same recursive inference strategy proposed for AR1 [Le Gratiet and Garnier, 2014]. It also requires to satisfy the same hypotheses, especially on the nested DoE assumption. In the inference, the GP prior of $f_{t-1}(\cdot)$ is replaced with the GP posterior $\tilde{f}_{t-1}(\cdot)$ obtained with the previous fidelity level. Following this assumption and considering an independence hypothesis between $\gamma_t(\cdot)$ and $\varrho_{t-1}(\cdot)$, NARGP model may be expressed by:

$$f_t(\mathbf{x}) = \varphi_t\left(\left[\mathbf{x}, \tilde{f}_{t-1}(\mathbf{x})\right]\right) \tag{3.49}$$

with $\varphi_t \sim \mathcal{GP}\left(0, k_t\left([\mathbf{x}, \tilde{f}_{t-1}(\mathbf{x})], [\mathbf{x}', \tilde{f}_{t-1}(\mathbf{x}')]\right)\right)$.

The authors proposed a specific covariance function for $\varphi_t(\cdot)$ that reflects the non-linear structure:

$$k_t\left([\mathbf{x}, \tilde{f}_{t-1}(\mathbf{x})], [\mathbf{x}', \tilde{f}_{t-1}(\mathbf{x}')]\right) = k_{\rho_{t-1}}(\mathbf{x}, \mathbf{x}') \times k_{f_{t-1}}\left(\tilde{f}_{t-1}(\mathbf{x}), \tilde{f}_{t-1}(\mathbf{x}')\right) + k_{\gamma_{t-1}}(\mathbf{x}, \mathbf{x}') \tag{3.50}$$

where $k_{\rho_{t-1}}(\cdot,\cdot)$ and $k_{\gamma_{t-1}}(\cdot,\cdot)$ are covariance functions with respectively an input space-dependent scaling effect and an input space-dependent bias effect, whilst $k_{f_{t-1}}(\cdot,\cdot)$ is the covariance function between the evaluated outputs at the previous layer. Hence, NARGP extends the capabilities of AR1 and enables to capture non-linear, non-functional and space-dependent cross-correlations between the low and high-fidelity models [Perdikaris et al., 2017].

NARGP resumes to a disjointed architecture in which a GP for each fidelity is fitted in an isolated hierarchical manner. Therefore, inference in NARGP comes back to inference of n_{fi} GPs in a sequential manner from the lower to the higher fidelity. However, this means that GPs at lower fidelities are not updated once they have been trained given the higher fidelities. To avoid this limitation, [Cutajar et al., 2019] proposed to extend NARGP to a deep Gaussian process by keeping the exact form of Eq (3.48).

Multi-Fidelity Deep Gaussian Process (MF-DGP)

A functional composition of GP priors is obtained by keeping the exact relationship in Eq. (3.48). This functional composition of GPs gives rise to a Deep Gaussian Process (DGP) with n_{fi} layers as described in Chapter 2, Section 2.3. In the classic formulation of DGPs for regression, the intermediate layers are latent and serve as Bayesian non-parametric mappings to capture complex and non-stationary responses (Chapter 4). However, in this formulation of DGPs, the intermediate layers have a physical signification. In fact, each layer corresponds to a fidelity and is connected to

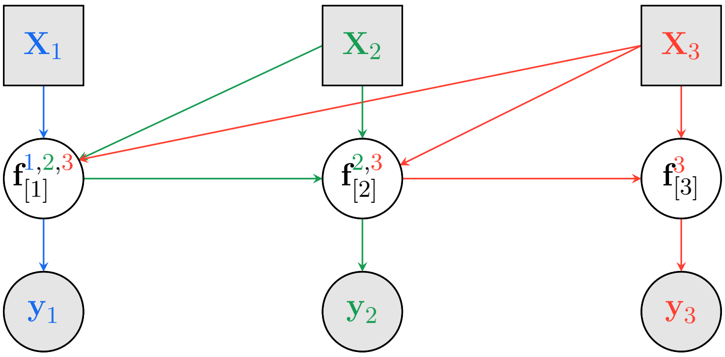


Fig. 3.15 Graphical representation of MF-DGP model for three fidelities (the lower fidelity represented in blue, the medium fidelity in green, and the high fidelity in red). Each layer of the DGP corresponds to a fidelity.

a couple of observed inputs/outputs. Moreover, the GP at each layer depends not only on the input data at this fidelity but also on all the previous fidelity evaluations for the same input data. To this end, $\mathbf{f}^t_{[i]}$ denotes the evaluation at layer i of $\mathbf{X}_t$ the input data at fidelity t (Fig. 3.15). This formulation of DGPs in the context of multi-fidelity is called Multi-Fidelity Deep Gaussian Process (MF-DGP) [Marmin and Filippone, 2018; Cutajar et al., 2019]. MF-DGP like NARGP imposes the definition of a combination of covariance functions at each layer taking into account the correlation between the inputs as well as the correlation between the outputs as expressed in Eq. (3.50)

The MF-DGP inference follows the variational approximation presented in [Salimbeni and Deisenroth, 2017] (see Chapter 2, Section 2.3.2). At each layer, a set of inducing inputs / outputs $(\mathbf{Z}_{[i]}, \mathbf{u}_{[i]})$ are introduced and the following variational approximation is considered:

$$q\left(\{\{\mathbf{f}^t_{[i]}\}_{i=1}^{t}\}_{t=1}^{n_{\mathrm{fi}}}, \{\mathbf{u}_{[i]}\}_{i=1}^{n_{\mathrm{fi}}}\right) = \prod_{t=1}^{n_{\mathrm{fi}}}\prod_{i=1}^{t} p(\mathbf{f}^t_{[i]}|\mathbf{u}_{[i]}, \{\mathbf{X}_t, \mathbf{f}^t_{[i-1]}\}, \mathbf{Z}_{[i-1]}) \times \prod_{i=1}^{n_{\mathrm{fi}}} q(\mathbf{u}_{[i]}) \quad (3.51)$$

where $q(\mathbf{u}_{[i]})$ is the approximated variational distribution of $\mathbf{u}_{[i]}$. Following a classical variational approach (Chapter 2, Section 2.1.2), the variational evidence lower bound (ELBO) is then obtained:

$$\mathcal{L} = \sum_{t=1}^{n_{\mathrm{fi}}}\sum_{i=1}^{n_t} \mathbb{E}_{q(f^{(i),t}_{[t]})}\left[\log p(y^{(i),t}|f^{(i),t}_{[t]})\right] - \sum_{t=1}^{n_{\mathrm{fi}}} \mathbb{KL}\left[q(\mathbf{u}_{[t]})||p(\mathbf{u}_{[t]}|\mathbf{Z}_{[t-1]})\right] \quad (3.52)$$

This bound is optimized with respect to the inducing inputs $\{\mathbf{Z}_{[t]}\}_{t=1}^{n_{\mathrm{fi}}}$, the variational parameters $\{\boldsymbol{\theta}_{q(\mathbf{u}_{[t]})}\}_{t=1}^{n_{\mathrm{fi}}}$, and the GP hyperparameters at each layer $\{\boldsymbol{\theta}_{[t]}\}_{t=1}^{n_{\mathrm{fi}}}$. However, optimizing the variational parameters using ordinary gradient may be not appropriate

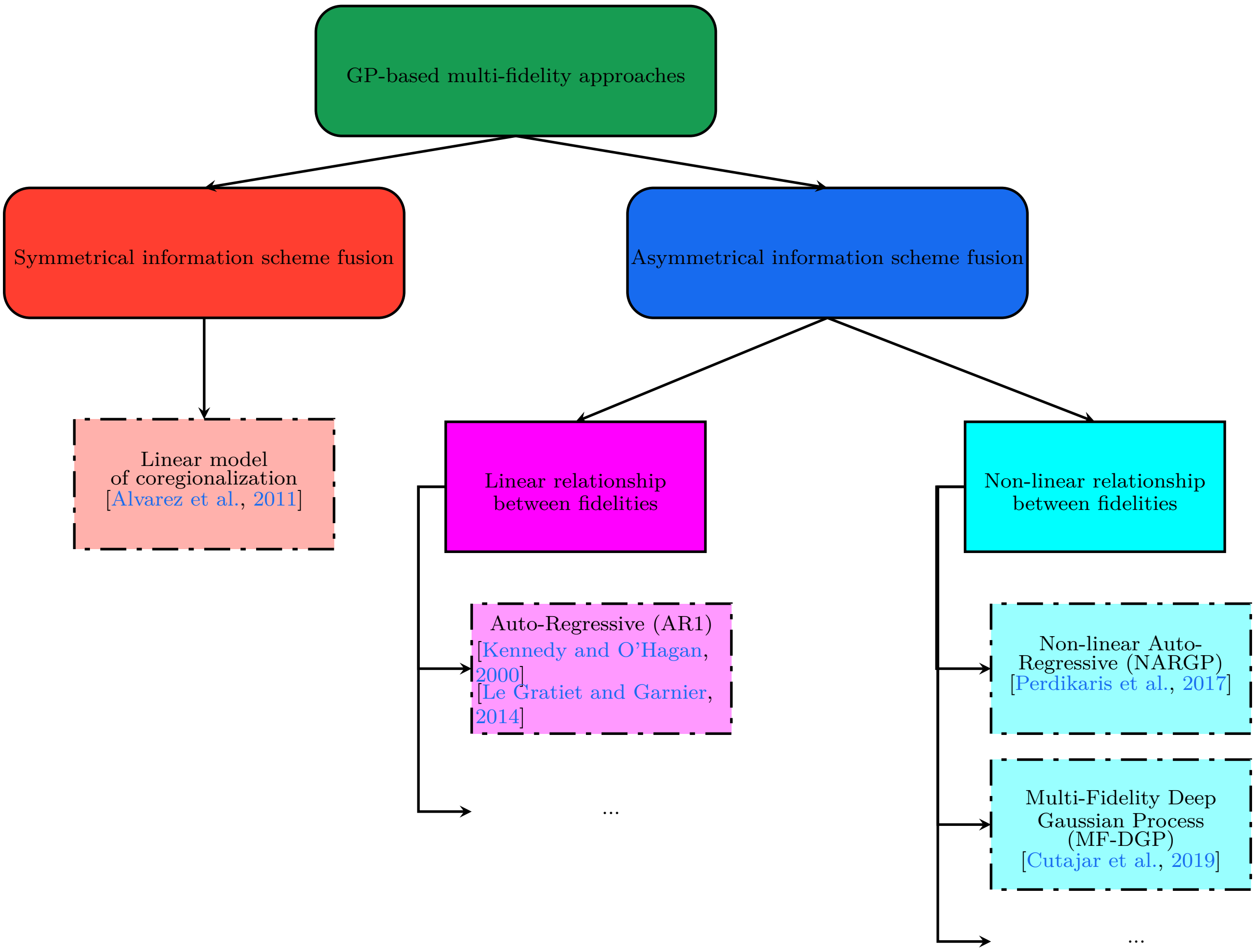

GP-based multi-fidelity approaches
Symmetrical information scheme fusion
Asymmetrical information scheme fusion
Linear model of coregionalization [Alvarez et al., 2011]
Linear relationship between fidelities
Non-linear relationship between fidelities
Auto-Regressive (AR1) [Kennedy and O'Hagan, 2000] [Le Gratiet and Garnier, 2014]
...
Non-linear Auto-Regressive (NARGP) [Perdikaris et al., 2017]
Multi-Fidelity Deep Gaussian Process (MF-DGP) [Cutajar et al., 2019]
...

Fig. 3.16 Classication of GP-based multi-fidelity approaches.

(Chapter 2, Section 2.1.2). Moreover, in the case of MF-DGP, the inputs at each layer are a combination of inputs in the original input space with the outputs of the previous layer (see Fig. 3.15), hence, optimizing freely the inducing inputs is not adequate. An optimization framework of MF-DGP is proposed in Chapter 6 to overcome these limitations.

For the considered GP-based multi-fidelity approaches described above, a classification is illustrated in Figure 3.16. The main distinctions correspond to the symmetrical or asymmetrical treatment of the fidelity information and the linear or non-linear relationship between the fidelities. In the first part of Chapter 6, a numerical comparison between these different approaches, as well as the proposed improved MF-DGP, on analytical and physical problems in different scenarios is carried out.

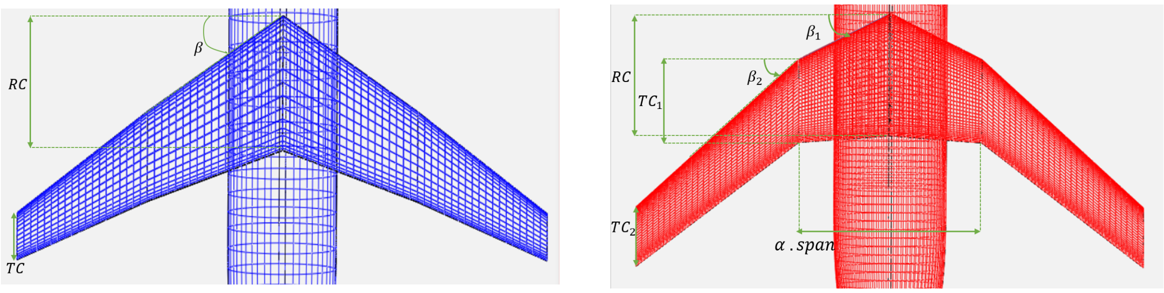


Fig. 3.17 A one-section wing characterized by 3 design variables: its root chord (RC), tip chord (TC), and the sweep angle (β) (left) can be used as a low-fidelity model of a two-section wing characterized by 6 design variables: its root chord (RC), tip chord of the first section (TC_1), tip chord of the second section (TC_2), sweep angle of the first section (β_1), sweep angle of the second section (β_2) and the relative span of the first section (α) (right).

3.4.2 Multi-fidelity with variable input space parameterization

The majority of multi-fidelity approaches assume that fidelities share the same input space. However, in practice, this is not always the case. In fact, due to either different modeling approaches from one fidelity to another, or omission of some variables in the lower-fidelity models, the input spaces may have distinct parameterization forms and/or dimensionality. For instance, in aerodynamics, to model multiple-section wing, simplified planform characterization can be used considering one section with average chords and sweep angles (Fig. 3.17).

In the literature, the main multi-fidelity approaches that address this issue belong to the space mapping multi-fidelity class [Bandler et al., 2004]. The space mapping multi-fidelity methods act on the inputs rather than the outputs of the models. The basic concept is to transform the high-fidelity inputs using a parametric function in order to minimize a distance between the corresponding low-fidelity outputs of this mapping and the exact high-fidelity outputs [Bandler et al., 2004]. In the space mapping approaches two fidelities are considered, hence, instead of using the n_{fi} levels of fidelity notation, the couple of high-fidelity and low-fidelity inputs/outputs data are respectively noted ($\mathbf{X}_{hf}, \mathbf{y}_{hf}$) and ($\mathbf{X}_{lf}, \mathbf{y}_{lf}$) for the space mapping approaches:

$$\hat{\beta} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(||y_{hf}^{(i)} - f_{lf}^{\text{exact}} \left(\boldsymbol{\psi}_{\beta}(\mathbf{x}_{hf}^{(i)}) \right) || \right) \tag{3.53}$$

where m denotes the size of the set of mapped points which is a subset of the HF data chosen based on trust region optimization algorithms, $\boldsymbol{\psi}_{\boldsymbol{\beta}}(\cdot)$ corresponds to the mapping, and $\boldsymbol{\beta}$ to its vector of parameters (in most applications $\boldsymbol{\psi}_{\boldsymbol{\beta}}(\cdot)$ is considered linear). The space mapping has been extensively used for multi-fidelity [Bandler et al., 2004, 2006; Robinson et al., 2008; Koziel, 2010] and different parametric mappings have been used, for instance, aggresive space mapping [Rayas-Sanchez, 2016] and neural networks [Rayas-Sánchez, 2004].

The space mapping approaches were first used in the case of variable-size input parameterization in [Robinson et al., 2008]. However, they are used in an optimization context, and the mapping is performed around the optimum candidates and is updated at each iteration of the trust-region optimization. This is not suited from a modeling point of view where the analysis of the high-fidelity function is performed for the whole input space.

A nominal mapping $\boldsymbol{\psi}_0(\cdot)$, based on practical insights of the multi-fidelity problem, is usually required. It expresses the assumed relationship between the different input spaces. In some cases, this nominal mapping is trivial. For instance, if the set of high and low-fidelity inputs are from the same set of physical equations, the low-fidelity inputs can then be a subset of the high-fidelity ones. Usually, the nominal mapping is problem specific and is defined based on expert opinion. A multi-fidelity approach as a bias correction [Li et al., 2016] (BC) can then be used based on this nominal mapping:

$$f_{hf}(\mathbf{x}) = f_{lf}(\boldsymbol{\psi}_0(\mathbf{x})) + \gamma(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^d \tag{3.54}$$

The Input Mapping Calibration (IMC) [Tao et al., 2019] is an approach that seeks to obtain a potentially better mapping than the nominal mapping. As the space mapping approach, it consists in finding a parametric mapping $\boldsymbol{\psi}_{\boldsymbol{\beta}}(\cdot)$. However, here the mapping is considered for the whole input space and the parameters of the mapping are obtained by minimizing the difference between the LF and HF model outputs on the HF data points plus a regularization term $\tau(\boldsymbol{\beta}, \boldsymbol{\beta}_0)$ based on the nominal mapping parameters $\boldsymbol{\beta}_0$:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\sum_{i=1}^{n_{hf}} \left(y_{hf}^{(i)} - f_{lf}^{\text{exact}} \left(\boldsymbol{\psi}_{\boldsymbol{\beta}} \left(\mathbf{x}_{hf}^{(i)} \right) \right) \right)^2 + \tau(\boldsymbol{\beta}, \boldsymbol{\beta}_{\mathbf{0}}) \right) \tag{3.55}$$

where n_{hf} corresponds to the number of HF training data points. The high-fidelity input data is then projected with the obtained mapping on the low-fidelity input space, and a multi-fidelity model with the same input spaces can be used (Fig. 3.18).

This optimization of the mapping parameters is done previously to the training of the multi-fidelity model, which prevents the parameters of the mapping to be updated, once the multi-fidelity model is optimized. Besides, the optimization is done using the exact low-fidelity model, which is considered as computationally free to evaluate, however, in many applications, it may not be the case. Moreover, the correlations over the original HF input space are not taken into account, since the multi-fidelity model is trained only on the lower-fidelity input space. Finally, the mapping parameters are estimated based on the concept that the low-fidelity model shares a similar trend with the high-fidelity one. This is the case in some applications as microwave applications where space mapping has emerged. However, in many multi-fidelity problems, minimizing the distance between the outputs does not guarantee an appropriate mapping (See Chapter 6).

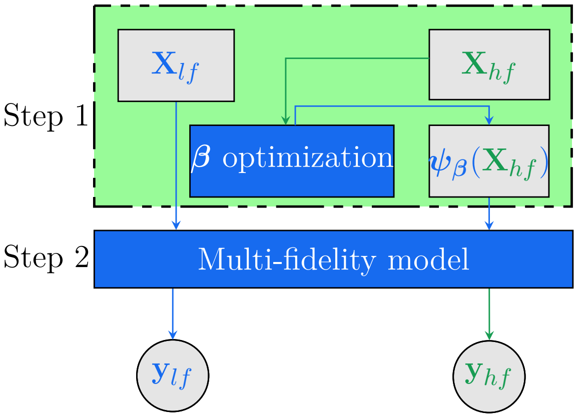


Fig. 3.18 Graphical representation of the IMC approach.

Up until now, the mapping from the high to the low-fidelity inputs is based on parametric deterministic functions and is usually trained sequentially with the multi-fidelity model. In Chapter 6, a novel approach to handle multi-fidelity modeling in the case of different input space parameterizations is proposed based on a non-parametric Bayesian mapping which is learned and is embedded in the multi-fidelity model.

3.5 Conclusion

This chapter reviewed the existing Gaussian process-based approaches for problems relative to the analysis and optimization of complex systems. In fact, non-stationarity, optimization of computationally expensive black-box functions, multiple antagonistic objectives to consider, and different levels of fidelities of code available, are recurrent and major investigated axes in the analysis and optimization of complex systems. Gaussian process-based approaches have been continuously developed for these problems in the literature these last two decades. These different approaches are summarized in the next paragraphs with their respective limitations that introduce the contribution chapters of this thesis.

For non-stationarity, three class of approaches adapt GPs to problems with input-dependent variations: direct formulation of non-stationary kernels, local stationary covariance functions, and non-linear mapping approaches. In this manuscript, non-stationarity is considered in the context of Bayesian optimization where data is scarce and may be high-dimensional. A direct formulation of non-stationary kernels in high-dimensional spaces is difficult due to the high-parametrization of the non-stationary kernels, while local stationary covariance functions may not be adapted to a configuration where data is scarce due to partition of the training data. Non-linear mapping approaches have been used in the context of Bayesian optimization. However, using parameterized functions as mappings may limit the flexibility of this class of approaches. For that, in Chapter 4, we propose to couple deep Gaussian processes (which use the hidden layers as non-parametric Bayesian mappings) to Bayesian optimization in order to address computationally expensive black-box optimization problems with non-stationary behaviors.

Bayesian optimization has been adapted to multiple objectives by using Pareto-dominance based infill crtieria such as the expected hyper-volume improvement. In the classic formulation of multi-objective Bayesian optimization, the objectives are considered independent, however, often the objectives show negative correlation and assuming independence may yield to loss of information. An adaptation of multi-objective Bayesian optimization to take into account these correlations have been developed using the linear model of coregionalization. However, this model considers only linear correlations between the objectives. The contribution of Chapter 5 addresses this issue by developing a new model based on deep Gaussian processes called multi-objective deep Gaussian process model which can exhibit non-linear correlation between the objectives.

Different Gaussian process-based multi-fidelity approaches have been developed. These methods are used in the context of the same input space definition for all the fidelities. However, in some applications, different parameterizations are used for each fidelity input space. To handle these applications, multi-fidelity space mapping approaches are usually used. However, they are based on deterministic parameterized mapping functions. Therefore, they have limited flexibility and may be problem-dependent. To overcome this issue, in Chapter 6, another model is developed based on multi-fidelity deep Gaussian process model. This model called multi-fidelity deep Gaussian process embedded mapping, includes a Bayesian non-parametric mapping between the input spaces of the different fidelities within the multi-fidelity deep Gaussian process model, thus, allowing a joint optimization of the multi-fidelity model and the input mappings.

Part II

Single and Multi-Objective Bayesian Optimization using Deep Gaussian Processes

Chapter 4

Bayesian Optimization with Deep Gaussian Processes for Non-Stationary Problems

> *"Experience with real-world data, however, soon convinces one that both stationarity and Gaussianity are fairy tales invented for the amusement of undergraduates."*
>
> Thomson (1994)

Chapter contributions

- Coupling of Bayesian optimization and deep Gaussian processes to handle computationally intensive black box and non-stationary constrained optimization problems.
- Assessment of this coupling with respect to state-of-the-art non-stationary approaches on analytical test problems.
- Application of this coupling on an extensive benchmark including representative aerospace optimization design problems.

$\mathcal{CH}_4$

Bayesian Optimization (BO) is a widely used approach to handle the optimization of computationally intensive and black-box problems (Chapter 3, Section 3.2). Generally, BO is based on Gaussian Process (GP) regression as a Bayesian model. The advantage of GP is that it gives an analytically tractable Gaussian predictive posterior distribution and its non-parametric form allows flexible modeling capabilities (Chapter 2, Section 2.2). However, standard GPs are used with a stationary covariance function *i.e.* they are based on the relative distance between the inputs and do not depend on the respective regions where the inputs lie. This induces some difficulties for GPs to handle some optimization problems in engineering design [Xiong et al., 2007]. In fact, in multiple engineering design problems, the response of interest varies with different degrees of smoothness depending on the input values. For instance, the design of a rocket booster involves different discipline behaviors depending on the input region *e.g.* the transition from subsonic to supersonic induces abrupt changes in the aerodynamic discipline [Gramacy and Lee, 2008]. The different approaches that have been developed to overcome this issue and adapt GPs to non-stationarity can be classified into three categories: direct formulation of a non-stationary covariance function [Higdon et al., 1999; Paciorek and Schervish, 2006; Plagemann et al., 2008; Heinonen et al., 2016], local stationary covariance function [Haas, 1990; Tresp, 2001; Rasmussen and Ghahramani, 2002; Bettebghor et al., 2011; Trapp et al., 2020], and non-linear mapping [Sampson and Guttorp, 1992; Gibbs, 1998; Xiong et al., 2007; Snoek et al., 2014; Marmin et al., 2018] (Chapter 3, Section 3.1 for details on these approaches). The non-linear mapping approach, consisting of a warping of the input space, has been used in a BO framework [Toal and Keane, 2012; Snoek et al., 2014], showing interesting performance in the context of optimization where data is scarce and eventually high-dimensional. However, parametric functions as mapping are problem dependent and do not provide intrinsically a measure of uncertainty about the mapping. One way to overcome this issue is to use a non-parametric Bayesian mapping. For that, GPs are natural candidates. Using GPs as input warping for a GP yields to a functional composition of GPs that is a Deep Gaussian Process (DGP) (Chapter 2, Section 2.3). Therefore, DGPs allow an automatic non-parametric Bayesian mapping of the input space.

The capacity of a DGP to handle non-stationarity has been first elucidated by [Damianou, 2015], where it has been used to fit a step function. This type of functions characterized by a flat region broken with a discontinuity are common in optimization problems due to some constraints where there is an abrupt transition from feasible to unfeasible regions of the input space. In [Damianou, 2015], it is shown that unlike GPs for which modeling this discontinuity is difficult, DGPs based on their intermediate layers are able to capture the discontinuity. This confirms the deep learning theory intuition, that is, it offers the possibility to capture multiple variations through the composition of multiple functions. This composition allows to use simple functions to learn a highly varying function [LeCun et al., 2015]. In Fig. 4.1, unlike a regular GP, a 2-layer DGP (henceforth, a DGP with l hidden layers is referenced as a l-layer DGP) is able to capture the non-stationarity of the modified Xiong function (see Eq. C.1, Appendix C for its definition). Therefore, a hierarchical composition of GPs presents better results than a shallow GP in the approximation of complex functions as described in [Damianou and Lawrence, 2013; Dai et al., 2015; Salimbeni and Deisenroth, 2017; Dutordoir et al., 2017]. In fact, a DGP allows a Bayesian and flexible way of kernel construction through input warping and dimensionality expansion to better fit the response in a scarce data context [Damianou and Lawrence, 2013].

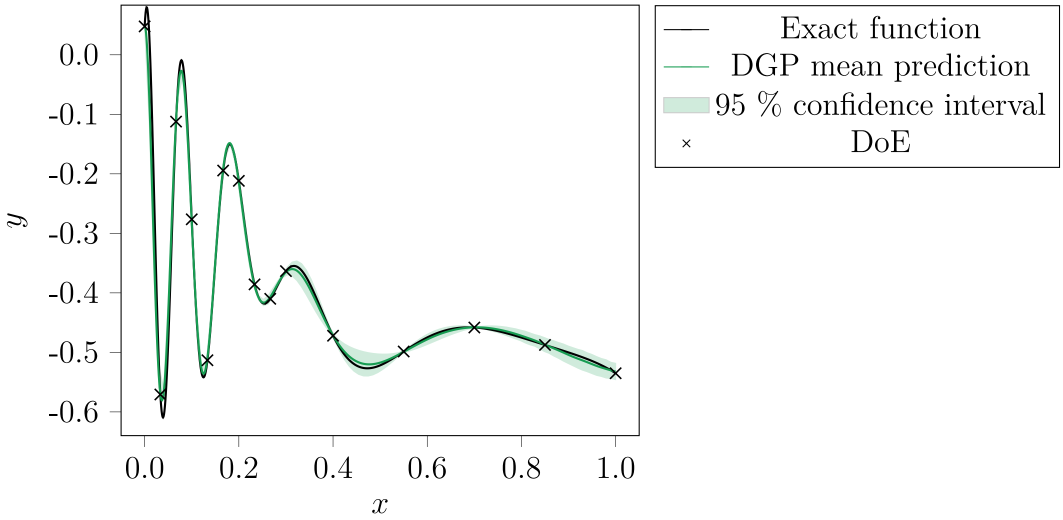


Fig. 4.1 Approximation of the modified Xiong-function, a non-stationary 1-dimensional function by a 2-layer DGP model. The model captures the non-stationarity of the exact function.

The coupling of DGPs and BO has been briefly introduced previously in [Dai et al., 2015], and it was directly applied on an analytical 2D problem. However, some issues may arise from this coupling that have not been investigated yet. In fact, as shown in Section 4.1.1, one of the limitation of the current training of DGPs is the under-estimation of the predictive uncertainty which can be penalizing when used within a BO framework. Additionally, a DGP has an architecture to be defined with respect to the data in hand, and given the BO iterative structure a trade-off between complexity and power of representation may be made at each iteration of the algorithm. Moreover, in contrast with a GP, the predictive distribution of a DGP is not necessary Gaussian, therefore, some infill criteria in BO such as the EI can not be directly used (Fig 4.2).

In this chapter, the key contribution is to investigate the application of DGPs for non-stationarity optimization problems in a BO framework. For that, an improved training technique is proposed to obtain a better predictive uncertainty as well as a more adapted training of DGP in the BO framework. Moreover, the influence of

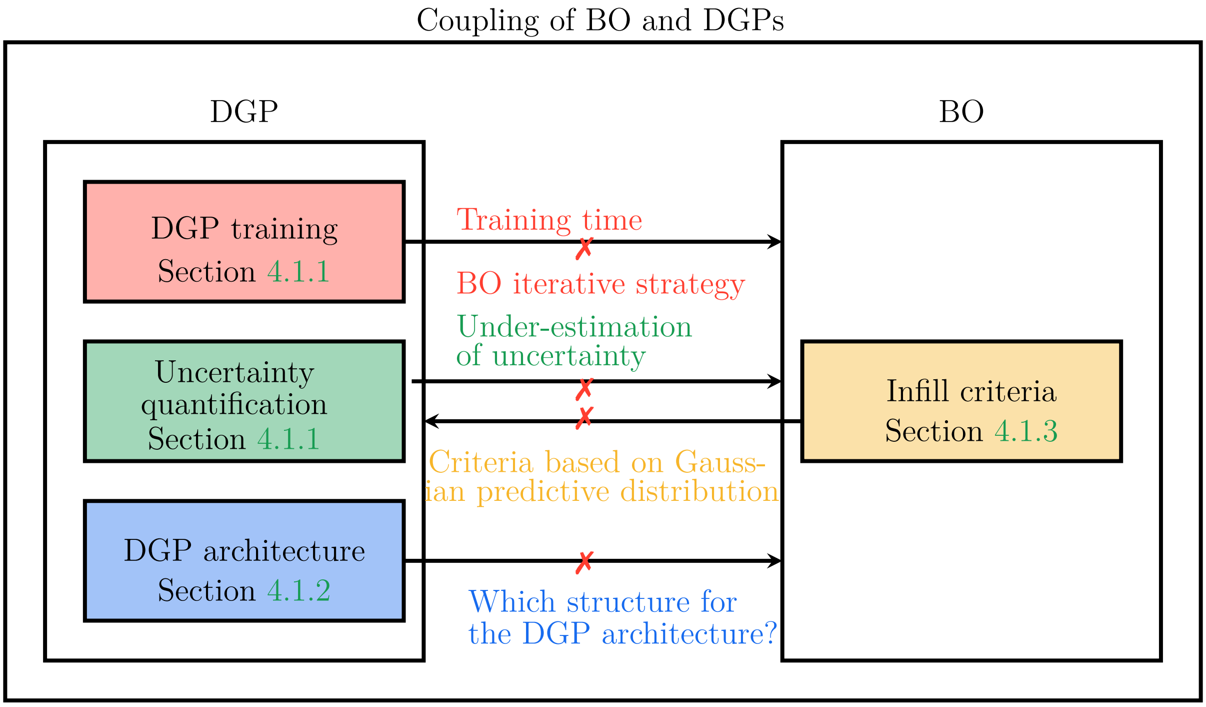


Fig. 4.2 The coupling of BO and DGP arises some issues within DGPs from the training, the predictive uncertainty, the DGP architecture perspectives, and within BO from the infill criteria perspective.

DGP architecture is investigated within the perspective of BO. The infill criteria used in BO are also discussed when coupled with DGPs. These different investigations allow us to propose an algorithm for BO coupled with DGPs for the optimization of non-stationary problems. Eventually, the proposed framework for DGP and BO is numerically evaluated on a benchmark of analytical test problems and representative aerospace design problems.

This chapter is organized in two main sections. In Section 4.1, a framework for coupling BO and DGPs is proposed. The proposed framework is based on an investigation covering several aspects, such as the training approach of DGP in the context of BO, uncertainty quantification, architecture of the DGP and infill criteria. Section 4.2 presents experimentations on analytical optimization problems and on aerospace optimization test problems, to assess the performance of BO & DGP compared to relevant existing approaches.

4.1 Bayesian Optimization using Deep Gaussian Processes

In this section, a deep investigation is followed in order to highlight the different challenges that may rise in the BO & DGP coupling and to propose contributions to overcome them. This concerns the training approach for the DGP, the uncertainty quantification of DGP, the infill criteria, the induced variables in each layer and the architecture of the DGP (number of layers, number of units, *etc.*). In this section, different analytical functions are used to illustrate the analyses made. These functions are described in Appendix C.

4.1.1 Training

Different inference approaches have been developed for DGP as reviewed in Chapter 2, Section 2.3.2. The doubly stochastic inference approach proposed in [Salimbeni and Deisenroth, 2017] is preferred in the present study since it keeps the dependence between layers and does not assume a particular form of the kernels used. The loss of analytical tractability may be compromising, since a Monte Carlo sampling approach is required. However, the form of the Evidence Lower Bound (ELBO) is fully factorizable over the data set allowing important parallelization.

In this section, an optimization approach of the ELBO based on natural gradient [Amari, 1998] is proposed which is adapted to the context of BO since it enables a more adapted predictive uncertainty quantification of the model and reduces the number of optimization iterations needed in the training. Empirical experimentations are carried out to demonstrate these improvements compared to the classical training approach based on ordinary stochastic gradient [Salimbeni and Deisenroth, 2017].

Optimization of the ELBO in the context of BO

The ELBO for a DGP configuration of l layers obtained using the doubly stochastic inference approach can be written as follows (details on this derivation are presented in Chapter 2, Section 2.3.2)

$$\mathcal{L} = \sum_{j=1}^{n} \mathbb{E}_{q\left(h_{[l]}^{(j)}\right)} \left[\log p\left(y^{(j)}|h_{[l]}^{(j)}\right)\right] - \sum_{i=1}^{l} \mathbb{KL}\left[q(\mathbf{U}_{[i]})||p(\mathbf{U}_{[i]})\right] \tag{4.1}$$

The ELBO is usually optimized using an ordinary stochastic gradient descent [Salimbeni and Deisenroth, 2017] with respect to the hyper-parameters of the GPs $\{\boldsymbol{\theta}_{[i]}\}_1^l$, the induced inputs $\{\mathbf{Z}_{[i]}\}_1^l$, and also the variational parameters $\{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}\}_1^l$ of the variational distributions $\{q(\mathbf{U}_{[i]}) = \mathcal{N}(\mathbf{U}_{[i]}|\bar{\mathbf{U}}_{[i]}, \boldsymbol{\Gamma}_{[i]}\}_1^l$. The ordinary gradient descent considers the steepest direction with respect to the euclidean distance:

$$\boldsymbol{\Theta}_{t+1} = \boldsymbol{\Theta}_t - \gamma_t \nabla_{\boldsymbol{\Theta}} \mathcal{L}|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}_t} \tag{4.2}$$

where $\boldsymbol{\Theta}_t$ is the set of all the parameters of the ELBO $\mathcal{L}$, γ_t the step size, and t denotes the iteration number of the ordinary gradient algorithm. In the case of the ELBO, since the variational parameters define a distribution, the parameter space is not characterized by an euclidean norm. The ordinary gradient descent in this case is not a suitable direction to follow in optimization. To illustrate this, consider two uni-variate Gaussian distributions parameterized by their mean and variance $p_1(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mu_1, \sigma_1^2)$ and $p_2(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mu_2, \sigma_2^2)$. A similar change $\mathrm{d}\sigma^2$ in the variance will update equivalently the distributions in terms of euclidean distance. However, the obtained distributions $p_{1,\text{updated}}$ and $p_{2,\text{updated}}$ are not equivalently updated in terms of KL divergence that is a measure usually used to quantify dissimilarities between distributions:

$$\mathbb{KL}[p_i||p_{i,\text{updated}}] = \log\left(\frac{\sqrt{\sigma_i^2 + \mathrm{d}\sigma^2}}{\sigma_i}\right) + \frac{\sigma_i^2}{2(\sigma_i^2 + \mathrm{d}\sigma^2)} - 0.5 \tag{4.3}$$

This equation shows that the dissimilarity induced between the original distribution and the updated one depends on the original variance of the distribution. For a low variance, the update yields to a large dissimilarity, while for a high variance the same update yields to a low dissimilarity. Moreover, changing the parameterization (for instance, using the precision (inverse of the variance) instead of the variance) yields to a different result.

This non-adaptation of ordinary stochastic gradient to the parameter space may induce several issues. In fact, since the direction proposed by the ordinary gradient does not point to the steepest descent, the training may take a large number of iterations. Secondly, even if it is expected to have an under-estimation of the predictive uncertainty when using variational inference (Chapter 2, Section 2.1.2), this under-estimation can be further aggravated due to a poor optimization of the ELBO. In fact, in the expression of the ELBO in Eq. (4.1), the minimization of the second term, corresponding to the KL divergence, which is particularly difficult for ordinary gradient, has a direct incidence on the predictive uncertainty. In fact, the prediction for locations $\mathbf{X}^*$ using a DGP is obtained by sampling samples from the first layer, through the inner layers, until reaching the final layer, using the variational posterior distribution at each layer i,

$q\left(\mathbf{F}^*_{[i]}|\mathbf{F}^*_{[i-1]},|\bar{\mathbf{U}}_{[i]},\mathbf{\Gamma}_{[i]},\mathbf{Z}_{[l]},\mathbf{X}^*\right)=\mathcal{N}\left(\mathbf{F}^*_{[i]}|\boldsymbol{F}_{[i]},\mathbf{\Sigma}_{[i]}\right)$ with $\mathbf{F}^*_{[0]}=\mathbf{X}^*$ and :

$$\boldsymbol{F}_{[i]}=\tilde{\mathbf{K}}\bar{\mathbf{U}}_{[i]} \tag{4.4}$$

and

$$\mathbf{\Sigma}_{[i]}=\mathbf{K}_{[i]}\left(\mathbf{F}^*_{[i-1]},\mathbf{F}^*_{[i-1]}\right)-\tilde{\mathbf{K}}_{[i]}\left(\mathbf{K}\left(\mathbf{Z}_{[i]},\mathbf{Z}_{[i]}\right)-\mathbf{\Gamma}_{[i]}\right)\tilde{\mathbf{K}}^{\intercal}_{[i]} \tag{4.5}$$

with

$$\tilde{\mathbf{K}}_{[i]}=\mathbf{K}_{[i]}\left(\mathbf{F}^*_{[i-1]},\mathbf{Z}_{[i]}\right)\mathbf{K}_{[i]}\left(\mathbf{Z}_{[i]},\mathbf{Z}_{[i]}\right)^{-1} \tag{4.6}$$

where $\mathbf{K}_{[i]}(\cdot,\cdot)$ corresponds to the kernel function at layer i. In Eq. (4.5) the red colored term shows the importance of the calibration of the variational distribution $q(\mathbf{U}_{[i]})=\mathcal{N}\left(\mathbf{U}_{[i]}|\bar{\mathbf{U}}_{[i]},\mathbf{\Gamma}_{[i]}\right)$ with respect to the prior distribution $p(\mathbf{U}_{[i]})=\mathcal{N}\left(\mathbf{U}_{[i]}|\mathbf{0},\mathbf{K}(\mathbf{Z}_{[i]},\mathbf{Z}_{[i]})\right)$ for the predictive uncertainty estimation. This calibration is performed by the trade-off between the minimization of the KL term and the maximization of the expectation term in Eq. (4.1). Therefore, the importance of an adapted optimization algorithm to avoid poor predictive uncertainty estimation.

To overcome these issues, the differential geometry of the distribution parameter space (its local curvature) is taken into account. For that, a distribution parameter space is characterized as a Riemannian manifold endowed with the Fisher information metric that is called statistical manifold. The Fisher information metric is a measure of the curvature of the distribution parameter space and is defined for a distribution $q(\mathbf{U}_{[i]})$ as:

$$\mathbf{F}_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}}=-\mathbb{E}_{q(\mathbf{U}_{[i]})}\left[\nabla^2_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}}\log q(\mathbf{U}_{[i]})\right]$$

The Fisher information is invariant to parameterization and depends only on the distribution. The steepest direction of a loss function defined on a statistical manifold is given by the ordinary gradient rescaled by the inverse Fisher information matrix and is called natural gradient [Amari, 1998]:

$$\begin{aligned}\boldsymbol{\theta}_{t+1,q(\mathbf{U}_{[i]})}&=\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}-\gamma_t\mathbf{F}^{-1}_{\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}}\nabla_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}}\mathcal{L}|_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}=\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}}\\&=\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}-\gamma_t\tilde{\nabla}_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}}\mathcal{L}|_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}=\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}}\end{aligned} \tag{4.7}$$

where $\tilde{\nabla}_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}}\mathcal{L}|_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}=\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}}=\mathbf{F}^{-1}_{\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}}\nabla_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}}\mathcal{L}|_{\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}=\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})}}$. Therefore, the optimization update comes back to the computation of the Fisher information matrix at each iteration. Since $\{q(\mathbf{U}_{[i]})\}_1^l$ are considered as Gaussian distributions, the Fisher information matrix has a simple form when using the natural parameterization $\boldsymbol{\theta}_{q(\mathbf{U}_{[i]})} = \left[\boldsymbol{\Gamma}_{[i]}^{-1}\bar{\mathbf{U}}_{[i]}, -\frac{1}{2}\boldsymbol{\Gamma}_{[i]}^{-1}\right]$ [Hensman et al., 2013]. In fact, the Fisher information matrix comes back to the gradient of the expectation parameters with respect to the natural parameters $\frac{\partial \boldsymbol{\nu}_{q(\mathbf{U}_{[i]})}}{\partial \boldsymbol{\theta}_{q(\mathbf{U}_{[i]})}}$ where the expectation parameters are $\boldsymbol{\nu}_{q(\mathbf{U}_{[i]})} = \left[\bar{\mathbf{U}}_{[i]}, \bar{\mathbf{U}}_{[i]}\bar{\mathbf{U}}_{[i]}^{\intercal} + \boldsymbol{\Gamma}_{[i]}\right]$.

Natural gradient has been used in the case of conjugate variational inference in GP [Hensman et al., 2013] and also in the non-conjugate case [Salimbeni et al., 2018]. In these works, it is shown that natural gradient performs better in the case of ill-conditioned posteriors where ordinary gradient is not able to converge. These ill-conditioned cases are recurrent in BO due to sequential addition of data in a non-uniform way.

Following these works, a generalization of natural gradients to the training of DGP is proposed. For this, for each layer, a natural gradient descent is performed for the variational parameters. More specifically, a loop is performed between an optimization step with the natural gradient to perform the optimization with respect to the parameters of the variational distributions $\{q(\mathbf{U}_{[i]})\}_1^l$ while fixing the other parameters, and an optimization step using a stochastic gradient descent optimizer (Adam optimizer [Kingma and Ba, 2014]) to perform the optimization with respect to the Euclidian space parameters $\{\boldsymbol{\theta}_{[i]}\}_1^l, \{\mathbf{Z}_{[i]}\}_l^l$ (Fig. 4.3). Using the natural gradient for all the distributions of the inner layers in the case of DGPs is tricky. Indeed, the Fisher information matrix of the inner layer variational distributions may show high ill-conditioning behavior. This is illustrated in Fig 4.4 where the condition number (the ratio between the highest eigenvalue by moduli and the lowest eigenvalue by moduli) of the Fisher information matrix of the distribution parameter space increases from the last to the first layer of the DGP. This ill-conditioning of the first layers may lead to amplification of the round-off error when inverting the Fisher information matrix. Additionally to the numerical issue, an important condition number implies large statistical fluctuations [Vallisneri, 2008]. It is therefore more cautious to use smaller steps when optimizing the parameter values of the inner layers to avoid instability. This is shown in Fig 4.5 where a smaller step size for the inner layers compared to the last layer yields to a stabilized convergence of the parameter value. Thus, a simple scheme is proposed where the step-size is constrained to a decreasing order from the last to the first layer.

In the context of BO, this optimization procedure has to be repeated after each added point using an initialization scheme for the parameters (random initialization, Latin Hyper-cube Sampling (LHS), data-dependent initialization [Ulapane et al., 2020]), which may be time consuming. In fact, these strategies do not take into account the

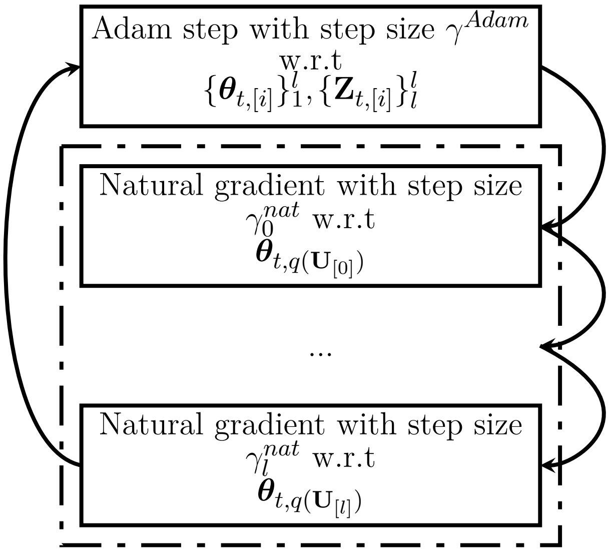

Adam step with step size γ^{Adam} w.r.t $\{\boldsymbol{\theta}_{t,[i]}\}_1^l, \{\mathbf{Z}_{t,[i]}\}_l^l$
Natural gradient with step size γ_0^{nat} w.r.t $\boldsymbol{\theta}_{t,q(\mathbf{U}_{[0]})}$
...
Natural gradient with step size γ_l^{nat} w.r.t $\boldsymbol{\theta}_{t,q(\mathbf{U}_{[l]})}$

Fig. 4.3 Loop procedure for the optimization of the ELBO. The loop consists of an optimization step using an ordinary stochastic gradient (Adam optimizer) for the deterministic parameters $\{\boldsymbol{\theta}_{t,[i]}\}_1^l, \{\mathbf{Z}_{t,[i]}\}_l^l$ and an optimization step using the natural gradient for the variational parameters $\boldsymbol{\theta}_{t,q(\mathbf{U}_{[0]})}$ of each layer i. The step size of the natural gradient are taken in this order $\gamma_0^{Nat} \leq \ldots \leq \gamma_i^{Nat} \leq \ldots \leq \gamma_l^{Nat}$ to avoid overly large step in the first layers.

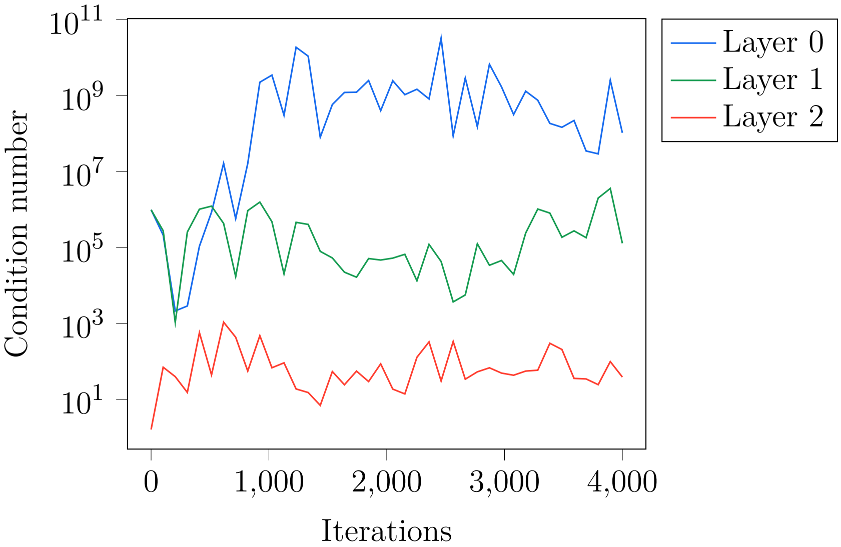


Fig. 4.4 Evolution of the condition number of the Fisher information matrix of the variational parameter space at each layer of a 2 layer-DGP. The inner layers show ill-conditioning behavior compared to the last layer.

data-additive structure of BO *i.e.*, after adding one data-point to the data-set the optimum of the parameters may not move much from its previous location. Hence, to

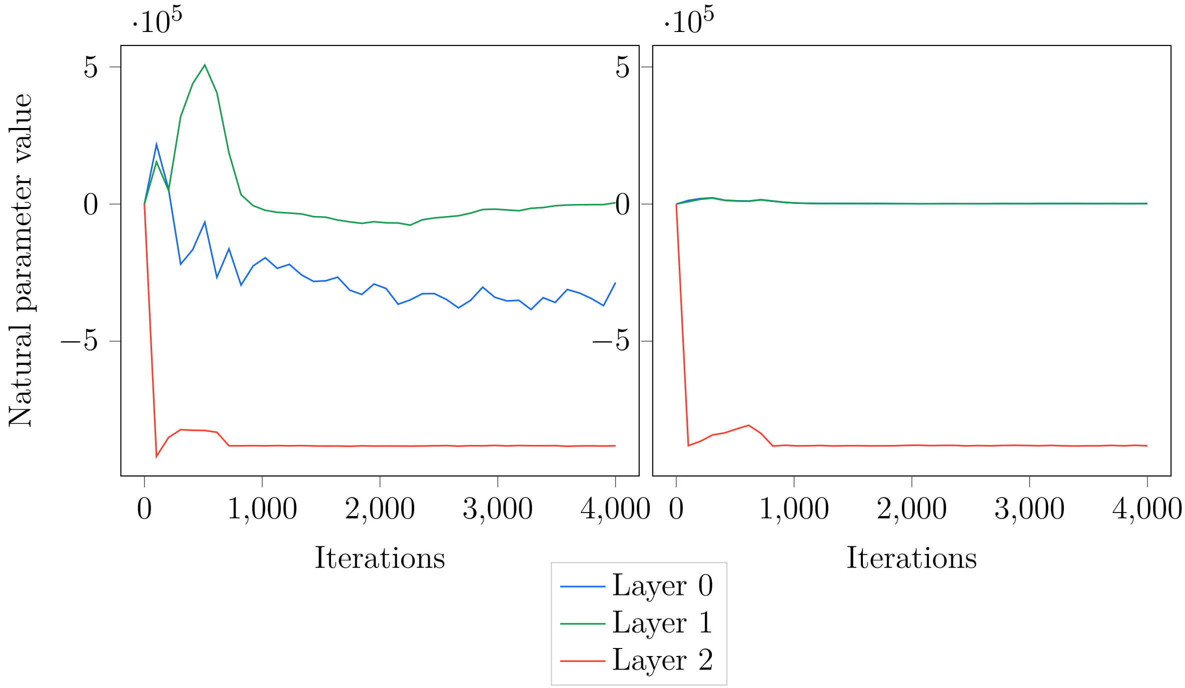


Fig. 4.5 Evolution of the first natural parameter of three induced variables (one at each layer) throughout the training of a 2-layer DGP. (left) The evolution of the parameters when using a step-size of 0.1 for the different layers shows that the parameter of the last layer quickly stabilizes unlike the two first layer parameters. (right) The evolution of the parameters when using a step-size of 0.1 for the last layer and 0.01 for the two inner layers shows that when reducing the step size of the inner layers the optimization is more stable.

take advantage of BO, the optimization can be initialized using the optimal values of the previously trained DGP model. As shown in Fig. 4.6, this allows faster convergence. However, this can make the optimization converge to a poor local optimum. Therefore, a complete training of the model is recommended after a certain number of BO iterations depending on the problem at hand. Moreover, using the previous parameter values requires that the number of parameters does not change from one iteration to another. Hence, the architecture of BO has to be fixed when initializing from the previous DGP model, otherwise if the architecture changes in the next iteration the initialization has to be done from scratch or a specific initialization for the added parameters have to be proposed. How to choose this architecture is discussed in details in Section 4.1.2.

The pseudo algorithm (Algorithm 1) describes the proposed training strategy of the DGP model within the context of BO.

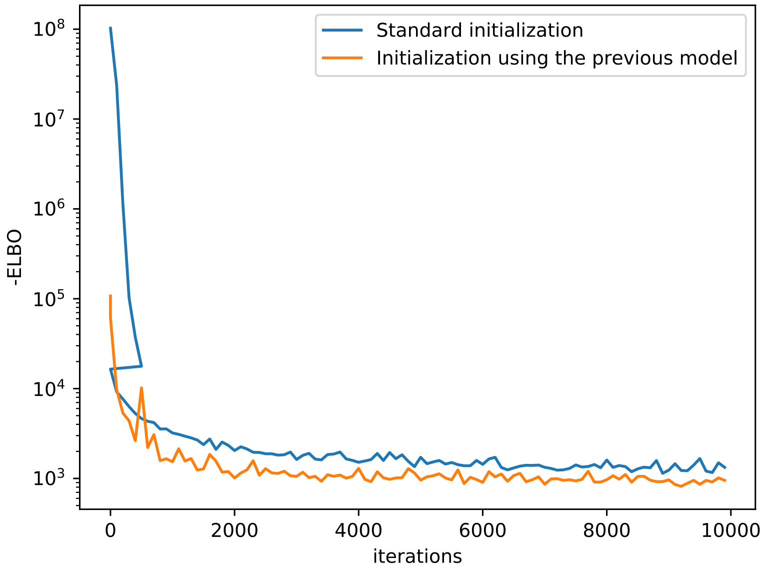


Fig. 4.6 Comparison of the evolution of the optimization of the ELBO in the case of using the standard initialization procedure (in blue) and in the case of using the previous model optimal parameters as the initialization (in orange). A 2 layer-DGP is used on a data set with a size of 100 points on the Trid function. Using the previous model allows better and faster convergence.

Experimental comparison on the DGP training

Comparison with respect to ELBO convergence

The evolution of the ELBO using three different optimization approaches is presented in Fig. 4.7 for three different problems (Appendix C). The optimization using the proposed optimization procedure named as Nat Grads in Fig. 4.7 gives the best results compared to the classical approach using only stochastic gradient (Adam). In fact, natural gradient for all the layers is faster and converge to a better value than the other two approaches. For the Hartmann 6d and the Trid functions, the size of the step of the natural gradient for the first layers is reduced compared to the step size of the last layer, in order to avoid overlarge step size.

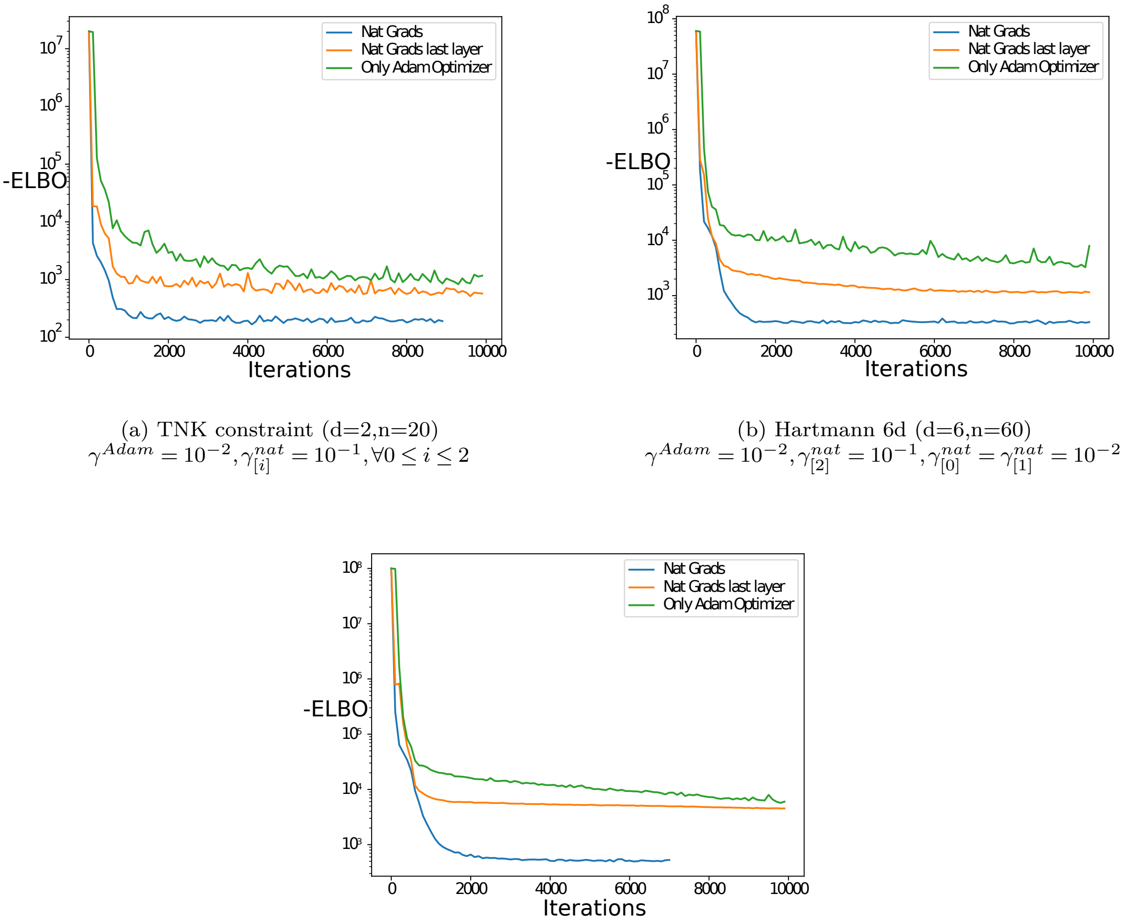
Comparison with respect to prediction accuracy and uncertainty quantification

A test set to estimate the Root Mean Square Error (RMSE) and the Mean Negative test Log-Likelihood (MNLL) is used to assess the prediction and the uncertainty estimator performance of the models trained by the proposed optimization approach (DGP Nat)

Algorithm 1: DGP model training

```
Require: $\mathbf{X},\mathbf{y}$.
Require: The number of induced variables, $m$.
Require: The number of layers, $l$.
Require: The number of loop iterations, $iter$.
Require: The step sizes $\gamma^{Adam}$, $\gamma_i^{Nat}, \forall i = 0,\ldots,l$
Require: $\{\boldsymbol{\theta}^*_{[i]}\}_1^l$, $\{\mathbf{Z}^*_{[i]}\}_1^l$, $\{\boldsymbol{\theta}^*_{q(\mathbf{U}_{[i]})}\}_1^l$ previous model optimal hyper-parameters
  and variational parameters to initialize from if available.
if $\{\boldsymbol{\theta}^*_{[i]}\}_1^l$, $\{\boldsymbol{Z}^*_{[i]}\}_1^l$, $\{\boldsymbol{\theta}^*_{q(\boldsymbol{U}_{[i]})}\}_1^l$ available then
    Initialize parameters:
    $\{\boldsymbol{\theta}_{0,[i]}\}_1^l \leftarrow \{\boldsymbol{\theta}^*_{[i]}\}_1^l$
    $\{\mathbf{Z}_{0,[i]}\}_1^l \leftarrow \{\mathbf{Z}^*_{[i]}\}_1^l$
    $\{\boldsymbol{\theta}_{0,q(\mathbf{U}_{[i]})}\}_1^l \leftarrow \{\boldsymbol{\theta}^*_{q(\mathbf{U}_{[i]})}\}_1^l$
else
    Initialize using another procedure (random initialization, principal
      component analysis, etc.)
end
$ELBO_0 \leftarrow \mathbf{X},\mathbf{y},\{\boldsymbol{\theta}_{0,[i]}\}_1^l,\{\mathbf{Z}_{0,[i]}\}_1^l,\{\boldsymbol{\theta}_{0,q(\mathbf{U}_{[i]})}\}_1^l$
$t \leftarrow 0$
while $t < iter$ do
    $\{\boldsymbol{\theta}_{t+1,[i]}\}_1^l,\{\mathbf{Z}_{t+1,[i]}\}_1^l \leftarrow$
      Adam optimizer step$(ELBO_t,\{\boldsymbol{\theta}_{t,[i]}\}_1^l,\{\mathbf{Z}_{t,[i]}\}_1^l,\gamma^{Adam})$
    $\boldsymbol{\theta}_{t+1,q(\mathbf{U}_{[i]})} \leftarrow$ Nat grad optimizer step$(ELBO_t,\boldsymbol{\theta}_{t,q(\mathbf{U}_{[i]})},\gamma_{[i]}^{Nat}), \forall i = 1,\ldots,l$
    $t \leftarrow t+1$
end
return $DGP(\mathbf{X},\mathbf{y},\{\boldsymbol{\theta}_{t-1,[i]}\}_1^l,\{\mathbf{Z}_{t-1,[i]}\}_1^l,\{\boldsymbol{\theta}_{t-1,q(\mathbf{U}_{[i]})}\}_1^l)$
```

compared to a classic training of DGP using only stochastic gradient descent (DGP Adam) and to standard GPs (Table 4.1). In order to highlight the increase in representation accuracy by composition, the same kernel (RBF also called squared exponential) is used for all the models. The models optimized by the proposed optimization approach (DGP Nat) provide the best results. It is interesting to notice that the models optimized by the Adam algorithm on all the variables give comparable results on the prediction. However, when optimizing the variational variables with ordinary stochastic gradient, it happens that the predictive uncertainty is under-estimated as illustrated in Fig. 4.8 where the uncertainty collapses to very low values. This explains the poor value obtained of the test log-likelihood in the case of the DGP model optimized by



(a) TNK constraint (d=2,n=20)
$\gamma^{Adam} = 10^{-2}, \gamma^{nat}_{[i]} = 10^{-1}, \forall 0 \leq i \leq 2$

(b) Hartmann 6d (d=6,n=60)
$\gamma^{Adam} = 10^{-2}, \gamma^{nat}_{[2]} = 10^{-1}, \gamma^{nat}_{[0]} = \gamma^{nat}_{[1]} = 10^{-2}$

(c) Trid (d=10,n=100)
$\gamma^{Adam} = 10^{-2}, \gamma^{nat}_{[2]} = 10^{-1}, \gamma^{nat}_{[0]} = \gamma^{nat}_{[1]} = 10^{-2}$

Fig. 4.7 Comparison of the evolution of the optimization of the ELBO on three different problems using three different optimizations: the proposed approach using the natural gradient for all the variational parameters and the Adam optimizer for the deterministic parameters (Nat Grads), an alternative optimization using the natural gradient for the variational parameters of the last layer and the Adam optimizer for the rest of the parameters (Nat Grads last layer), and an optimization using the Adam optimizer for all the parameters (Only Adam optimizer). γ^{adam} is the step size of the Adam optimizer and $\gamma^{nat}_{[i]}$ is the step size of the natural gradient for the variational parameters at layer i.

only an Adam optimizer compared to the ones given by a DGP trained by the proposed approach.

In the context of BO, this uncertainty measure is crucial for the construction of infill criteria. An underestimated uncertainty will favor the BO algorithm sampling

Table 4.1 Comparison of the Root Mean Squared Error (RMSE) and the Mean Negative test Log-Likelihood (MNLL) and their standard deviations (std) on three different problems with a training data size of density 10 ($n = 10 \times d$ where n is the data size and d the input dimension of the function) over 50 repetitions and a test set of 1000 data points using a Latin Hypercube Sampling (LHS). GP: Gaussian Process with an RBF kernel. DGP Adam: DGP with 2 hidden layers with all its parameters optimized by an Adam optimizer. DGP Nat: DGP with 2 hidden layers with all the variational parameters optimized by a Natural gradient and the hyper-parameters by an Adam otimizer.

Function	Approach	mean RMSE	std RMSE	mean test log-likelihood	std test log-likelihood
TNK constraint	GP	0.18832	0.0305	8866.59	20166.95
	DGP Adam	0.17746	0.0482	467207	400777
	DGP Nat	**0.1659**	**0.013**	**3671**	**1766.48**
Hartmann 6d	GP	0.3010	0.0311	**566.27**	**451.798**
	DGP Adam	0.3166	0.0252	2595.70	2393.99
	DGP Nat	**0.2921**	**0.0200**	1386.12	1111.29
Trid	GP	12934	965	10912	114
	DGP Adam	11978	496.71	12690	808
	DGP Nat	**11151**	**388**	**10342**	**109**

around the current minimum limiting thereby its exploration capabilities. Hence, a combination of the natural gradient on all the variational parameters and the Adam optimizer on the hyper-parameters is used in BO & DGP for training the models enabling a better uncertainty quantification and a faster training.

4.1.2 Architecture of the DGP

The architecture of the DGP is a key question when using a DGP in BO. The configuration of the architecture of the DGP includes the number of layers, the number of units in each inner layer (*i.e.* the input dimension of the inner layers) and the number of induced variables in each layer (Z_l).

Increasing the number of these architecture parameters enables a more powerful ability of representation. However, these variables are directly related to the computa-

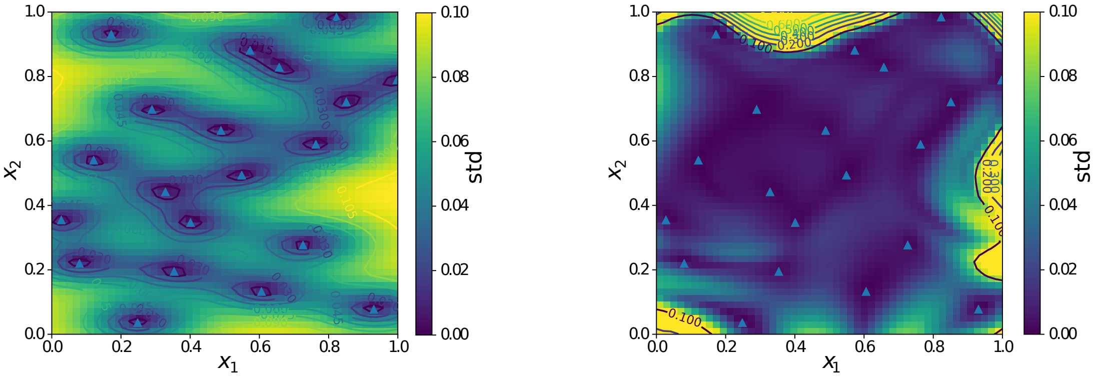


Fig. 4.8 Standard deviation on the prediction given by a 2 layer-DGP model on the TNK constraint function. The markers represent the positions of the training points. (left) standard deviation given by a model optimized using natural gradient on all the variational parameters and Adam on the deterministic parameters. (right) standard deviation given by a model optimized using ordinary stochastic gradient (Adam) for all the parameters. An underestimation of the uncertainty happens in the second approach.

tional complexity of the algorithm. Indeed, the computational complexity of a BO with DGP is given by $\mathcal{O}(j \times s \times t \times n \times (m_{[1]}^2 d_{[1]} + \ldots + m_{[i]}^2 d_{[i]} + \ldots + m_{[l]}^2 d_{[l]}))$ where j is the number of added points in BO, s the number of propagated samples, t the number of optimization steps in the DGP training, n the size of the data set, l the number of layers, $m_{[i]}$ the number of induced inputs at the layer i and $d_{[i]}$ the number of units at the layer i. The total number of parameters to optimize depends on the structure. In fact, it includes the kernel hyper-parameters of each GP of size $\sum_{i=1}^{l}(d_{[i]}+2)$ (for an ARD kernel), the induced inputs at each layer $\sum_{i=1}^{l} m_{[i]} \times d_{[i]}$, as well as the variational parameters $\sum_{i=1}^{l}(m[i]+m_{[i]}(m_{[i]}+1)/2)$. Moreover, the number of optimization steps t needed usually increases according to the number of optimized parameters.

Within the context of BO, the overhead computational cost of DGP training has to be limited compared to the exact evaluation of the objective and constraint functions (see Section 4.2 for computational times). Usually, in the early iterations of BO since few data is available there is not enough information to use complex models which are more time consuming, therefore a standard GP may be sufficient. Then, along the iterations by adding more data to the DoE, more complex and non-stationary distribution of the data can be encountered, hence, the number of layers is increased in order to enhance the power of representation of the model. In the deep learning theory, the structure of deep architecture is an active topic [Bengio, 2012; LeCun et al., 2015; Goodfellow et al., 2016] where the goal is to obtain a trade-off between generalization-error and training-error. These works highlight the fact that there is no structure of a deep architecture for all problems and rather that these structures are problem dependent. For a DGP, it is also difficult to specify its depth and width for a specific problem without using computationally expensive approaches such as cross-validation. However, the difference of DGPs from standard deep neural networks is their Bayesian formulation. This allows DGPs to avoid over-fitting even with complex structures. However, complex structures yield to a high-dimensional parameter space making the training task difficult. Hence, the trade-off in DGP must be done between the power of representation of the structure and its complexity. As discussed in Chapter 2, Section 2.2.3 a layer of DGP can be seen as an infinitely wide neural network, this highlights the power of representation of each layer of a DGP and therefore its depth is relatively less important than standard deep neural networks. In fact, in the experimentation section (Section 4.2) a two-layer DGP shows enough power of representation to represent different non-stationarity behaviors.

It is interesting to observe that the number of inducing variables is the preponderant term in the complexity of the BO with DGP. Induced inputs were first introduced in the framework of sparse GPs (Chapter 2, Section 2.2.2). By choosing a number of induced inputs m with $m << n$ and n the number of data points, the complexity of the inference becomes $O(nm^2)$ instead of $O(n^3)$. This allows computational speed ups in the training of the model. In sparse GP, increasing the number of inducing inputs increases the accuracy until reaching $m = n$ when the full GP model is recovered. In DGPs, the interpretation of the induced inputs is more complicated. Firstly, it is essential to use induced inputs to obtain the Evidence Lower Bound for the inference in DGP. Secondly, the variables $\mathbf{H}_{[i]}, i = 1, \dots, l$ are random variables and not deterministic as $\mathbf{X}$. So, it is possible to gain more precision even if $m_{[i]} > n$, since an infinite number of points is needed to define a distribution.

However, the functional composition of GPs within a DGP makes each layer an approximation of a simpler function. In Fig. 4.9, a 2-layer DGP is used to approximate the modified Xiong function with 15 induced inputs (marked by blue triangles) in each layer, the input-output of each layer and the position of the induced inputs are plotted. The intermediate layers try to deform the input space by stretching it, in order that the last layer approximates a stationary function, achieving an unparameterized mapping. Hence, the inner layers have a less complex behavior than the whole model. It is interesting to notice that in the inner layers, the induced inputs positions are overlapping, meaning that only a reduced number of induced inputs can capture the features of the inner layers, hence, allowing computational speed ups.

To adapt the number of induced variables to the training framework proposed in Section 4.1.1 for DGP within BO, the number of induced inputs is fixed along BO to the total number of data-points at the end of the algorithm. This allows the number of parameters to be constant along the BO algorithm and hence allowing the use of the previous optimal values of the model parameters at the next iteration of BO. Moreover, in the early iterations of BO, the latent variables $\mathbf{H}_{[i]}, i = 1, \ldots, l$ have an important variance, hence a higher number of induced inputs compared to the observed data-points allows a gain in precision in the beginning of BO.

4.1.3 Infill criteria

To use DGP in BO, it is essential to adapt the considered infill criteria to DGP. In fact, some infill criteria can not be used directly with DGP. For example, the popular Expected Improvement (EI) formula in Eq.(3.20) is based on the fact that the prediction is Gaussian. However, in DGP the prediction does not necessary follows a normal distribution. The EI is the expected value of $\mathcal{I}(\mathbf{x}) = \max(0, y_{min} - f(\mathbf{x}))$. Therefore, the direct approach is to use sampling techniques to approximate this expectation value (Eq.(3.19)). A computation of EI using MCMC has been previously used in [Snoek et al., 2012] for a full Bayesian treatment of the hyper-parameters. However, in the case of DGP, sampling has to be used to address the non-Gaussianity of the predictive distribution. This concerns also other infill criteria, for instance infill criteria formula used to handle constraints such as the Probability of Feasibility (PoF) Eq. (3.25) and the Expected Violation Eq. (3.23) (EV) formula are based on the Gaussian distribution of the model. Therefore, for a candidate $\mathbf{x}$ and a constraint DGP model $g(\cdot)$, to compute the PoF, sampling on the indicator function of feasibility $\mathbb{I}(g(\mathbf{x}) \leq 0)$ is performed, and to compute the EV, sampling on the violation $\mathcal{V}(\mathbf{x}) = \max(0, g(\mathbf{x}))$ is performed.

In some cases sampling can be avoided, in fact, as observed in Fig. 4.9, the inner layers are often simple functions, almost linear, with a last layer that approximates a deformed stationary function. This allows the prediction from the composition of GPs to be reasonably considered as Gaussian most of the time (see Fig. 4.10). Hence, to predict using DGPs, a Gaussian approximation can be made after verification of its Gaussian behavior, in order to directly use the analytical formula of the infill criteria used for GPs.

Infill criteria such as EI are highly multi-modal, especially in high-dimensional problems. For this reason, an evolutionary algorithm such as a differential evolution algorithm [Price et al., 2006] can be preferred for the optimization of the infill criterion. The DGP allows parallel prediction which makes it possible to evaluate the infill

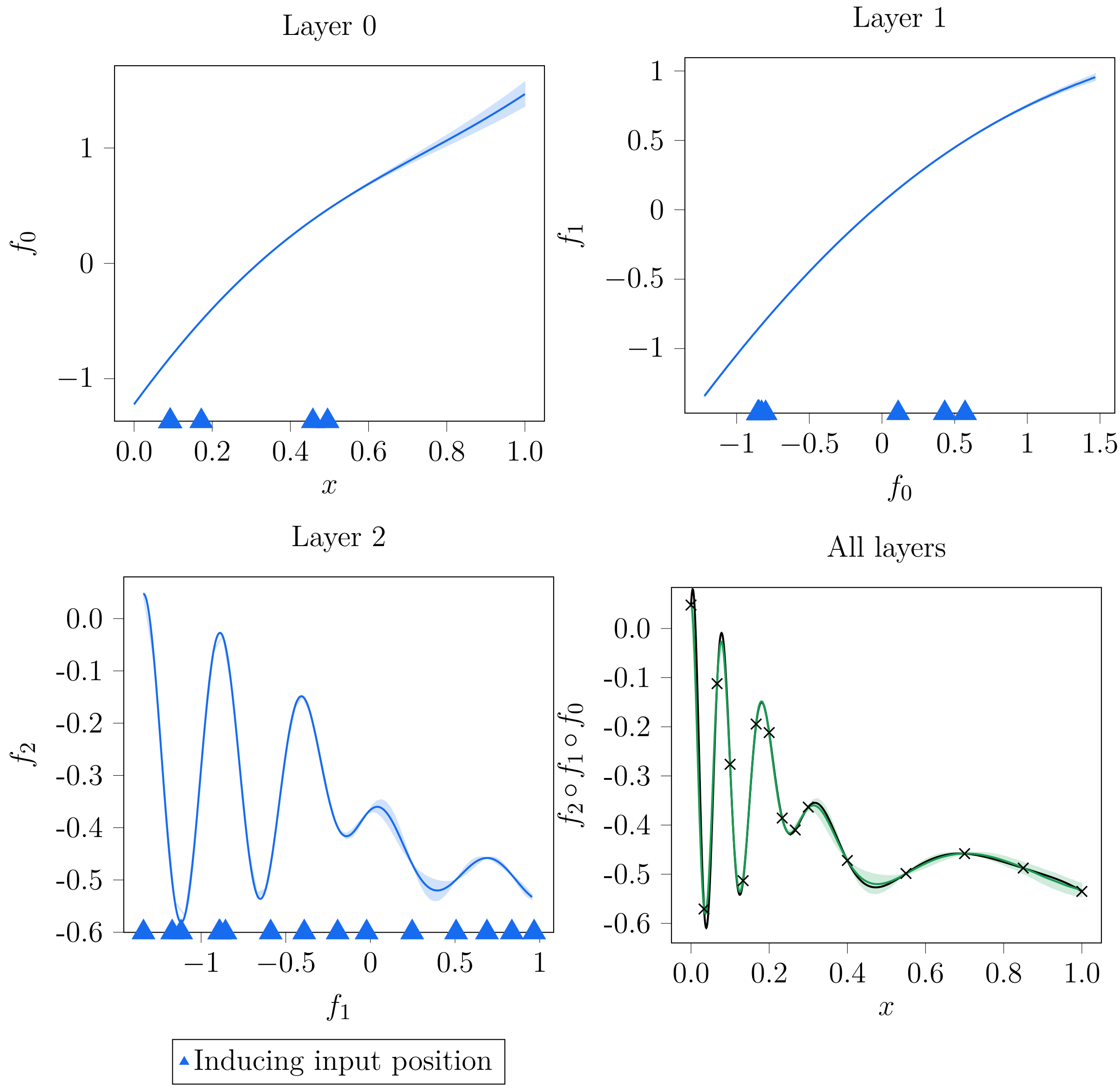


Fig. 4.9 The input-output signal of each layer of a 2 layer-DGP used to approximate the modified Xiong function. DGPs allow unparameterized non linear mapping. The intermediate layers stretch the input space, in order that the last layer approximates a stationary function. The combination of the inner layers gives a non-stationary function. The markers represent the induced input locations. (top left) first layer, (top right) second layer, (lower left) output layer, (lower right) DGP prediction.

criteria for all the population of the optimization algorithm simultaneously. The result obtained using the evolutionary algorithm can then be optimized by a local optimizer. This hybridization has been preferred in this study to the use of multiple local searches whose number increases exponentially with the dimension of the problem.

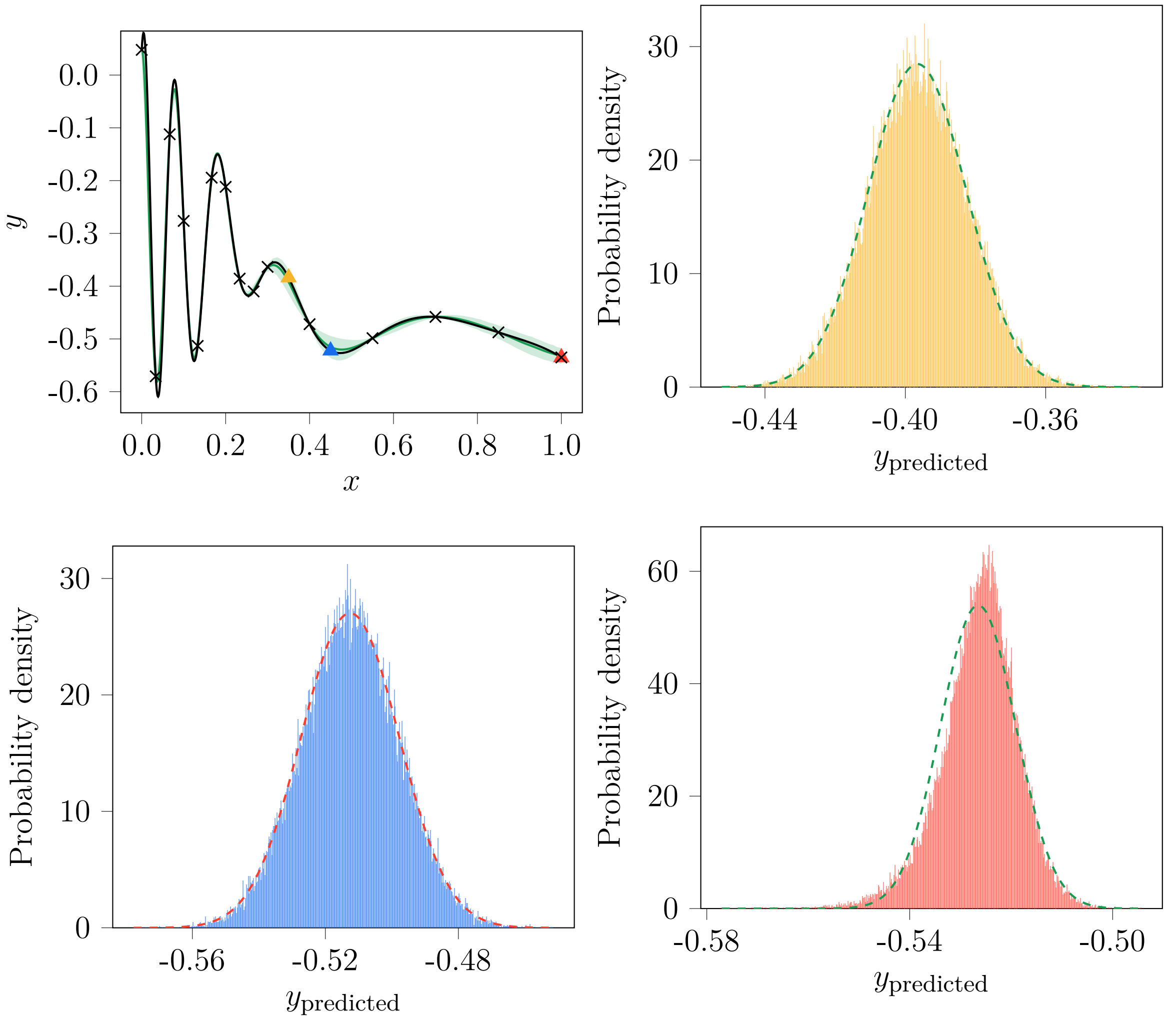


Fig. 4.10 50 000 samples drawn on the value of the prediction of a 2-layer DGP model in three different locations (triangles in top left figures). In the top right and lower left figures, the predictions are almost Gaussian. In the lower right figure, the distribution of the prediction is slightly asymmetric, but it is still well approximated by a Gaussian distribution.

4.1.4 Synthesis of DGP adaptations proposed in the context of BO

To summarize the proposed adaptations of DGP to BO, Algorithm 2 describes the steps previously discussed. The Expected Improvement is used as the infill criterion, but other infill criteria may be used. If approximation of the DGP prediction by a Gaussian is not valid, sampling techniques are used to compute the infill criterion. Some empirical rules can be used to determine the number of points in the initial DoE and the number of added points during the BO algorithm depending on the dimension of the problem d (for the experimentations in Section 4.2, for all the problems an initial DoE of size $5 \times d$ is considered and $10 \times d$ points are added in the BO process). The size of the induced variables is fixed along all the BO iterations to the total number of points at the end of the BO. This allows the models to keep the same number of parameters along the iterations, making it possible to initialize them from the previous models. Moreover, as discussed previously, setting the number of induced variables to a number larger than the number of points in the training data set for DGP may allow a better representation. The model is trained using the described loop of a natural gradient step for the variational parameters of all layers and an Adam optimization step for the deterministic parameters. The model at a given iteration of the BO process is updated from the model optimal parameter values at the previous BO iteration for a certain number of consecutive iterations allowing speed ups in the DGP training, and then initialized from scratch every n_{update} iterations to avoid being tricked in some bad local minima. These different adaptations for the BO and DGP coupling are summarized in Fig. 4.11.

In Algorithm 2, the unconstrained optimization problem case is considered. However, the generalization to the constrained case is straightforward, since it comes back to create DGP models also for the constraints and to use sampling for a constrained infill criterion as the Probability of Feasibility or the Expected Violation.

Algorithm 2: Unconstrained BO with DGP algorithm

```
Require: Expensive black-box objective function of dimension $d$ to optimize, $f^{exact}$
Require: Number of initial points $n$ in data set.
Require: Number of total added points $n_{add}$.
Require: Number of layers $l$ (default $l = 2$).
Require: Number of loop iterations in the training of the DGP model $iter$.
Require: Number of consecutive DGP updates using the previous model optimal values $n_{update}$.
$\mathbf{X}_0 \leftarrow LHS(d,n)$ (or another design of experiments method)
$\mathbf{y}_0 \leftarrow f^{exact}(\mathbf{X}_0)$ (evaluate)
$m \leftarrow n + n_{add}$ (set the number of induced variables to the final number of points)
$t \leftarrow 0$
$model_0 \leftarrow$ DGP model training Algorithm 1$(\mathbf{X}_0, \mathbf{y}_0, m, l, iter)$ (optimize model from scratch)
while $t \leq n_{add}$ do
    $t \leftarrow t + 1$
    $\mathbf{x}^{(t)} \leftarrow argmax(EI_{model_{t-1}}(\mathbf{x}))$ (use sampling to estimate the $EI$, in the constrained case the PoF or the EV are also estimated)
    $y^{(t)} \leftarrow f^{exact}(\mathbf{x}^{(t)})$ (evaluate)
    $\mathbf{X}_t \leftarrow \begin{bmatrix} \mathbf{X}_{t-1} \\ \mathbf{x}^{(t)} \end{bmatrix}$ (add a row to the matrix)
    $\mathbf{y}_t \leftarrow \begin{bmatrix} \mathbf{y}_{t-1} \\ y^{(t)} \end{bmatrix}$ (add an element to the vector)
    if $t \not\equiv 0 \pmod{n_{update}}$ then
        $model_t \leftarrow$ DGP model training Algorithm 1$(\mathbf{X}_t, \mathbf{y}_t, m, l, iter, model_{t-1})$ (optimize model using the optimal parameter values of the previous model as initialization)
    else
        $model_t \leftarrow$ DGP model training Algorithm 1$(\mathbf{X}_t, \mathbf{y}_t, m, l, iter)$ (optimize model from scratch)
    end
end
return $\boldsymbol{X}_t, \boldsymbol{y}_t$
```

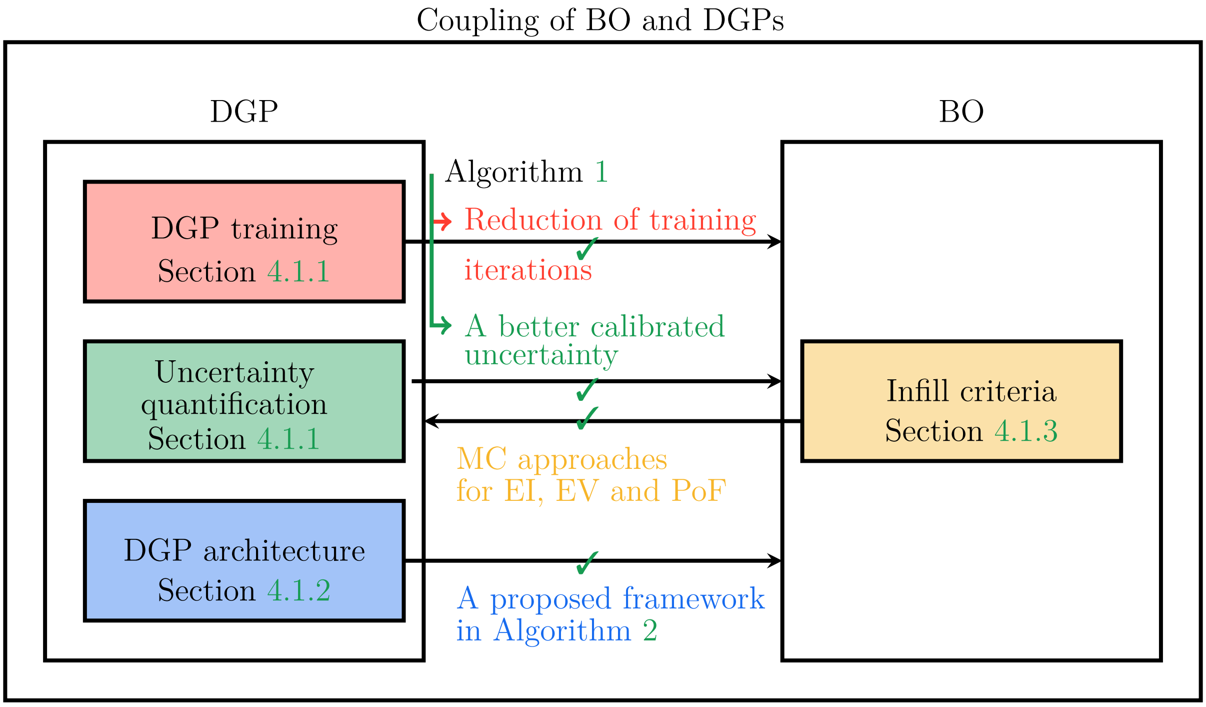

Coupling of BO and DGPs
DGP
BO
DGP training
Section 4.1.1
Uncertainty quantification
Section 4.1.1
DGP architecture
Section 4.1.2
Algorithm 1
Reduction of training iterations
A better calibrated uncertainty
✓
✓
✓
✓
MC approaches for EI, EV and PoF
A proposed framework in Algorithm 2
Infill criteria
Section 4.1.3

Fig. 4.11 Proposed adaptation for the coupling of BO and DGP for non-stationary problems. It includes a DGP training approach to accelerate the training and to obtain a well-calibrated predictive uncertainty quantification, infill criteria estimated by sampling instead of exact analytic equations, and a default 2 layer-DGP architecture.

4.2 Experimentations

In this section, experimentations are carried out in order to evaluate the performance of BO with DGPs. Firstly, analytical test functions are considered to compare BO & DGP with repetitions to evaluate the robustness to the initial DoE. Then, the most competitive algorithms are applied to two aerospace vehicle design test problems.

4.2.1 Analytical test problems

Experimentations on three different analytical optimization problems (Appendix C) have been carried out to assess the performance of the proposed framework of BO & DGP detailed in the previous section. The first test case is a 2-d constrained problem is used to compare different architectures of DGPs in the BO process in order to highlight the trade-off between model complexity and the time budget available. The other two optimization problems are used to compare BO & DGP to state of the art BO algorithms with models that also use a non-linear mapping to handle non-stationarity (NS kriging [Xiong et al., 2007], APNS [Toal and Keane, 2012], Bayesian NLM [Snoek et al., 2014]) in two different scenarios. The first one (Trid $10d$) is when a regular BO with GP algorithm has issues to reach the optimum and the second one (Hartmann-6d) is when a regular BO with GP algorithm is able to reach the optimum. This allows to evaluate the robustness of the algorithm on the characteristics of the problem in hand, and its application to a problem with no assumption about its stationarity. The same BO loop is used for the different models experimented. The results of a random optimization (random grid search) with the same number of evaluations as BO are also presented for the analytical functions to highlight the difficulty of the problems in the context of a limited budget of evaluations. Details on the numerical setup are presented in Appendix D.

Test case 1: 2-d constrained problem

The function to optimize is a simple two dimensional quadratic function. The constraint is non-stationary and is feasible when equal to zero. An important discontinuity between the feasible and non feasible regions breaks the smoothness of the constraint (Fig. 4.12). Therefore, the problem is challenging for standard GP, since the optimal region is exactly at the boundary of the discontinuity, requiring an accurate modeling of the non-stationarity. This type of functions characterized by a flat region broken with a discontinuity are common as constraints in engineering design problems due to abrupt transition from feasible to unfeasible regions of the input space.

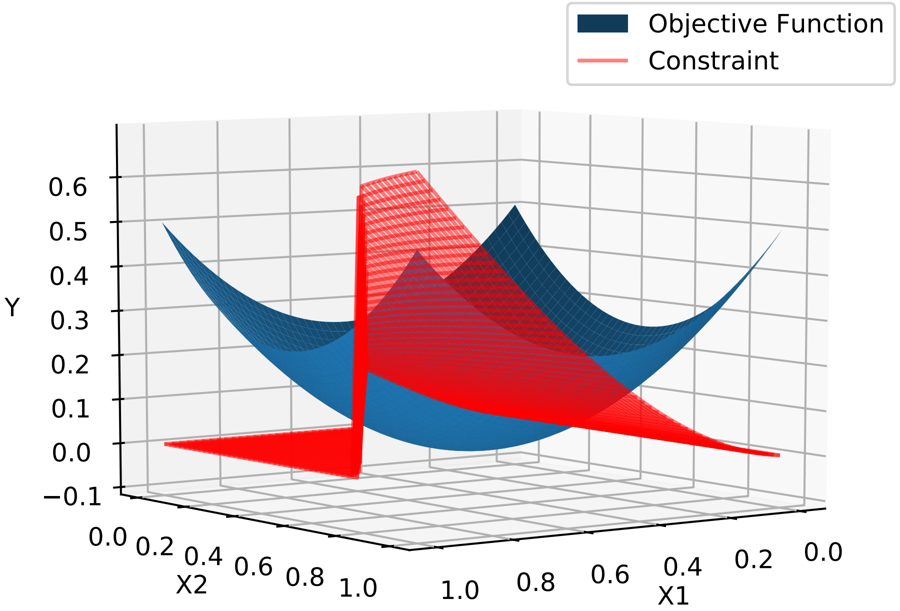


Fig. 4.12 Objective and constraint functions 2d problem. The constraint is non-stationary. An important discontinuity separates between the feasible and unfeasible space, making it difficult for a classic GP to model.

A DoE of 10 initial data points is initialized using a Latin Hypercube Sampling. Then, 20 points are added using the Expected Violation criterion (EV) to handle the constraint. A standard Gaussian Process with a RBF kernel is used to approximate the objective function. The DGPs are considered with a RBF kernel in each layer and are trained using 5000 optimization steps of Algorithm 1. To assess the robustness of the BO algorithms, 50 repetitions are performed from different randomized LHS DoEs.

The convergence plots of the BO algorithms with GP, DGP 2, 3, 4 and 5 layers are displayed in Fig. 4.13. As expected, the BO with GP is not well-suited for this problem. At the end of the algorithm, the median is still far from the actual minimum and there is an important variation. This is due to the fact that the GP can not capture the discontinuity and the feasible tray region of the constraint and considers a large area as unfeasible (Fig. 4.15). However, BO with DGP accurately capture the frontier between the feasible and unfeasible regions (Fig. 4.16), which makes it able to give efficient results with a median at the end of the optimization algorithms near

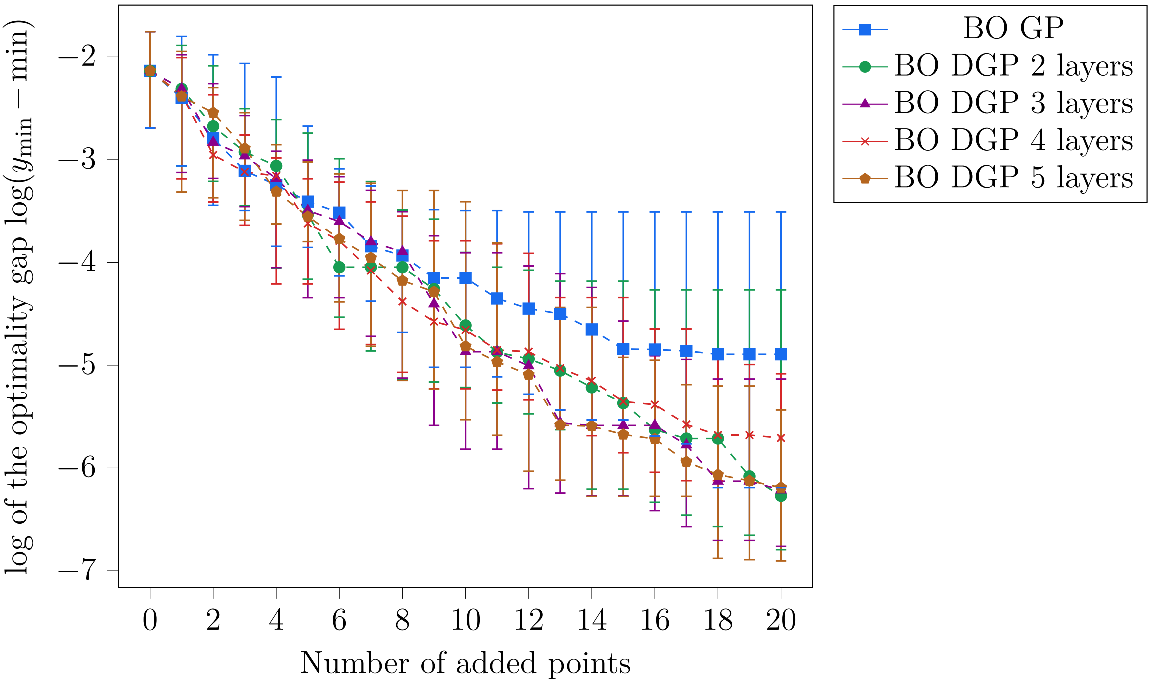


Fig. 4.13 Plot of convergence of BO using different architectures of DGPs with 5000 training steps for 50 different initial DoE and a standard GP. The markers indicate the median of the minimum obtained while the error-bars indicate the first and third quartiles.

to the actual minimum and better robustness to the initial DoEs. Furthermore, the 3-layer DGP provides the best results as can be analyzed from the mean and standard deviation of best found points given in Table 4.2. Increasing the number of layers deteriorates the quality of the results. This is explained by the fact that 5000 steps in the training of DGPs with more than three layers in this case is insufficient. In fact, in this configuration, the number of parameters to optimize increases by 274 parameters by adding another layer *i.e.*, while for the 2-layer DGP with a DoE of 20 points the number of parameters is 822, for the 3-layer DGP it is 1096, for the 4-layer DGP it is 1370, and for the 5-layer DGP it is 1644. This makes it necessary to increase the number of optimization steps in the training of deeper models, since the parameter space increases in dimensionality. However, increasing the number of layers and the number of steps induces additional computational time (Fig. 4.14) which quickly becomes a large burden for high dimensional problems. Consequently, for the remaining test cases only a DGP with two layers is considered.

Table 4.2 Performance of BO (values of the minimum found) with standard GP and different DGP configurations on the constrained 2d problem. 50 repetitions are performed.

Algorithm	**average minimum obtained**	**standard deviation on the minimum obtained**	**average optimality gap**
BO & GP	0.09356	0.0605	0.03336
BO & DGP 2 L	0.08468	0.059793	0.02448
BO & DGP 3 L	**0.073918**	**0.04293**	**0.01371**
BO & DGP 4 L	0.08066	0.05073	0.02046
BO & DGP 5 L	0.08204	0.05707	0.02184
Random optimization	0.26320	0.10808	0.20120
Global minimum	0.0602	-	-

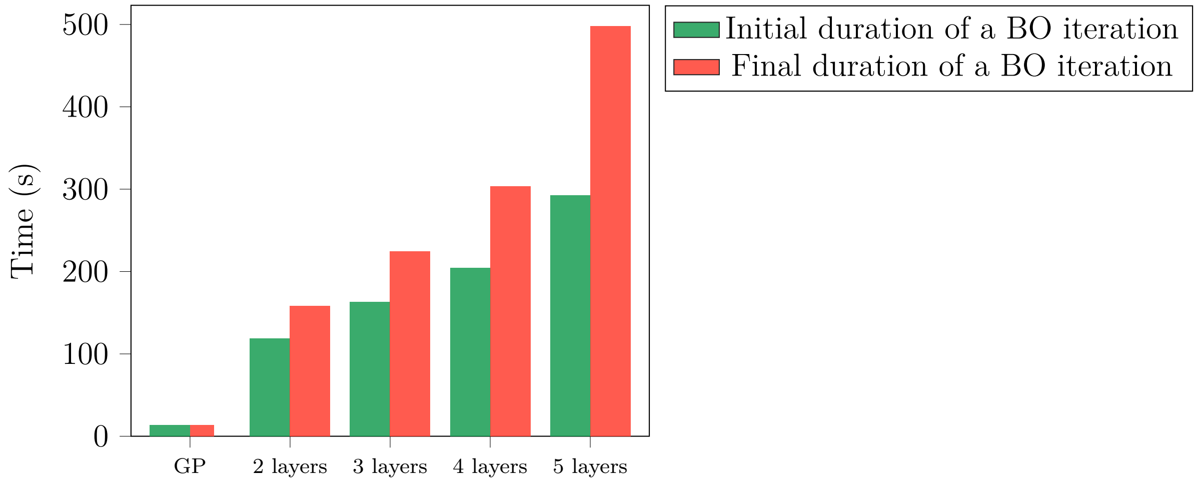


Fig. 4.14 Average time in one iteration of BO according to the number of layers in DGP at the start of the algorithm (data set of 10 points) and at the exhaustion of the evaluation budget (data set of 30 points). A GP is faster than the other DGP architectures due to its fast training, however, it has poor modeling performance in non-stationary problems.

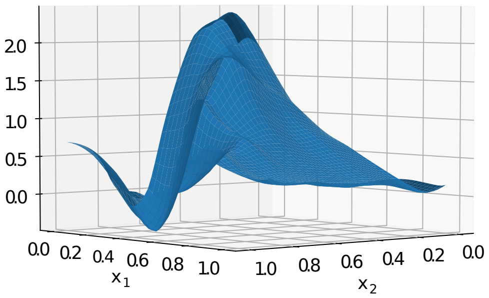


Fig. 4.15 GP approximation of the constraint with a DoE of 40 points. The GP can not model correctly the tray region and considers a large feasible area as unfeasible.

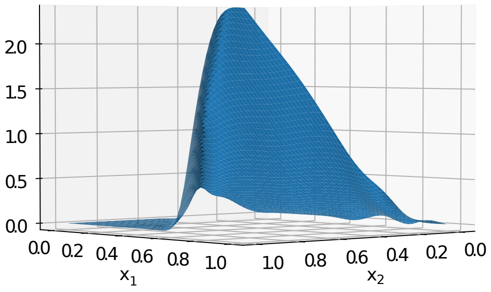


Fig. 4.16 DGP 2 layers approximation of the constraint with a DoE of 40 points. The DGP is able to model correctly the tray feasible region, and the discontinuity is modeled accurately.

Test case 2: Trid function

The Trid function is considered in 10 dimensions Eq.(C.3) in Appendix C. It is an unconstrained optimization problem. The range of variation of the 10d Trid function values is large. It varies from 10^5 to its global minimum $f(\mathbf{x}^*) = -210$ (Fig. C.3 in Appendix C). This large variation range makes it difficult for BO with stationary GP to find the global minimum.

The results of BO with a DGP of 2 hidden layers are compared to the Bayesian input warping used by Snoek *et al.* (Bayesian NLM) and to the results found in [Toal and Keane, 2012] (NS kriging and APNS with the tuning of the algorithms involved by the authors of the paper) over 50 different repetitions with different initial DoEs (Table 4.3). The initial DoEs are initialized with a Latin Hypercube Sampling with 50 initial points, and 100 points are added during the BO using the EI criterion.

The minimum given by BO & GP, NS kriging (non-stationary kriging) and APNS (Adaptive Partial Non-Stationary kriging) for this problem are not close to the global minimum. Moreover, there is a high variation in the obtained minimum values, showing the difficulty of these approaches to handle this optimization problem. BO & Bayesian NLM and BO & DGP provide the best results. A slight advantage for BO & DGP is observed compared to BO & Bayesian NLM with an average minimum obtained -206.739 which is very close to the actual global minimum -210 with a standard deviation of 1.5521, hence illustrating the robustness of the proposed approach.

Table 4.3 Performance of BO (values of the minimum found) with standard GP, non-stationary kriging with two knots (NS kriging), adaptive partial non-stationary kriging (APNS), Deep Gaussian Processes with two hidden layers (DGP) on the Trid function.

Algorithm	**average minimum obtained**	**standard deviation on the minimum obtained**	**average optimality gap**
BO & GP	-20.730	75.654	189.27
BO & NS kriging	-57.727	59.920	152.273
BO & APNS	-49.112	62.746	160.888
BO & Bayesian NLM	-203.71	30.79	6.29
BO & DGP	**-206.739**	**1.5521**	**3.261**
Random optimization	7086.5	1747.7	7296.5
Global minimum	-210	-	-

Test case 3: Hartmann-6d function

The Hartmann-6d is a 6d function Eq.(C.4) in Appendix C. It is an unconstrained optimization problem. The Hartmann-6d is smooth and does not show non-stationary behavior (Fig. C.4). The interest of this function is that BO coupled with some non-stationary approaches can not reach its global minima while BO & classic GP presents good performance on it [Toal and Keane, 2012]. Hence, using BO with DGP on this function allows to demonstrate the robustness of this non-stationary BO algorithm on stationary functions. This is representative of real industrial cases when there is no information about the stationarity of the problem at hand.

The results of BO with a DGP of 2 hidden layers are compared to the Bayesian input warping used by Snoek *et al.* (Bayesian NLM) and to the results found in [Toal and Keane, 2012] (NS kriging and APNS with the tuning of the algorithms involved by the authors of this paper) over 50 different repetitions with different initial DoEs (Table 4.4). The initial DoEs are initialized using a Latin Hypercube Sampling with 30 initial points and 60 points are added during the BO process using the EI criterion.

The results obtained by BO & NS kriging and APNS are relatively far from the global optimum and show larger variation of found optimum. The stationary GP gives better and more robust results, since it is adapted to the stationary behavior of the Hartmann function. However, the minimum obtained by BO & DGP is closer to the global optimum and the optimization is more robust to the initial DoE than standard

Table 4.4 Performance of BO (values of the minimum found) with standard GP, non-stationary kriging with two knots (NS kriging), adaptive partial non-stationary kriging (APNS), and Deep Gaussian Processes with two hidden layers (DGP) on the Hartmann 6d function.

Algorithm	**average minimum obtained**	**standard deviation on the minimum obtained**	**average optimality gap**
BO & GP	-3.148	0.275	0.174
BO & NS kriging	-2.818	0.570	0.504
BO & APNS	-3.051	0.415	0.271
BO & Bayesian NLM	-3.1713	0.3256	0.1507
BO & DGP	**-3.250**	**0.098**	**0.072**
Random optimization	-1.9760	0.4268	1.3459
Global minimum	-3.322	-	-

GP even if the function is stationary. Moreover, BO & DGP presents also better results compared to Bayesian NLM. This shows the interest of using the BO & DGP even for functions without any information on their stationary behavior unlike BO & NS kriging and APNS that are ill-suited for stationary functions.

4.2.2 Application to industrial test case: design of aerospace vehicles

In this subsection, experimentations on two industrial test cases are presented. The first test case is a $4d$ booster optimization design problem. The complexity is increased in the second test case by considering the optimization of a three stage sounding rocket with 15 design variables. BO with a two layer DGP is applied and compared to GP using the same kernels (RBF) to highlight the increase of the representation accuracy by composition of the same GPs, and based on the experimentations done on the analytical test cases, the Bayesian NLM which gave competing results to DGP is chosen for comparison. BO with GPs is also applied, it corresponds to the reference approach usually applied in these problems.

Engineering test case 1: optimization of a solid propellant booster

To confirm the interest of the application of BO with DGP, an aerospace vehicle design optimization problem is considered. It consists of the maximization of the velocity increment (ΔV) of a solid-propellant booster. It is a representative physical problem for solid booster design with simulation models fast enough to compute the exact minimum to compare and illustrate the efficiency of the proposed algorithm.

The optimization of ΔV for a solid propellant booster is considered (Fig. 4.17). Four design variables are involved:

- Propellant mass: $5 \text{ t} < m_{\text{prop}} < 15 \text{ t}$
- Combustion chamber pressure: $5 \text{ bar} < p_c < 100 \text{ bar}$
- Throat nozzle diameter: $0.2 \text{ m} < d_c < 1 \text{ m}$
- Nozzle exit diameter: $0.5 \text{ m} < d_s < 1.2 \text{ m}$

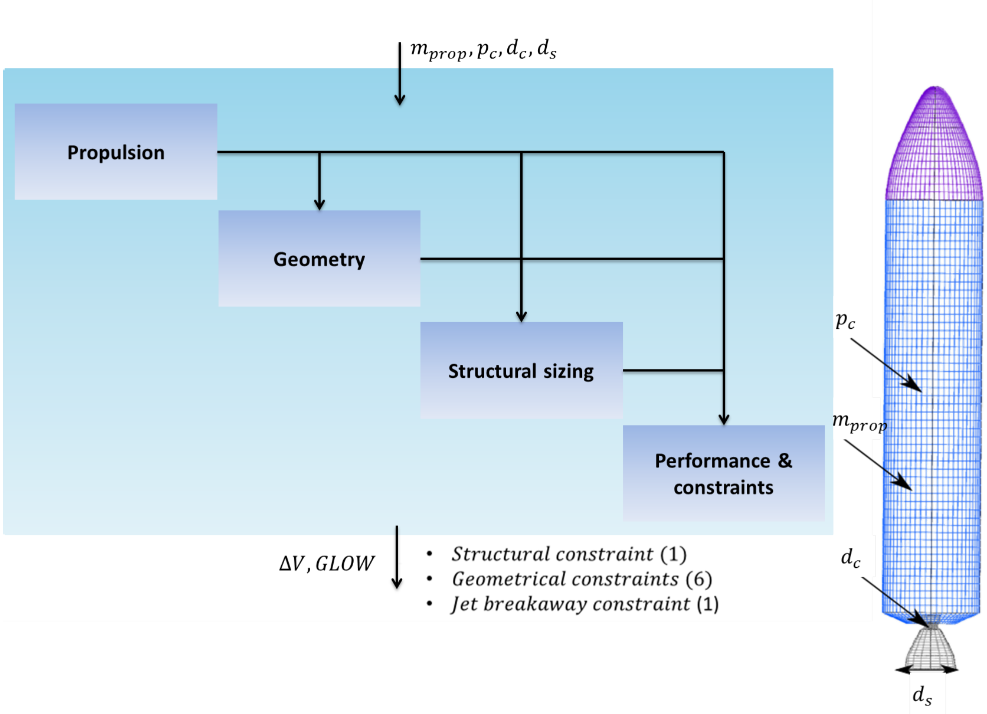
Nine constraints are also considered including a structural one limiting the combustion pressure according to the motor case, 6 geometrical constraints on the internal vehicle layout for the propellant and the nozzle, a jet breakaway constraint concerning the nozzle throat diameter and the nozzle exit diameter, and a constraint on the maximal Gross Lift-Off Weight (GLOW) allowed. The optimization problem may be written as:

$$\begin{array}{ll}
\text{Minimize:} & -\Delta V(\mathbf{x}) \\
\text{w.r.t:} & \mathbf{x} = [m_{prop}, p_c, d_c, d_s] \\
\text{s.t:} & \begin{cases} \text{1 structural constraint} \\ \text{6 geometrical constraints} \\ \text{1 jet breakaway constraint} \\ \text{1 constraint on the maximal GLOW allowed} \end{cases}
\end{array}$$

This problem involves non-stationarity behaviors due to some constraints. In fact, the constraints may have a different behavior in the feasible and unfeasible regions. Moreover, the objective function which is the velocity increment may have a tray region when it is equal to zero, due to insufficient initial thrust (Fig. 4.18).

The initial DoE are set using a Latin Hypercube Sampling of 30 points and 50 points are added with BO using EI for the objective function and EV for the constraints. To assess the robustness of the results, 10 repetitions are performed.

The plots of convergence of the BO algorithms are displayed in Fig. 4.19. After adding 50 points, all the algorithms reach the global minimum. However, BO with



m_{prop}, p_c, d_c, d_s
Propulsion
Geometry
Structural sizing
Performance & constraints
$\Delta V, GLOW$
• *Structural constraint* (1)
• *Geometrical constraints* (6)
• *Jet breakaway constraint* (1)
p_c
m_{prop}
d_c
d_s

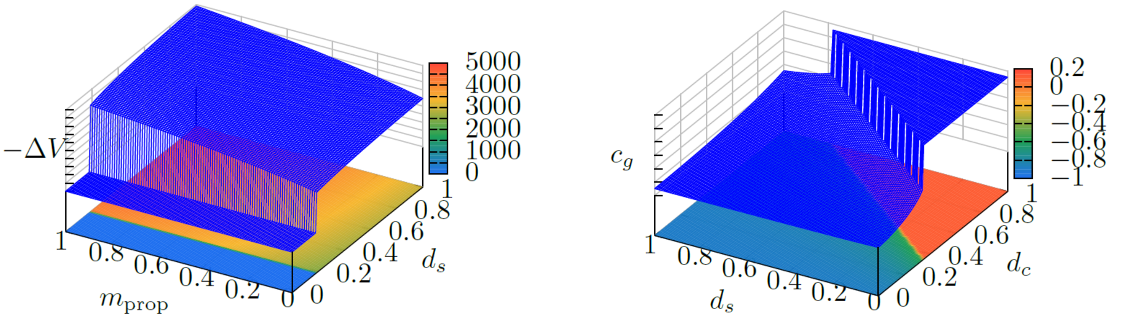
Fig. 4.17 Optimization problem of a solid-propellant booster engine. The formulation of the problem involves different disciplines (propulsion, geometry, structural sizing and performance). The problem considers the maximization of the velocity increment subject to 9 constraints.

DGP presents faster convergence than the competing algorithms. BO with DGP shows robust results near the global optimum $4738 m/s$ after only 6 iterations, while the BO with GP is not stabilized until 24 iterations (Table 4.5). BO with Bayesian non linear mapping (NLM) gives better results than BO with GP but it is still slower than BO with DGP in the first iterations of BO.

Table 4.5 Performance of the algorithms after 12 added points, after 24 added points and after 50 added points.

	After 6 added points		After 24 added points		After 50 added points	
Algorithm	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
BO & GP	-4543	145	-4709	41.33	-4725	10.63
BO & NLM	-4624	92.80	**-4721**	**17.53**	-4734	8.77
BO & DGP	**-4670**	**74.53**	-4718	22.53	**-4736**	**7.49**

The convergence speed is important in case of expensive black-box functions. Indeed, one evaluation of the objective function or the constraints can cost multiple hours, even multiple days. Hence, BO with DGP is interesting even for problems where BO



A sectional view of the velocity increment (m/s) according to the exit diameter of the nozzle d_s and the Propellant mass m_{prop}.

A sectional view of a geometric constraint (normalized values) according to the diameters of the nozzle d_c and d_s.

Fig. 4.18 Sectional views of the non-stationary behaviors of some functions involved in the booster problem

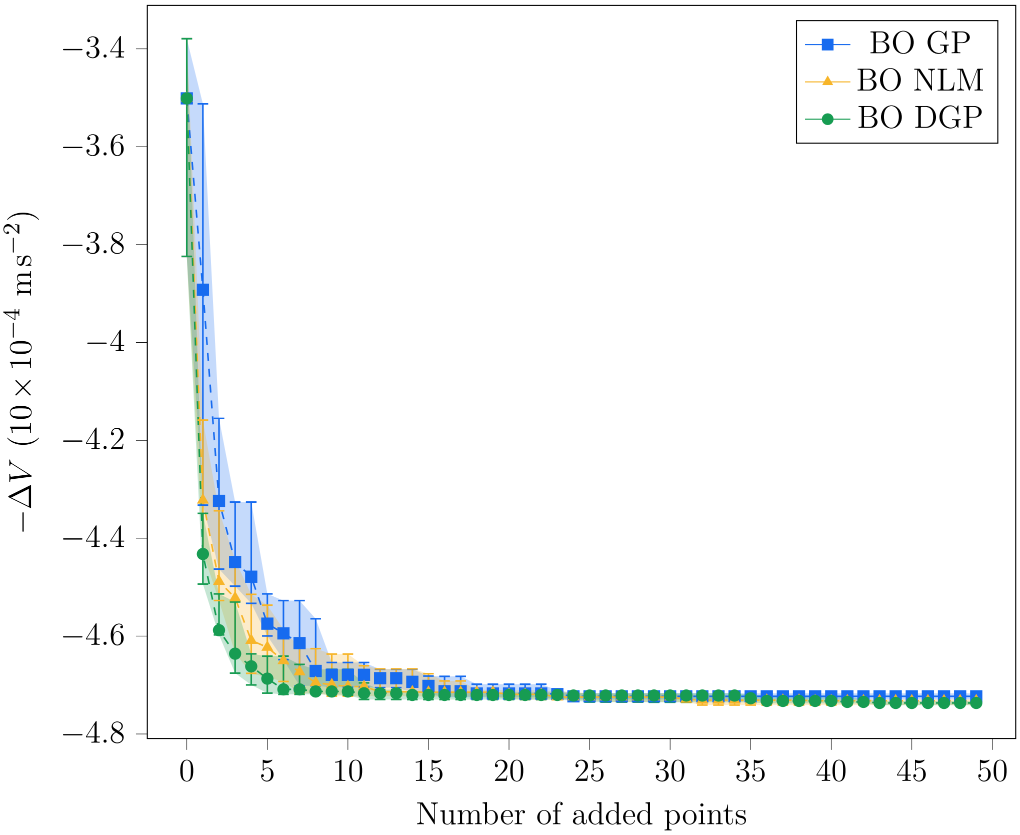


Fig. 4.19 Convergence curve of $-\Delta V$ of BO with GP, BO with Bayesian NLM and BO with a 2-layer DGP. BO with DGP gives the better result in term of speed of convergence.

with GP can reach the global minimum, due to its speed of convergence which can reduce drastically the number of evaluations needed to converge.

Engineering test case 2: three stage sounding rocket

The last test case of the study is the optimization of the design of a three stage sounding rocket with 15 design variables. The goal of the design problem is to find the optimal architecture of the rocket able to maximize the final altitude $h_{\max}$ that can be reached by the rocket after the propelled phase to release the payload experiments given a constraint on the $GLOW < 3$ t. The first stage of this vehicle is a solid propellant one whereas the second and third stages use liquid propellant (LOx/RP1, Liquid Oxygen and Rocket Propellant 1).

The performance of the launch vehicle are estimated through the use of multidisciplinary design process composed of trajectory, structure, aerodynamics and liquid and solid propulsions. The design process is implemented using openMDAO [Gray et al., 2019]. The N2 Chart of the overall design problem is represented in Figure 4.20.

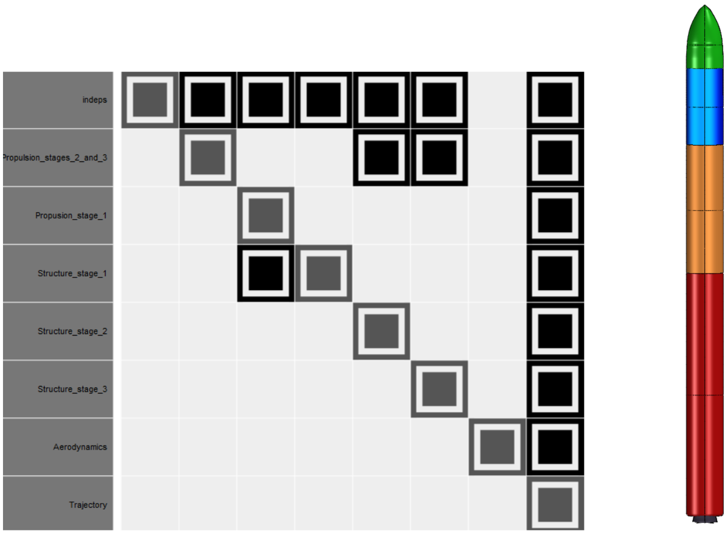


Fig. 4.20 N2 Chart of the three stage sounding rocket design process (left) and illustration of the rocket with in red the first stage, in orange the second stage, in blue the third stage and in green the fairing (right)

The dimensionality of this test case has been increased with respect to the previous test cases in order to assess the performance of BO & DGP in complex test case. The optimization problem involves 15 design variables and is subjected to 12 constraints. The design variables are the following :

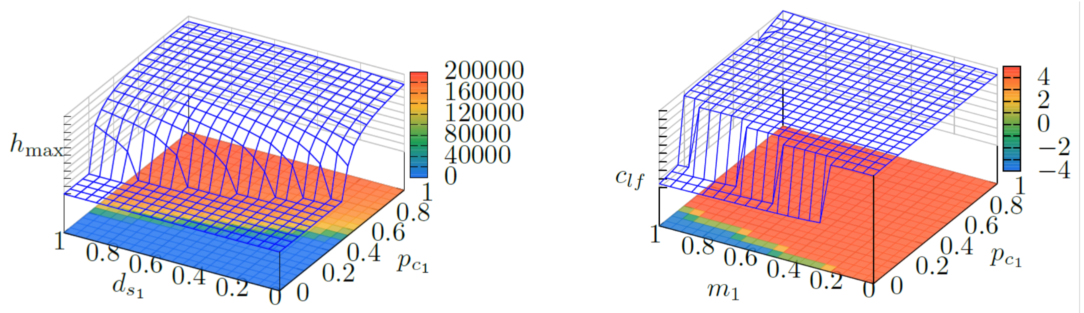
- the diameters of the different stages: $0.7\text{m} \leq d_1 \leq 1.1\text{m}$, $0.5\text{m} \leq d_2 \leq 0.8\text{m}$, $0.5\text{m} \leq d_3 \leq 0.8\text{m}$,
- the propellant masses of the different stages: $0.5 \text{ t} \leq m_1 \leq 4 \text{ t}$, $0.4 \text{ t} \leq m_2 \leq 1.2 \text{ t}$, $0.2 \text{ t} \leq m_3 \leq 0.5 \text{ t}$,
- the chamber pressures of the different stage engines: $25\text{bar} \leq p_{c_1} \leq 50\text{bar}$, $5\text{bar} \leq p_{c_2} \leq 15\text{bar}$, $5\text{bar} \leq p_{c_3} \leq 15\text{bar}$,
- the throat and exit nozzle diameters of the first stage: $0.1\text{m} \leq d_{c_1} \leq 0.3\text{m}$, $0.5\text{m} \leq d_{s_1} \leq 0.9\text{m}$,
- the mass flow rates of the stages 2 and 3: $10\text{kg}\,\text{s}^{-1} \leq q_2 \leq 30\text{kg}\,\text{s}^{-1}$, $5\text{kg}\,\text{s}^{-1} \leq q_3 \leq 20\text{kg}\,\text{s}^{-1}$,
- the oxidizer to fuel ratio of the stages 2 and 3: $3.2 \leq OF_2 \leq 4$, $3.2 \leq OF_3 \leq 4$.

The inequality constraints are relative to the integrity of the first stage (structural and geometrical constraints about the solid propellant stage), the maximal axial load factor that can be endured by the three different stages and the maximal GLOW allowed. The optimization problem may be formulated as follows:

$$\begin{array}{ll}
\text{Minimize:} & -h_{\max}(\mathbf{x}) \\
\text{w.r.t:} & \mathbf{x} = [d_1, d_2, d_3, m_1, m_2, m_3, p_{c_1}, p_{c_2}, p_{c_3}, d_{c_1}, d_{s_1}, q_2, q_3, OF_2, OF_3] \\
\text{s.t:} & \left\{ \begin{array}{l}
\text{1 structural constraint} \\
\text{8 constraints on the integrity of the first stage} \\
\text{3 constraints on the maximal axial load factor} \\
\text{1 constraint on the maximal GLOW allowed}
\end{array} \right.
\end{array}$$

The objective and constraints show non-stationary behaviors according to different variables as illustrated in Fig. 4.21. In fact, there is an abrupt change between the feasible and unfeasible regions of several constraints. Furthermore, the objective function which is the maximum altitude has a tray region equal to zero when the sounding rocket can not lift off and go up abruptly once there is enough initial thrust compared to its GLOW.

Since in a 15 dimensional design space the optimization of the EI can be problematic, WB2S criterion [Bartoli et al., 2019] is preferred in this test case and the EV is used for the constraints. The initial DoE are set using a Latin Hypercube Sampling of 75 points and 80 points are added. The functions involved in this problem are computationally expensive, hence, only 5 repetitions are performed to assess the robustness.



A sectional view of the final altitude $h_{\max}$ according to the exit nozzle diameter d_{s_1} and to the chamber pressure p_{c_1} of the first stage.

A sectional view of the constraint of the maximal axial load factor of the three stages according to the combustion pressure p_{c_1} and the propellant mass m_1 of the first stage.

Fig. 4.21 Sectional views of some non-stationary behaviors involved in the 3 stages sound rocket problem

The plots of convergence of the BO algorithms are displayed in Fig. 4.22. The optimization problem is very constrained as can be concluded from the initial DoEs where there is no feasible solution. The challenge in the early iterations of the BO algorithms is to find feasible solutions, which makes the accurate modeling of the constraints extremely determinant. At this level, BO with DGP does better than the two other algorithms. In fact, BO with DGP obtains a feasible solution after a maximum of 18 added points for the different repetitions, however, it takes for BO with NLM and BO with GP respectively 29 and 34 iterations to reach the feasible design space. After the exhaustion of the evaluation budget (80 added points) there is a notable difference between the three different algorithms. In fact, BO with DGP dominates completely BO with GP (worst final optimal value obtained by BO with DGP is better than the best final optimal value obtained by BO with GP). Moreover, BO with DGP gives a better average on the minimum obtained and is more robust than BO with NLM as can be seen when analyzing the standard deviation of the results (Table 4.6). Thus, BO with DGP provides better results compared with the other algorithms in this test problem where the design space is large 15d and under strong constraints.

The evolution of the altitude, the mass, the load factor and the velocity according to time for the optimal sounding rocket design obtained by BO with DGP is given in Figure 4.23.

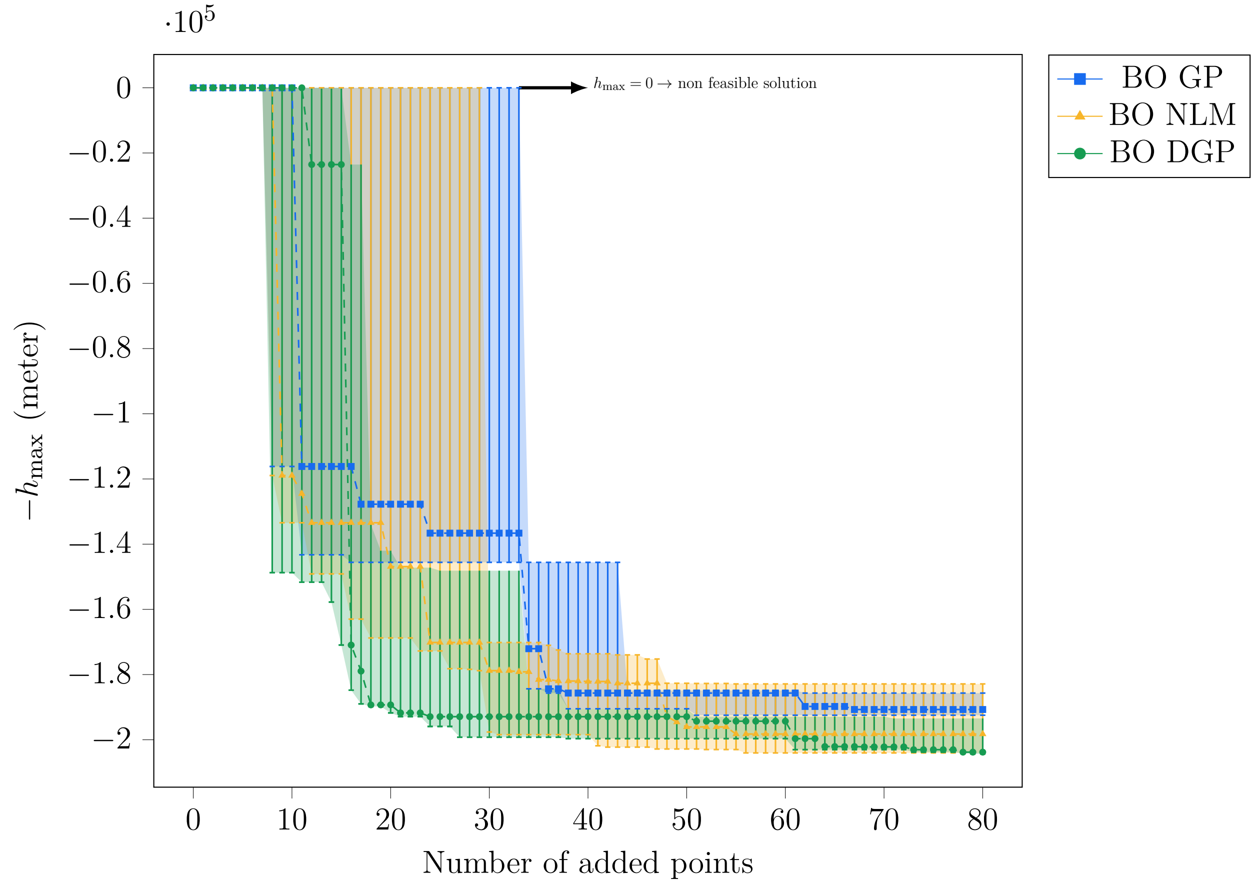


Fig. 4.22 Convergence curve of the negative altitude $-h_{\max}$ of BO with GP, BO with Bayesian NLM and BO with a 2 layers DGP. BO with DGP gives the better result in term of speed of convergence and dispersion of the results

Table 4.6 Comparison of the performance of BO with GPs, Bayesian NLM and DGP with 2 layers in terms of the speed to reach the feasibility design space and of the quality of the optimal value obtained after adding 80 points.

Algorithm	**Max number of iterations for feasibility**	**Average optimal value obtained (-km)**	**Standard deviation on the optimal value obtained**
BO & GP	34	-191.747 km	6969
BO & NLM	29	-195.406 km	10325
BO & DGP	**18**	**-200.110** km	**6183**

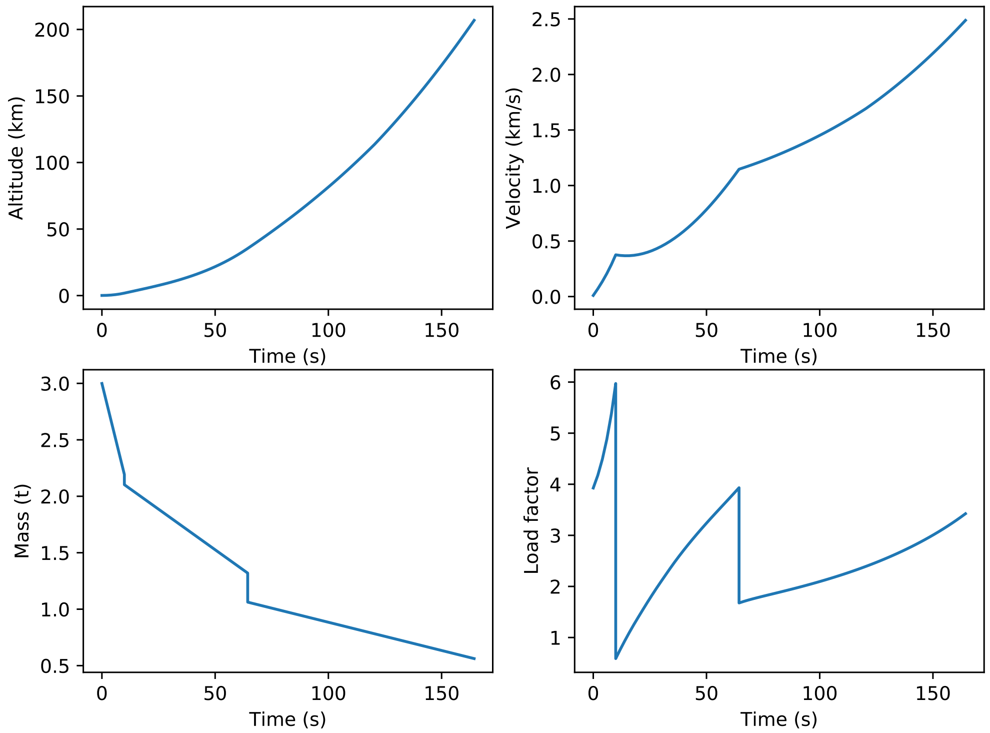


Fig. 4.23 Illustrations of optimal trajectory, Altitude, Relative Velocity, Mass and Axial Load factor as functions of time for the optimal solution given by the BO & DGP algorithm.

4.3 Conclusion

The application of DGP to Bayesian optimization has been discussed in this chapter. This coupling requires some adaptations of the handling of DGPs and BO. For that, a framework for BO & DGP has been developed. This framework, proposes adaptations of DGPs for BO (training approach, uncertainty quantification, architecture of the DGP) and also of BO for DGPs (the iterative structure of BO, infill criteria). These adaptations were described and illustrated through some analytical examples. Following these propositions, BO with DGP was assessed on analytical test optimization problems. The experimentations showed its better efficiency and robustness compared with standard BO & GP and approaches using non-linear mapping to handle non-stationarity. Finally, this algorithm was applied to aerospace engineering design problems. This illustrated its efficiency on constrained problems and also proved the dimension scaling of BO with DGP up to $15d$. Moreover, these test cases also highlighted a better handling of the constraints by BO with DGPs, where it reaches the feasible domain faster than the compared algorithms and obtains a better optimal value at the exhaustion of the evaluation budget available.

The contribution of this chapter is to couple between BO and DGP and also to highlight the tangible interest of this coupling. In fact, the results of this coupling are compared to state-of-the-art algorithms in BO for non-stationary functions and its application on real industrial optimization problems. This chapter also provides some design choices for the coupling of BO with DGP based on experimentation analysis. However, these experimentations are not generalizable, and theoretical analysis is needed. Thus, the discussion presented in this chapter on the design choices for the coupling of BO with DGP leads to interesting theoretical research tracks. An important one is how does the natural gradient optimization method of DGP affect its uncertainty model as experienced in this study. Moreover, infill criteria such as Thompson Sampling or criteria using information theory may be more adapted to DGP than the EI. More parallelism can also be integrated at different levels of the coupling of BO with DGP.

In this chapter, only the single objective case has been taken into account. However, in design optimization problems, often different objectives are considered [Arias-Montano et al., 2012]. Moreover, these different objectives are antagonistic and solving the optimization problem comes back to finding trade-off between these objectives. The approach of BO described in this chapter can be used by considering each objective independently and using a multi-objective infill criteria such as the expected hyper-volume improvement [Hebbal et al., 2019]. However, considering each objective independently may be sub-optimal [Shah and Ghahramani, 2016]. In the next chapter, a multi-objective deep Gaussian process model is proposed that enables a joint modeling of the different objectives, thus, exhibiting an objective correlation instead of the classic approach of modeling independently each objective.

Chapter 5

Multi-Objective Bayesian Optimization taking into account correlation between objectives

> *" There are no solutions; there are only trade-offs."*
>
> Thomas Sowell

Chapter contributions

- Development of a novel model called the Multi-Objective Deep Gaussian Process (MO-DGP) for jointly modeling of correlated functions.
- Exhibition of the limits of the existing approach to compute the correlated Expected Hyper-Volume Improvement, and proposition of a novel computational approach.
- Application of the proposed model and the Expected Hyper-Volume Improvement computational approach to an extensive benchmark including a representative aerospace multi-objective optimization design problem.

$\mathcal{CH}_5$

For engineering design problems, single objective optimization may result in an over-optimized objective to the detriment of other performance. Indeed, multiple objectives have to be taken into account in order to find a trade-off between the different criteria of interest [Arias-Montano et al., 2012; Brevault et al., 2020a]. For instance, in the design of an aerospace launch vehicle, different objectives may be considered such as the minimization of the gross-lift-off-weight, the maximization of the payload mass, and the maximization of the change in velocity. The trade-offs between these objectives called Pareto dominant solutions are obtained using multi-objective optimization algorithms [Deb, 2001; Talbi et al., 2012]. In the context of black-box computationally intensive objectives, Multi-Objective Bayesian Optimization (MO-BO) is the extension of Bayesian Optimization (BO) to the multi-objective case [Emmerich et al., 2006]. This consists in using infill criteria that take into account multiple objectives such as the Expected Hyper-Volume Improvement (EHVI) [Emmerich and Klinkenberg, 2008], while using Bayesian models such as Gaussian Processes (GPs) independently for each objective (see Chapter 3, Section 3.3 for details). The contribution of the previous chapter that is Deep Gaussian Processes (DGPs) for the optimization of non-stationary functions can be extended to the multi-objective case by using an independent DGP for each objective and considering the EHVI as an infill criterion [Hebbal et al., 2019].

The limitation of MO-BO is that modeling each objective independently does not take advantage of the potential correlation between the objectives. In fact, in a multi-objective optimization setting, the objectives are usually antagonistic especially around the Pareto front, that is the set of the Pareto dominant solutions. Moreover, in addition to the modeling, the computation of the infill criteria such as the EHVI also considers the objectives as independent. To overcome these limitations, [Shah and Ghahramani, 2016] proposed to model the different objectives jointly using a Linear Model of Coregionalization (LMC) where each output corresponds to an objective, and the coregionalization matrix is used to encode the correlation between the objectives. Moreover, an approximation scheme is developed to compute a correlated-objective EHVI (see Chapter 3, Section 3.3.3 for details). This approach, while it overcomes the limitation of independency between the objectives in MO-BO, still presents some limitations. In fact, using LMC considers only the linear correlation between the objectives, hence, more complicated correlations may not be exhibited using this model. Moreover, the approximation scheme used to compute the correlated EHVI approximates piece-wise linear functions and step functions with Gaussian distributions. This may yield to limited approximation of the correlated EHVI.

The contribution of this chapter is at two levels of the MO-BO framework that are the model used and the computation of the infill criterion. At the first level, a novel model based on DGPs is proposed that takes into account correlations between the objectives to improve its predictive capability. At the second level, an investigation is carried out on the computation of EHVI while taking into account the correlation between the objectives in order to propose an adapted approach to compute the correlated EHVI. The performance of the proposed model and the proposed approach to compute the correlated EHVI is assessed on analytical test problems as well as on an aerospace optimization problem.

This chapter is organized in three main sections. In the first section (Section 5.1), the proposed model is developed with a focus on its training and its predictive capabilities. The second section (Section 5.2) discusses the computation of the EHVI in the context of correlated objectives and a novel approach to compute the correlated EHVI is proposed. The final section (Section 5.3) presents an analytical benchmark as well as a representative aerospace multi-objective design problem to evaluate the performance of the proposed model with respect to the approach proposed by [Shah and Ghahramani, 2016] as well as classic MO-BO algorithms.

5.1 Multi-Objective Deep Gaussian Process Model (MO-DGP)

In this section, the Multi-Objective Deep Gaussian Process model (MO-DGP) is proposed to take into account the correlation between the objectives. An inference approach for MO-DGP is developed, and its prediction capability is compared to independent models for each objective and to the Linear Model of Coregionalization (LMC). The notations used in Chapter 3, Section 3.3 are adopted in the remaining of this chapter. A multi-objective problem is considered characterized by n_o objectives potentially optimized under n_c constraints in a d-dimensional design space (minimization is considered without loss of generality). Let $\mathbf{X}$ be the input data of size n and $(\mathbf{y}_1, \ldots, \mathbf{y}_{n_o})$ its evaluations on the different n_o objectives.

5.1.1 Model specifications

The classic approach for MO-BO is to consider a Bayesian model for each objective independently [Emmerich et al., 2006; Zhang et al., 2010; Emmerich et al., 2016; Yang et al., 2019]. This is illustrated in Fig. 5.1 where a GP is used for each objective. However, the objectives are usually antagonistic especially around the Pareto front. Therefore, taking into account this correlation instead of considering them independent may result in a better learning of these objectives. The proposed MO-DGP model aims to exhibit this correlation by modeling jointly these objectives. Compared to the use of DGP in Chapter 4, where the functional composition of GPs was used to model non-stationary behavior, in this chapter, DGP are used to model correlation between objective functions.

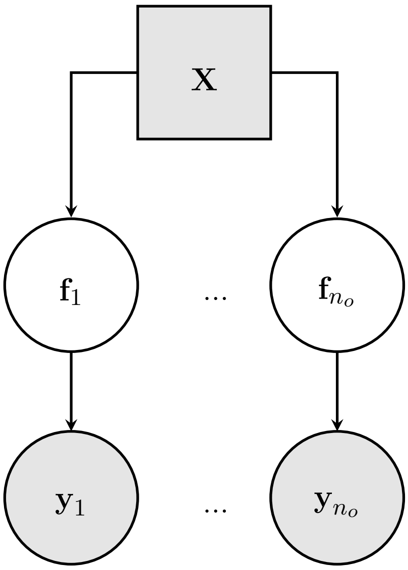


Fig. 5.1 Illustration of the independent modeling of the objectives in a MO-BO framework. An independent Bayesian model is used for each objective i for $1 \leq i \leq n_o$.

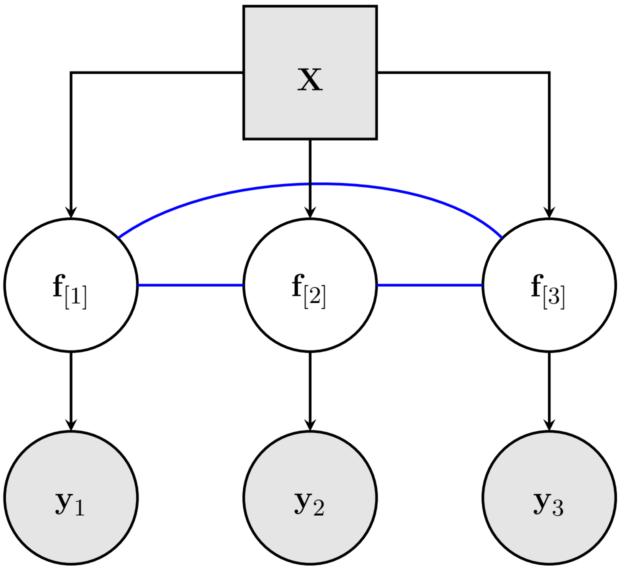


Fig. 5.2 Illustration of the MO-DGP model in the case of three objectives. The objectives f_i for $1 \leq i \leq n_o$ are connected by non-oriented edges (colored blue) and constitute a clique (all the nodes are adjacent).

MO-DGP considers a DGP model of n_o layers where a layer i corresponds to the objective i and is conditioned on the observed values of this objective $\mathbf{y}^i$. Instead of classic DGPs which are represented as directed graphs with a Markov-Chain structure meaning that a layer i depends only on the previous one, in MO-DGP, the unobserved nodes $f_{[i]}$ are connected with non-oriented edges and constitute a clique meaning that each layer i interacts with every other layer j (Fig. 5.2). In fact, there is no prior known structure about the interaction between the objectives to consider an oriented direction between them. Therefore, non-oriented edges are used to connect between the layers. Notice that, unlike DGPs, MO-DGP cannot be written as a functional composition of GPs since there is no actual starting function. Moreover, in each layer the input is augmented with the outputs of all the other layers. Therefore, the input space dimension of each layer is $d + n_o - 1$.

The covariance function for each GP has to take into account the augmented input space. The proposed kernel inspired by [Perdikaris et al., 2017] allows to exhibit the correlation between the objectives as follows:

$$k_i\left([\mathbf{x}, \tilde{\mathbf{f}}_{[-i]}(\mathbf{x})], [\mathbf{x}', \tilde{\mathbf{f}}_{[-i]}(\mathbf{x}')]\right) = k_{\rho_i}(\mathbf{x}, \mathbf{x}') \times k_{f_{[i]}}\left(\tilde{\mathbf{f}}_{[-i]}(\mathbf{x}), \tilde{\mathbf{f}}_{[-i]}(\mathbf{x}')\right) + k_{\gamma_i}(\mathbf{x}, \mathbf{x}') \quad (5.1)$$

where $\tilde{\mathbf{f}}_{[-i]}(\mathbf{x})$ stands for the vector-valued evaluation of $\mathbf{x}$ by all the GP posteriors $\tilde{f}_{[j]}$ expect $\tilde{f}_i$. $k_{\rho_i}(\cdot,\cdot)$ and $k_{\gamma_i}(\cdot,\cdot)$ are covariance functions with respectively an input space-dependent scaling effect and an input space-dependent bias effect, whilst $k_{f_{[i]}}(\cdot,\cdot)$ is the covariance function between the evaluated posteriors of the other objectives.

5.1.2 Inference in MO-DGP

For the inference in MO-DGP, first, the doubly stochastic approach proposed in [Salimbeni and Deisenroth, 2017] is followed (see Chapter 2, Section 2.3.2). For that, at each GP layer, a set of input/output induced variables $\{\mathbf{Z}_{[i]}, \mathbf{u}_{[i]}\}_{i=1}^{n_o}$ is introduced and the evidence of the model is written as follows:

$$\begin{aligned} p(\{\mathbf{y}_i\}_{i=1}^{n_o}|\mathbf{X}) &= \int\int p(\{\mathbf{y}_i\}_{i=1}^{n_o}, \{\mathbf{f}_{[i]}\}_{i=1}^{n_o}, \{\mathbf{u}_{[i]}\}_{i=1}^{n_o}|\mathbf{X}, \{\mathbf{Z}_{[i]}\}_{i=1}^{n_o}) \mathrm{d}\{\mathbf{f}_{[i]}\}_{i=1}^{n_o} \mathrm{d}\{\mathbf{u}_{[i]}\}_{i=1}^{n_o} \\ &= \int\int \prod_{i=1}^{n_o} \left[p(\mathbf{y}_i|\mathbf{f}_{[i]})\right] \times p(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}|\{\mathbf{u}_{[i]}\}_{i=1}^{n_o}, [\mathbf{X}, \mathbf{f}_{[-i]}], \mathbf{Z}_{[i]}) \\ &\times \prod_{i=1}^{n_o} \left[p(\mathbf{u}_{[i]}|\mathbf{Z}_{[i]})\right] \mathrm{d}\{\mathbf{f}_{[i]}\}_{i=1}^{n_o} \mathrm{d}\{\mathbf{u}_{[i]}\}_{i=1}^{n_o} \end{aligned} \tag{5.2}$$

where $\mathbf{f}_{[i]}$ represents the layer i GP evaluation of $\mathbf{X}$. Then, the following variational approximation is considered:

$$q\left(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}, \{\mathbf{u}_{[i]}\}_{i=1}^{n_o}\right) = p(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}|\{\mathbf{u}_{[i]}\}_{i=1}^{n_o}, [\mathbf{X}, \mathbf{f}_{[-i]}], \mathbf{Z}_{[i]}) \times \prod_{i=1}^{n_o} q(\mathbf{u}_{[i]}) \tag{5.3}$$

Notice here that unlike the variational approximation presented in [Salimbeni and Deisenroth, 2017] where the chain rule is used for $p(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o})$, in MO-DGP, the $\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}$ are connected by non-oriented edges, thus, there is no starting point to use the chain rule to express the joint distribution by conditional distributions. Therefore, the joint distribution is kept in the variational approximation. This variational approximation is introduced in the evidence in Eq. (5.2) as follows:

$$\begin{aligned} p(\{\mathbf{y}_i\}_{i=1}^{n_o}|\mathbf{X}) = &\int\int \prod_{i=1}^{n_o} \left[p(\mathbf{y}_i|\mathbf{f}_{[i]})\right] \times p(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}|\{\mathbf{u}_{[i]}\}_{i=1}^{n_o}, [\mathbf{X}, \mathbf{f}_{[-i]}], \mathbf{Z}_{[i]}) \\ &\times \prod_{i=1}^{n_o} \left[p(\mathbf{u}_{[i]}|\mathbf{Z}_{[i]})\right] \times \frac{q\left(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}, \{\mathbf{u}_{[i]}\}_{i=1}^{n_o}\right)}{q\left(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}, \{\mathbf{u}_{[i]}\}_{i=1}^{n_o}\right)} \mathrm{d}\{\mathbf{f}_{[i]}\}_{i=1}^{n_o} \mathrm{d}\{\mathbf{u}_{[i]}\}_{i=1}^{n_o} \end{aligned} \tag{5.4}$$

Then, by introducing the log and using Jensen inequality, the log evidence of the model is bounded by an Evidence Lower bound (ELBO):

$$\begin{aligned}\log p(\{\mathbf{y}_i\}_{i=1}^{n_o}|\mathbf{X}) \geq & \mathcal{L}_{\text{MO-DGP}} \\ \mathcal{L}_{\text{MO-DGP}} = & \int\int q\left(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o},\{\mathbf{u}_{[i]}\}_{i=1}^{n_o}\right)\times \\ & \log\left(\frac{\prod_{i=1}^{n_o}\left[p(\mathbf{y}_i|\mathbf{f}_{[i]})\right]\times p(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}|\{\mathbf{u}_{[i]}\}_{i=1}^{n_o},[\mathbf{X},\mathbf{f}_{[-i]}],\mathbf{Z}_{[i]})\prod_{i=1}^{n_o}\left[p(\mathbf{u}_{[i]}|\mathbf{Z}_{[i]})\right]}{q\left(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o},\{\mathbf{u}_{[i]}\}_{i=1}^{n_o}\right)}\right) \\ & \mathrm{d}\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}\mathrm{d}\{\mathbf{u}_{[i]}\}_{i=1}^{n_o}\end{aligned} \tag{5.5}$$

Replacing the variational approximation by its expression in Eq. (5.3) allows to simplify the expression of this lower bound:

$$\begin{aligned}\mathcal{L}_{\text{MO-DGP}} = & \int\int p(\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}|\{\mathbf{u}_{[i]}\}_{i=1}^{n_o},[\mathbf{X},\mathbf{f}_{[-i]}],\mathbf{Z}_{[i]}) \\ & \times\prod_{i=1}^{n_o}\left[q(\mathbf{u}_{[i]})\right]\times\log\left(\frac{\prod_{i=1}^{n_o}\left[p(\mathbf{y}_i|\mathbf{f}_{[i]})p(\mathbf{u}_{[i]}|\mathbf{Z}_{[i]})\right]}{\prod_{i=1}^{n_o}q(\mathbf{u}_{[i]})}\right)\mathrm{d}\{\mathbf{f}_{[i]}\}_{i=1}^{n_o}\mathrm{d}\{\mathbf{u}_{[i]}\}_{i=1}^{n_o}\end{aligned} \tag{5.6}$$

Then, by separating the log expressions and identifying the expectation term and the KL divergence term, the factorized expression of the ELBO over the observed variables is obtained:

$$\mathcal{L}_{\text{MO-DGP}} = \sum_{i=1}^{n_o}\sum_{j=1}^{n}\mathbb{E}_{q(\{f_{[k]}^{(j)}\}_{k=1}^{n_o})}\log\left(p\left(y_i^{(j)}|f_{[i]}^{(j)}\right)\right)-\sum_{i=1}^{n_o}\mathbb{KL}\left[q\left(\mathbf{u}_{[i]}\right)||p(\mathbf{u}_{[i]}|\mathbf{Z}_{[i]})\right] \tag{5.7}$$

The main difference with the ELBO obtained in regular DGPs is that the expectation term in this ELBO is computed with respect to the joint distribution $q(\{f_{[k]}^{(j)}\}_{k=1}^{n_o})$ since the chain rule cannot be used. While the conditioned sampling from a layer given the other layers is straightforward by using the posterior predictive distribution of a Gaussian process, sampling from the joint distribution of all the layers is more challenging. In fact, every layer interacts with every other layer, hence, there is no starting layer, unlike regular DGPs. To overcome this issue, Gibbs sampling can be used [Geman and Geman, 1984; Gelman et al., 2013]. In fact, Gibbs sampling allows to obtain samples from a joint distribution, which is difficult to directly sample from, by using the conditional distributions. Practically, given the DoE $(\mathbf{X},\mathbf{y}_1,\dots,\mathbf{y}_{n_o})$, the layer evaluations $\mathbf{f}_{[1]},\dots,\mathbf{f}_{[n_o]}$ are initialized to some value $\mathbf{f}_{[1],\{0\}},\dots,\mathbf{f}_{[n_o],\{0\}}$ (*e.g.*, 0 for normalized data), where the subscript $_{\{j\}}$ stands for the iteration number j of the Gibbs sampling procedure. Now that all layers output have a specific value, one can use the distribution of each layer conditioned on all the other layers output. More specifically, s samples $(\mathbf{f}^{[1]}_{[1],\{1\}}, \ldots, \mathbf{f}^{[s]}_{[1],\{1\}})$ are drawn (in parallel) following $q(\mathbf{f}_{[1]}|\mathbf{f}_{[2],\{0\}}, \ldots, \mathbf{f}_{[n_o],\{0\}})$ (where the superscript $^{[t]}$ stands for the sample number t), then using the updated values of layer 1 a sample is drawn from $q(\mathbf{f}_{[2]}|\mathbf{f}^{[t]}_{[1],\{1\}}, \mathbf{f}_{[3],\{0\}}, \ldots, \mathbf{f}_{[n_o],\{0\}})$ for $1 \leq t \leq s$, and so on until reaching the final layer. This loop over the different layers is repeated until stabilization of the distribution (and therefore the samples). The expectation term is then estimated by averaging over the samples obtained at the final loop iteration. This is summarized in Algorithm 3 where n_{Gibbs} corresponds to the number of loops in the Gibbs sampling procedure. This enables to estimate the ELBO and therefore perform its optimization with respect to the inducing inputs $\{\mathbf{Z}_{[i]}\}_{i=1}^{n_{\text{o}}}$, the parameters $\{\boldsymbol{\theta}_{q(\mathbf{u}_{[i]})}\}_{i=1}^{n_{\text{o}}}$ of the variational distributions $\{q(\mathbf{u}_{[i]}) = \mathcal{N}(\mathbf{u}_{[i]}|\bar{\mathbf{u}}_{[i]}, \boldsymbol{\Gamma}_{[i]})\}_{1}^{n_{\text{o}}}$, and the GP hyperparameters at each layer $\{\boldsymbol{\theta}_{[i]}\}_{i=1}^{n_{\text{o}}}$. The challenging part of the MO-DGP training is the optimization of the induced inputs $\{\mathbf{Z}_{[i]}\}_{i=1}^{n_{\text{o}}}$. In fact, the induced inputs lie in the augmented input space of dimension $d + n_o - 1$ where the $n_o - 1$ last components depend on the d first ones making it not suitable to optimize them freely. To overcome this issue, the last $n_o - 1$ dimensions of the inducing inputs are not considered in the optimization, and rather inferred by propagation through the other layers of the d first dimensions that are optimized freely. More details on this procedure are developed in Section 6.1.1.

The complexity of this ELBO is $\mathcal{O}\left(s \times n_{\text{Gibbs}} \times \sum_{i=1}^{n_o} n_i^3\right)$. In the numerical experiments performed in Section 5.3, for a number of samples $s = 1000$, a number of loops $n_{\text{Gibbs}} = 4$ is found to be sufficient for the stabilization of the samples.

5.1.3 MO-DGP prediction

To predict the response of the different objectives for a new data-point $\mathbf{x}^*$ using MO-DGP, Gibbs sampling is used as described in Algorithm 3. Instead of the DoE $\mathbf{X}$, it is performed on the new data-point $\mathbf{x}^*$.

MO-DGP, by jointly modeling the different objectives, improves the prediction capability compared to independent modeling. To illustrate this feature of MO-DGP,

Algorithm 3: Sampling algorithm from the joint distribution $q(\{\mathbf{f}_{[i]}\}_{k=1}^{n_o})$

```
Initialize: $\mathbf{f}^{[t]}_{[1],\{0\}}, \dots, \mathbf{f}^{[t]}_{[n_o],\{0\}}$ $1 \leq t \leq s$
for $j = 1 \dots n_{Gibbs}$ do
    $\mathbf{f}^{[t]}_{[1],\{j\}} \sim q(\mathbf{f}_{[1]} | \mathbf{f}^{[t]}_{[2],\{j-1\}}, \mathbf{f}^{[t]}_{[3],\{j-1\}} \dots, \mathbf{f}^{[t]}_{[n_o],\{j-1\}})$ $1 \leq t \leq s$
    $\mathbf{f}^{[t]}_{[2],\{j\}} \sim q(\mathbf{f}_{[2]} | \mathbf{f}^{[t]}_{[1],\{j\}}, \mathbf{f}^{[t]}_{[3],\{j-1\}} \dots, \mathbf{f}^{[t]}_{[n_o],\{j-1\}})$ $1 \leq t \leq s$
    $\vdots$
    $\mathbf{f}^{[t]}_{[i],\{j\}} \sim q(\mathbf{f}_{[i]} | \mathbf{f}^{[t]}_{[1],\{j\}}, \dots, \mathbf{f}^{[t]}_{[i-1],\{j\}}, \mathbf{f}^{[t]}_{[i+1],\{j-1\}}, \dots, \mathbf{f}^{[t]}_{[n_o],\{j-1\}})$ $1 \leq t \leq s$
    $\vdots$
    $\mathbf{f}^{[t]}_{[n_o],\{j\}} \sim q(\mathbf{f}_{[n_o]} | \mathbf{f}^{[t]}_{[1],\{j\}}, \mathbf{f}^{[t]}_{[2],\{j\}} \dots, \mathbf{f}^{[t]}_{[n_o-1],\{j\}})$ $1 \leq t \leq s$
end
return $\{\mathbf{f}^{[t]}_{[1],\{n_{\text{Gibbs}}\}}, \dots, \mathbf{f}^{[t]}_{[n_o],\{n_{\text{Gibbs}}\}}\}_{t=1}^{s}$
```

the following two-objective toy-problem is considered:

$$\begin{aligned} \text{min} \quad & [f_1(x), f_2(x)] \\ \text{s.t.} \quad & 0 \leq x \leq 1 \\ \text{with} \quad & f_1(x) = \exp\left(\cos\left(15(2x - 0.2)\right)\right) - 1 \\ \text{and} \quad & f_2(x) = -x \exp\left(\cos\left(15(2x - 0.2)\right)\right) - 1 \end{aligned} \tag{5.8}$$

The objective space of this two-objective problem is represented in Fig. 5.3. This figure illustrates the negative correlation between the two objectives. The modeling of these

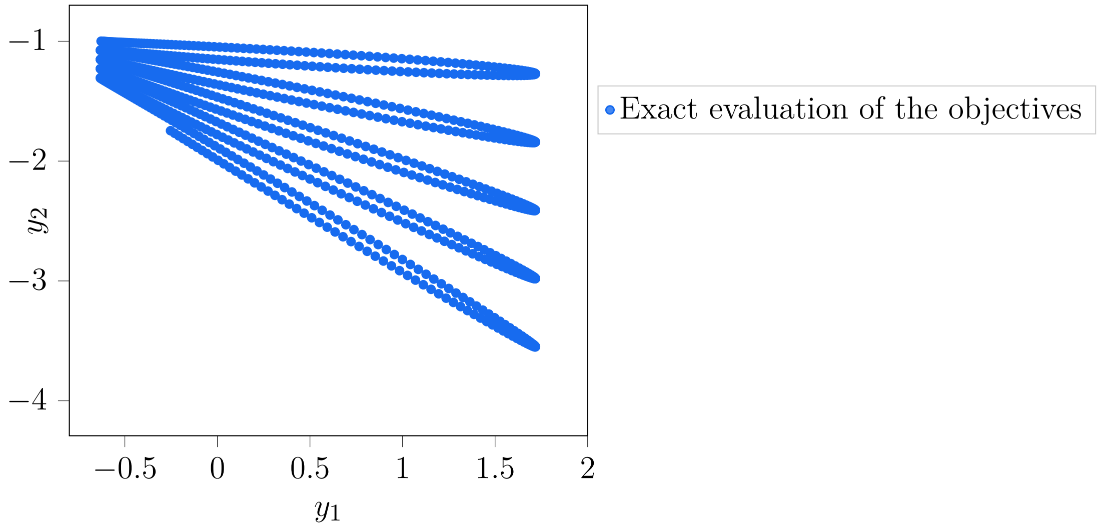


Fig. 5.3 Objective space of the two-objective problem defined in Eq. (5.8)

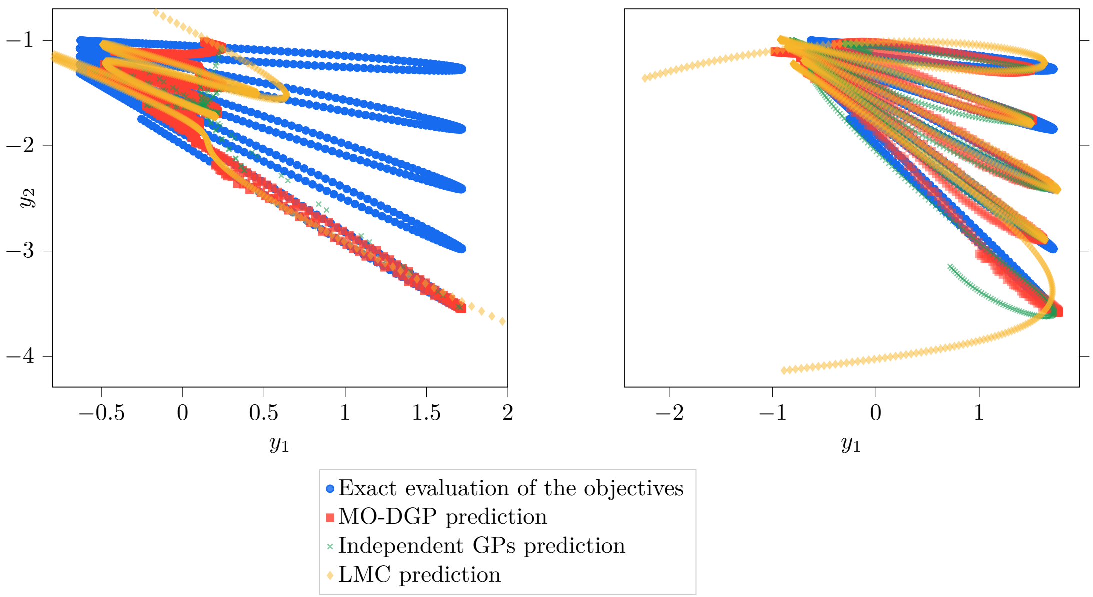


Fig. 5.4 Prediction of MO-DGP (colored red), independent GPs (colored green), and LMC (colored orange) on the objective space of the defined problem in Eq. (5.8). (left) Prediction with a DoE size of 10 data-points, MO-DGP captures well the Pareto front compared to the two other models. (right) Prediction with a DoE size of 15 data-points, the prediction of the three models is improved, however, LMC and GPs still provides an inaccurate approximated Pareto front.

objectives (without considering the optimization problem solving) is performed with two different sizes of Design of Experiments (DoE) of availability of data (10 and 15 observations) using independent GPs, LMC, and MO-DGP (Fig. 5.4). With only 10 data-points, MO-DGP well captures the exact Pareto front compared to the two other models. When increasing the number of data-points, the prediction performance over all the objective space is improved for the three models. However, the approximated Pareto fronts obtained by the independent GPs and LMC are still inaccurate with respect to the exact Pareto front.

5.2 Computation of the Expected Hyper-Volume Improvement (EHVI)

To integrate MO-DGP within a BO-MO framework, it has to be coupled to a multi-objective infill criterion. One of the widely used infill criteria is the Expected Hyper-

Volume Improvement (EHVI). The EHVI considers the Lebesgue measure of the expected hyper-volume dominated by the approximated Pareto front obtained by the Bayesian models Eq. (3.31). In the remaining of this chapter, the two-objective case is considered where the expression of the EHVI can be written as follows (see Chapter 3, Section 3.3 for details):

$$\begin{aligned} EHVI(\mathbf{x}) = &\sum_{i=1}^{n_p+1} \int_{y_1^{\mathrm{lb}}}^{y_1^{(i)}} \int_{y_2^{\mathrm{lb}}}^{y_2^{(i)}} \left(y_1^{(i-1)} - y_1^{(i)}\right) \left(y_2^{(i)} - f_2(\mathbf{x})\right) p\left(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) + \\ &\sum_{i=1}^{n_p+1} \int_{y^{(i)}}^{y_1^{(i-1)}} \int_{y_2=y_2^{\mathrm{lb}}}^{y_2^{(i)}} \left(y_1^{(i)} - f_1(\mathbf{x})\right) \left(y_2^{(i)} - f_2(\mathbf{x})\right) p\left(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) \end{aligned} \tag{5.9}$$

where n_p is the size of the approximated Pareto front, y_i^{lb} and y_i^{ub} are respectively a chosen lower-bound and upper-bound on objective i, and $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n_p)}$ are the DoE approximated Pareto front evaluations sorted in a decreasing order of the first objective, $\mathbf{y}^{(0)} = [y_1^{\mathrm{ub}}, y_2^{\mathrm{lb}}]^\intercal$, and $\mathbf{y}^{(n_p+1)} = [y_1^{\mathrm{lb}}, y_2^{\mathrm{ub}}]^\intercal$. This expression is analytically tractable when considering the objectives as independent. However, this is not the case when using a joint model for the objectives such as MO-DGP or LMC.

5.2.1 Approximation of piece-wise functions with Gaussian distributions

To compute the EHVI when using a joint model with a Gaussian predictive distribution for the objectives, [Shah and Ghahramani, 2016] proposed an approximation approach. This approximation fully described in Chapter 3, Section 3.3.3, consists in rewriting the bounds of the integrals in Eq. (5.9) on $\mathbb{R}$ by introducing the indicator function $\mathbb{I}[\cdot]$, then, in approximating the piece-wise linear functions $(y_1^{(i)} - f_1(\mathbf{x}))\mathbb{I}\left[y^{(i)} \leq f_1(\mathbf{x}) \leq y_1^{(i-1)}\right], (y_2^{(i)} - f_2(\mathbf{x}))\mathbb{I}\left[y_2^{\mathrm{lb}} \leq f_2(\mathbf{x}) \leq y_2^{(i)}\right]$ and the constant piece-wise function $(y_1^{(i-1)} - y_1^{(i)})\mathbb{I}\left[y_1^{(i-1)} \leq y_1^{(i)}\right]$ by Gaussian distributions using moment matching. Next, since the predictive distribution is considered as Gaussian, the integrands come back to the product of two multi-variate Gaussians which is a scaled multi-variate Gaussian distribution. Therefore, the expression in Eq. (5.9) comes back to integrals of scaled Gaussian distributions on their full support that is equal to the scaling factor.

The limit of this approximation approach is that the estimation of a piece-wise linear and constant functions by Gaussian distributions may be inaccurate, as displayed in

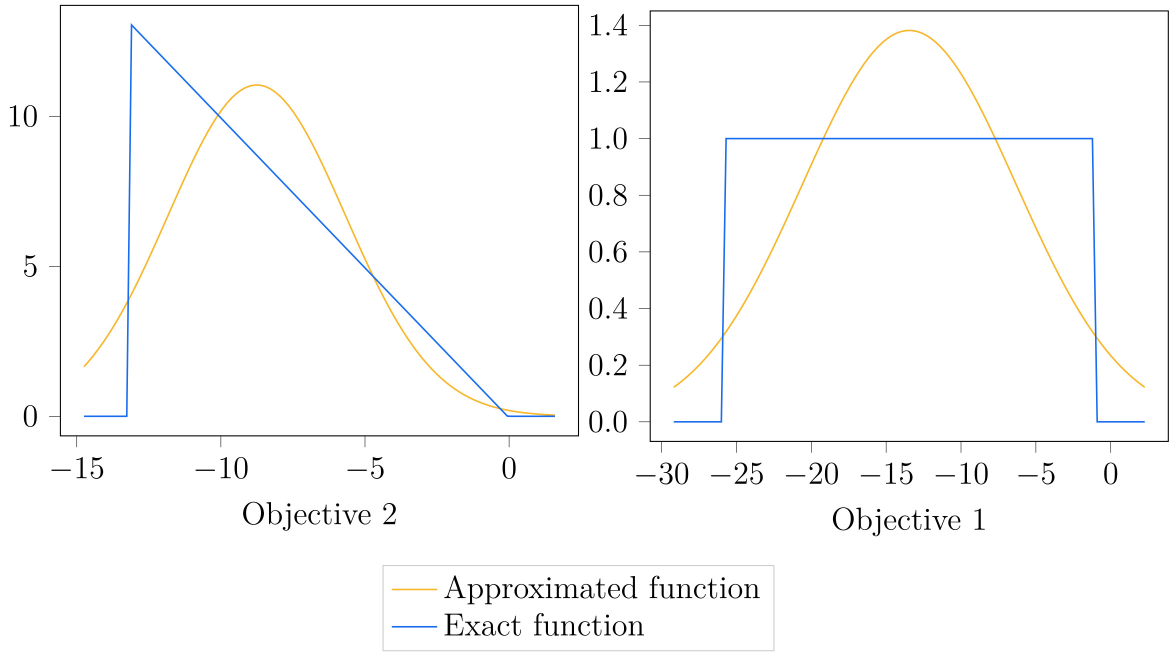


Fig. 5.5 Piece-wise functions are approximated by Gaussian distributions using moment matching, (left) a piece wise linear function approximated by a Gaussian distribution, (right) a piece wise constant function approximated by a Gaussian distribution.

Fig. 5.5, yielding to a limited approximation of the correlated EHVI. To illustrate this, consider the previous multi-objective problem in Eq. (5.8) and a LMC model trained on a DoE of 10 observations. The predictive distribution given by the LMC model is a two-variate Gaussian distribution with a correlation between the two objectives determined by the LMC oregionalization matrix. To assess the quality of this approximation, the obtained correlated EHVI is compared to the independent EHVI using the same approximation, and to the exact independent EHVI in Fig. 5.6. This allows to identify the part of the approximated correlated EHVI induced by the approximation from the one induced by taking into account the correlation between the objectives. As it can be seen in Fig. 5.6, the approximation itself induces a more important change in value than the correlation. In fact, the difference between the exact independent EHVI and the approximated independent EHVI is larger than the one between the latter and the approximated correlated EHVI. Moreover, the input variable value corresponding to the maximum of the EHVI given by the approximated EHVI differs from the one given by the exact independent EHVI due to the approximation and not the correlation between the objectives. Therefore, the approximation may lead to less interesting data point to be added to the DoE. Moreover, increasing the integral bounds on which the EHVI is computed makes the approximation coarser since the segments on which the approximation is performed are wider and therefore the approximation error is larger. This is illustrated in the right figure of Fig. 5.6, where the same EHVI is computed on larger bounds. Since in BO the initial DoE is small, the segments on which the approximation is performed are wide, this may yield to large errors in the approximation.

Hence, this example illustrates the limits of this approximation in the context of MO-BO at two levels. The first one is that the approximation itself may lead to a different maximum of the EHVI and hence a non-desirable added-point to the data-set. Second, the approximation is deteriorated when the objective space is populated with only a few solutions, which is usually the case in MO-BO.

One may argue that using a mixture of Gaussian distributions would give a better approximation. In fact a mixture of Gaussian distributions will better approximate the piece-wise functions as illustrated in Fig. 5.7 in which a mixture of four Gaussian distributions is used. The parameters of the Gaussian distributions are optimized in order to minimize the quadratic error between the approximation and the exact piece-wise functions. The approximation of the EHVI is improved as illustrated in Fig. 5.8. However, it still depends on the segments on which the approximation is performed, an using wider segments yields to larger errors in the approximation. Moreover, the parameters of the Gaussians are obtained using multiple optimization for each segment which is time consuming.

Another limitation of this approach to compute the EHVI is that it assumes that the predictive distribution of the model is Gaussian. While the predictive distribution of LMC is Gaussian, it is not necessary the case for MO-DGP as illustrated in Fig. 5.9. In the next subsection, a more accurate approach to compute the correlated EHVI is proposed which does not assume a particular form of the predictive distribution.

5.2.2 Proposed computational approach for correlated EHVI

Instead of approximating the piece-wise functions, another way to compute the EHVI would be to approximate the density of the predictive distribution. To estimate a density, there are extensive works in the literature [Silverman, 1986; Sheather, 2004; Scott, 2015; Wand and Yu, 2020]. One of the most popular approaches is Kernel Density Estimation (KDE) [Parzen, 1962; Simonoff, 2012]. For a set of s samples of dimension n_o, $(\mathbf{f}^{[1]}, \ldots, \mathbf{f}^{[s]})$ drawn from an unknown distribution with density $p(\mathbf{f}(\mathbf{x}))$,

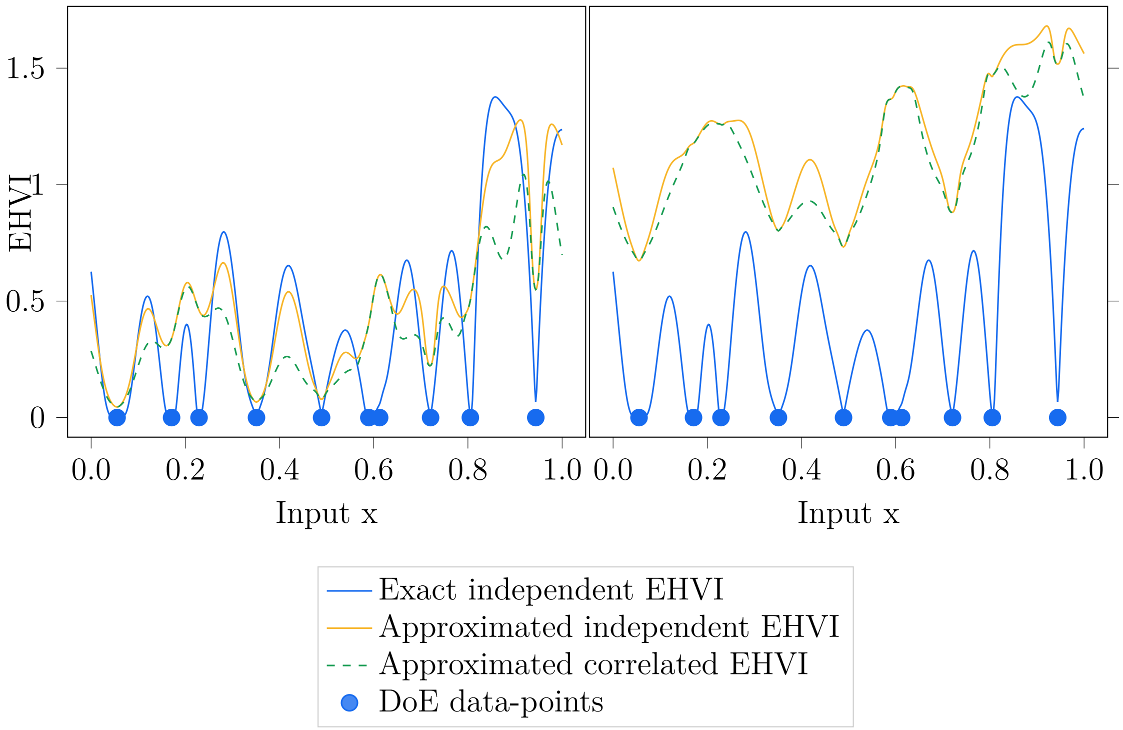


Fig. 5.6 Comparison on the design space of three different computations of the EHVI for the multi-objective problem in Eq. (5.8) with a DoE of 10 data-point. The blue colored curve corresponds to the exact computation of the EHVI with the assumption of independence between the objectives, the orange colored curve corresponds to the approximated computation of the EHVI with the assumption of independence between the objectives and the dashed green curve corresponds to the approximated computation of the EHVI with correlation between the objectives. In the right figure, the bounds on which the EHVI is computed are widened to show the degradation of the approximation with respect to the wideness of the bounds.

the KDE is defined as follows:

$$\hat{p}(\mathbf{f}(\mathbf{x})) = \frac{1}{s}\sum_{i=1}^{s} k_{\mathbf{B}}\left(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[i]}\right) \tag{5.10}$$

where $k_{\mathbf{B}}$ is a kernel function to be specified and $\mathbf{B}$ is a positive definite $n_o \times n_o$ matrix called the bandwidth [Jones et al., 1996].

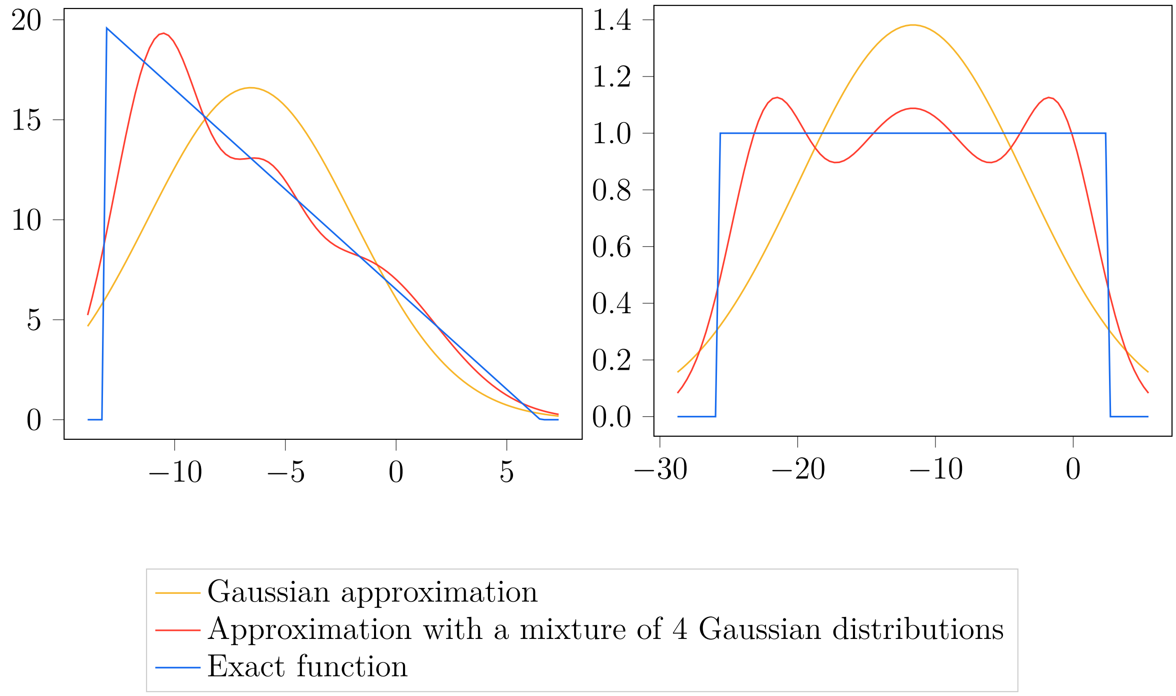


Fig. 5.7 Piece-wise functions (colored blue) are approximated by Gaussian distributions (colored orange) using moment matching and a mixture of four Gaussian distributions (colored red) by minimizing the error between the exact function and the mixture with respect to the parameters of the distributions, (left) the approximation of a piece wise linear function, (right) the approximation of a piece wise constant function

Using the KDE to estimate $p\left(\mathbf{f}(\mathbf{x})|\mathbf{Y},\mathbf{x},\mathbf{X}\right)$ in the expression of the EHVI in Eq. (5.9) yields to the following expression:

$$
\begin{aligned}
EHVI(\mathbf{x}) = & \sum_{i=1}^{n_p+1} \int_{y_1^{\text{lb}}}^{y_1^{(i)}} \int_{y_2^{\text{lb}}}^{y_2^{(i)}} \left(y_1^{(i-1)} - y_1^{(i)}\right)\left(y_2^{(i)} - f_2(\mathbf{x})\right) \frac{1}{s}\sum_{j=1}^{s} k_{\mathbf{B}}\left(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[t]}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) + \\
& \sum_{i=1}^{n_p+1} \int_{y^{(i)}}^{y_1^{(i-1)}} \int_{y_2=y^{\text{lb}_2}}^{y_2^{(i)}} \left(y_1^{(i)} - f_1(\mathbf{x})\right)\left(y_2^{(i)} - f_2(\mathbf{x})\right) \frac{1}{s}\sum_{j=1}^{s} k_{\mathbf{B}}\left(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[t]}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) \\
= & \frac{1}{s}\sum_{j=1}^{s} \left(\sum_{i=1}^{n_p+1} \int_{y_1^{\text{lb}}}^{y_1^{(i)}} \int_{y_2^{\text{lb}}}^{y_2^{(i)}} \left(y_1^{(i-1)} - y_1^{(i)}\right)\left(y_2^{(i)} - f_2(\mathbf{x})\right) k_{\mathbf{B}}\left(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[t]}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) + \right. \\
& \left. \sum_{i=1}^{n_p+1} \int_{y^{(i)}}^{y_1^{(i-1)}} \int_{y_2=y^{\text{lb}_2}}^{y_2^{(i)}} \left(y_1^{(i)} - f_1(\mathbf{x})\right)\left(y_2^{(i)} - f_2(\mathbf{x})\right) k_{\mathbf{B}}\left(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[t]}\right) \mathrm{d}\mathbf{f}(\mathbf{x}) \right)
\end{aligned}
\tag{5.11}
$$

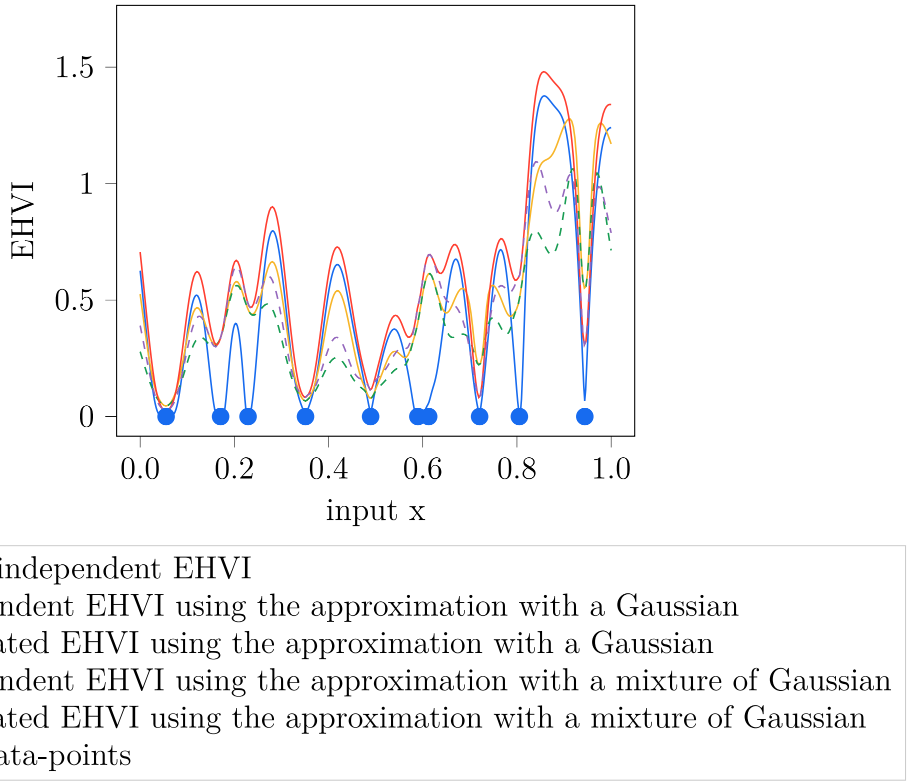


Fig. 5.8 Comparison on the design space of five different computations of the EHVI for the multi-objective problem in Eq. (5.8) with a DoE of 10 data-point. Despite the approximation of the piece-wise functions being better using the mixture of Gaussians, the corresponding approximation of the EHVI (colored red with the assumption of independency and dashed purple with correlated objectives) did not improve.

where $\mathbf{f}^{[t]}, j = 1, \dots, s$ are samples drawn from the predictive distribution of the model $p(\mathbf{f}(\mathbf{x})|\mathbf{Y}, \mathbf{x}, \mathbf{X})$. To obtain an analytical tractable form of the EHVI, the multivariate normal kernel $k_{\mathbf{B}}(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[t]}) = \frac{1}{\sqrt{(2\pi)^{n_o}|\mathbf{B}|}} \exp \frac{-1}{2} \left(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[t]}\right)^{\intercal} \mathbf{B}^{-1} \left(\mathbf{f}(\mathbf{x}) - \mathbf{f}^{[t]}\right)$ is used, where $|\mathbf{B}|$ corresponds to the determinant of $\mathbf{B}$. For the bandwidth matrix $\mathbf{B}$, the Silverman rule [Silverman, 1986] is used that is:

$$\begin{aligned} \sqrt{\mathbf{B}_{ii}} &= \sigma_i \left(\frac{4}{s(n_o + 2)} \right)^{\frac{1}{n_o+4}} \\ \sqrt{\mathbf{B}_{it}} &= 0 \text{ for } i \neq t \end{aligned} \tag{5.12}$$

where σ_i is the standard deviation of the marginal predictive distribution of the objective i. This allows to obtain a diagonal covariance matrix for the Gaussian kernel and therefore to estimate the predictive distribution as a mixture of the product of

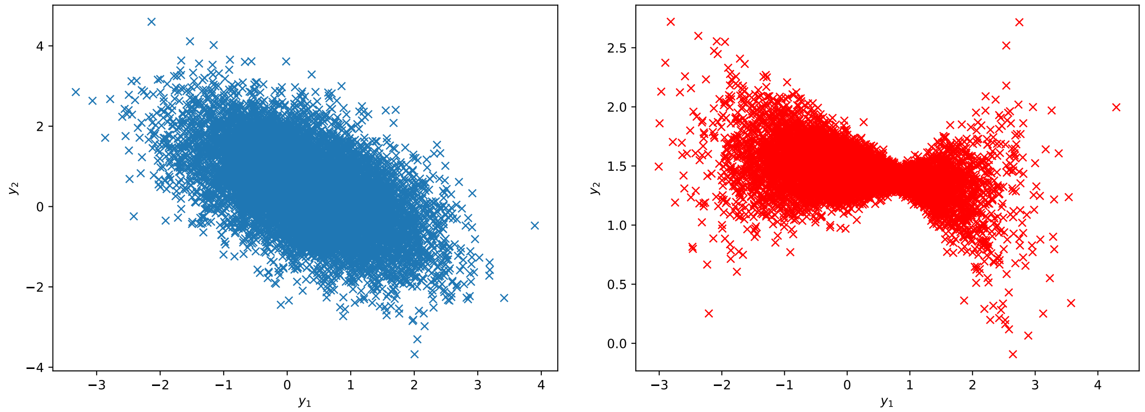


Fig. 5.9 Samples drawn in the objective space from, (left) the predictive posterior distribution of LMC which is a multivariate Gaussian distribution, (right) the predictive posterior distribution of MO-DGP which is not necessary Gaussian.

univariate Gaussian distributions:

$$\begin{aligned}
\hat{p}(\mathbf{f}(\mathbf{x})) =& \frac{1}{s}\sum_{j=1}^{s} k_{\mathbf{B}}\left(\mathbf{f}(\mathbf{x})-\mathbf{f}^{[t]}\right)\\
=& \frac{1}{s}\sum_{j=1}^{s}\frac{1}{\sqrt{(2\pi)^{n_o}|\mathbf{B}|}}\exp\frac{-1}{2}\left(\mathbf{f}(\mathbf{x})-\mathbf{f}^{[t]}\right)^{\intercal}\mathbf{B}^{-1}\left(\mathbf{f}(\mathbf{x})-\mathbf{f}^{[t]}\right)\\
=& \frac{1}{s}\sum_{j=1}^{s}\prod_{i=1}^{n_o}\frac{1}{\sqrt{(2\pi)\mathbf{B}_{ii}}}\exp\frac{-1}{2}\left(f_{[i]}(\mathbf{x})-f_{[i]}^{[t]}\right)^{\intercal}\mathbf{B}_{ii}^{-1}\left(f_{[i]}(\mathbf{x})-f_{[i]}^{[t]}\right)\\
=& \frac{1}{s}\sum_{j=1}^{s}\prod_{i=1}^{n_o}\phi_{\mathcal{N}(f_{[i]}^{[t]},\mathbf{B}_{ii})}\left(f_{[i]}(\mathbf{x})\right)
\end{aligned} \tag{5.13}$$

To illustrate this approximation, in Fig. 5.10, the predictive distribution of MODGP in Fig. 5.9 is estimated using Eq. (5.13), yielding to an accurate fit of the true distribution.

By injecting the estimate obtained in Eq. (5.13) in the expression of the EHVI in Eq. (5.11) which is considered in the two-objective case ($n_o = 2$), the following is

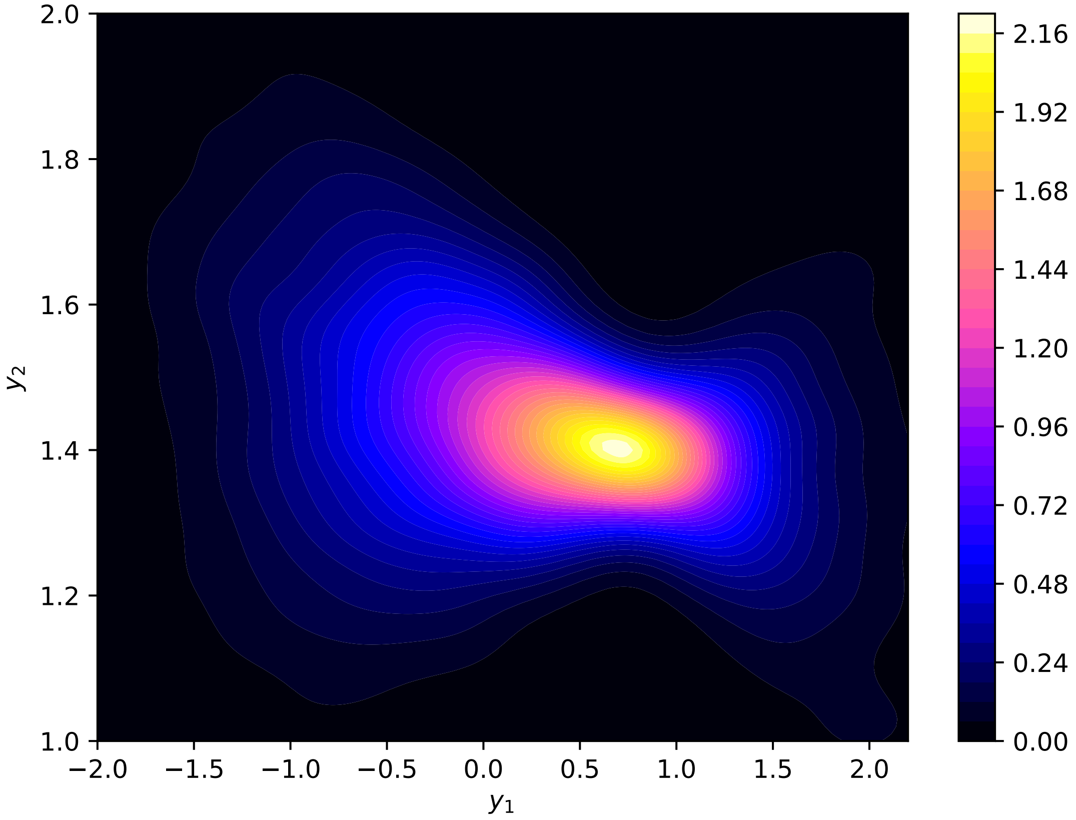


Fig. 5.10 KDE of the posterior predictive distribution of MO-DGP, whose samples are drawn in Fig. 5.9.

obtained:

$$
\begin{aligned}
&EHVI(\mathbf{x}) = \frac{1}{s}\sum_{j=1}^{s}\sum_{i=1}^{n_p+1} \\
&\left(\int_{y_1^{\mathrm{lb}}}^{y_1^{(i)}}\int_{y_2^{\mathrm{lb}}}^{y_2^{(i)}}\left(y_1^{(i-1)} - y_1^{(i)}\right)\left(y_2^{(i)} - f_{[2]}(\mathbf{x})\right)\phi_{\mathcal{N}(f_{[1]}^{[t]},\mathbf{B}_{11})}\left(f_1(\mathbf{x})\right)\phi_{\mathcal{N}(f_{[2]}^{[t]},\mathbf{B}_{22})}\left(f_{[2]}(\mathbf{x})\right)\mathrm{d}\mathbf{f}(\mathbf{x}) + \right. \\
&\left.\int_{y^{(i)}}^{y_1^{(i-1)}}\int_{y_2=y^{\mathrm{lb}_2}}^{y_2^{(i)}}\left(y_1^{(i)} - f_{[1]}(\mathbf{x})\right)\left(y_2^{(i)} - f_{[2]}(\mathbf{x})\right)\phi_{\mathcal{N}(f_1^{[t]},\mathbf{B}_{11})}\left(f_{[1]}(\mathbf{x})\right)\phi_{\mathcal{N}(f_{[2]}^{[t]},\mathbf{B}_{22})}\left(f_{[2]}(\mathbf{x})\right)\mathrm{d}\mathbf{f}(\mathbf{x})\right)
\end{aligned}
\tag{5.14}
$$

The computation of the EHVI comes back to the computation of the following integral $\int_{-\infty}^{b}(a - f_t(\mathbf{x}))\frac{1}{\sqrt{\mathbf{B}_{tt}}}\phi_{\mathcal{N}(0,1)}\left(\frac{f_t(\mathbf{x})-f_t^{[t]}}{\sqrt{\mathbf{B}_{tt}}})\right)\mathrm{d}f_t(\mathbf{x})$ which can be computed as follows:

$$\begin{aligned}\xi(a,b,\mu,\sigma) &= \int_{-\infty}^{b}(a - f_t(\mathbf{x}))\frac{1}{\sigma}\phi_{\mathcal{N}(0,1)}\left(\frac{f_t(\mathbf{x})-\mu}{\sigma}\right)\mathrm{d}f_t(\mathbf{x}) \\ &= \sigma\phi_{\mathcal{N}(0,1)}\left(\frac{b-\mu}{\sigma}\right) + (a-\mu)\Phi_{\mathcal{N}(0,1)}\left(\frac{b-\mu}{\sigma}\right)\end{aligned} \tag{5.15}$$

Therefore, the final expression of the EHVI comes back to:

$$\begin{aligned}&EHVI(\mathbf{x}) = \frac{1}{s}\sum_{j=1}^{s}\sum_{i=1}^{n_p+1} \\ &\left(\left(y_1^{(i-1)} - y_1^{(i)}\right)\left(\Phi_{\mathcal{N}(0,1)}\left(\frac{y_1^{(i)} - f_1^{[t]}(\mathbf{x})}{\sqrt{\mathbf{B}_{11}}}\right) - \Phi_{\mathcal{N}(0,1)}\left(\frac{y_1^{\mathrm{lb}} - f_1^{[t]}}{\sqrt{\mathbf{B}_{11}}}\right)\right)\right. \\ &\left(\xi(y_2^{(i)}, y_2^{(i)}, f_2^{[t]}, \sqrt{\mathbf{B}_{22}}) - \xi(y_2^{(i)}, y_2^{\mathrm{lb}}, f_2^{[t]}, \sqrt{\mathbf{B}_{22}})\right) \\ &+ \left(\xi(y_1^{(i-1)}, y_1^{(i-1)}, f_1^{[t]}, \sqrt{\mathbf{B}_{11}}) - \xi(y_1^{(i-1)}, y_1^{(i)}, f_1^{[t]}, \sqrt{\mathbf{B}_{11}})\right) \\ &\left.\left(\xi(y_2^{(i)}, y_2^{(i)}, f_2^{[t]}, \sqrt{\mathbf{B}_{22}}) - \xi(y_2^{(i)}, y_2^{\mathrm{lb}}, f_2^{[t]}, \sqrt{\mathbf{B}_{22}})\right)\right)\end{aligned} \tag{5.16}$$

By using a well established approach that is KDE for the estimation of the predictive distribution instead of approximating the piece-wise functions, this method to compute the EHVI is more accurate and is suitable for non-Gaussian predictive distributions. The accuracy of this approach is illustrated in Fig. 5.11. In the left figure, the correlation given by the predictive distribution (for instance, for $x^* = 0.85$, the full predictive covariance matrix is $\begin{pmatrix} 0.867 & -0.811 \\ -0.811 & 0.921 \end{pmatrix}$, see Fig. 5.12) of the model induces a difference between the EHVI computed with the assumption of independency and the one without this assumption. In the right figure, by changing the DoE, the predictive distribution of the LMC model provides a weak covariance between the objectives (for instance, for $x^* = 0.85$ the full predictive covariance matrix is $\begin{pmatrix} 0.0104 & -0.003 \\ -0.003 & 0.0096 \end{pmatrix}$, see Fig. 5.12), inducing a similar EHVI in the case of the assumption of independency to the one when the correlation is taken into account. However, it is interesting here to notice that the correlated EHVI using KDE comes back to the exact EHVI, highlighting the low bias in this approximation.

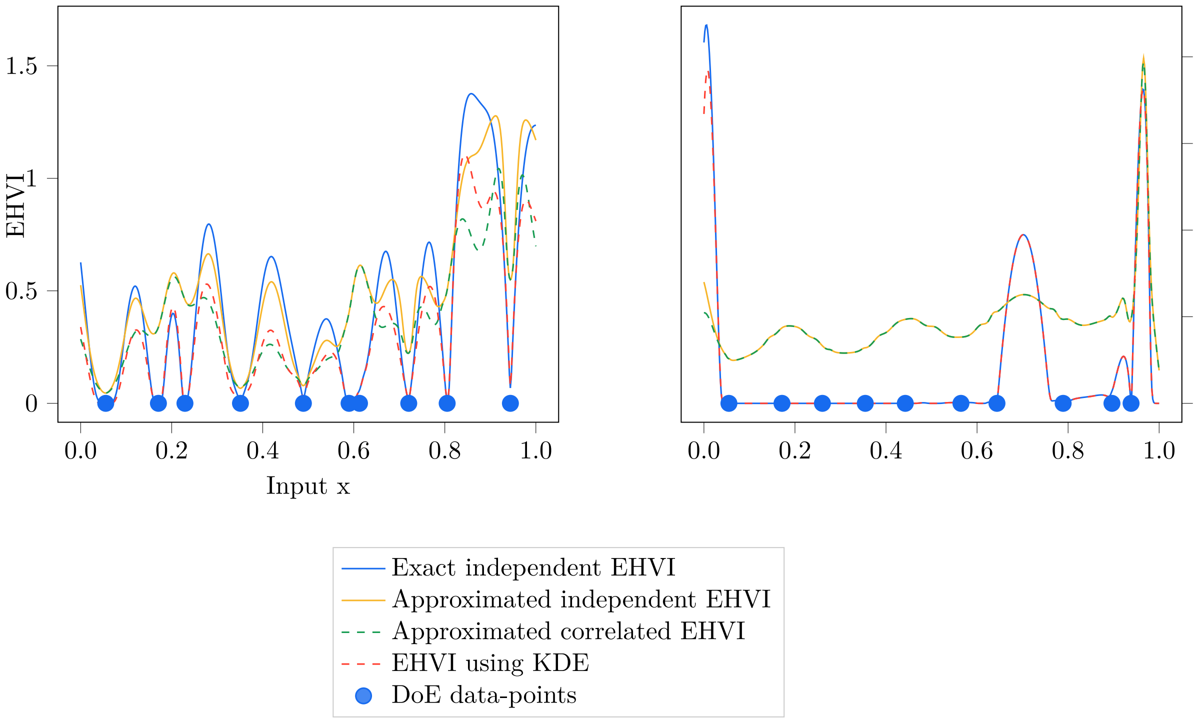


Fig. 5.11 Comparison on the design space of four different computations of the EHVI for the multi-objective problem in Eq. (5.8) with a DoE of 10 data-points. The blue colored curve corresponds to the exact computation of the EHVI with the assumption of independence between the objectives, the orange colored curve corresponds to the approximated computation of the EHVI [Shah and Ghahramani, 2016] with the assumption of independence between the objectives and the dashed green curve corresponds to the approximated computation [Shah and Ghahramani, 2016] of the EHVI with correlation between the objectives, and the dashed red curve corresponds to the proposed approach to compute the EHVI using KDE. In the right figure, a different DoE is used to train the LMC model, the obtained correlation in the predictive distribution using this DoE is not decisive and it can be seen that the EHVI using KDE comes back to the exact independent EHVI highlighting its accuracy.

The EHVI is directly used for unconstrained multi-objective problems. In the case of multi-objective problems with constraints, the EHVI is coupled with a constrained infill-criterion such as the Probability of Feasibility (PoF) and the Expected Violation (EV) in the same way as the Expected Improvement (EI) (these infill criteria are detailed in Chapter 3, Section 3.2.2.)

The two developed approaches that are MO-DGP to jointly model the objectives and the computational method for the correlated EHVI intervene at two different levels

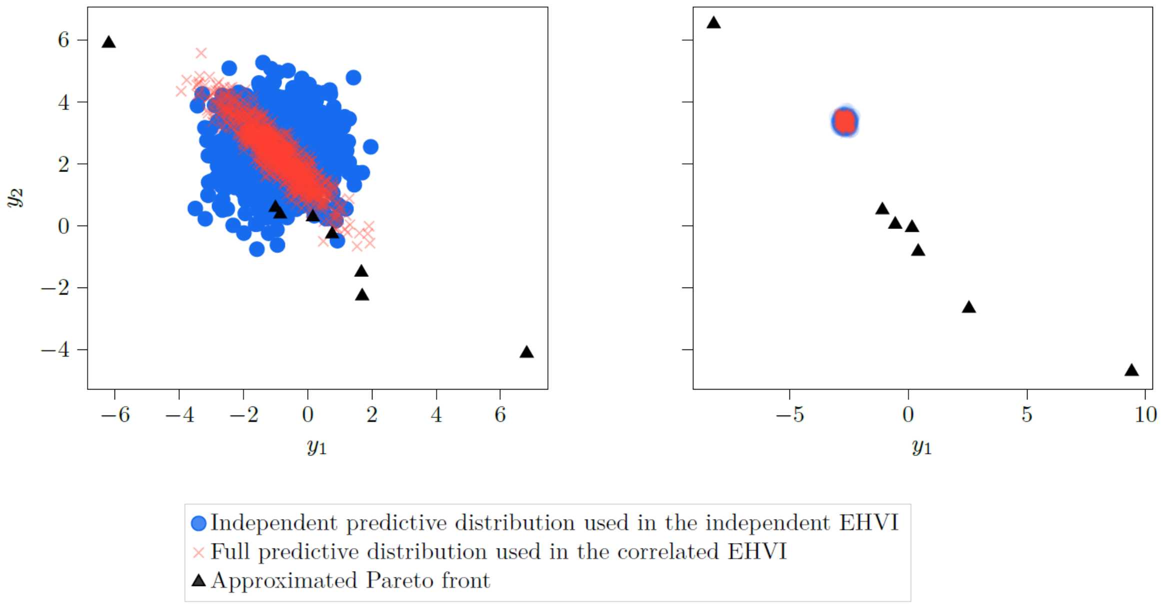


Fig. 5.12 Samples obtained by a LMC model trained on the two different DoEs (Fig. 5.11) in the objective space for a test point $x^* = 0.85$, with the full predictive distribution (red coded) used to compute the correlated EHVI, and with the predictive distribution using only the diagonal covariance (blue code) used to compute the independent EHVI. (left) The off-diagonal terms of the the predictive distribution covariance matrix given by the LMC model $\begin{pmatrix} 0.867 & -0.811 \\ -0.811 & 0.921 \end{pmatrix}$ induce a difference between the the correlated and the independent EHVI (left figure in Fig. 5.11). (right) The weak covariance of the predictive distribution given by the LMC model $\begin{pmatrix} 0.0104 & -0.003 \\ -0.003 & 0.0096 \end{pmatrix}$ induces a correlated EHVI similar to the independent EHVI (right figure in Fig. 5.11).

of the MO-BO algorithm. The next section, is devoted to numerical experiments to evaluate these two methodological approaches in MO-BO.

5.3 Numerical Experiments

In this section, experimentations are carried out in order to evaluate the performance of MO-BO with MO-DGP. A benchmark of analytical functions and a representative aerospace problem are considered to compare different MO-BO algorithms including the proposed MO-BO with MO-DGP. Each MO-BO algorithm consists in a coupling between a model and a computational approach of the EHVI (Table 5.1). Moreover, NSGA-II [Deb, 2001] a classic multi-objective evolutionary algorithm, is also run with the same number of evaluations as the MO-BO algorithms to highlight the interest of BO.

Table 5.1 The different MO-BO algorithms compared. For each MO-BO algorithm, a model (MO-DGP, independent GPs, or LMC) is coupled with a computational approach for the EHVI (independent objectives, correlated EHVI using KDE, correlated using the Gaussian approximation in [Shah and Ghahramani, 2016]).

Model	**EHVI computation**
MO-DGP	Independent
MO-DGP	Correlated EHVI using KDE
GPs	Independent
LMC	Independent
LMC	Correlated EHVI using KDE
LMC	Correlated using the Gaussian approximation in [Shah and Ghahramani, 2016] (Correlated GA)

For each problem of input dimension d, an initial DoE of $5 \times d$ is initialized using random Latin Hyper-cube Sampling (LHS) and a maximum of $10 \times d$ data-points are added using the MO-BO algorithms. To assess the robustness of the algorithms, 20 repetitions with different initial DoEs are performed. Details on the numerical setup are presented in Appendix D.

The obtained results are displayed using a table for each problem where the average and standard deviation (std) of the final Hyper-Volume (HV) (the higher the better) for each algorithm are presented. The hyper-volume indicator evolution over the iterations of the algorithms is also displayed using convergence curves with quartile bars. This allows to assess the speed of convergence of each algorithm, the quality of the final approximated Pareto front obtained, and the robustness to the initial DoE. Moreover, for each algorithm the final approximated Pareto front corresponding to the median repetition in terms of hyper-volume is plotted to assess the quality of the approximated fronts with respect to the exact Pareto front.

In the first part of this section, the different algorithms are compared on unconstrained analytical test functions. In the second part, a representative two-objective constrained aerospace problem is used to show the applicability of the proposed approach on representative physical problem.

Table 5.2 Performance of MO-BO on the 1-d test problem (values of the final hyper-volume obtained and its standard deviation on 20 repetitions) using MO-DGP, independent GPs, or LMC as a model, and with EHVI computed either with the assumption of independence or with correlation.

Model	EHVI computation	average HV	HV standard deviation
MO-DGP	**Independent**	**0.508**	**0.014**
MO-DGP	Correlated KDE	0.500	0.022
GPs	Independent	0.484	0.034
LMC	Independent	0.484	0.029
LMC	Correlated KDE	0.479	0.027
LMC	Correlated GA	0.478	0.054
NSGA-II		0.420	0.047

5.3.1 Analytical functions

A 1d test problem

In this section, the one-dimensional two-objective problem presented in Eq. (5.8) is optimized. The two objectives are negatively correlated in the objective space (Fig. 5.4). The hyper-volume is computed in the objective space within the rectangle ([-1,-4],[2.5,-0.5]).

The final hyper-volume for each algorithm is presented in Table 5.2 (for the sake of of clarity, the plots of LMC with correlated EHVI using GA are not represented, since it gives similar results to LMC with correlated EHVI using KDE). MO-BO with MO-DGP out-performs MO-BO with the other models in terms of the final HV (average HV for MO-BO with MO-DGP/EHVI computed independently: 0.508, and average HV for MO-BO with GPs and LMC: 0.484) and the robustness to the DoE (std dev of HV for MO-BO with MO-DGP/EHVI computed independently: 0.014, and std dev for MO-BO with GPs: 0.034 and with LMC: 0.029). Taking into account the correlation in the computation of the EHVI does not improve the results neither for MO-DGP nor for LMC where the final HV stays roughly the same (average HV for MO-BO with MO-DGP/EHVI correlated: 0.500).

The evolution of HV with respect to the number of added points is displayed in the graphic at the top of Fig. 5.13. It is interesting to notice that with only 6 added points, the results of MO-BO with MO-DGP is already better than the final results obtained by the other models. This highlights its speed of convergence compared to the other algorithms which is important in the case of expensive function evaluations. It illustrates also the interest of learning jointly the different objectives with a single multi-task DGP model. The graphic at the bottom of Fig. 5.13 displays the approximated Pareto front for the median repetition in terms of hyper-volume for different algorithms (for the sake of clarity, for the LMC only the approximated front obtained by the independent EHVI is displayed). The different solutions of the approximated Pareto front obtained by MO-BO with MO-DGP are well spreaded around and are all of rank 1 meaning they belong to the exact Pareto front. Alternatively, GP and LMC solutions are not all of rank 1.

Kursawe problem

The Kursawe is a 3-d two-objective problem [Kursawe, 1990]. The two objectives are defined as follows:

$$\begin{array}{ll} \text{min} & [f_1(\mathbf{x}), f_2(\mathbf{x})] \\ \text{s.t.} & -5 \leq x_i \leq 5 \qquad\qquad\qquad\qquad i = 1, \ldots, 3 \\ \text{with} & f_1(\mathbf{x}) = \sum_{i=1}^{2} \left(-10 \exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right) \\ \text{and} & f_2(\mathbf{x}) = \sum_{i=1}^{3} \left(|x_i|^{0.8} + 5 \sin\left(x_i^3\right)\right) \end{array} \tag{5.17}$$

From the analytic expression of the two objectives in Eq. (5.17), there is no clear correlation between the two objectives. The hyper-volume is computed in the objective space within the rectangle ([-22,-14],[-5,5]).

The final hyper-volume for each algorithm is presented in Table 5.3. For this problem, MO-BO with independent GPs performs better than all the other algorithms (average HV for MO-BO with independent GPs: 0.372, average HV for MO-BO with MO-DGP 0.350, average HV for MO-BO with LMC: 0.273). The deterioration of the results by the joint objective models that are MO-DGP and LMC may be explained by the fact that there is not a correlation in the objective space between the two objectives. Still, MO-BO with MO-DGP gives competitive results to MO-BO with GPs in this case compared to MO-BO with LMC. As in the previous test problem, the approach of computation of the EHVI is not decisive (average HV for MO-BO with MO-DGP/ EHVI computed independently: 0.350, average HV for MO-BO with MO-DGP/ correlated EHVI 0.354). For the same number of function evaluations NSGA-II achieves in average a HV of 0.173 which highlights the difficulty of the problem.

The evolution of HV with respect to the number of added points is displayed in the graphic at the top of Fig. 5.14. MO-BO with LMC struggles from the early iterations to improve the hyper-volume while MO-BO with independent GPs dominates the other

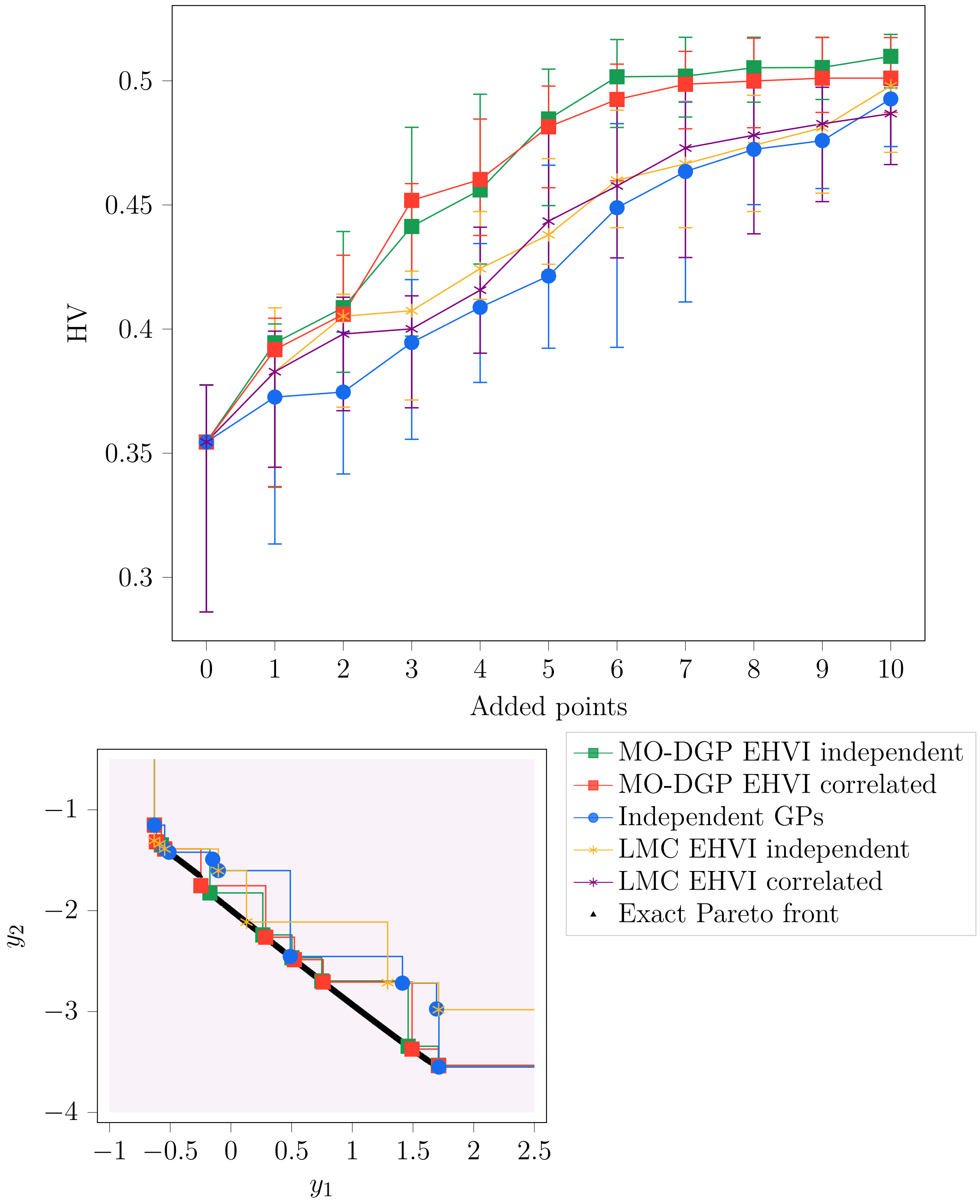


Fig. 5.13 Convergence curve and approximated Pareto fronts for the 1-d test problem, (top) hyper-volume evolution of each algorithm with respect to the number of added points with the MO-BO framework, (bottom) approximated Pareto front obtained in the median repetition in terms of hyper-volume of the different algorithms.

Table 5.3 Performance of MO-BO on Kursawe problem (values of the final hyper-volume obtained and its standard deviation on 20 repetitions) using MO-DGP, independent GPs, or LMC as a model, and with EHVI computed either with the assumption of independence or with correlation.

Model	**EHVI computation**	**average HV**	**HV standard deviation**
MO-DGP	Independent	0.350	**0.060**
MO-DGP	Correlated KDE	0.354	0.076
GPs	Independent	**0.372**	0.072
LMC	Independent	0.273	0.078
LMC	Correlated KDE	0.260	0.096
LMC	Correlated GA	0.255	0.090
NSGA-II		0.173	0.0322

algorithms. The graphic at the bottom of Fig. 5.14 displays the approximated Pareto front for the median repetition in terms of hyper-volume for different algorithms. Even after adding 30 data-points to the DoE, hence, increasing its size to 45 data-points, the approximated Pareto front of the different algorithms does not exceed 6 data-points. The MO-BO with GPs is able to capture the three discontinued parts of the exact Pareto-front with at least one data-point in each part. However, MO-BO with MO-DGP has more difficulty to capture the bottom part of the exact Pareto-front. MO-BO with LMC is unable to obtain a solution in the exact Pareto front.

DTLZ1-modified problem

A modified version of the DTLZ1 which is a multi-dimensional multi-objective problem [Deb et al., 2005] is considered in this section. This modified version of DTLZ1 yields to a concave Pareto front which is more difficult to approximate. The problem is considered with 5 dimensions and two objectives with the following expressions:

$$\begin{aligned} \text{min} \quad & [f_1(\mathbf{x}), f_2(\mathbf{x})] \\ \text{s.t.} \quad & 0 \leq x_i \leq 1 \qquad\qquad i = 1, \ldots, 5 \\ \text{with} \quad & f_1(\mathbf{x}) = -0.5x_1(1 + h(\mathbf{x})) \\ \text{and} \quad & f_2(\mathbf{x}) = -0.5(1 - x_1)(1 + h(\mathbf{x})) \\ \text{and} \quad & h(\mathbf{x}) = 100\left(5 + \textstyle\sum_{i=1}^{5}\left((x_1 - 0.5)^2 - \cos(2\pi(x_i - 0.5))\right)\right) \end{aligned} \tag{5.18}$$

This expression highlights the negative-correlation between the two objectives which depends on $h(x)$. The hyper-volume is computed in the objective space within the rectangle ([-600,-600],[25,25]).

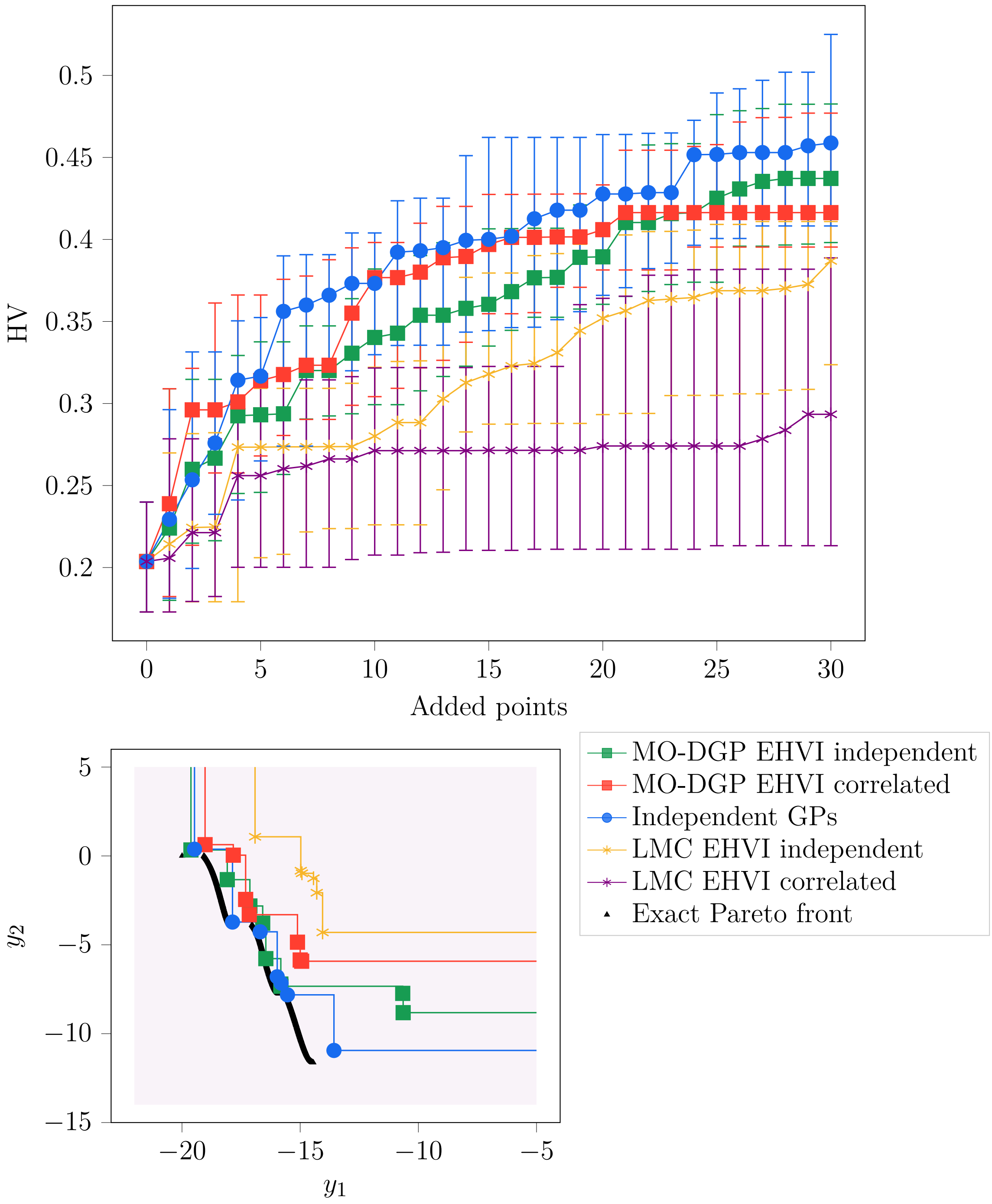


Fig. 5.14 Convergence curve and approximated Pareto fronts for the Kursawe problem, (top) hyper-volume evolution of each algorithm with respect to the number of added points with the MO-BO framework, (bottom) approximated Pareto front obtained in the median repetition in terms of hyper-volume of the different algorithms.

Table 5.4 Performance of MO-BO on the modified DTLZ 1 problem (values of the final hyper-volume obtained and its standard deviation on 20 repetitions) using MO-DGP, independent GPs, or LMC as a model, and with EHVI computed either with the assumption of independence or with correlation.

Model	**EHVI computation**	**average HV**	**HV standard deviation**
MO-DGP	Independent	0.381	**0.0031**
MO-DGP	Correlated	**0.382**	0.0035
GPs	Independent	0.365	0.0041
LMC	Independent	0.378	0.0039
LMC	Correlated	0.360	0.0736
NSGA-II		0.223	0.0275

The final hyper-volume for each algorithm is presented in Table 5.4. The MO-BO with a joint modeling approach (MO-DGP and LMC) out-performs MO-BO with independent GPs in terms of the final HV (average HV for MO-BO with MO-DGP: 0.381, and average HV for MO-BO with LMC: 0.378, an with GPs: 0.365) and the robustness to the DoE (std dev of HV for MO-BO with MO-DGP: 0.0031, and std dev for MO-BO with LMC: 0.0039, and with GPs: 0.0041). This illustrates the large correlation between the objectives that improves the modeling when using a joint model. Moreover, MO-DGP performs better than LMC which is explained by a more sophisticated exhibition of the correlation between the objectives. Taking into account the correlation between the objectives in the computation of the EHVI for MO-BO with MO-DGP does not change the final results. However, for MO-BO with LMC the final HV obtained is lower and less robust to the initial DoE. This is due to the off-diagonal values of the predicted covariance matrix that are not well-predicted.

The evolution of HV with respect to the number of added points is displayed in the graphic at the top of Fig. 5.15. From the 20-th added data-point, MO-BO with MO-DGP has already stood-out from the other algorithms. Actually, it is faster in terms of speed of convergence, with 25 added data-points it already outperforms MO-DGP with GPs. An interesting remark is that while MO-BO with LMC gives comparable results to MO-BO with MO-DGP at the end of the iterations, it is slower in terms of speed of convergence. In fact, its is outperformed by MO-BO with independent GPs in the early iterations. This may be explained by the fact that it need more data-points to learn the correlation between the objectives than MO-DGP as illustrated in Fig. 5.9. The graphic at the bottom of Fig. 5.15 displays the approximated Pareto front for the median repetition in terms of hyper-volume for different algorithms. The different

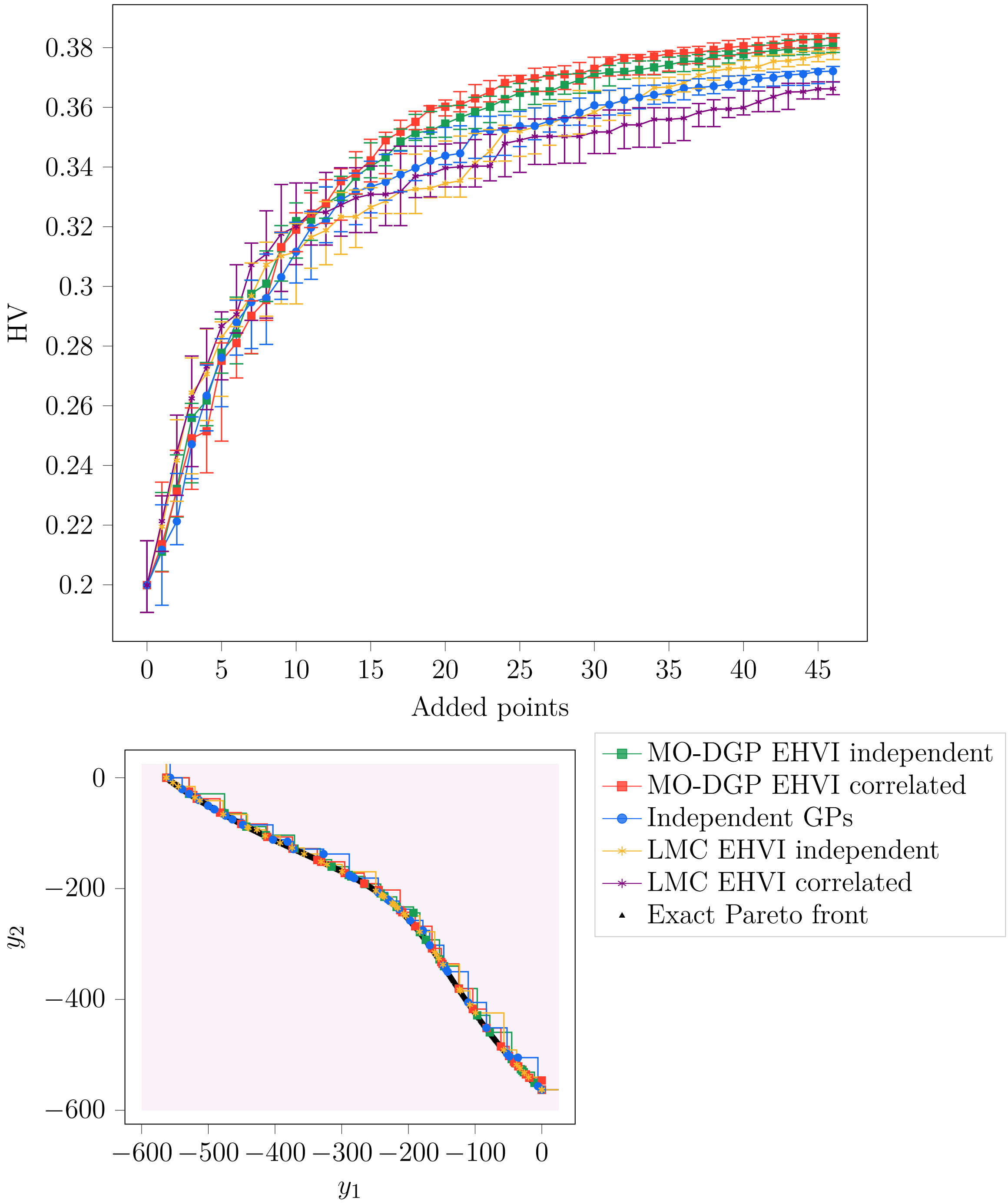


Fig. 5.15 Convergence curve and approximated Pareto fronts for the modified DLTZ1 problem, (top) hyper-volume evolution of each algorithm with respect to the number of added points with the MO-BO framework, (bottom) approximated Pareto front obtained in the median repetition in terms of hyper-volume of the different algorithms.

Table 5.5 Performance of MO-BO on ZDT 6 problem (values of the final hyper-volume obtained and its standard deviation on 20 repetitions) using MO-DGP, independent GPs, or LMC as a model, and with EHVI computed either with the assumption of independence or with correlation.

Model	**EHVI computation**	**average HV**	**HV standard deviation**
MO-DGP	Independent	**0.350**	**0.063**
MO-DGP	Correlated	0.265	0.1075
GPs	Independent	0.315	0.0699
LMC	Independent	0.061	0.0655
LMC	Correlated	0.046	0.0669
LMC	Correlated GA	0.043	0.0676
NSGA-II		0.	0.0

solutions of the approximated Pareto fronts reaches the exact Pareto front. Therefore, the differences in terms of hyper-volume are mainly due to the diversity and number of solutions in the approximated Pareto front.

Zitzler–Deb–Thiele 6 problem

The Zitzler–Deb–Thiele 6 (ZDT 6) problem is a 10-dimensionnal two-objective problem [Deb et al., 2005]. The expression of the two objectives is as follow:

$$
\begin{aligned}
\min \quad & [f_1(\mathbf{x}), f_2(\mathbf{x})] & \\
\text{s.t.} \quad & 0 \le x_i \le 1 & i = 1,\ldots,10 \\
\text{with} \quad & f_1(\mathbf{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1) & \\
\text{and} \quad & f_2(\mathbf{x}) = \varphi(\mathbf{x}) h\left(f_1(\mathbf{x}), \varphi(\mathbf{x})\right) & \\
\text{and} \quad & \varphi(\mathbf{x}) = 1 + 9\left(\frac{\sum_{i=2}^{10} x_i}{9}\right)^{0.25} & \\
\text{and} \quad & h(f_1(\mathbf{x}), \varphi(\mathbf{x})) = 1 - \sqrt{\frac{f_1(\mathbf{x})}{\varphi(\mathbf{x})}} &
\end{aligned}
\tag{5.19}
$$

In this problem, the correlation between the two objectives is more complicated than in the 1-D test problem and in the modified DTLZ 1 where the two objectives are the product of the same function. In fact, in this problem the second objective is written as a functional composition of the first objective and a second function $\varphi(x)$. The hyper-volume is computed in the objective space within the rectangle ([0,0],[1.1,1.1]).

The final hyper-volume for each algorithm is presented in Table 5.5. The high-dimensionality of the problem makes it difficult to optimize with few function evaluations. In fact, with the 150 function evaluations NSGA-II is unable to obtain a solution that lies in the considered rectangle for the hyper-volume in the 20 repetition runs. In the MO-BO algorithms, MO-DGP with EHVI computed independently gives the best results both in terms of the final HV (average HV : 0.35) and the robustness to the DoE (std dev of HV: 0.063). It is followed next by MO-BO with GPs (average HV: 0.315, std dev of HV: 0.069). In this problem, taking into account the correlation in the computation of the EHVI for MO-BO with MO-DGP degrades severely the performance, which drops to an average HV of 0.265. This may be explained by the fact that in the context of few data (50 → 130 data-points) compared to the dimension of the problem (10), the two first moments of the marginals of the predictive distribution, which are sufficient to compute the independent EHVI, need less data to be well predicted than the full predictive distribution used to computed the EHVI correlated. MO-BO with LMC performs poorly (average HV of 0.061), this is due to the complicated correlation between the objectives that cannot be captured by the linear relations involved in LMC. Moreover, the remark stated previously about the full predictive distribution needing more data to be well-predicted is also observed here, since the LMC with correlated EHVI (average HV 0.046) deteriorates the performance of LMC with independent EHVI.

The evolution of HV with respect to the number of added points is displayed in the graphic at the top of Fig. 5.16. The different MO-BO with joint modeling of the objectives (MO-DGP and LMC) have a slow start in the early iterations. This is due to the difficulty to exhibit the correlations between the objectives with few data. However, with enough data, MO-BO with MO-DGP using EHVI computed independently goes ahead MO-BO with GPs. Moreover, in terms of robustness to the initial DoE, MO-DGP using EHVI computed independently offers better results than MO-BO with independent GPs. MO-BO with MO-DGP using EHVI correlated is slower due to the reasons stated previously. The graphic at the bottom of Fig. 5.16 displays the approximated Pareto front for the median repetition in terms of hypervolume for different algorithms. The MO-BO with LMC struggles to obtain solutions within the rectangle on which the EHVI is computed. The approximated Pareto front by MO-DGP with EHVI computed independently is well spread around all the exact Pareto front, while the one obtained by MO-DGP with EHVI correlated does not reach the lower part of the exact front.

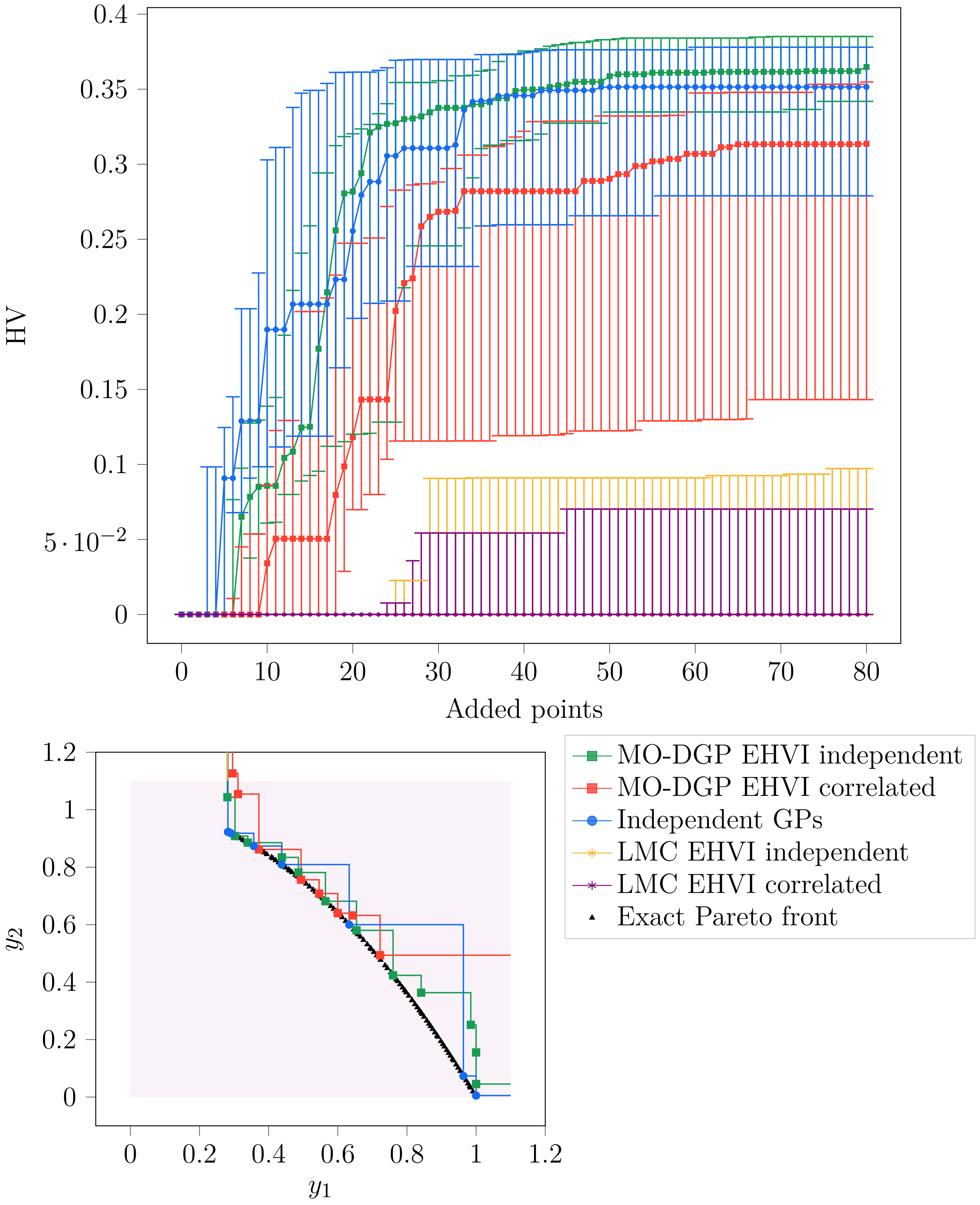


Fig. 5.16 Convergence curve and approximated Pareto fronts for the modified ZDT 6 problem, (top) hyper-volume evolution of each algorithm with respect to the number of added points with the MO-BO framework, (bottom) approximated Pareto front obtained in the median repetition in terms of hyper-volume of the different algorithms.

5.3.2 Multi-objective aerospace design problem

To confirm the interest of the MO-BO with MO-DGP for engineering applications, a representative aerospace vehicle design optimization problem is considered consisting of a two-objective constrained optimization of a solid-propellant booster.

Problem formulation

The optimization of two objectives for a solid propellant booster is considered (Fig 5.17). The objectives are :

- Minimization of the Gross Lift-off Weight (GLOW)
- Maximization of the change in velocity (ΔV)

In addition, four design variables are considered:

- Propellant mass: 2t $\leq m_{prop} \leq$ 20t
- Combustion chamber pressure: 5bar $\leq p_c \leq$ 500bar
- Throat nozzle diameter: 0.2m $\leq d_c \leq$ 1m
- Nozzle exit diameter: 0.5m $\leq d_s \leq$ 1.5m

The two objectives are directly correlated through Tsiolkovsky equation:

$$\Delta_V = g_0 \times I_{sp} \times \log\left(\frac{GLOW}{GLOW - m_{prop}}\right) \tag{5.20}$$

where g_0 is the standard gravity and I_{sp} is the specific impulse. Different constraints are also considered including a structural one limiting the combustion pressure according to the motor case, 6 geometrical constraints on the internal vehicle layout for the propellant and the nozzle, and a jet breakaway constraints concerning the throat nozzle diameter and the nozzle exit diameter. Making a total of 8 constraints.

$$\begin{array}{ll} \text{Minimize:} & -\Delta V(\mathbf{x}) \\ \text{Minimize:} & GLOW(\mathbf{x}) \\ \text{w.r.t:} & \mathbf{x} = [m_{prop}, p_c, d_c, d_s] \\ \text{s.t:} & \begin{cases} \text{1 structural constraint} \\ \text{6 geometrical constraints} \\ \text{1 jet breakaway constraints} \end{cases} \end{array} \tag{5.21}$$

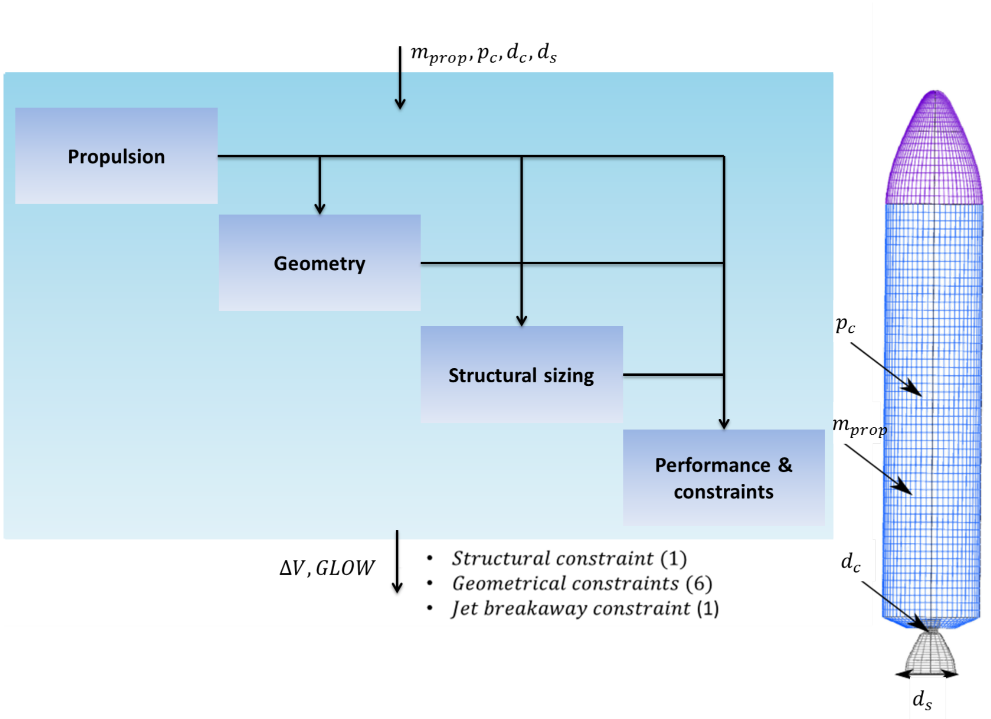

m_{prop}, p_c, d_c, d_s
Propulsion
Geometry
Structural sizing
Performance & constraints
$\Delta V, GLOW$
• *Structural constraint* (1)
• *Geometrical constraints* (6)
• *Jet breakaway constraint* (1)
p_c
m_{prop}
d_c
d_s

Fig. 5.17 Optimization problem of a solid-propellant booster engine. The formulation of the problem involves different disciplines (propulsion, geometry, structural sizing and performance). The problem considers the maximization of the velocity increment and minimization of the GLOW subject to 8 constraints.

Results

The different MO-BO algorithms used previously are applied to this problem with the considerations of the different constraints. In fact, in addition to the models for the objectives, a GP is used for each constraint. The EHVI is coupled to the Probability of Feasibility to handle the constraints (see Chapter 3, Section 3.2.2).

The final hyper-volume for each algorithm is presented in Table 5.6. The constraints make it difficult for an evolutionary algorithm such as NSGA-II to obtain results for few function evaluations (average HV of 0.087). For the MO-BO algorithms, MO-BO with joint modeling for the objectives (MO-DGP and LMC) outperforms MO-BO with independent GPs in terms of final HV (average HV for MO-DGP 0.473, for LMC 0.465, and for GPs 0.435) and also in terms of robustness to the DoE (HV std dev for MO-DGP 0.033, for LMC 0.0244, and for GPs 0.091). Therefore, the joint models are able to capture the physical correlation between the change in velocity (ΔV) and the gross lift-off weight. The correlation between the objectives is not complex enough to make a notable difference between LMC and MO-DGP as in ZDT 6 or the 1D test

Table 5.6 Performance of MO-BO on the aerospace test problem (values of the final hyper-volume obtained and its standard deviation on 20 repetitions) using MO-DGP, independent GPs, or LMC as a model, and with EHVI computed either with the assumption of independence or with correlation.

Model	**EHVI computation**	**average HV**	**HV standard deviation**
MO-DGP	Independent	**0.473**	0.033
MO-DGP	Correlated KDE	0.469	**0.0243**
GPs	Independent	0.435	0.0910
LMC	Independent	0.465	0.0244
LMC	Correlated KDE	0.457	0.0483
LMC	Correlated GA	0.452	0.0541
NSGA-II		0.087	0.0852

problem. For MO-BO with the correlated EHVI, the results are roughly the same using MO-DGP, however, when using LMC the results are less robust to the initial DoE with an increase of the HV std dev to 0.483. This means that the full predictive distribution given by MO-DGP is more adapted than the one obtained by LMC.

The evolution of HV with respect to the number of added points is displayed in the graphic at the top of Fig. 5.18. In the case of EHVI computed independently, the final result given by LMC is comparable to the one obtained by MO-DGP, however, LMC is slower to converge. In fact, with 20 added data-points MO-BO with MO-DGP already out-stands itself from the other algorithms. When computing the EHVI correlated, the improvement is slower in the early iterations. As stated previously, this is due to the fact that the full predictive distribution needs more data to be well-predicted. The graphic at the bottom of Fig. 5.18 displays the approximated Pareto front for the median repetition in terms of hyper-volume for different algorithms. In terms of diversity and number of solutions in the approximated Pareto front the different algorithms show comparable results. However, MO-BO with GP struggles to reach its solutions to the exact Pareto front compared to the MO-BO with a joint model for the objectives where the majority of the solutions are on the exact Pareto front.

5.3.3 Synthesis of the results

The results obtained in these numerical experiments allowed us to draw conclusions in terms of the chosen model and also the approach to compute the EHVI in MO-BO.

- **In terms of the chosen model:**

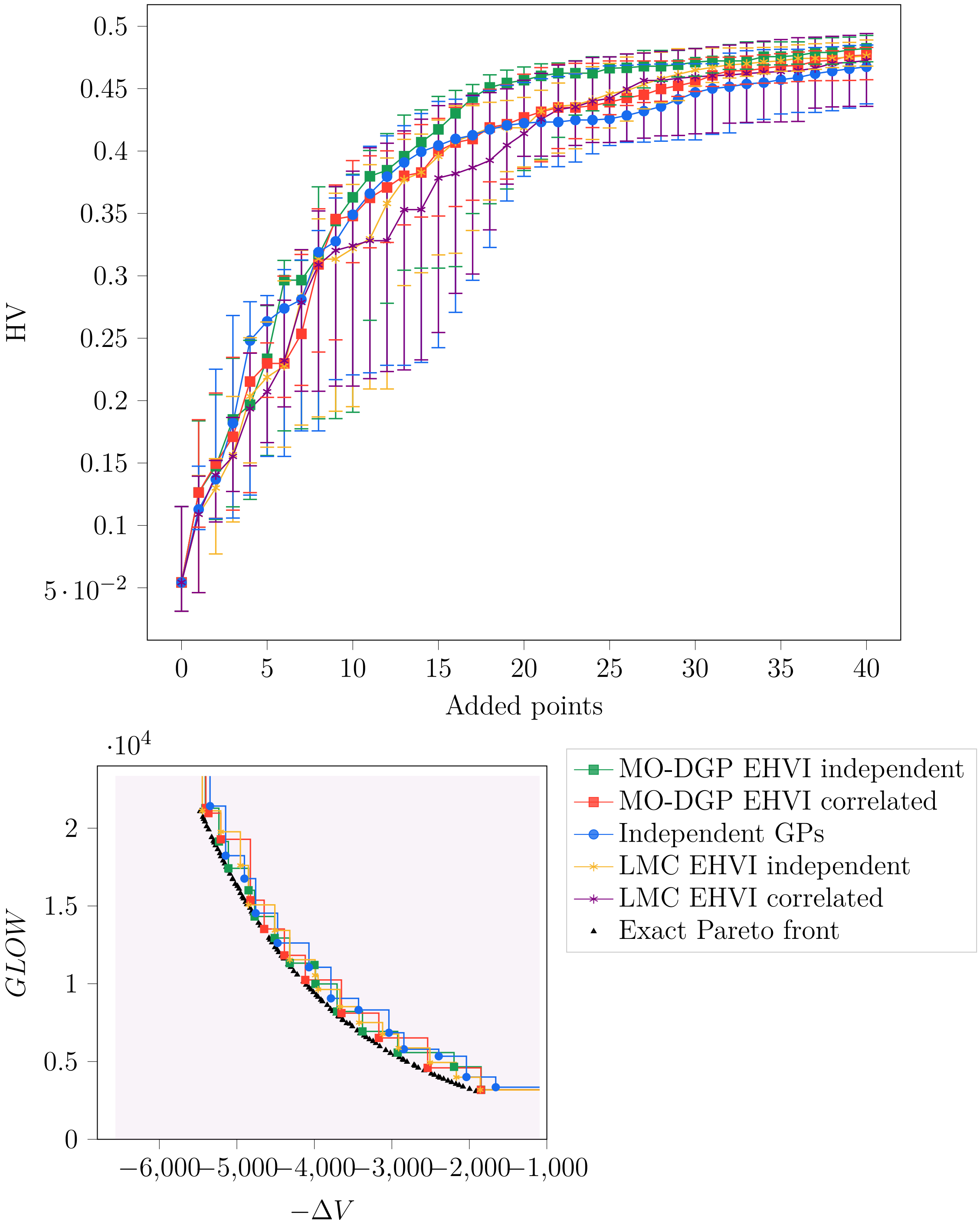


Fig. 5.18 Convergence curve and approximated Pareto fronts for the aerospace problem, (top) hyper-volume evolution of each algorithm with respect to the number of added points with the MO-BO framework, (bottom) approximated Pareto front obtained in the median repetition in terms of hyper-volume of the different algorithms.

  - **MO-BO with MO-DGP:** for problems with correlations between the objectives (1D problem, DTLZ 1, and the aerospace problem) and even with complex correlations (ZDT 6) the model is able to take advantage of this correlation and outperforms the other algorithms in terms of the final HV obtained, the robustness to the initial DoE, and to the speed of convergence. For the problem where there is no correlation between the objectives (Kursawe) it still gives comparable results to MO-BO with GPs. However, the major drawback of MO-DGP compared to the other models is its complexity and training time.
  - **MO-BO with LMC:** for problems with correlations between the objectives (1D problem, DTLZ 1, and the aerospace problem) the model gives competitive results to MO-BO with MO-DGP, but, with a slower speed of convergence. For complex correlations between the objectives (ZDT 6) and no correlation (Kursawe), it is largely outperformed by MO-BO with MO-DGP and with independent GPs.
  - **MO-BO with independent GPs:** is the classic approach and as expected it performs decently in the different configurations. However, not taking into account correlations between the objectives makes it at a disadvantage to the compared algorithms in multiple problems (1D problem, DTLZ 1, and the aerospace problem) resulting in less performing results.

- **In terms of the EHVI computational approach:**
  - For the different test problems, using the correlated EHVI in an algorithm did not improve clearly the performance. However, it happened that it deteriorates the performance. This is explained by the fact that, unlike independent EHVI which uses the two first moments of the marginal predictive distribution, the correlated EHVI uses the full predictive distribution which may need more data-points to be well predicted.
  - Taking into account the correlation in the EHVI for MO-BO with MO-DGP yields to comparable results for the different test problems except for ZDT 6. However, for MO-BO with LMC, the results when using the correlated EHVI are deteriorated more frequently (ZDT 6, DTLZ 1, aerospace problem, Kursawe). This means that the full predictive distribution obtained by MO-DGP is better calibrated than the one obtained by LMC.
  - The correlated EHVI computed using KDE and the one computed using the Gaussian approximation proposed in [Shah and Ghahramani, 2016] yields

approximatively to the same result (for Gaussian predictive distributions obtained by LMC) with a slight advantage to the approach using KDE.

5.4 Conclusions

Multi-objective optimization considers antagonistic performance to optimize. In this context, there is usually a strong negative correlation between the objectives especially around the Pareto front. However, classic MO-BO approaches consider the objectives independently and do not take advantage of potential correlations. Moreover, the popular infill criterion used that is the EHVI is computed with the assumption of independency between the objectives.

In this chapter, a DGP based model which allows a joint modeling of the objectives in order to exhibit a potential correlation has been developed. In addition, an accurate computing approach for the EHVI without the assumption of objective independency is proposed.

Unlike in Chapter 4, where the intermediate layers of DGPs are hidden and used as non-parametric Bayesian mapping of the input space to handle non-stationary problems, in this chapter, MO-DGP is a DGP based-model where each layer corresponds to an objective, therefore, inducing interpretability for the intermediate layers and also making DGP a multi-task model. Moreover, the nodes are connected with undirected nodes and are fully connected with each others. While this increases the complexity of the model, it also increases its power of representation. In fact, each layer takes the other objectives as inputs to improve its prediction. The ELBO of this model has been derived and a Gibbs sampling approach is proposed for estimating it. This model proves to better predict correlated functions than independent GPs or LMC.

To use MO-DGP within a MO-BO framework, it has to be coupled with the EHVI. The EHVI has been adapted to the case of correlated objectives in [Shah and Ghahramani, 2016]. However, this computational approach is based on an approximation which is usually not tight and induce important dissimilarities. Moreover, it is not adapted to the case where the predictive distribution of the model is not a multi-variate Gaussian which is usually not the case for MO-DGP. Instead, another computational approach has been proposed for the EHVI where it is the predictive distribution that is approximated using KDE. This computational approach proves to be more accurate.

Experimentations on analytical and on an aerospace design problem were carried out to prove the interest of using MO-DGP and the correlated EHVI within a MO-BO framework. These numerical experiments highlight the fact that MO-BO with MO-

DGP outperforms MO-BO with independent GPs and with LMC over different test problems in terms of the final hyper-volume obtained, the robustness to the initial DoE, and the speed of convergence. However, the correlated EHVI does not prove to be decisive. This is caused by the fact that the full predictive distribution needs more data to be well-approximated.

In this chapter, only the two-objective case was considered. MO-DGP is formulated for multiple objectives and it can be used for more than two. However, it is not conceivable to use it in a many-objective context due to its complexity. Therefore, it would be interesting to see if the interest of MO-DGP shown in the two-objective case is confirmed in the three-objective case. For the EHVI, the computing approach followed is specific to the two-objective case. However, there are exact approaches to compute the EHVI in the case of more objectives with the assumption of independency [Hupkens et al., 2015]. To compute the correlated EHVI in that case, the proposed approach using KDE to approximate the predictive distribution still holds.

In the numerical experiments carried in this chapter, the correlated EHVI does not improve the obtained results. Numerical experiments with larger initial DoEs would allow the model to better approximate the full predictive distribution, hence, the correlated EHVI may be more decisive in this context.

MO-DGP model was used in the context of multi-objective optimization. However, it can also be seen as a general multi-task model where there is no known hierarchy between the functions in order to improve the prediction and uncertainty quantification that would be used in a context of analysis.

For known hierarchy between the functions, one of the classic approaches are multi-fidelity modeling and DGP can also be used in this context as it is developed in the next part of this thesis.

Part III

Multi-fidelity analysis

Chapter 6

Multi-fidelity analysis using Deep Gaussian Processes

“ The question you need to ask is not "Is the model true?" (it never is) but "Is the model good enough for this particular application?".”

Alberto Luceño and Maria del Carmen Paniagua-Quiñones (2009)

Chapter goals

- Proposition of an improved training technique for the multi-fidelity deep Gaussian process model.
- Benchmark of Gaussian process-based multi-fidelity approaches with identically defined fidelity input spaces on aerospace test cases.
- Proposition of a deep Gaussian process multi-fidelity model for different input domain definitions.
- Assessment of the proposed model performance on analytical test cases and engineering design problems.

$\mathcal{CH}_6$

The analysis of complex systems is usually characterized by different levels of analysis in terms of the design variables taken into account and the complexity of the response to be modeled. These levels of analysis depend on the considered design phase. In the early design stage, the analysis is not as thorough as in the detailed design phase or the manufacturing phase. This yields to the use of different physical models, each one characterized by its own accuracy and computational cost. Generally, the more precise is the model, the more computationally intensive it is. In the early design phase, computationally efficient (but imprecise) physical models called Low-Fidelity

(LF) models are often used in order to explore a large design space. In the detailed design phase, High-Fidelity (HF) models are employed to capture complex physical phenomena and to refine the response obtained but with an intensive computational cost.

In the previous part of this manuscript, only the HF data were used to construct the machine learning regression models. However, due to the computational cost of the HF model, its data are scarce and may be insufficient to capture the response of the unobserved function in the whole design space. To overcome this issue, multi-fidelity methods enrich the HF data with LF data. Actually, the correlations between the LF and HF models are exhibited within a multi-fidelity model enabling the improvement of the high-fidelity prediction.

Gaussian Processes (GPs) are a popular approach for multi-fidelity modeling (see Chapter 3, Section 3.4). One of the recent multi-fidelity approaches based on GPs is the Multi-Fidelity Deep Gaussian Process (MF-DGP) [Cutajar et al., 2019]. It has the advantage of describing non-linear correlation between fidelities by considering a DGP in which each layer corresponds to a fidelity level. MF-DGP is based on the sparse DGP approximation proposed in [Salimbeni and Deisenroth, 2017]. However, one of the limitations of MF-DGP is that the inputs of the intermediate layers are the combination of the data-set in the original input space with their corresponding function evaluation. Therefore, freely optimizing the inducing inputs is not adequate, as they are related by a deterministic mapping (corresponding to the engineering model). In [Cutajar et al., 2019], the inducing inputs are fixed to arbitrary values, limiting the power of representation of the model. Another limitation of MF-DGP as well as the other GP-based approaches presented in Chapter 3, Section 3.4.1 is that it assumes that the input spaces of all the fidelities are identically defined in terms of input variables. However, this is not always the case. Actually, due to either different modeling approaches from one fidelity to another, or an omission of some variables in the lower fidelity models for instance, the input spaces may differ in the form of the parameterization and also in the dimensionality.

The contribution of this chapter is two-fold: addressing the limits of the induced inputs optimization in MF-DGP and proposing a DGP-based model for multi-fidelity in the case of different input spaces. Specifically, the chapter is decomposed into two main sections. In the first section (Section 6.1), a new training approach for MF-DGP is proposed to train the inducing inputs. Then, this improvement of MF-DGP is evaluated with respect to the different approaches reviewed in Chapter 3, Section 3.4.1 on an extensive benchmark of analytical and aerospace design problems. In the second section (Section 6.2), a new model based on MF-DGP is proposed for multi-fidelity problems with different input spaces. This is accomplished by a new model formulation of MF-DGP incorporating a mapping between the different fidelity input spaces in a non-parametric way. Similarly, the proposed approach is compared to literature techniques on analytical and engineering design problems.

6.1 Multi-fidelity with identically defined fidelity input spaces

In this section, the input spaces of all fidelities are identically defined in terms of input variables. The first subsection (Section 6.1.1) develops a training approach for the inducing inputs of MF-DGP in order to improve its modeling capability. In the second subsection (Section 6.1.2), analytical and aerospace benchmark problems are presented to compare the proposed MF-DGP improvement to the standard MF-DGP and also to the other GP based multi-fidelity approaches presented in Chapter 3, Section 3.4.1 in different scenarios of availability of HF data and dimensionality of the inputs.

6.1.1 Improvement of Multi-Fidelity Deep Gaussian Process Model (MF-DGP)

Let $(\mathbf{X}_t, \mathbf{y}_t)$ be the couple of inputs/outputs of each fidelity $t \in \{1, \ldots, n_{\text{fi}}\}$, where n_{fi} is the number of fidelities sorted in an increasing order of fidelities *i.e.* $(\mathbf{X}_1, \mathbf{y}_1)$ corresponds to the lowest fidelity data-set and $(\mathbf{X}_{n_{\text{fi}}}, \mathbf{y}_{n_{\text{fi}}})$ to the highest fidelity data-set. Let d and n_t be respectively the dimension of the input data and the size of the training data at fidelity t.

MF-DGP [Cutajar et al., 2019] described in details in Chapter 3, Section 3.4.1, is a DGP in which each layer corresponds to a fidelity. Moreover, the GP at each layer t depends not only on the input data at this fidelity $\mathbf{X}_t$ but also on the previous fidelity GP $t-1$ evaluation of the same input data $\mathbf{X}_t$ (Fig. 3.15). Therefore, the input dimension for all the layers, except the first one, are augmented by the output scalar response of the previous layer. The structure of this augmented input space $[\mathbf{X}_t, f_{t-1}(\mathbf{X}_t]$ is particular since its last dimension depends on the d first dimensions. This yields to a difficulty when training MF-DGP. Actually, MF-DGP inference follows the variational approximation used in [Salimbeni and Deisenroth, 2017], which leads to the following ELBO:

$$\mathcal{L} = \sum_{t=1}^{n_{\text{fi}}} \sum_{i=1}^{n_t} \mathbb{E}_{q(f_{[t]}^{(i),t})} \left[\log p(y_t^{(i)} | f_{[t]}^{(i),t}) \right] - \sum_{t=1}^{n_{\text{fi}}} \mathbb{KL} \left[q(\mathbf{u}_{[t]}) || p(\mathbf{u}_{[t]} | \mathbf{Z}_{[t-1]}) \right] \tag{6.1}$$

where $f_{[j]}^{(i),t})$ is the evaluation of the observation i of the inputs at fidelity t ($\mathbf{X}_t$) by the GP at layer j ($f_{[j]}(\cdot)$), $\mathcal{KL}$ corresponds to the KL divergence, and the bound is optimized with respect to the inducing inputs $\{\mathbf{Z}_{[t]}\}_{t=1}^{n_{\text{fi}}}$, the variational parameters $\{\boldsymbol{\theta}_{q(\mathbf{u}_{[t]})}\}_{t=1}^{n_{\text{fi}}}$ of the variational distributions $\{q(\mathbf{u}_{[t]}) = \mathcal{N}(\mathbf{u}_{[t]} | \bar{\mathbf{u}}_{[t]}, \boldsymbol{\Gamma}_{[t]}\}_1^{n_{\text{fi}}}$, and the GP hyperparameters at each layer $\{\boldsymbol{\theta}_{[t]}\}_{t=1}^{n_{\text{fi}}}$. However, the induced inputs at a layer t for $2 \leq t \leq n_{\text{fi}}$ lie in the augmented input space. Therefore, freely optimizing these inducing inputs does not take into account the specificity of the augmented input space where the last dimension depends on the d first components (Fig. 6.1). To avoid this issue, [Cutajar et al., 2019] proposed to fix the inducing inputs during the training to arbitrary values. By choosing the d first components of the arbitrary induced inputs from the observed inputs at the previous layer and the last component as the corresponding output, this approach keeps a dependence between the coordinates of the induced inputs. However, not optimizing the induced inputs limits the capacity of the model.

An optimization framework is proposed in order to optimize the inducing inputs in the augmented input space and hence increasing the learning capability of MF-DGP. For that, the augmented inducing inputs $\{\mathbf{Z}_{[t]}\}_{t=2}^{n_{\text{fi}}}$ are constrained as follows:

$$\mathbf{Z}_{[t]} = \left[\mathbf{Z}_{[t],1:d}, \hat{f}_{[t-1]} \left(\mathbf{Z}_{[t]}^{t-1} \right) \right]; \forall 2 \leq t \leq n_{\text{fi}} \tag{6.2}$$

where $\hat{f}_{[t-1]}(\cdot)$ corresponds to the posterior mean of the previous layer and $\mathbf{Z}_{[t]}^{i}$ is defined with the following recursive equation:

$$\mathbf{Z}_{[t]}^{i} = \left[\mathbf{Z}_{[t],1:d}, \hat{f}_{[i-1]} \left(\mathbf{Z}_{[t]}^{i-1} \right) \right]; \forall 2 \leq i \leq t \text{ and } 2 \leq t \leq n_{\text{fi}} \tag{6.3}$$

and

$$\mathbf{Z}_{[t]}^{1} = \mathbf{Z}_{[t],1:d} \tag{6.4}$$

This constraint allows to express $\mathbf{Z}_{[t],d+1}$ as a function of $\mathbf{Z}_{[t],1:d}$ and hence collapses the $d+1$ coordinate of the inducing inputs. This is accomplished by propagating $\mathbf{Z}_{[t],1:d}$ from the first layer whose input space is not augmented (of dimension d) then using at each inner layer the previous posterior mean evaluation to augment $\mathbf{Z}_{[t],1:d}$ as expressed in Eq. (6.3) until reaching the layer of fidelity t (Fig. 6.2). Therefore, it allows the

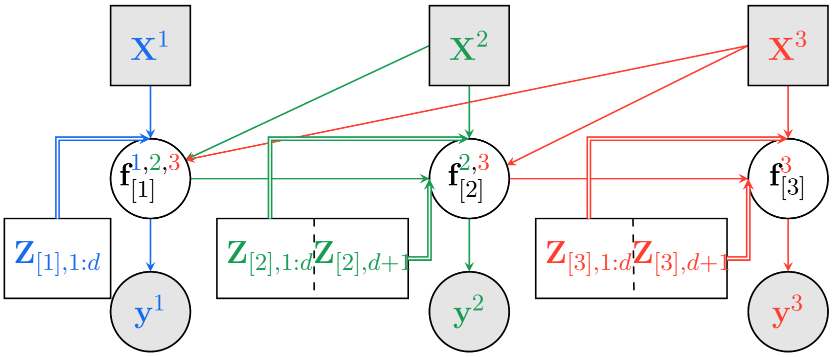


Fig. 6.1 Representation of the induced inputs in MF-DGP. The regular lines and double lines correspond respectively to the observed inputs and induced input dependences. Except for the first layer, the induced inputs lie in an augmented input space of $d+1$ dimensions where the $d+1$ coordinate depends on the d first coordinates making it non-suitable to freely optimize the induced inputs. [Cutajar et al., 2019] fix $\mathbf{Z}_{[t],1:d} = \mathbf{X}_{t-1}$ and $\mathbf{Z}_{[t],d+1} = \mathbf{y}_{t-1}$ along the training.

dependence in the augmented inducing input space to be kept during the training by optimizing only the first d coordinates and inferring the $d+1$ component using the propagation mechanism (Eq. (6.2), Eq. (6.3)).

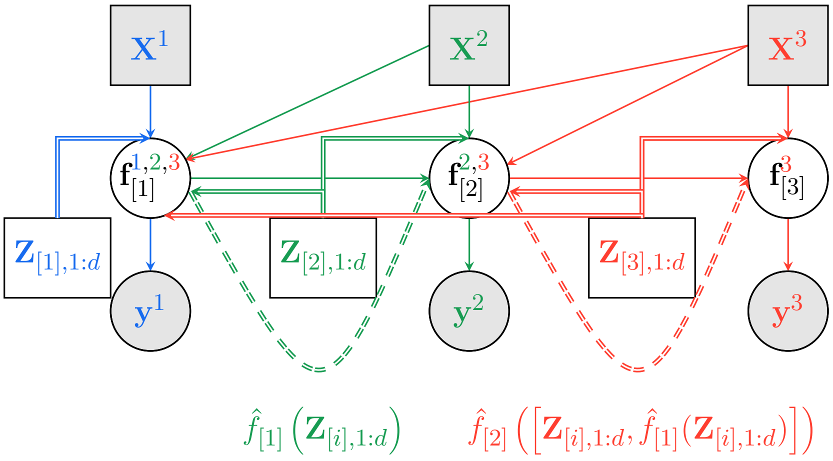


Fig. 6.2 Graphical representation of the proposed induced input optimization framework for MF-DGP. $\mathbf{Z}_{[t],d+1}$ is collapsed by constraining it to the result of the propagation of $\mathbf{Z}_{[t],1:d}$ through the posterior mean prediction $\hat{f}_i(\cdot)$ of the previous fidelity layers $1 \leq i \leq t$. This allows to freely optimize the d first coordinates of the inducing inputs while keeping a dependence with the $d+1$ component.

Algorithm 4 summarizes the proposed optimization of the ELBO for the MF-DGP using the optimization framework for the inducing inputs, as well as using the optimizer based on the natural gradient described in Chapter 4, Section 4.1.1.

```
Algorithm 4: MF-DGP ELBO optimization
Initialization of the number of maximum iterations. $\mathit{maxiter}$
Initialization of the hyperparameters of the kernels $\{\boldsymbol{\theta}_{[t]}\}_{t=1}^{n_{\text{fi}}}$.
$q(\mathbf{u}_{[t]}) \leftarrow \mathcal{N}\left(\mathbf{y}_t, \boldsymbol{\Gamma}_{[t]}\right), \forall 1 \leq t \leq n_{\text{fi}}$. For optimization stability, $\Gamma_{[t]}$ is initialized
  at low values.
$\mathbf{Z}_{[t],1:d} \leftarrow \mathbf{X}_t, \forall 1 \leq t \leq n_{\text{fi}}$
$j \leftarrow 0$
while $0 \leq j \leq \mathit{maxiter}$ do
    $\mathbf{Z}^{1}_{[t]} \leftarrow \mathbf{Z}_{[t],1:d}; \forall 1 \leq t \leq n_{\text{fi}}$
    $\mathbf{Z}^{i}_{[t]} \leftarrow \left[\mathbf{Z}_{[t],1:d}, \hat{f}_{i-1}\left(\mathbf{Z}^{i-1}_{[t]}\right)\right]; \forall 2 \leq i \leq t$ and $2 \leq t \leq n_{\text{fi}}$
    $\mathbf{Z}_{[t]} \leftarrow \left[\mathbf{Z}_{[t],1:d}, \hat{f}_{[t-1]}\left(\mathbf{Z}^{t-1}_{[t]}\right)\right]; \forall 2 \leq i \leq n_{\text{fi}}$
    $ELBO, \{\mathbf{Z}_{[t],1:d}\}_{t=1}^{n_{\text{fi}}}, \{\boldsymbol{\theta}_{[t]}\}_{t=1}^{n_{\text{fi}}} =$
      Adam step $\left(ELBO, \{\boldsymbol{\theta}_{[t]}\}_{t=1}^{n_{\text{fi}}}, \{\mathbf{Z}_{[t],1:d}\}_{t=1}^{n_{\text{fi}}}\right)$
    $ELBO, \{\boldsymbol{\theta}_{q(\mathbf{u}_{[t]})}\}_{t=1}^{n_{\text{fi}}} \leftarrow$ Natural step $\left(ELBO, \{\boldsymbol{\theta}_{q(\mathbf{u}_{[t]})}\}_{t=1}^{n_{\text{fi}}}\right)$
    $j \leftarrow j+1$
end
return $ELBO, \{\mathbf{Z}_{[t],1:d}\}_{t=1}^{n_{\text{fi}}}, \{\boldsymbol{\theta}_{[t]}\}_{t=1}^{n_{\text{fi}}}, \{\boldsymbol{\theta}_{q(\mathbf{u}_{[t]})}\}_{t=1}^{n_{\text{fi}}}$
```

In the next section, MF-DGP using this optimization approach of the inducing inputs along with the use of natural gradients for the variational distributions is compared to regular MF-DGP and other multi-fidelity GP approaches on analytical multi-fidelity problems as well as a benchmark of aerospace problems.

6.1.2 Numerical experiments of the improved MF-DGP on analytical and aerospace multi-fidelity problems

MF-DGP trained using Algorithm 4, henceforth, referenced as MF-DGP improved is compared to regular MF-DGP to highlight the increase of its learning capacity as well as the other multi-fidelity approaches presented in Chapter 3, Section 3.4.1. The multi-fidelity methods are compared with respect to the three metrics used throughout this manuscript: the coefficient of determination (R2), the Root Mean Square Error (RMSE) and the Mean Negative test LogLikelihood (MNLL). A large HF test set is used to compute the metrics. For GP-based multi-fidelity methods, it is important to compare the prediction accuracy metrics (R2 the higher, the better and RMSE, the lower, the better) and the predictive uncertainty of the multi-fidelity model to accurately explain the test set (MNLL, the lower, the better). Indeed, the predictive uncertainty of GP-based techniques is often used either for model refinement, uncertainty propagation or optimization (as seen in Bayesian Optimization in Chapter 4). Therefore, the multi-fidelity model has to be accurate both in terms of prediction and of uncertainty model associated to the prediction.

Analytical numerical experiments

The proposed approach has been compared with other GP multi-fidelity based approaches AR1, NARGP, and regular MF-DGP in the same benchmark of 4 analytical functions used in [Cutajar et al., 2019] (Currin, Park, Borehole, Branin, see Appendix C). To assess the robustness of the methods, 20 repetitions on different Design of Experiments (DoE) have been performed. The obtained results are displayed on Table 6.1. The improved MF-DGP shows the best results in prediction accuracy and in uncertainty quantification with a large robustness to different DoE. On the Borehole problem, it gives comparable results to the AR1. This is explained by the fact that the Borehole problem shows strong linearity between the fidelities. The results given by the improved MF-DGP are better than the regular MF-DGP on the four problems, in prediction accuracy, uncertainty quantification, and robustness to DoE, illustrating the improvement of the learning capacity of MF-DGP.

In the next section, experimentation on aerospace multi-fidelity applications using GP-based multi-fidelity approaches including the improved MF-DGP is carried out in different scenarios of data availability in order to conclude on the efficiency of each approach with respect to the problem at hand and the considered scenario.

Application to an aerospace multi-fidelity benchmark

Seven techniques are compared: a GP using only the HF dataset (GP HF), the auto-regressive model (AR1) with inference scheme introduced by Kennedy and O'Hagan [Kennedy and O'Hagan, 2000], the co-kriging linear model of corregionalization (LMC), the non-linear auto-regressive multi-fidelity gaussian process without nested DoE (NARGP) and with nested DoE (NARGP-nest), multi-fidelity Deep Gaussian Process (MF-DGP) (more details on these approaches in Chapter 3, Section 3.4.1), and multi-fidelity Deep Gaussian Process with the proposed improvements (MF-DGP improved).

For the considered problems, several sizes of design experiments are considered for the HF dataset to analyze the influence availability of HF data. In order to assess the robustness of the methods to the LF and HF datasets, the experimentations are repeated on 20 different DoEs using Latin Hypercube Sampling (LHS) for each size of the dataset. For the nested DoE of NARGP-nest, the same DoE as other techniques

Table 6.1 Performance of the different multi-fidelity models on 4 different problems using 20 repetitions with different DoE. R^2 refers to the R squared error, MNLL to the mean negative test log likelihood, RMSE to the root mean squared error, and std to the standard deviation. Currin and Park $(d = 2)$ problems are modeled with 12 input data on the LF and 5 input data on the HF. Borehole $(d = 8)$ is modeled with 60 input data on the LF and 5 input data on the HF. Branin $(d = 2)$ is used with 80 input data on the lower fidelity, 30 on the medium fidelity and 5 input data on the higher fidelity.

Approach	AR1				NARGP			
Functions	R^2	MNLL	RMSE	std RMSE	R^2	MNLL	RMSE	std RMSE
Currin	0.8994	46.400	0.7355	0.2131	0.8743	123.398	0.8147	0.2572
Park	0.9831	299.463	0.5809	0.2100	0.8792	213.759	1.0986	1.2420
Borehole	**0.9998**	**-3.777**	**0.0047**	**0.00097**	0.9968	-2.503	0.0226	0.0120
Branin	0.1944	15810.3	0.1765	0.0658	0.0241	4285.26	0.2034	0.0344

Approach	MF-DGP				MF-DGP improved			
Functions	R^2	MNLL	RMSE	std RMSE	R^2	MNLL	RMSE	std RMSE
Currin	0.8856	1.6834	0.7427	0.3398	**0.9148**	**1.4165**	**0.6735**	**0.2056**
Park	0.8436	1.1616	1.1364	1.496	**0.9852**	**0.8807**	**0.5693**	**0.0969**
Borehole	0.9986	-2.006	0.0168	0.0032	0.9994	-2.733	0.0107	0.0016
Branin	0.3592	3.5977	0.1541	0.0665	**0.5865**	**5.0382**	**0.1256**	**0.0480**

are considered except that the HF DoE is included in the LF DoE. To have the same number of samples in LF DoE, the same number of HF samples are removed from LF DoE.

The obtained results are presented through numerical tables and boxplot figures. The tables present the mean value and the standard deviation for R2, RMSE and MNLL considering 20 repetitions from different LHS for the training multi-fidelity set. Moreover, an indicator providing the improvement of RMSE of the multi-fidelity techniques with respect to the single fidelity GP HF is added. A negative value means that the multi-fidelity technique improves the RMSE compared to GP HF by an amount of x%.

All GP-based multi-fidelity techniques are implemented with a squared exponential kernel. Co-kriging with LMC is based on a coregionalization matrix of rank 2 (corresponding to two independent latent functions).

Single-Stage-To-Orbit trajectory simulation

Problem definition

This trajectory problem is based on the "Time-Optimal Launch of a Titan II" example defined by Longuski *et al.* [Longuski et al., 2014]. It is an optimal control problem which consists in finding the pitch angle profile for a Single-Stage-To-Orbit (SSTO) launch vehicle that minimizes the time required to reach orbit injection under considering a constant thrust. A 2D Cartesian simulation with a planar trajectory, non rotating Earth is considered. The corresponding equations of motion are derived from [Balesdent et al., 2012a]. The target final orbit is a circular orbit at the altitude of 185km.

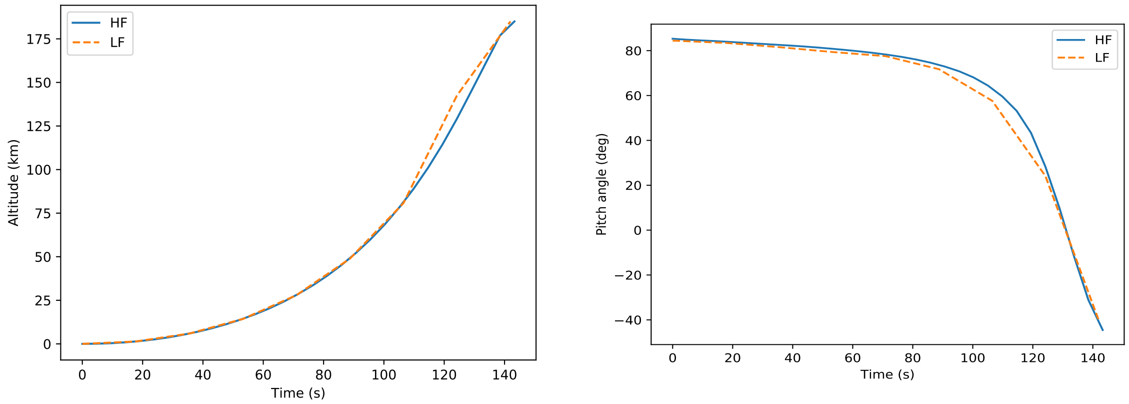
The trajectory simulation is carried out using Dymos [Falck and Gray, 2019] which is an open-source tool for solving optimal control problems involving multidisciplinary systems. It is built on top of the OpenMDAO framework [Gray et al., 2019; van Gent and La Rocca, 2019]. It uses pseudospectral collocation method that is classically used to solve such a type of problems [Li et al., 2020]. A high order Gauss-Lobatto collocation method [Herman and Conway, 1996] is used to solve this optimization problem. Gauss-Lobatto is a generalization of the Hermite-Simpson optimization scheme developed by Herman and Conway [Herman and Conway, 1996]. In this approach, for solving optimal control problem, polynomials are considered to represent the state variable time history over segments (subintervals) of the total time of interest. The polynomial family follows the Gauss-Lobatto rules. Each segment is discretized according to the Legendre-Gauss-Lobatto polynomial nodes. The value of each state variable and each control variable at each state discretization node is a design variable. The higher the number of segments, the higher the accuracy of the optimal control solving but the higher the number of design variables in the optimization problem and therefore the associated computational cost.

For the SSTO problem, two fidelity models are considered. The input space is composed of five design variables: the thrust, the specific impulse, the diameter of the launch vehicle, the initial mass of the vehicle and the coefficient of drag (Table 6.2). The considered output is the fuel burnt mass during the flight.

Table 6.2 SSTO input design variable definition

Input variables	Domain of definition
Thrust (T)	[1800, 2400]kN
Specific impulse (Isp)	[210, 330]s
Launch vehicle diameter (d)	[2.5, 4.4]m
Launch vehicle initial mass (m_0)	[120, 124]t
Coefficient of drag (C_d)	[0.1, 0.9]

The two fidelities are distinguished by the number of segments of the Gauss-Lobatto collocation. The LF model assumes a low number of segments $num_{segments} = 4$ corresponding to a discretization scheme enabling fast optimal control solving but limited simulation accuracy. The HF model assumes a higher number of segments $num_{segments} = 15$ providing a high accuracy for the trajectory simulation but a more complex and more computationally intensive optimal control problem to be solved.



(a) Altitude as a function of time

(b) Pitch angle as a function of time

Fig. 6.3 Illustrations for SSTO trajectory with Low Fidelity (LF) and High Fidelity (HF) models

The difference between LF and HF models are illustrated on Figures 6.3a and 6.3b representing the altitude as a function of time and the pitch angle as a function of range. The LF model provides a reasonable approximation of the HF model but with substantial simplification in the trajectory.

Results

Boxplots illustrating the results for SSTO problem are displayed in Figure 6.4. In addition, results including the different comparison metrics are provided in Table 6.3. The non-linear multi-fidelity techniques perform less accurately compared to linear approaches for a small number of samples in the HF DoE. For instance, for a HF sample size of 5 points, AR1 and LMC provide the same prediction accuracy (R2 of 0.993) and LMC provides the best model of prediction uncertainty (MNLL of −5.03 for LMC compared to −3.33 for AR1). While the improved MF-DGP provides the best results among the non-linear approaches both in prediction accuracy (R2 of 0.949) and uncertainty quantification (MNLL of −3.615), the regular MF-DGP performs poorly with a R2 of −7.865. For HF sample size of 10 and 20, both NARGP and NARGP-nest degrade the RMSE performance compared to GP HF. Furthermore, once enough HF samples are available for regular MF-DGP, it provides comparable prediction accuracy as MF-DGP improved and as linear approaches (R2 of 0.986 for AR1 compared to 0.982 for MF-DGP) but the MF-DGP approaches provide a better uncertainty quantification (MNLL of −5.038 for MF-DGP improved compared to 2.92 for AR1). Considering the results with a DoE for HF of 20 samples, this test case is a representative illustration of the trade-off between the prediction accuracy and the quality of the uncertainty model for the prediction. AR1 tends to provide a better prediction against the HF test set, however, the quality of the uncertainty model associated to MF-DGP improved is better and therefore future use of such a model for optimization, uncertainty propagation or refinement strategies might present more advantages. Eventually, considering the best multi-fidelity model for each size of HF samples, the addition of HF samples reaches a limit in terms of RMSE improvement compared to GP HF, as for 5 HF samples the best improvement is of 83% while for 20 HF samples it decreases to 19%.

Table 6.3 Summary of the results obtained on the SSTO problem

Function	Method	R2 (std)	RMSE (std)	MNLL (std)	Evolution of RMSE wrt GP HF	DOE size (LF, HF)
SSTO	GP HF	0.642(0.425)	9.123e-3(5.248e-3)	7.566e+2(1.367e+3)	-	100, 5
		0.969(0.041)	2.757e-3(1.442e-3)	1.064e+1(1.748e+1)	-	100, 10
		0.972(0.068)	2.036e-3(2.095e-3)	4.044(8.781)	-	100, 20
	LMC	0.993(0.002)	1.464e-3(2.187e-4)	**-5.027(9.996e-2)**	−83%	100, 5
		0.891(0.438)	2.727e-3(5.141e-3)	-3.973(4.730)	−1%	100, 10
		0.975(0.061)	1.925e-3(2.004e-3)	-4.266(3.153)	−5%	100, 20
	AR1	**0.993(0.001)**	**1.462e-3(5.668e-5)**	-3.326(2.460)	−**83**%	100, 5
		0.991(0.007)	**1.583e-3(4.667e-4)**	-3.227(2.874)	−**42**%	100, 10
		0.986(0.028)	**1.639e-3(1.283e-3)**	2.919(6.476)	−19%	100, 20
	NARGP	0.951(0.032)	3.754e-3(1.051e-3)	1.312e+1(7.274e+1)	−58%	100, 5
		0.958(0.052)	3.227e-3(1.655e-3)	8.761(2.934e+1)	+17%	100, 10
		0.971(0.069)	2.127e-3(2.115e-3)	7.208(1.143e+1)	+4%	100, 20
	NARGP-nest	0.949(0.033)	3.808e-3(1.134e-3)	-7.133e-1(1.262e+1)	−58%	100, 5
		0.961(0.044)	3.129e-3(1.462e-3)	4.939(2.128e+1)	+13%	100, 10
		0.974(0.062)	2.102e-3(1.896e-3)	5.533(1.192e+1)	+3%	100, 20
	MF-DGP	-7.865(28.921)	2.651e-2(4.518e-2)	-2.675(9.035e-1)	190%	100, 5
		0.968(0.071)	2.521e-3(1.893e-3)	**-4.640(4.315e-1)**	−8%	100, 10
		0.982(0.035)	1.876e-3(1.426e-3)	**-5.014(5.717e-1)**	−8%	100, 20
	MF-DGP improved	0.949(0.153)	2.66e-3(2.9e-3)	-3.615(3.99e-1)	−71%	100, 5
		0.968(0.088)	2.36e-3(2.11e-3)	**-4.48(4.23e-1)**	−14%	100, 10
		0.983(0.034)	1.832e-3(1.415e-3)	**-5.038(5.554e-1)**	−10%	100, 20

SuperSonic Business Jet multidisciplinary problem

Problem definition

For the second aerospace design application, a multidisciplinary design is considered of a SuperSonic Business Jet (SSBJ) based on the problem defined by Sobieszczanski *et al.* [Langley et al., 1998]. The multidisciplinary analysis is composed of four disciplinary modules: structures, aerodynamics, propulsion and performance estimation. All the disciplines are modeled with an analysis level typical for an early conceptual design

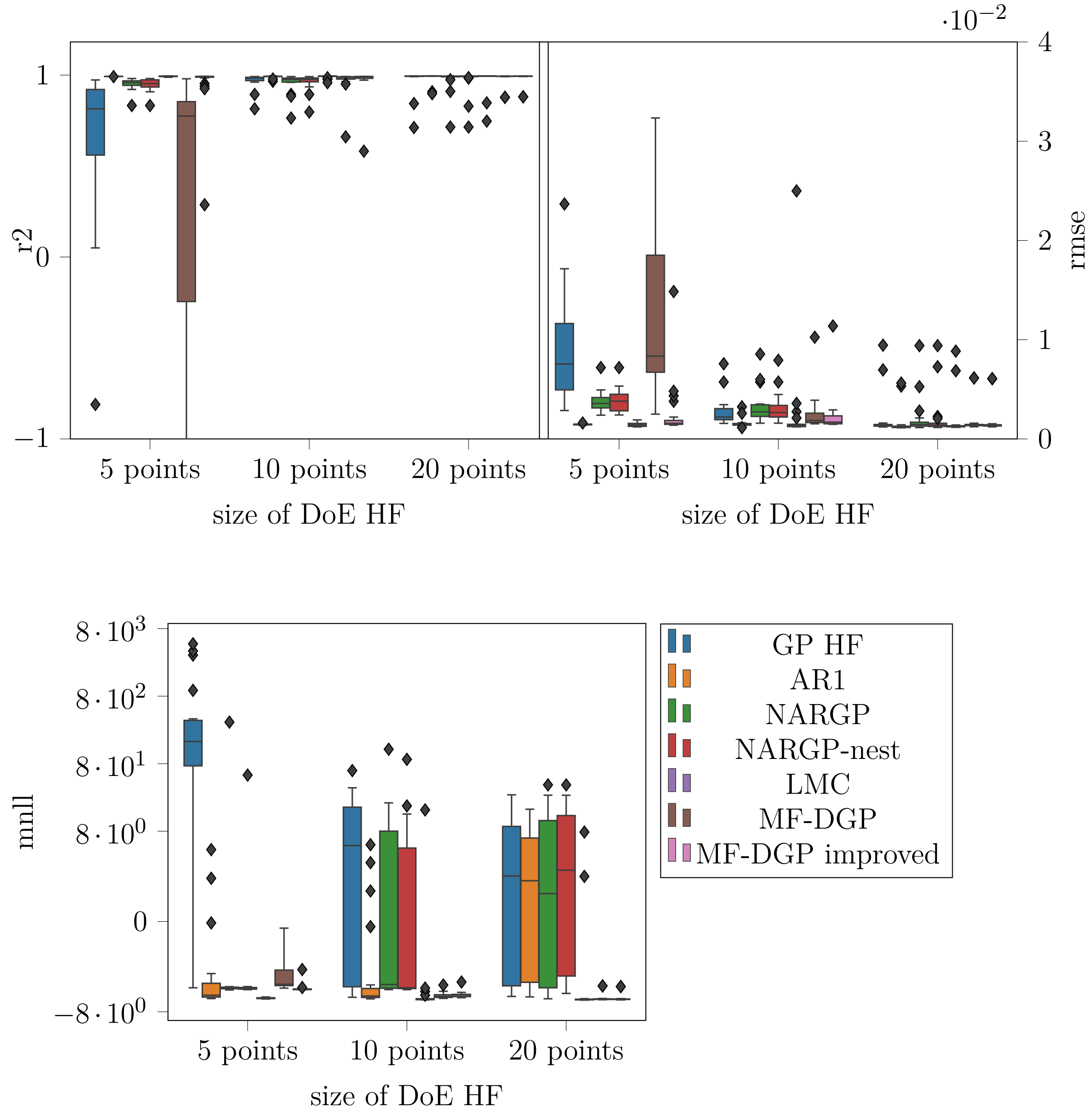


Fig. 6.4 Boxplots of R2, RMSE and MNLL for SSTO problem. From let to the right: GP HF, AR1, NARGP, NARGP nested, LMC, MF-DGP, MF-DGP improved.

stage. The aircraft simulation allows to estimate its range through the Breguet range equation. Each discipline implements early design models (analytical formula). The structure discipline computes the stresses undertaken by the wings of the aircraft and the mass of the different components of the vehicle (*e.g.*, fuselage, wing, fuel). It takes as inputs the definition of the characteristics of the wings (thickness to chord ratio, aspect ratio, sweep angle), the lift coefficient (from the aerodynamics discipline) and the engine mass (from the propulsion discipline). The aerodynamics discipline computes the lift and drag of the vehicle. It takes as inputs the wing characteristics, flight conditions and the size of engine from the other disciplines. The propulsion discipline aims at defining the dimension, mass and consumption of the engine from the flight conditions and drag of the vehicle. Finally, the performance discipline computes the range of the vehicle from the outputs of the other disciplines: the lift over drag ratio, the engine consumption, the cruise Mach number, the altitude and the weights of the aircraft. The range is considered as the output of the design process for the training of the multi-fidelity surrogate model. For more details on SSBJ simulation, refer to [Langley et al., 1998]. The SSBJ problem is simulated using OpenMDAO framework [Gray et al., 2019]. As the SSBJ is a multidisciplinary problem, it requires a multidisciplinary analysis (MDA) in order to satisfy the coupling consistency between the different disciplines. This MDA can be performed using Fixed-Point-Iteration that is an iterative process between the different disciplines. This process is considered as converged when the discrepancy of the output disciplines between two iterations is less than a given tolerance ϵ. The lower the tolerance, the higher the accuracy of the response but the higher the duration of the MDA. For that context, two tolerances $\epsilon_{lf} > \epsilon_{hf}$ have been considered to define the two fidelities of the design process. The low-fidelity considers a coarse convergence of the MDA (only one iteration) whereas the high-fidelity considers a very restrictive tolerance and requires a dozen of iterations between the disciplines. The design input parameters are defined in Table 6.4.

Table 6.4 SSBJ input design variable definition

Input variables	Domain of definition
Thickness to chord ratio	[0.025, 0.085]
Altitude	[20, 50]km
Mach number	[1.0, 2.0]
Aspect ratio	[1.5, 6.0]
Wing sweep	[20, 70]deg
Wing surface area	[93, 163]m^2

Results

Boxplots illustrating the results for SSBJ test case are displayed in Figure 6.5. Furthermore, numerical results including the comparison metrics are provided in Table 6.5. Similarly to the previous test case, linear approaches (AR1 and LMC) provide more accurate results considering the limited HF sample size case (for 5 points, R2 of 0.963 for LMC) and regular MF-DGP gives poor results compared to MF-DGP improved. However, by slightly increasing the number of HF samples from 5 to 10, the prediction accuracy of MF-DGP becomes comparable to MF-DGP improved. Over all the models, once MF-DGP improved gets enough HF data it provides the best results in terms of prediction accuracy (for 10 samples, R2 of 0.97, for 20 samples, R2 of 0.982) and uncertainty quantification (for 10 samples, MNLL of -1.929, for 20 samples, MNLL of -2.15).

Table 6.5 Summary of the results obtained on the SSBJ problem

Function	Method	R2 (std)	RMSE (std)	MNLL (std)	Evolution of RMSE wrt GP HF	DOE size (LF, HF)
SSBJ	GP HF	0.131(0.336)	2.300e-1(4.591e-2)	3.298e+3(9.908e+3)	-	100, 5
		0.48(0.329)	1.727e-1(5.192e-2)	2.223e+2(6.797e+2)	-	100, 10
		0.836(0.074)	9.980e-2(2.000e-2)	1.404(2.947)	-	100, 20
	LMC	**0.963(0.014)**	**4.753e-2(9.139e-3)**	**-1.514(1.762e-1)**	−**79**%	100, 5
		0.968(0.015)	4.367e-2(9.459e-3)	**-1.762(3.060e-1)**	−74%	100, 10
		0.980(0.005)	3.541e-2(4.494e-3)	-1.550(5.549e-1)	−64%	100, 20
	AR1	0.957(0.024)	5.067e-2(1.334e-2)	1.953(3.185)	−77%	100, 5
		0.970(0.008)	4.298e-2(5.946e-3)	3.726e-1(1.647)	−75%	100, 10
		0.980(0.006)	3.513e-2(4.680e-3)	-5.478e-1(1.467)	−64%	100, 20
	NARGP	0.716(0.433)	1.093e-1(7.774e-2)	2.201e+3(9.555e+3)	−52%	100, 5
		0.875(0.143)	7.791e-2(4.308e-2)	9.731e-1(2.269)	−54%	100, 10
		0.904(0.105)	7.003e-2(3.430e-2)	2.588e-1(2.109e+00)	−29%	100, 20
	NARGP-nest	0.791(0.273)	9.819e-2(5.988e-2)	2.791(6.395)	−57%	100, 5
		0.921(0.073)	6.499e-2(2.835e-2)	-4.889e-1(2.018)	−62%	100, 10
		0.950(0.039)	5.296e-2(1.911e-2)	-1.257(1.358)	−47%	100, 20
	MF-DGP	0.679(0.433)	1.110e-1(8.951e-2)	1.129e+1(2.936e+1)	−51%	100, 5
		0.966(0.012)	4.569e-2(7.180e-3)	-1.750(1.303e-1)	−73%	100, 10
		0.974(0.012)	3.945e-2(8.025e-3)	**-1.931(1.525e-1)**	−60%	100, 20
	MF-DGP improved	0.857(0.334)	6.740e-2(6.73e-2)	10.16e+1(3.988e+1)	−70%	100, 5
		0.970(0.012)	**4.223e-2(9.004e-3)**	**-1.929(1.840e-1)**	**-76%**	100, 10
		0.982(0.0038)	**3.337e-2(3.677e-3)**	**-2.15(1.229e-1)**	**-66%**	100, 20

It is interesting to notice that LMC tends to perform as well as AR1 technique in terms of prediction accuracy but presents better results regarding the uncertainty model. The differences between the two approaches are in the symmetrical (LMC) and asymmetrical (AR1) fusion schemes. Multi-fidelity problems are asymmetrical by nature (information provided by HF are more accurate than by LF) so AR1 should be more suited for such a type of problems. However, it appears that LMC provides robustness to DoE and accurate predictions that are similar to AR1 or even better, but also provides an accurate uncertainty model for the prediction.

Aerostructural problem

Problem definition

The aerostructural problem is based on OpenAeroStruct [Jasa et al., 2018] which is a tool that performs aerostructural simulation and optimization using OpenMDAO [Gray et al., 2019]. It couples a vortex-lattice method (VLM) [Anderson, 1991] and a finite-element method (FEM) using six degree-of-freedom spatial beam elements with axial, bending, and torsional stiffness to simulate aerodynamic and structural analyses using lifting surfaces [Jasa et al., 2018]. The aerodynamics submodel involves VLM to

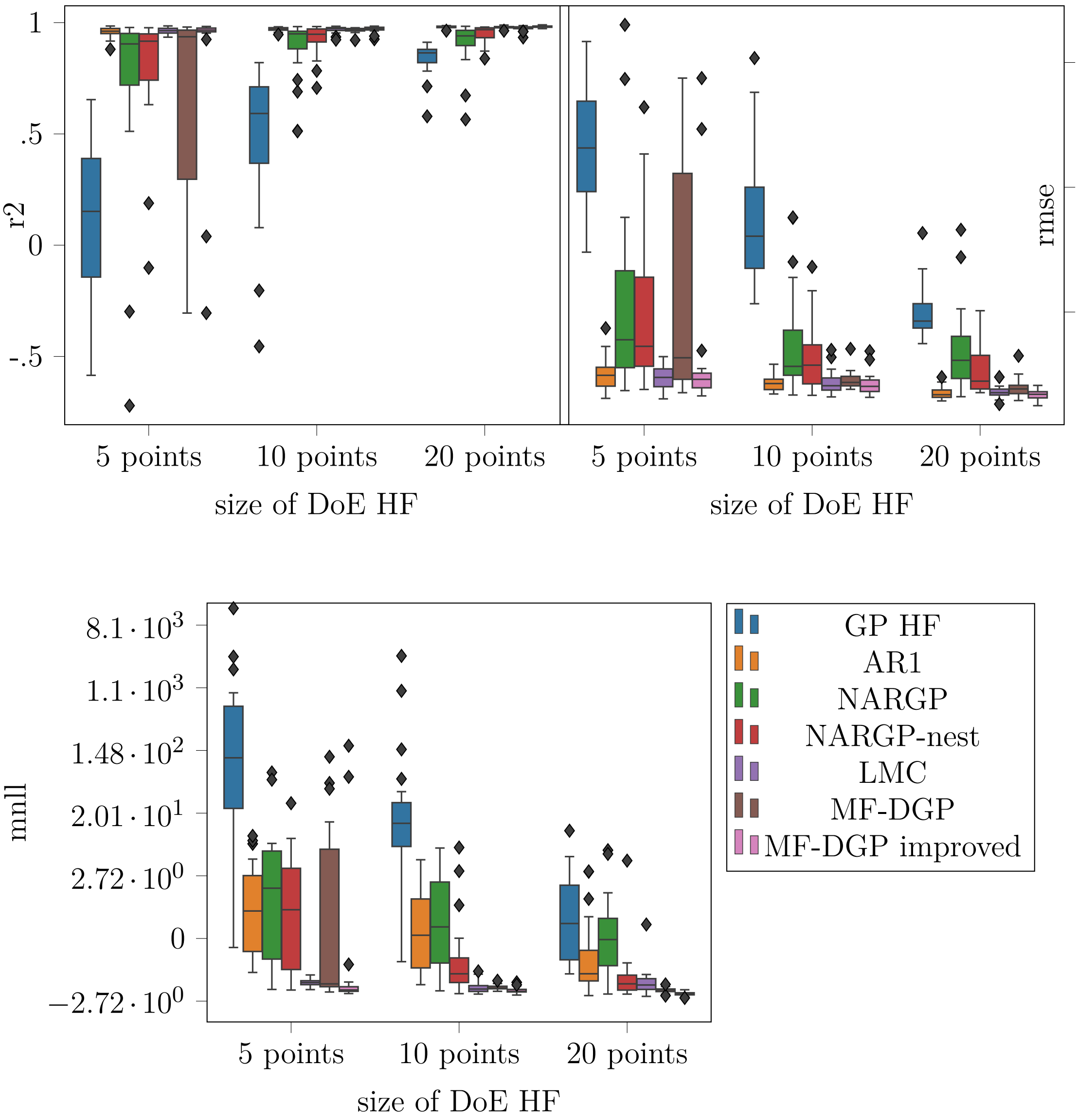


Fig. 6.5 Boxplots of R2, RMSE and MNLL for SSBJ problem. From let to the right: GP HF, AR1, NARGP, NARGP nested, LMC, MF-DGP, MF-DGP improved.

estimate the aerodynamic loads acting on the lifting surfaces. Provided a structured mesh defining a lifting surface, the aerodynamic properties are estimated using the circulation distribution. The lifting surface is modeled using horseshoe vortices to represent the vortex system of a wing. A vortex filament implies a flow field in the surrounding space. The strength of a vortex filament is its circulation, which induces lift on a surface.

For the structural submodel, a FEM technique is involved using spatial beam elements, resulting in six degree-of-freedom per node. The spatial beam element is a combination of beam, torsion and truss elements, therefore it simultaneously carries axial, bending, and torsional loads.

In OpenAeroStruct, the structures and aerodynamics are two separate submodels that receive inputs and compute outputs. The aerodynamics submodel takes a mesh as an input and outputs aerodynamic loads, whereas the structural group takes as input aerodynamic loads and outputs structural displacements. The load and displacement exchange is simplified as the same spanwise discretization is used for the aerodynamic and structural submodels. A Gauss-Seidel algorithm [Salkuyeh, 2007] is used to solve the multidisciplinary analyses and satisfy the interdisciplinary couplings.

For the multi-fidelity modeling problem, two fidelities are considered to estimate the lift coefficient CL of a wing. The difference between the models consists in the mesh refinement, a sparse mesh for the LF model and a dense mesh for the HF model (Figure 6.6).

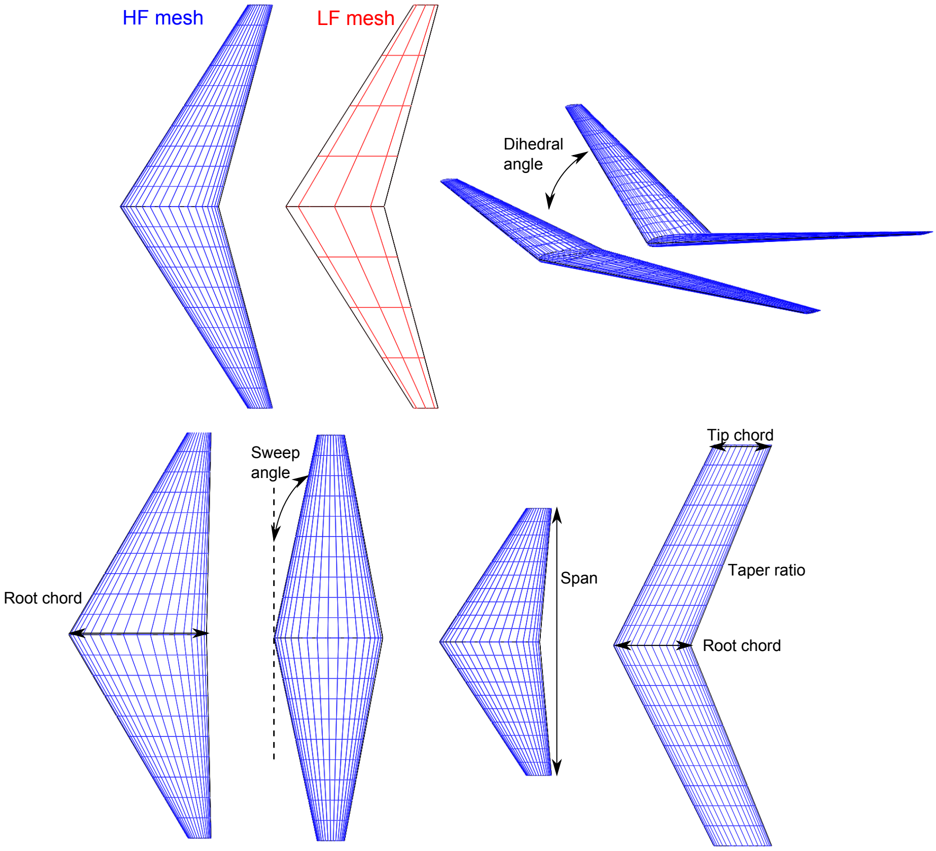


Fig. 6.6 Geometrical parameter definition and HF/LF meshes for the aerostructual problem

The input space is composed of eight design variables: the angle of attack, the span, the sweep angle, the dihedral angle, the taper ratio, and the root chord at three location along the space (Table 6.6). The geometrical input parameters are illustrated in Figure 6.6.

Table 6.6 Aerostructural input design variable definition

Input variables	Domain of definition
Angle of attack	[1.0, 5.0]deg
Span	[5.0, 10.0]m
Sweep angle	[0., 20.]deg
Dihedral angle	[0., 20.]deg
Taper ratio	[0.7, 1.4]
Root chord at three locations	$[1.0, 5.0]^3$m

Results

Boxplots illustrating the results for the aerostructural test case are displayed in Figure 6.7. Furthermore, numerical results including the comparison metrics are provided in Table 6.7. These last experimentations confirm the previous results, that are with enough HF data, MF-DGP improved outperforms the other multi-fidelity methods both in terms of prediction accuracy (improvement of the GP HF RMSE by 76% for 10 HF samples, by 57% for 20 HF samples compared to 67% and 57% for LMC respectively) as well as the predictive uncertainty (MNLL of −3.2 for MF-DGP improved compared to −2.77 for LMC). Moreover, even if regular MF-DGP performs better than the other models, improved MF-DGP increases the accuracy of the model.

Table 6.7 Summary of the results obtained on the OpenAeroStruct problem

Function	Method	R2 (std)	RMSE (std)	MNLL (std)	Evolution of RMSE wrt GP HF	DOE size (BF, HF)
OAS	GP HF	0.613(0.379)	4.891e-2(1.842e-2)	4.298e+1(6.880e+1)	-	160, 10
		0.939(0.028)	2.032e-2(4.372e-3)	-9.421e-3(4.046)	-	160, 20
	LMC	0.958(0.044)	1.609e-2(6.403e-3)	-2.774(3.113e-1)	−67%	160, 10
		0.983(0.009)	1.059e-2(2.441e-3)	-2.575(9.259e-1)	−47%	160, 20
	AR1	0.956(0.022)	1.721e-2(4.192e-3)	7.482(8.182)	−64%	160, 10
		0.982(0.008)	1.102e-2(2.346e-3)	-2.040e-1(1.712)	−45%	160, 20
	NARGP	0.914(0.085)	2.272e-2(9.700e-3)	-1.004(4.699)	−53%	160, 10
		0.963(0.026)	1.534e-2(5.092e-3)	-2.379(1.190)	−25%	160, 20
	NARGP-nest	0.921(0.084)	2.147e-2(9.738e-3)	-1.221(4.731)	−56%	160, 10
		0.956(0.036)	1.635e-2(6.425e-3)	-2.258(1.149)	−19%	160, 20
	MF-DGP	0.973(0.018)	1.331e-2(3.934e-3)	-2.986(3.612e-1)	−72%	160, 10
		0.987(0.007)	9.453e-3(2.008e-3)	**-3.354(2.363e-1)**	−**53**%	160, 20
	MF-DGP improved	**0.980(0.009)**	**1.114e-2(2.72e-3)**	**-3.199(2.630e-1)**	−**76**%	160, 10
		0.989(0.003)	**8.707e-3(1.120e-3)**	**-3.463(1.591e-1)**	−**57**%	160, 20

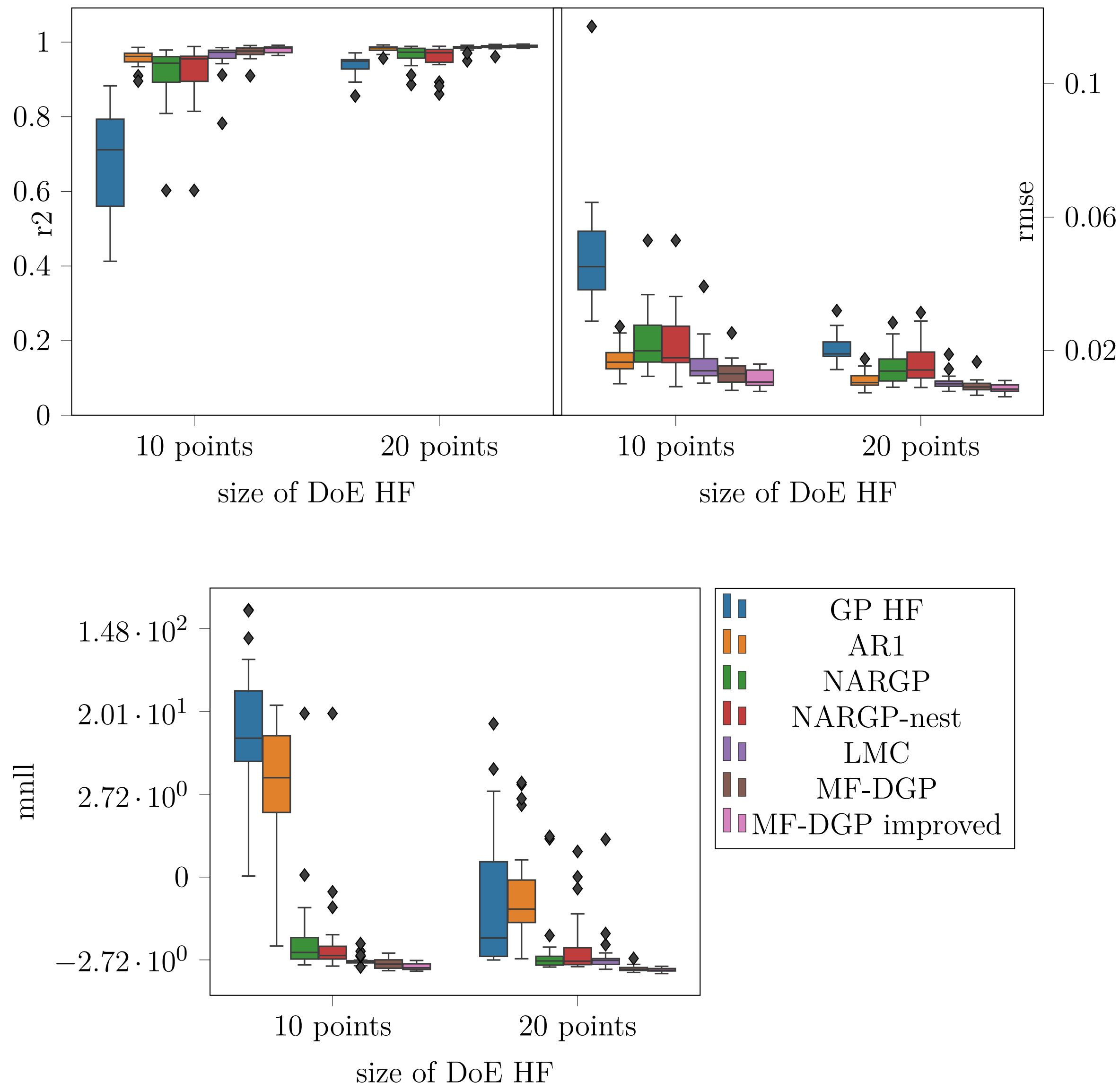


Fig. 6.7 Boxplots of R2, RMSE and MNLL for OpenAeroStruct problem. From let to the right: GP HF, AR1, NARGP, NARGP nested, LMC, MF-DGP, MF-DGP improved.

Results synthesis

Some general trends can be drawn about the multi-fidelity GP-based approaches studied in this section. When a limited number of HF samples is available due to the computational cost associated to such models, multi-fidelity techniques allow to reduce the prediction error compared to a single high-fidelity GP model. When the number of available HF samples increases, the relative improvement of multi-fidelity methods compared to single fidelity approach decreases up to a point where low-fidelity information do not offer improvement to the prediction accuracy and therefore a single fidelity model is preferable.

Moreover, when a very limited number of HF samples with respect to the problem dimension is considered, linear mapping between fidelities (AR1 and LMC) tends to provide better results than non-linear mapping approaches (NARGP and MF-DGP) which are more difficult to train when not enough HF information is available to model this relationship. Indeed, the non-linear multi-fidelity techniques, due to their higher complexity of definition (nested composition of Gaussian processes), offer higher capability of modeling but with a higher number of hyperparameters to be tuned. Still, when there is not enough HF information (5 HF samples for the SSTO and SSBJ problems) MF-DGP improved provides competitive results to the other models unlike regular MF-DGP which provides poor results in these cases. With enough HF data (10 and 20 data points) in the three described problems, MF-DGP improved provides the best results compared to the other models, over the three investigated metrics that are, prediction accuracy, predictive uncertainty, and robustness to the DoE. However, the main drawback of the improved MF-DGP is its computational complexity. In fact, besides the inherent complexity to train a DGP, the improved MF-DGP, at each iteration of the training, propagates the induced inputs throughout all the previous layers which is computationally expensive. Still, since the HF is considered computationally expensive, the training time of the improved MF-DGP is far lower than the evaluation time of an HF data point.

6.2 Multi-fidelity with different input domain definitions

The previously presented multi-fidelity problems and the used methods considered the input spaces of the different fidelities as identically defined. However, in several engineering multi-fidelity problems, each fidelity may be defined on its own input space. In fact, to overcome the issue of high-dimensional HF design-spaces which require a large size of training data to be approximated, the LF model may consider only a subset of design variables. For that, the variables that have less influence may be neglected and not taken into account in the LF model yielding to a different input-space between the fidelity models. This allows a reduction of the complexity of the problem at the cost of the accuracy. For instance, when modeling the stress or the aerodynamic forces on a thick-surface (such as an aircraft wing), in a first approximation, the thickness of this surface may be neglected, thus, the studied structure is considered as

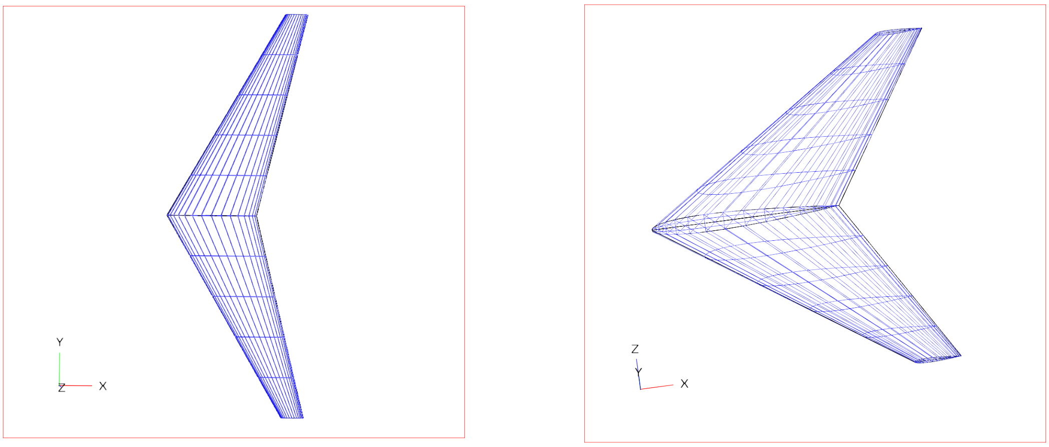


Fig. 6.8 A multi-fidelity problem where the input spaces are not identically defined. (left) Wing represented as a 2D object in low-fidelity. (right) Wing represented as a 3D object in high-fidelity.

two-dimensional in the LF model (Fig 6.8). Another way of reducing the dimensionality of the problem, is by averaging the effects of some variables to create a small-sized set of variables. Unlike the case where some variables are abstracted, in this framework, the variables describing the LF model are not a subset of the HF variables. For instance, in aerodynamics, to model a multiple-section wing, simplified plan-form characterization can be used considering only one section with average chords and sweep angles (Section 6.2.4). Moreover, each fidelity may use different physical theories (*e.g.*, Euler-Bernoulli beam theory *vs.* Timoshenko-Ehrenfest beam theory), different frames of reference (*e.g.*, inertial frame *vs.* local frame), or different coordinate systems (*e.g.*, Cartesian coordinates *vs.* spherical coordinates *vs.* cylindrical coordinates). These specificities for each fidelity may result in different parameterizations of the input variables and therefore different input-spaces. For instance, for geometrical input variables, in one fidelity a Cartesian formulation can be used, whilst in the other fidelity, a spherical formulation is preferred.

The classic approach to deal with such variable input spaces is to use a nominal mapping from the HF to LF input space based on practical insights of the multi-fidelity problem. [Tao et al., 2019] developed an Input Mapping Calibration approach (IMC) that seeks to improve the nominal mapping using a calibrated parametric mapping (see Chapter 3, Section 3.4.2 for details on IMC and other input mapping approaches). However, in IMC as well as in the classic input mapping approaches, the optimization of the mapping parameters is done previously to the training of the multi-fidelity model, hence, preventing the mapping to be updated once the multi-fidelity model is trained. Moreover, the multi-fidelity model uses only the projection of the HF data on the lower-fidelity input space obtained by the mapping, hence, it does not take into account the correlations in the original HF input space.

In this section, a model is developed based on MF-DGP that embeds a non-parametric Bayesian mapping within the multi-fidelity model (Section 6.2.1, 6.2.2). The associated inference approach is detailed in Section 6.2.3. This proposed model called Multi-Fidelity Deep Gaussian Process Embedded Mapping (MF-DGP-EM) is then assessed on analytical and engineering multi-fidelity problems (Section 6.2.4) and its computational aspects are investigated (Section 6.2.6).

6.2.1 Multi-fidelity Deep Gaussian Process Embedded Mapping (MF-DGP-EM)

A multi-fidelity problem is considered in which each fidelity t is defined by its own input space of dimension d_t. Therefore, classic GP approaches including MF-DGP can not directly be used. To overcome this issue, multi-output GPs $\left(\mathbf{H}_{[t]}(\cdot)\right)_{1\leq t\leq n_{\mathrm{fi}}-1}$ are introduced in MF-DGP to map between the input-spaces of two successive fidelities t and $t+1$. For $t=1,\dots,n_{\mathrm{fi}}-1$ a multi-output GP $\mathbf{H}_{[t]}(\cdot):\mathbb{R}^{d_{t+1}}\rightarrow\mathbb{R}^{d_t}$, maps from the input-space of fidelity $t+1$ of dimension d_{t+1} to the lower space fidelity t with dimension d_t. The input mapping GPs $\mathbf{H}_{[t]}(\cdot)$ are conditioned on the nominal mapped values $\mathbf{X}_t^{t+1}$ (Section 6.2.2 for details on the nominal mapped values). The model obtained is a two-level DGP, where the first level maps between the different fidelity input-spaces and the second level propagates the fidelity evaluations (Fig. 6.9). Hence, the mapping between the input-spaces of the fidelities is defined within the multi-fidelity model. As the auto-regressive model [Kennedy and O'Hagan, 2001], non-linear auto-regressive [Perdikaris et al., 2017], and MF-DGP, the defined model has a regressive structure, meaning that information is considered in a non-symmetrical way. Therefore, MF-DGP-EM requires a known hierarchy of the fidelity levels. Moreover, in the case of MF-DGP-EM, the input-spaces may have different dimensionalities. The dimensionality is usually decreasing from the higher to the lower fidelities, thus, the input mapping GPs perform dimensionality reduction from the high-fidelity to the low-fidelity input-spaces.

This proposed model allows a concurrent optimization of the mapping and the multi-fidelity model. Besides, compared to IMC, only the input data nominal mapping values are used instead of nominal mapping functions over the whole input-space. This allows a more flexible mapping adequate in the case of computationally expensive mappings. Moreover, using GPs in the first level of the model enables a non-parametric

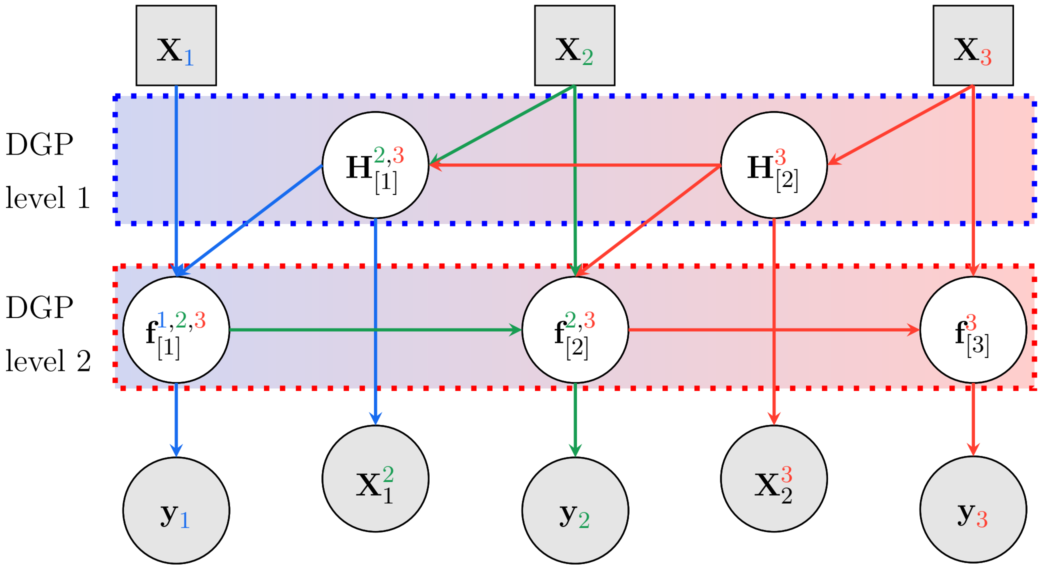


Fig. 6.9 Graphical representation of MF-DGP-EM with different input-spaces. $\mathbf{X}_t$ and $\mathbf{y}_t$ represent respectively the input data and the response at fidelity t. $\mathbf{X}^{t+1}_t$ corresponds to the nominal mapped values of the input data $\mathbf{X}_{t+1}$ to the lower fidelity t. $\mathbf{H}^t_{[i]}$ represents the GP mapping of the input data from fidelity t to the input-space of the fidelity i. $\mathbf{f}^t_{[i]}$ represents the evaluation at layer i of the inputs at fidelity t. The propagation of the inputs and evaluations is color-coded to indicate the associated fidelity. The blue, green, and red colors represent respectively the low, medium, and high fidelity levels. MF-DGP-EM embeds two DGP levels, a first level going from the highest fidelity to the lowest one maps between the input-spaces through the GP mappings $\{\mathbf{H}_{[t]}\}_{t=1}^{n_{\text{fi}}-1}$. The second level, as MF-DGP, propagates the GP fidelity evaluations from the lowest to the highest fidelity through the fidelity GPs $\{f_{[t]}\}_{t=1}^{n_{\text{fi}}}$.

mapping and induces uncertainty quantification on the latter, which differs from the space mapping approach that requires a deterministic parametric form of the mapping to be used. It avoids over-fitting compared to parametric mapping. Finally, this model keeps the original input-space correlations, since $\mathbf{X}_i$ is used as input for $\mathbf{f}_{[i]}(\cdot)$.

6.2.2 The input mapping GPs

The input mappings are performed with multi-output GPs that transform the input data from the higher-fidelity input-space into the lower-fidelity input-space. The input mapping GPs are conditioned on mapped nominal values of the training set. The nominal mapping is obtained based on physical insights of the relationship between the fidelities. For example, if the low-fidelity variables are a subset of the high-fidelity variables, the nominal mapping is simply the identity. However, it can be more complicated. For instance, it can map the design variables of the high-fidelity into the low-fidelity space to obtain an identical defined quantity of interest (*e.g.*, the volume defined by the HF variables equals to the volume defined by the mapped HF variables) this may induce computationally expensive input mappings. The proposed model is convenient in this case *i.e.*, the nominal mapping is not known in the whole input-space but only for the training HF data. In the case where the nominal mapping is uncertain due to lack of physical insight in the relationship between the input-spaces of the different fidelities, a white kernel can be added to the covariance function in order to take into account this uncertainty.

When mapping from a high-fidelity to a low-fidelity input space, the relevance of the HF variables is not isotropic. In fact, some variables are abstracted or averaged with small weights in the LF. For that, an Automatic Relevance Determination (ARD) kernel is preferred for the input mapping GPs. Meaning that, in the mapping each variable has its own length-scale parameter. Therefore, the variables neglected are automatically given high length-scale values. The specific form of the ARD kernel (*e.g.*, squared exponential, Matérn) depends on the problem in hand. Without loss of generality, due to its low hyper-parameterization and smoothness properties, an ARD squared exponential kernel is used in this work.

6.2.3 The Evidence Lower Bound

As in regular DGPs and MF-DGP, the computation of the marginal likelihood of MF-DGP-EM:

$$p\left(\mathbf{y}_{n_{\text{fi}}},\dots,\mathbf{y}_1,\mathbf{X}^{n_{\text{fi}}}_{n_{\text{fi}}-1},\dots,\mathbf{X}^2_1|\mathbf{X}_{n_{\text{fi}}},\dots,\mathbf{X}_1\right) \tag{6.5}$$

is analytically non-tractable. Approximations are necessary to obtain a lower bound on this marginal likelihood which is then maximized to train the multi-fidelity model.

As in MF-DGP, the DGP inference followed is the doubly stochastic inference scheme presented in [Salimbeni and Deisenroth, 2017]. At each layer i, a set of inducing inputs / outputs $(\mathbf{Z}_{[i]},\mathbf{u}_{[i]})$ are introduced for the fidelity GP $\mathbf{f}_{[i]}(\cdot)$, and similarly for each mapping $\mathbf{H}_{[i]}(\cdot)$, a set of inducing inputs / outputs $(\mathbf{W}_{[i]},\mathbf{V}_{[i]})$ are introduced.

Then, the following variational approximation is considered:

$$
\begin{aligned}
q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right) = &\prod_{t=1}^{n_{\text{fi}}}\prod_{i=1}^{t-1}\left[p(\mathbf{f}^{t}_{[i]}|\mathbf{u}_{[i]};\left[\mathbf{H}^{t}_{[i]},\mathbf{f}^{t}_{[i-1]}\right],\mathbf{Z}_{[i-1]})\; p\left(\mathbf{H}^{t}_{[i]}|\mathbf{V}_{[i]};\mathbf{H}^{t}_{[i+1]},\mathbf{W}_{[i+1]}\right)\right] \\
&\times\prod_{t=1}^{n_{\text{fi}}} p(\mathbf{f}^{t}_{[t]}|\mathbf{u}_{[t]};\left[\mathbf{X}_{t},\mathbf{f}^{t}_{[t-1]}\right],\mathbf{Z}_{[t-1]})\times\prod_{i=1}^{n_{\text{fi}}} q(\mathbf{u}_{[i]})\times\prod_{i=1}^{n_{\text{fi}}-1} q(\mathbf{V}_{[i]})
\end{aligned}
\tag{6.6}
$$

where $q(\cdot)$ is the variational distribution of the latent variables and:

$$
\begin{aligned}
\mathcal{F} &= \{\{\mathbf{f}^{t}_{[i]}\}_{i=1}^{t}\}_{t=1}^{n_{\text{fi}}} \\
&= \{\{\mathbf{f}^{1}_{[1]}\},\{\mathbf{f}^{2}_{[1]},\mathbf{f}^{2}_{[2]}\},\ldots,\{\mathbf{f}^{t}_{[1]},\ldots,\mathbf{f}^{t}_{[t]}\},\ldots,\{\mathbf{f}^{n_{\text{fi}}}_{[1]},\ldots,\mathbf{f}^{n_{\text{fi}}}_{[n_{\text{fi}}]}\}\} \\
\mathcal{U} &= \{\mathbf{u}_{l}\}_{i=1}^{n_{\text{fi}}} \\
&= \{\mathbf{u}_{[1]},\ldots,\mathbf{u}_{[n_{\text{fi}}]}\} \\
\mathcal{H} &= \{\{\mathbf{H}^{t}_{[i]}\}_{i=1}^{t-1}\}_{t=2}^{n_{\text{fi}}} \\
&= \{\{\mathbf{H}^{2}_{[1]}\},\ldots,\{\mathbf{H}^{t}_{[1]},\ldots,\mathbf{H}^{t}_{[t-1]}\},\ldots,\{\mathbf{H}^{n_{\text{fi}}}_{[1]},\ldots,\mathbf{H}^{n_{\text{fi}}}_{[n_{\text{fi}}-1]}\}\} \\
\mathcal{V} &= \{\mathbf{V}_{[i]}\}_{i=1}^{n_{\text{fi}}-1} \\
&= \{\mathbf{V}_{[1]},\ldots,\mathbf{V}_{[n_{\text{fi}}-1]}\}
\end{aligned}
\tag{6.7}
$$

By marginalizing the latent variable, the log evidence of the model is given by:

$$
\log p\left(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}},\{\mathbf{X}^{t}_{t-1}\}_{t=2}^{n_{\text{fi}}}|\{\mathbf{X}_t\}_{t=1}^{n_{\text{fi}}},\right) = \log\int\!\!\int\!\!\int\!\!\int p\left(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}},\{\mathbf{X}^{t}_{t-1}\}_{t=2}^{n_{\text{fi}}},\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}|\{\mathbf{X}_t\}_{t=1}^{n_{\text{fi}}}\right)\mathrm{d}\mathcal{F}\mathrm{d}\mathcal{U}\mathrm{d}\mathcal{H}\mathrm{d}\mathcal{V}
\tag{6.8}
$$

Then, the variational approximation used in Eq. (6.6) $q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)$ is introduced as follows:

$$
\log p\left(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}},\{\mathbf{X}^{t}_{t-1}\}_{t=2}^{n_{\text{fi}}}|\{\mathbf{X}_t\}_{t=1}^{n_{\text{fi}}}\right) = \log\int\!\!\int\!\!\int\!\!\int p\left(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}},\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}|\{\mathbf{X}_t\}_{t=1}^{n_{\text{fi}}},\{\mathbf{X}^{t}_{t-1}\}_{t=2}^{n_{\text{fi}}}\right)\frac{q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)}{q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)}\mathrm{d}\mathcal{F}\mathrm{d}\mathcal{U}\mathrm{d}\mathcal{H}\mathrm{d}\mathcal{V}
\tag{6.9}
$$

A lower bound on the log evidence of the model is obtained using Jensen inequality which relates a concave function of an integral (the logarithm in this case) to the concave function of the integral:

$$
\begin{aligned}
&\log p\left(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}},\{\mathbf{X}^{t}_{t-1}\}_{t=2}^{n_{\text{fi}}}|\{\mathbf{X}_t\}_{t=1}^{n_{\text{fi}}}\right) \geq \mathcal{L} \\
&\mathcal{L} = \int\!\!\int\!\!\int\!\!\int q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)\times\log\frac{p\left(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}},\{\mathbf{X}^{t}_{t-1}\}_{t=2}^{n_{\text{fi}}},\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}|\{\mathbf{X}_t\}_{t=1}^{n_{\text{fi}}}\right)}{q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right),}\mathrm{d}\mathcal{F}\mathrm{d}\mathcal{U}\mathrm{d}\mathcal{H}\mathrm{d}\mathcal{V}
\end{aligned}
\tag{6.10}
$$

Then, by replacing the variational distribution by its expression in Eq. (6.6) and canceling out equivalent terms in the numerator and denominator, the following expression is obtained (the dependence on $\{X^t\}_{t=1}^{n_{\text{fi}}}$ is dropped for notation simplicity):

$$\mathcal{L} = \int\int\int\int q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)\log\left(\frac{(p(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}}|\mathcal{F})p(\{\mathbf{X}_{t-1}^t\}_{t=2}^{n_{\text{fi}}}|\mathcal{H})\times p(\mathcal{U})\times p(\mathcal{V})}{q(\mathcal{U})\times q(\mathcal{V})}\right)\mathrm{d}\mathcal{F}\mathrm{d}\mathcal{U}\mathrm{d}\mathcal{H}\mathrm{d}\mathcal{V} \tag{6.11}$$

Next, the log expression is separated into a sum of four terms:

$$\begin{aligned}\mathcal{L} = \int\int\int\int & q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)\log\left(p(\{\mathbf{y}_t\}|\mathcal{F})\right)+\\ & q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)\log\left(p(\{\mathbf{X}_{t-1}^t\}_{t=2}^{n_{\text{fi}}})\right)+\\ & q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)\log\left(\frac{p(\mathcal{U})}{q(\mathcal{U})}\right)+\\ & q\left(\mathcal{F},\mathcal{U},\mathcal{H},\mathcal{V}\right)\log\left(\frac{p(\mathcal{V})}{q(\mathcal{V})}\right)\\ & \mathrm{d}\mathcal{F}\mathrm{d}\mathcal{U}\mathrm{d}\mathcal{H}\mathrm{d}\mathcal{V}\end{aligned} \tag{6.12}$$

The first term does not depend on the variables $\mathcal{H}$,$\mathcal{U}$ and $\mathcal{V}$, thus, it comes back to:

$$\mathcal{L}_1 = \int q\left(\mathcal{F}\right)\log\left(p(\{\mathbf{y}_t\}_{t=1}^{n_{\text{fi}}}|\mathcal{F})\right)\mathrm{d}\mathcal{F} \tag{6.13}$$

For the second term of the sum, the log expression does not depend on the variables $\mathcal{F}$,$\mathcal{U}$ and $\mathcal{V}$, thus, the second term comes back to:

$$\mathcal{L}_2 = \int q\left(\mathcal{H}\right)\log\left(p(\{\mathbf{X}_{t-1}^t\}_{t=2}^{n_{\text{fi}}}|\mathcal{H})\right)\mathrm{d}\mathcal{H} \tag{6.14}$$

For the third term, the log expression does not depend on the variables $\mathcal{F},\mathcal{H}$ and $\mathcal{V}$, thus, the third term comes back to:

$$\mathcal{L}_3 = \int q\left(\mathcal{U}\right)\log\left(\frac{p(\mathcal{U})}{q(\mathcal{U})}\right)\mathrm{d}\mathcal{U} \tag{6.15}$$

For the fourth term, the log expression does not depend on the variables $\mathcal{F},\mathcal{H}$ and $\mathcal{U}$, thus, the fourth term comes back to:

$$\mathcal{L}_4 = \int q\left(\mathcal{V}\right)\log\left(\frac{p(\mathcal{V})}{q(\mathcal{V})}\right)\mathrm{d}\mathcal{V} \tag{6.16}$$

By injecting these terms in eq. (6.12) and identifying the expectation and KL divergence terms, then factorizing over the training data-set, the final expression is obtained:

$$\begin{aligned}\mathcal{L} = &\sum_{t=1}^{n_{\text{fi}}}\sum_{i=1}^{n_t}\mathbb{E}_{q(f_{[t]}^{(i),t})}\left[\log p\left(y_t^{(i)}|f_{[t]}^{(i),t}\right)\right]+\\ &\sum_{t=1}^{n_{\text{fi}}-1}\sum_{i=1}^{n_t}\mathbb{E}_{q(\mathbf{H}_{[t]}^{(i),t+1})}\left[\log p\left(\mathbf{X}_t^{(i),t+1}|\mathbf{H}_{[t]}^{(i),t+1}\right)\right]-\\ &\sum_{t=1}^{n_{\text{fi}}}\mathbb{KL}\left[q\left(\mathbf{u}_{[t]}\right)||p\left(\mathbf{u}_{[t]};\mathbf{Z}_{[t-1]}\right)\right]-\\ &\sum_{t=1}^{n_{\text{fi}}-1}\mathbb{KL}\left[q\left(\mathbf{V}_{[t]}\right)||p\left(\mathbf{V}_{[t]};\mathbf{Z}_{[t+1]}\right)\right]\end{aligned} \tag{6.17}$$

The Kulback-Leibler divergence is analytically tractable for Gaussian distributions [Kullback and Leibler, 1951]. The expectation term is approximated using Monte-Carlo sampling with s samples. Therefore, the complexity of the model for a number of induced inputs equal to the number of observations at each layer is $\mathcal{O}\left(s\times\left(n_1^3+\sum_{t=2}^{n_{\text{fi}}}\left(d_t\times n_t^3\right)\right)\right)$. The difference with regular MF-DGP with complexity $\mathcal{O}\left(s\times\left(\sum_{t=1}^{n_{\text{fi}}}\left(n_t^3\right)\right)\right)$ comes from the fact that MF-DGP-EM uses GP input mappings that are multi-output.

Prediction

The prediction of a test data $\mathbf{X}_t^*$ belonging to the input-space of fidelity t using the two-level MF-DGP-EM is a two-step process. First, the test data $\mathbf{X}_t^*$ are propagated through the first level of the MF-DGP-EM allowing the projection of the test data on the lower-fidelity inputs spaces to obtain $H_{[t-1]}^*,\dots,H_{[1]}^*$. Then, propagation through the second level is carried out to obtain the evaluation at the different fidelities. Hence, a prediction of $\mathbf{X}_t^*$ with fidelity t is:

$$q(\mathbf{f}_{[t]}^*)=\frac{1}{s}\sum_{j=1}^{s}q\left(\mathbf{f}_{[t]}^*|q(\mathbf{u}_{[t]}),\{[\mathbf{H}_{[t]}^*,\mathbf{f}_{[t-1]}^*]\},\mathbf{Z}_{[t-1]}\right) \tag{6.18}$$

where s is the number of propagated samples.

6.2.4 Numerical experiments on multi-fidelity problems with different input space domain definitions

To evaluate the performance of the proposed model MF-DGP-EM, numerical experiments are carried out in this section. Firstly, analytical test problems are considered. The first analytical test case is an illustrative example to compare the different approaches and also to point out the efficiency of MF-DGP-EM on problems where classical fixed input-space parametrization approaches are used (MF-DGP). The two remaining analytical test problems address the cases where different dimensions and parametrizations are considered for each input-space. Two physical test problems are also presented: a structural multi-fidelity problem and an aerodynamic multi-fidelity problem. The prediction accuracy is assessed using the R squared metric (R2) and the Root Mean Square Error (RMSE). The test Mean Negative Log-Likelihood (MNLL) metric is used to validate the uncertainty quantification on the prediction which is important for the trade-off between exploration and exploitation for adaptive design of experiments and optimization. The final part of this section discusses the computational aspect of MF-DGP-EM with respect to the compared approaches.

Details on the numerical setup are presented in Appendix D.

Analytical problems

Illustrative test problem

For this toy problem, the non-linear multi-fidelity problem proposed in [Perdikaris et al., 2017] is used. The high-fidelity function $f_{hf}(\cdot)$ is defined as a function of the low-fidelity function $f_{lf}(\cdot)$:

$$f_{hf}(x_1) = x_1 \exp\Big(f_{lf}(2x_1 - 0.2)\Big) - 1 \tag{6.19}$$

where f_{lf} is:

$$f_{lf}(x_1) = \cos(15x_1) \tag{6.20}$$

where $0 \leq x_1 \leq 1$.

This multi-fidelity problem has been used previously in [Perdikaris et al., 2017] and [Cutajar et al., 2019] in the context of multi-fidelity modeling with the same input variable parametrization. However, one can argue that this problem can be interpreted as a multi-fidelity problem with different input-space parametrizations. In fact, based on Eq. (6.19) the nominal mapping $g_0(\cdot)$ between the two input spaces can be defined as:

$$g_0(x_{hf}) = 2x_{hf} - 0.2 \tag{6.21}$$

This nominal mapping is compared to the IMC mapping. The IMC approach is used to obtain a calibrated mapping that tries to minimize the distance between the outputs of the two fidelities according to the following equation:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \left(\sum_{i=1}^{n_{hf}} \left({y_{hf}}^{(i)} - f_{lf}^{\text{exact}} \left(\boldsymbol{\psi}_{\beta} \left({\mathbf{x}_{hf}}^{(i)} \right) \right) \right)^2 + \tau(\boldsymbol{\beta}, \boldsymbol{\beta_0}) \right) \tag{6.22}$$

where n_{hf} corresponds to the number of HF training data points, $\boldsymbol{\psi}_{\beta}$ to parametric mapping parameterized by a set of parameters $\boldsymbol{\beta}$, and $\tau(\boldsymbol{\beta}, \boldsymbol{\beta_0})$ to a regularization term based on the parameters $\boldsymbol{\beta_0}$ of the nominal mapping. For this problem, a linear parametric mapping is considered for the IMC. Moreover, 14 HF training data points are sampled using LHS. The obtained mapping by IMC $\psi(\cdot)$ is the following:

$$\psi(x_{hf}) = 0.0715x_{hf} + 0.65924 \tag{6.23}$$

The IMC is compared to the nominal mapping in Fig. 6.10. The IMC minimizes the distance between the outputs of the HF and LF following Eq. (6.22). However, in doing so, it maps the HF input-space to a small range interval in the LF input-space $[0,1] \rightarrow [0.659, 0.730]$. To analyze this mapping in the HF output space, Fig. 6.11 represents the exact output of the HF, the output of the LF composed with the nominal mapping, and the output of the LF composed with the IMC. The IMC results in a quadratic mean trend of the HF observations and looses the sinusoidal feature of the LF. To analyze the repercussions of such a behavior from a multi-fidelity model prediction accuracy point of view, a HF GP model prediction (Fig. 6.12) is compared to a bias correction approach (Eq. (3.54)) used with the nominal values (BC nominal) and a bias correction approach used with the IMC mapping (Fig. 6.13). The BC IMC (RMSE: 0.482) deteriorates the prediction accuracy obtained by a GP model using only the HF data (RMSE: 0.311) because of the non-adequate projection from the HF to the LF. However, the BC nominal improves the prediction accuracy (RMSE: 0.265) since the LF encodes exactly the oscillation phase information about the HF (Fig. 6.11).

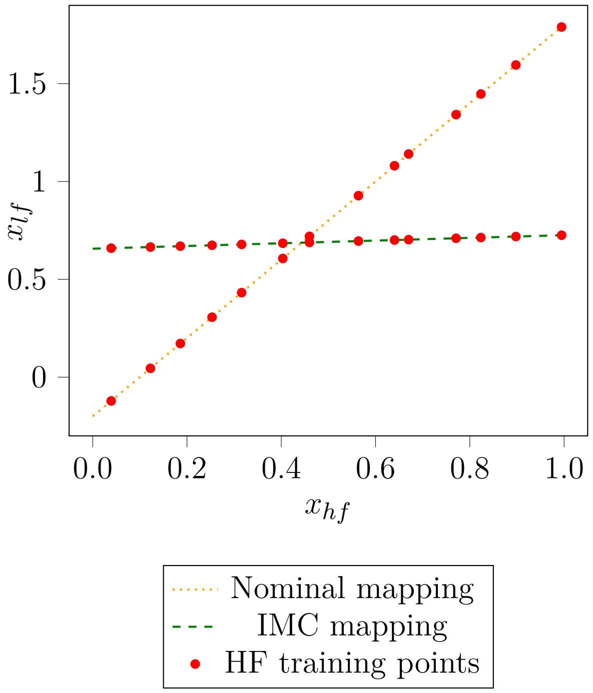


Fig. 6.10 Parametric linear mapping obtained by IMC compared to the nominal mapping in the input-space.

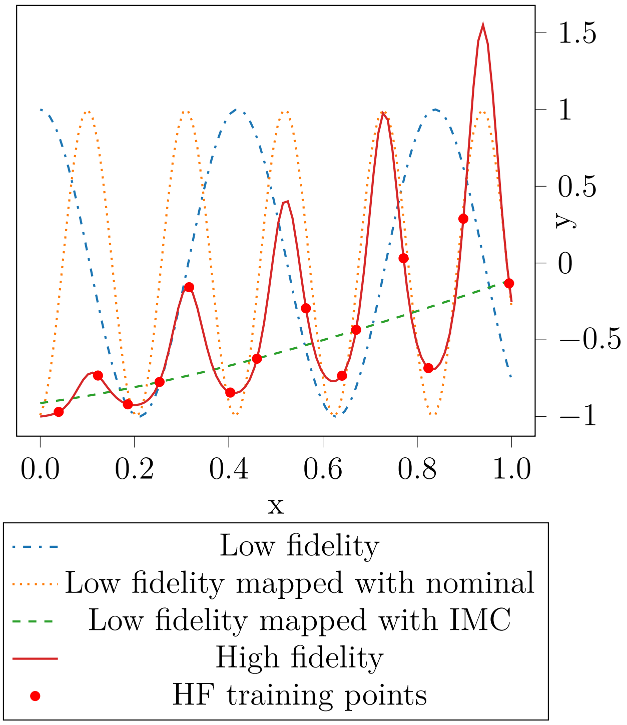


Fig. 6.11 The input mapping results in the output space. The IMC output minimizes the distance from the HF training data points, however, the correlation LF and HF is lost.

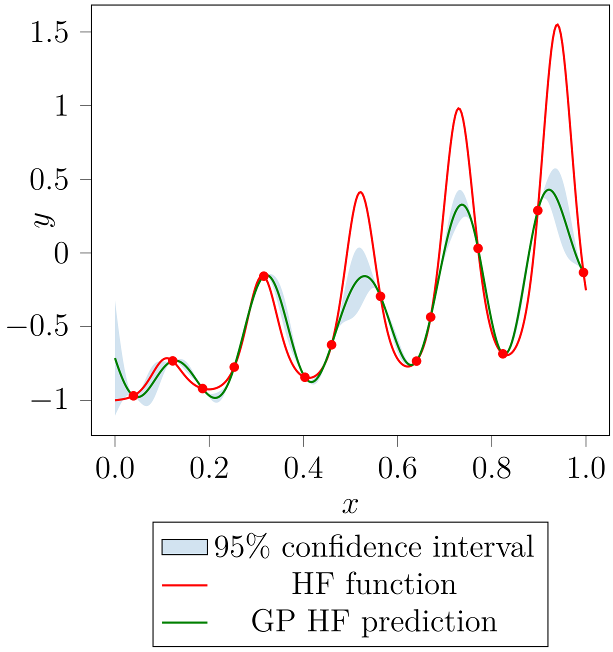


Fig. 6.12 The prediction and uncertainty obtained by a GP model of the HF. GP HF metrics: R^2 : 0.756, RMSE: 0.311, MNLL: -0.106

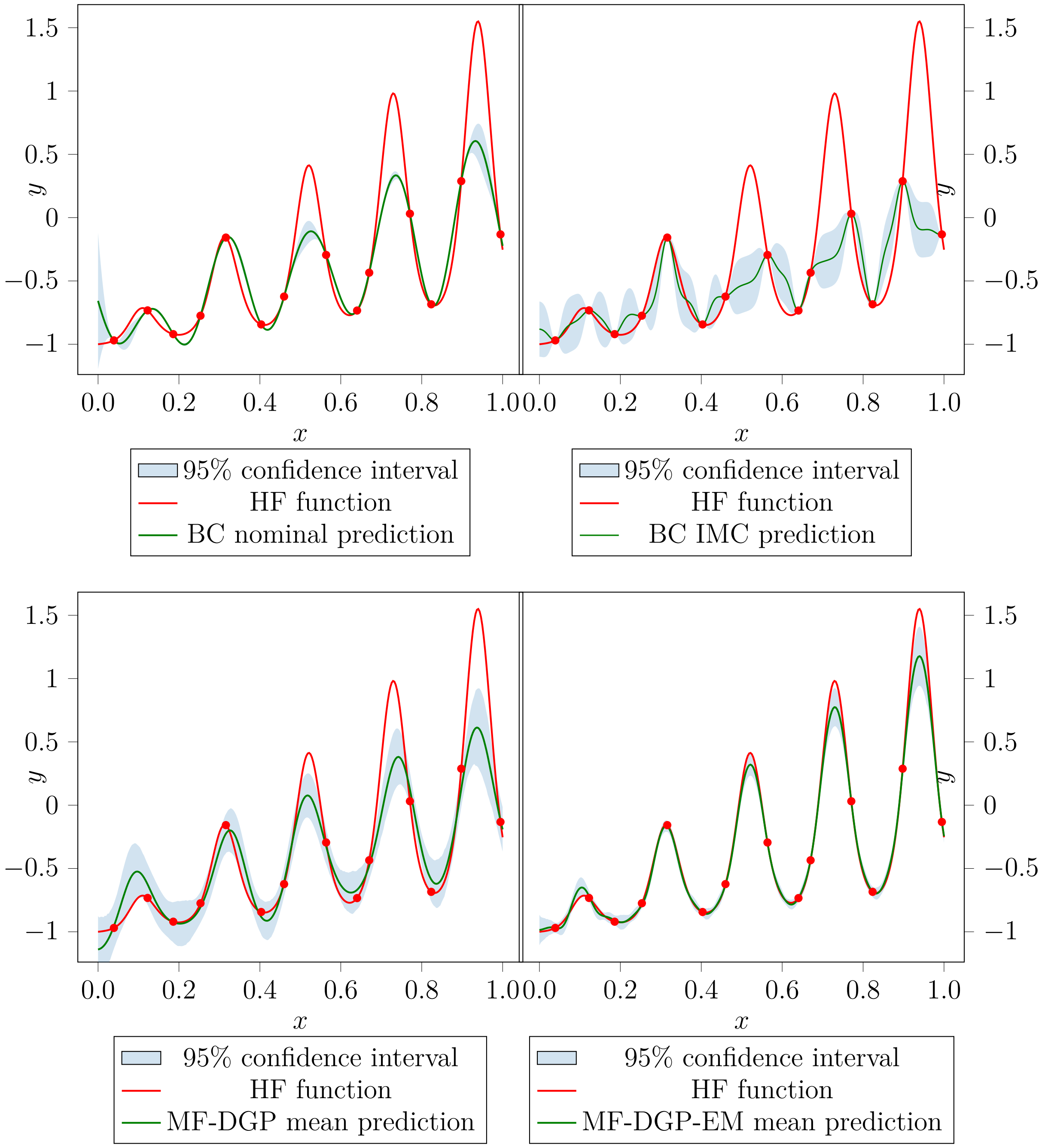


Fig. 6.13 The prediction and uncertainty obtained by a bias correction approach using nominal mapping values (top left) and by a bias correction approach using IMC values (top right). MF-DGP (bottom left) and by MF-DGP-EM (bottom right). MF-DGP-EM takes into account the relationship between the input-spaces which improves the results. BC nominal metrics: R^2 : 0.824, RMSE: 0.265, MNLL: 0.638. BC IMC metrics: R^2 : 0.416, RMSE: 0.482, MNLL: 0.938. MF-DGP metrics: R^2: 0.840, RMSE: 0.255, MNLL: 0.231. **MF-DGP EM metrics: R^2 : 0.984, RMSE: 0.077, MNLL: -1.921.**

MF-DGP-EM using the nominal mapped values of the HF training data and standard MF-DGP that considers the same input variable parametrization, are also compared in Fig. 6.13. For these two models, the exact LF model is not used and 30 training LF points are sampled using LHS. The MF-DGP-EM can embed the information of the input-space mapping to improve the prediction accuracy and uncertainty quantification (RMSE: 0.077, MNLL: −1.921) of the MF-DGP (RMSE: 0.255, MNLL: 0.231). This result is interesting since the MF-DGP-EM was applied to a problem that was previously treated as a multi-fidelity problem with the same input-space parametrization. Therefore, when there is some information about the input-space relationship between the different fidelities, the modeling can be improved using MF-DGP-EM even in problems with the same input dimensions.

As in this case, in the next problems, the different fidelities may not share the same trend. Moreover, the LF is not considered necessarily computationally free. Hence, for comparison, the BC approach with nominal mapping is preferred to the IMC approach.

Varying input-space test problems

To assess the efficiency of the proposed MF-DGP-EM, a comparison is carried out by modeling the high-fidelity using a GP (GP-HF) and to Bias Correction approach (BC) with nominal mapping [Li et al., 2016]. This comparison with BC is interesting since in this approach, the nominal mapping functions are used to define the relationship between the fidelities, in contrast with MF-DGP-EM where only nominal mapped values of the training HF data are known and the mapping has yet to be learned. Two problems described in the following are used for this analytical comparison.

Problem 1: The first test case is based on the Park multi-fidelity problem [Xiong et al., 2013]. The low-fidelity model is considered only with two variables (Eq. (6.24) and Eq. (6.25)). This problem depicts the case where some variables are neglected in the low-fidelity model for simplicity. The nominal mapping is naturally the identity mapping of the HF variables (Eq. (6.26)).

The high-fidelity function is four-dimensional with an input domain $[0,1]^4$:

$$\begin{aligned} f_{hf}(x_1,x_2,x_3,x_4) = &\frac{x_1}{2}\left(\sqrt{1+(x_2+x_3^2)\frac{x_4}{x_1^2}}-1\right) \\ &+(x_1+3x_4)\exp\left(1+\sin(x_3)\right) \end{aligned} \tag{6.24}$$

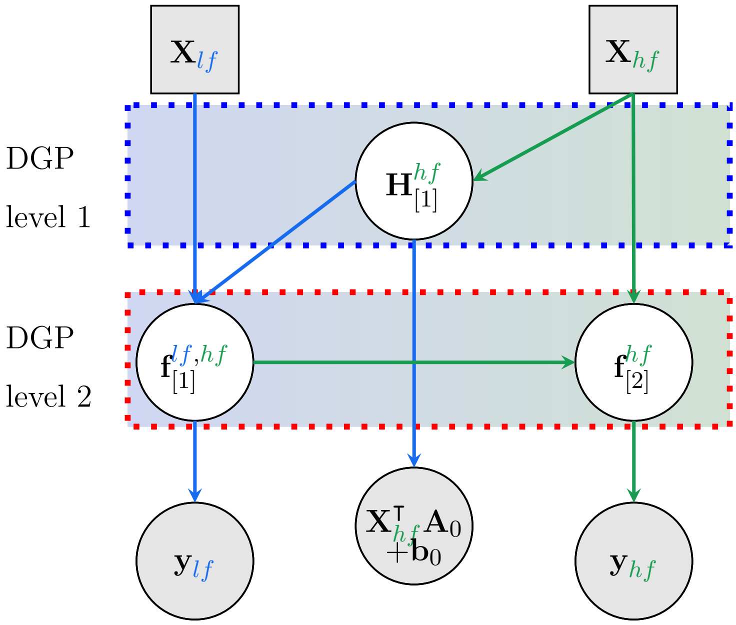


Fig. 6.14 Graphical representation of MF-DGP-EM for Problem 1. The fidelity GPs $f_{[1]}(\cdot)$ and $f_{[2]}(\cdot)$ are conditioned respectively on the LF and HF observations. The input mapping GP $\mathbf{H}_{[1]}(\cdot)$ is conditioned on the nominal mapping defined in eq. (6.26) $\mathbf{X}_{hf}^{\intercal}\mathbf{A}_0 + \mathbf{b}_0$.

The low-fidelity function is two-dimensional with an input domain $[0,1]^2$:

$$\begin{aligned} f_{lf}(x_1,x_2) = &\left(1+\frac{\sin(x_1)}{10}\right) f_{hf}(x_1,x_2,0.5,0.5) - 2x_1 \\ &+ x_2^2 + 0.75 \end{aligned} \tag{6.25}$$

The nominal mapping is a linear mapping $\mathbf{X}^{\intercal}\mathbf{A}_0 + \mathbf{b}_0$ with:

$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ and } \mathbf{b}_0 = [0,0] \tag{6.26}$$

The configuration of MF-DGP-EM for this problem is illustrated in Fig. 6.14. Since only two fidelity levels are considered, the first level DGP comes back to a multi-output GP.

Problem 2 : The second test case is a problem describing the situation in which the fidelities are parameterized in different input-spaces (cartesian and spherical parametrizations), in addition to different dimensionalities (Eq. (6.27) and Eq. (6.28)).

Table 6.8 Performance of the different multi-fidelity models on Problem 1 (Eqs. 6.24, 6.25) using 20 repetitions with different LHS generated DoE. Three scenarios on the available HF information are experimented (4, 6, and 8 input data on the HF). 30 training data points are used for the LF and 1000 test data points to compute the metrics in the HF space.

Analytical Problem 1				
HF DoE size	**Algorithms**	R2 (std)	RMSE (std)	MNLL (std)
4 data points	HF model	0.4381 (0.4511)	3.1673 (1.3076)	3974.3 (16921.3)
	BC model	0.7877 (0.3718)	1.8324 (1.0386)	2428.4 (4192.4)
	MF-DGP-EM	**0.9187 (0.1505)**	**1.1020 (0.6964)**	**15.756 (47.556)**
6 data points	HF model	0.9112 (0.1046)	1.2378 (0.5670)	**1.5146(0.5407)**
	BC model	0.9185 (0.0398)	1.2545 (0.3581)	921.27 (2775.2)
	MF-DGP-EM	**0.9731 (0.0200)**	**0.7146 (0.2230)**	3.8986 (5.4270)
8 data points	HF model	0.9037 (0.1686)	1.1389 (0.8453)	19.105 (75.129)
	BC model	0.9476 (0.0489)	0.9351 (0.4686)	13.875 (33.041)
	MF-DGP-EM	**0.9874 (0.0093)**	**0.4784 (0.1803)**	**1.3614 (1.5949)**

The high-fidelity function is three-dimensional with an input domain $[0,1]^3$:

$$\begin{aligned} f_{hf}(r,\theta,\phi) =& 3.5\left(r\cos\left(\frac{\pi}{2}\phi\right)\right)+2.2\left(r\sin\left(\frac{\pi}{2}\theta\right)\right) \\ &+0.85\left(\left|r\cos\left(\frac{\pi}{2}\theta\right)-2r\sin\left(\frac{\pi}{2}\theta\right)\right|\right)^{2.2} \\ &+\frac{2\cos(\pi\phi)}{1+3r^2+10\theta^2} \end{aligned} \tag{6.27}$$

The low-fidelity function is two-dimensional with an input domain $[0,1]^2$:

$$f_{lf}(x_1,x_2) = 3x_1+2x_2+0.7(|x_1-1.7x_2|)^{2.35} \tag{6.28}$$

The nominal mapping values are based on the transformation of the training high-fidelity points using:

$$x_1 = r\cos\left(\frac{\pi}{2}\phi\right) \tag{6.29}$$

$$x_2 = r\sin\left(\frac{\pi}{2}\theta\right) \tag{6.30}$$

To assess the performance of the algorithms on different scenarios depending on the available HF information, three different sizes of the HF DoE are experimented (4, 6, and 8 HF data points). The robustness concerning the distribution of the HF data points in the input-space is evaluated using 20 repetitions with different LHS for each size of the DoE. For all the scenarios, the number of LF training data points is fixed to 30 training data points. Fig. 6.15 and Fig. 6.16 present the results obtained by the different models.

Table 6.9 Performance of the different multi-fidelity models on Problem 2 (Eqs. 6.27, 6.28) using 20 repetitions with different LHS generated DoE. Three scenarios on the available HF information are experimented (4, 6, and 8 input data on the HF). 30 training data points are used for the LF and 1000 test data points to compute the metrics in the HF space.

Analytical Problem 2				
HF DoE size	**Algorithms**	R2 (std)	RMSE (std)	MNLL (std)
4 data points	HF model	0.2549 (0.3998)	1.5514 (0.4380)	8016.6 (31752)
	BC model	**0.6248 (0.2189)**	**1.0940 (0.3336)**	193.68 (732.25)
	MF-DGP-EM	0.4509 (0.4411)	1.2813 (0.5226)	**14.110 (17.801)**
6 data points	HF model	0.4958 (0.4079)	1.2187 (0.5225)	468.17 (1545)
	BC model	0.7412 (0.2343)	0.8742 (0.3718)	93.985 (262.30)
	MF-DGP-EM	**0.7946 (0.1996)**	**0.7850 (0.3158)**	**4.6228 (4.4710)**
8 data points	HF model	0.7867 (0.2299)	0.7959 (0.3320)	9.1492 (33.817)
	BC model	0.8821 (0.0431)	0.6302 (0.1171)	4.4421 (7.3884)
	MF-DGP-EM	**0.9111 (0.0465)**	**0.5372(0.1459)**	**3.9798(4.1756)**

On Problem 1 (Table 6.8), MF-DGP-EM is more efficient and robust to the DoE in each scenario than the other algorithms. In fact, with a DoE size of only 4 data points for HF, the MF-DGP-EM obtains better and more robust results both in terms of prediction accuracy (RMSE: 1.10, with a std of 0.69) and uncertainty quantification (MNLL: 15.75, with a std of 47.55) compared to the BC approach (RMSE: 1.83, with a std of 1.04, MNLL: 3974, with a std of 16921) and the GP HF (RMSE: 3.17, with a std of 1.31, MNLL: 2428, with a std of 4192). The BC approach improves the prediction accuracy of the GP HF in the case where there is not enough information in the HF (4 data points). However, the relative improvement with respect to GP HF decreases when the number of HF data points crosses the threshold of 6 data points (BC RMSE for 6 HF data points: 1.25 and for 8 HF data points: 0.93, GP HF RMSE for 6 HF data points: 1.24 and for 8 HF data points: 1.14). This is not the case for the MF-DGP-EM which continues to improve the prediction accuracy even when the HF information increases (MF-DGP-EM RMSE for 6 HF data points: 0.71 and for 8 HF data points: 0.48). The uncertainty quantification obtained by BC is less accurate than the other approaches in the three scenarios (MNLL for 4 HF data points: 3974, for 6 HF data points: 921 and for 8 HF data points: 19.10).

For Problem 2 (Table 6.9), it is interesting to observe that in the scenario of 4 data points, the MF-DGP-EM, whilst showing improvement compared to the GP HF in terms of prediction accuracy (MF-DGP RMSE: 1.28, GP HF RMSE: 1.55), it is not as good as the BC approach (RMSE: 1.09). This is due to the difficulty to learn the mapping with only 4 training data points. However, in terms of uncertainty quantification, MF-DGP-EM gives the better results in the three scenarios (MNLL for 4 HF data points: 14.11, for 6 HF data points: 4.62 and for 8 HF data points: 3.97) compared to either the BC approach (MNLL for 4 HF data points: 193.68, for 6 HF data points: 93.98 and for 8 HF data points: 4.44) or the GP HF (MNLL for 4 HF data points: 8016, for 6 HF data points: 468 and for 8 HF data points: 9.14). This is because even if there is not enough information to learn the input mapping (the case of 4 HF data points), the uncertainty quantification on this mapping is well balanced which enables the uncertainty on the prediction to be better. By increasing the HF data size (6 and 8 data points), the MF-DGP-EM better learns the mapping between the input-spaces and gives also the better results in terms of prediction accuracy (RMSE for 6 HF data points: 0.78 and for 8 HF data points: 0.54) compared to the BC approach (RMSE for 6 HF data points: 0.87 and for 8 HF data points: 0.63) and the GP HF approach (RMSE for 6 HF data points: 1.22 and for 8 HF data points: 0.79).

In conclusion of these two first numerical experiments, the MF-DGP-EM presents generally better results in terms of prediction accuracy, uncertainty quantification, and robustness to the DoE when the mapping relationship is well learned.

Structural problem

The first physical problem is a structural modeling problem. The objective is to model the maximum distortion criterion (also known as von Mises yield criterion) of a cantilever beam with a rectangular hole inside. This criterion expresses the needed elastic energy of distortion for the yielding of the structure to begin. The Euler-Bernoulli beam theory [Bauchau and Craig, 2009] is used for the low and high-fidelity models.

In the low-fidelity a standard solid rectangular cantilever beam (Fig. 6.17) characterized by its length L, its width d, and the applied force at its extremity F is considered (3 LF variables). In this case, the computation of the maximum distortion is computed analytically using the von Mises equation:

$$\sigma_{VM} = \sqrt{(\sigma_{ax} + \sigma_b)^2 + 3\tau_{sh}^2} \tag{6.31}$$

where σ_{ax} is the axial stress, σ_b the bending stress and τ_{sh} the shear stress. For this simplified cantilever beam problem, the maximal von Mises (VM) stress is reached at the basis of the beam (meaning at $x = 0$ on Fig. 6.18). At the basis, the axial stress is null, the shear stress is given by $\tau_{sh} = \frac{F}{l^2}$ and the bending stress is equal to $\sigma_b = \frac{6F \times L}{d^3}$. Therefore, given the parameters F, L and d, it is possible to easily estimate analytically the maximal VM within the beam.

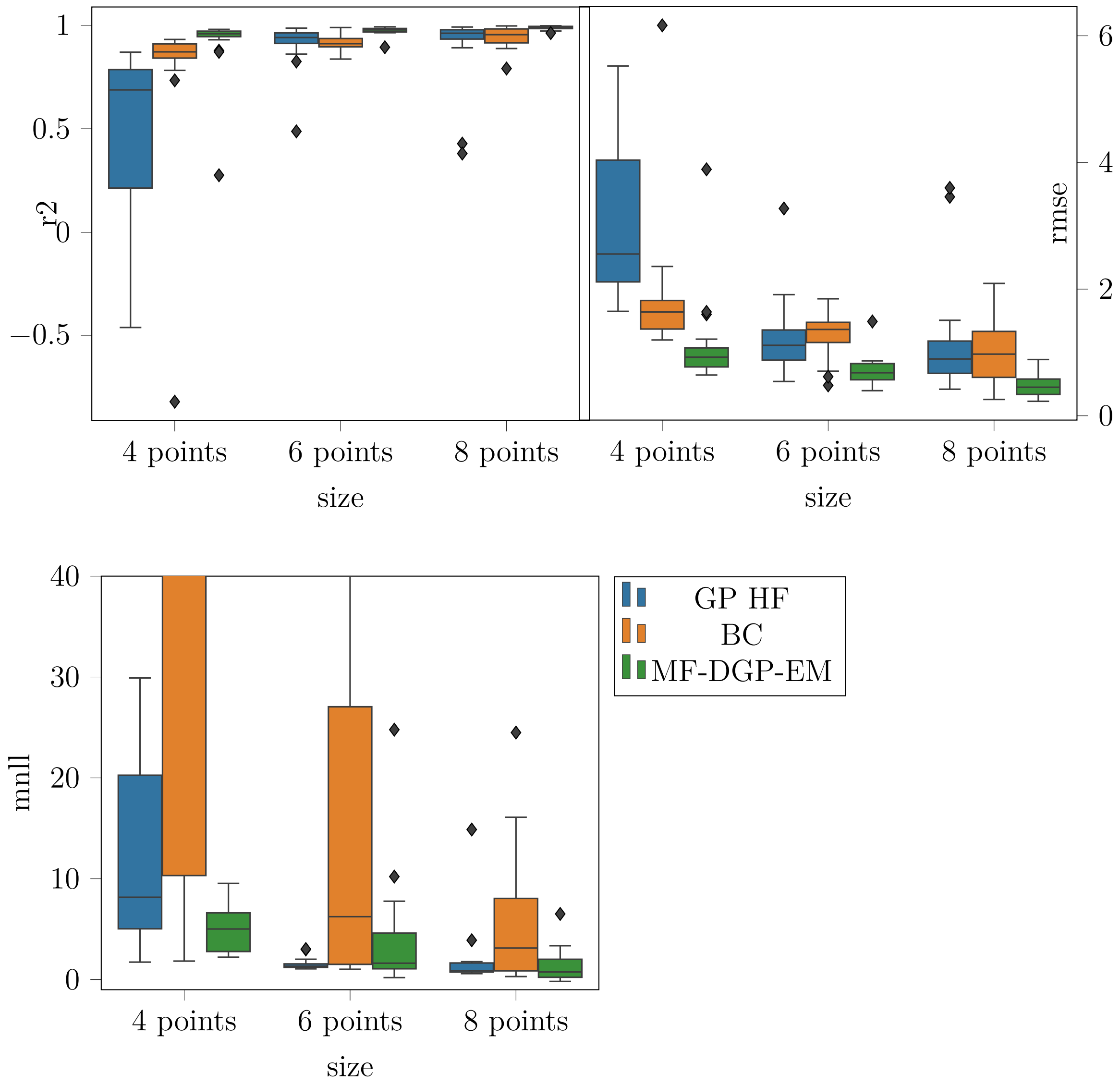


Fig. 6.15 Performance of the different models (GP HF for the high-fidelity GP, BC for the Bias correction approach, and MF-DGP-EM for the proposed model) on Problem 1 with 3 sizes of DoE (4, 6, and 8 data points on the HF and 30 data points on the LF) using 20 LHS repetitions.

In the high-fidelity, a rectangular cantilever beam with a rectangular bore along its horizontal axis is considered (Fig. 6.18). The HF variables are the length and width of the cantilever beam, the applied force at its end, and also the width and length of the rectangular bore (5 HF variables).

The maximum distortion can not be computed analytically in the case of the beam considered in the HF model. It is necessary to follow a finite element (FE) analysis

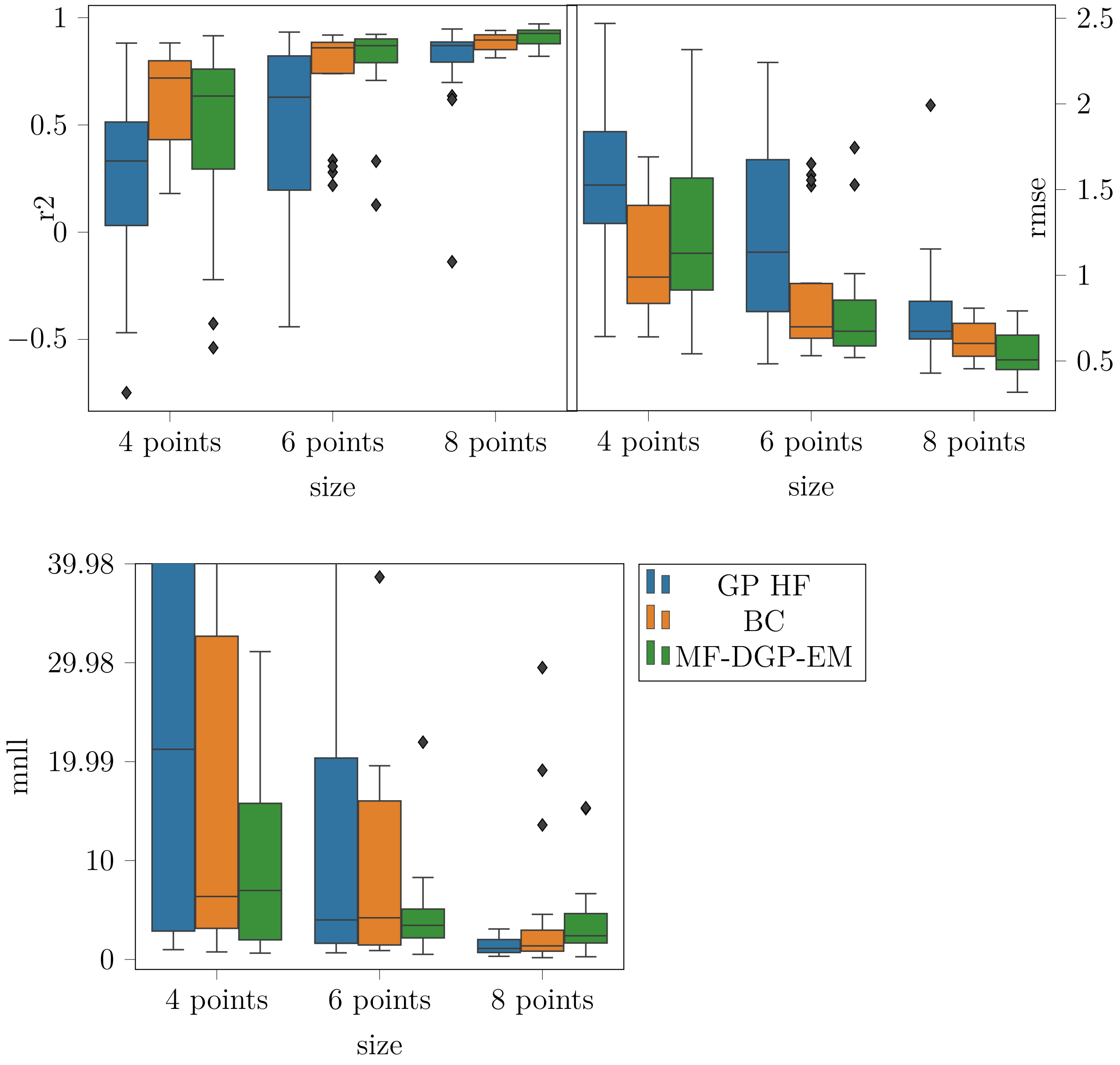


Fig. 6.16 Performance of the different models (GP HF for the high-fidelity GP, BC for the Bias correction approach, and MF-DGP-EM for the proposed model) on Problem 2 with 3 sizes of DoE (4, 6, and 8 data points on the HF and 30 input data on the LF) using 20 LHS repetitions.

approach. In this case, Caculix solver [Dhondt, 2017] is used. A FE analysis can be computationally expensive according to the mesh refinement used (Fig. 6.19 and Fig. 6.20). Hence, only a few evaluations of the HF are available. In the present case, the LF model provides an appropriate approximation of the HF model with a reduced computational cost, which makes interesting the use of multi-fidelity approaches to enrich the HF with LF information. However, the classical multi-fidelity approaches

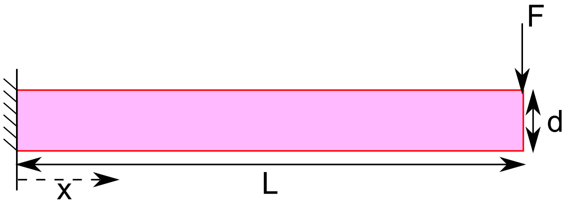


Fig. 6.17 low-fidelity beam representation. A standard solid rectangular cantilever beam characterized by its length L, its width d, and the applied force at its extremity F.

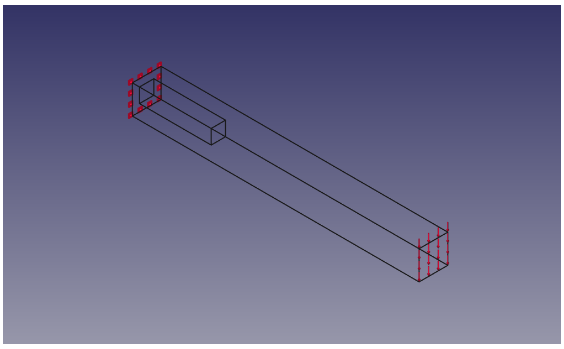


Fig. 6.18 high-fidelity beam representation. A rectangular cantilever beam with a rectangular bore along its horizontal axis.

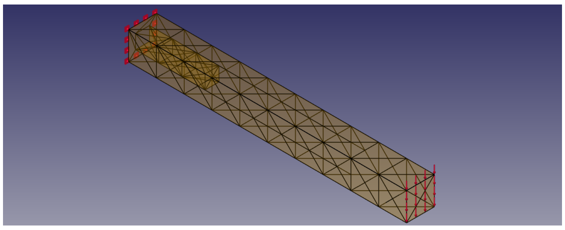


Fig. 6.19 Mesh grid used for the FE analysis of the high-fidelity cantilever beam.

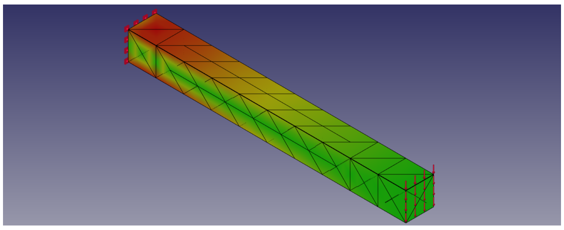


Fig. 6.20 Obtained distortion along the cantilever beam using the FE analysis.

can not be used because of the difference in dimensionality between the input-spaces of the HF and LF models (3 for the LF and 5 for the HF). Hence, MF-DGP-EM is used and compared to the BC approach and to using only the HF information (GP HF). Since in this case the LF design variables are included in the HF design variables, the nominal mapping is the identity with omission of 2 variables (the length and the width of the rectangular bore). The performance of the models is assessed on different scenarios of the available HF information. In fact, three different sizes of the HF DoE are experimented (4, 6, and 8 data points). The robustness with respect to the distribution of the HF data points in the input-space is evaluated using 20 repetitions with different LHS for each size of the DoE. For all the scenarios, the number of LF training data points is fixed to 30 training data points.

The results obtained are presented in Table 6.10 and illustrated in Fig. 6.21. In terms of prediction accuracy, the GP-HF is outperformed by the multi-fidelity approaches in the three scenarios which highlights the relevance of the low-fidelity model. With a DoE size of only 4 data points for HF, the BC approach outperforms the MF-DGP-EM

Table 6.10 Performance of the different multi-fidelity models on the structural problem (Section 6.2.4) using 20 repetitions with different LHS generated DoE. Three scenarios on the available HF information are experimented (4, 6, and 8 input data on the HF). 30 training data points are used for the LF and 1000 test data points to compute the metrics in the HF space.

Structural problem				
HF DoE size	**Algorithms**	R2 (std)	RMSE (std)	MNLL (std)
4 data points	HF model	0.1977 (1.1167)	0.8751 (0.4474)	7668.9 (21292)
	BC model	**0.8702 (0.1793)**	**0.3471 (0.1063)**	11542 (38744)
	MF-DGP-EM	0.4997 (0.2110)	0.8309 (0.4070)	**4.2601(7.0356)**
6 data points	HF model	0.6760 (0.4672)	0.5934 (0.2483)	53.055 (117.96)
	BC model	**0.9320 (0.0130)**	**0.3103 (0.0814)**	14866 (62243)
	MF-DGP-EM	0.9204 (0.0402)	0.3281 (0.1156)	**13.200 (18.66)**
8 data points	HF model	0.8032 (0.2375)	0.3895 (0.1750)	14.131 (30.054)
	BC model	0.9179 (0.0782)	0.2496 (0.0793)	76.925 (170.36)
	MF-DGP-EM	**0.9400 (0.0362)**	**0.2285 (0.0768)**	**5.7554 (7.308)**

approach in terms of prediction accuracy (BC: RMSE: 0.35, with a std of 0.10, MF-DGP-EM: RMSE: 0.83, with a std of 0.4). This can be explained by the fact that the relationship between the two fidelities is well approximated by a linear function, which makes it easier for the BC approach to capture the HF with only few information. By increasing the size of the training HF data (6 and 8 data points), the MF-DGP-EM gives comparable results to the BC approach in terms of prediction accuracy (MF-DGP-EM for 6 HF data points RMSE: 0.33 and for 8 HF data points RMSE: 0.23; BC for 6 HF data points RMSE: 0.31 and for 8 HF data points RMSE: 0.25). However, as observed in the analytical test problems, one of the main advantages of the MF-DGP-EM is the quality of the uncertainty quantification. In fact, even if the prediction accuracy is not as good as the one obtained by the BC approach (case of 4 HF data points) the added uncertainty on the nominal mapping allows the MF-DGP-EM to obtain better results in terms of uncertainty quantification (MF-DGP-EM MNLL: 4.26; BC MNLL: 11542 in the case of 4 HF data points). The BC approach gives less accurate results in the three scenarios when it comes to uncertainty quantification (MNLL for 6 HF data points: 14866 and for 8 HF data points: 76.9).

Aerodynamic problem

In this problem, the objective is to model the lift coefficient (CL) of a winged reusable launch vehicle composed of a core, two wings, and two canards [Brevault et al., 2020a] (Fig. 6.22 and Fig. 6.23). The Vortex Lattice Method (VLM), is used for the computation of CL using openVSP and VSPAERO [Gloudemans et al., 1996]. It is a computational fluid dynamics numerical approach that models lifting surfaces, using discrete vortices to compute lift and induced drag. The span of the main wings and

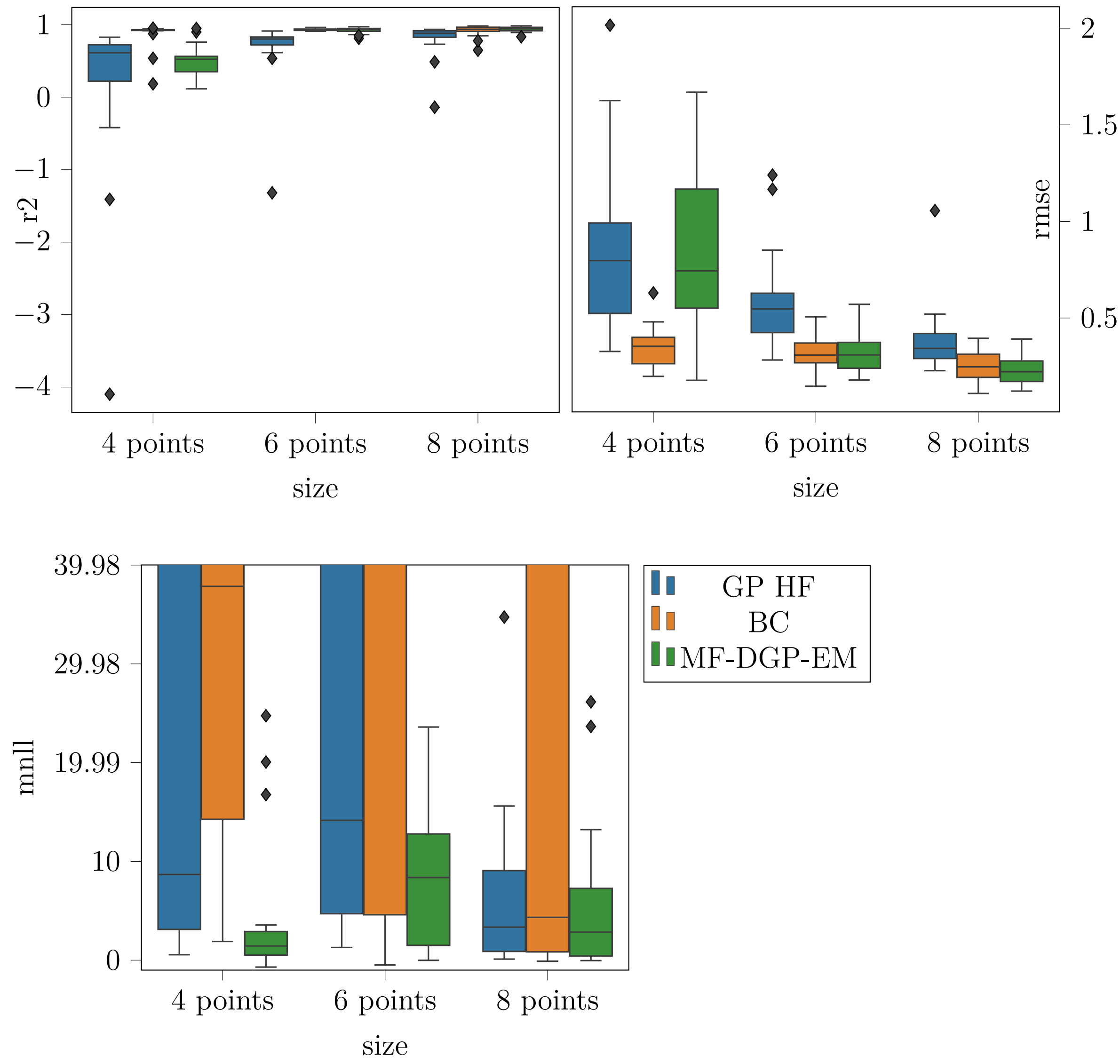


Fig. 6.21 Performance of the different models (GP HF for the high-fidelity GP, BC for the Bias correction approach, and MF-DGP-EM for the proposed model) on the structural test problem with 3 sizes of DoE (4, 6, and 8 data points on the HF and 30 data points on the LF) using 20 LHS repetitions.

the canards are fixed for the two fidelities and flight conditions of Mach number equal to 0.5 and angle of attack of 2 degrees are considered.

In the low-fidelity, wings and canards with only one section are considered. The variables involved in this case are:

- root chord (RC_m) of the main wings,

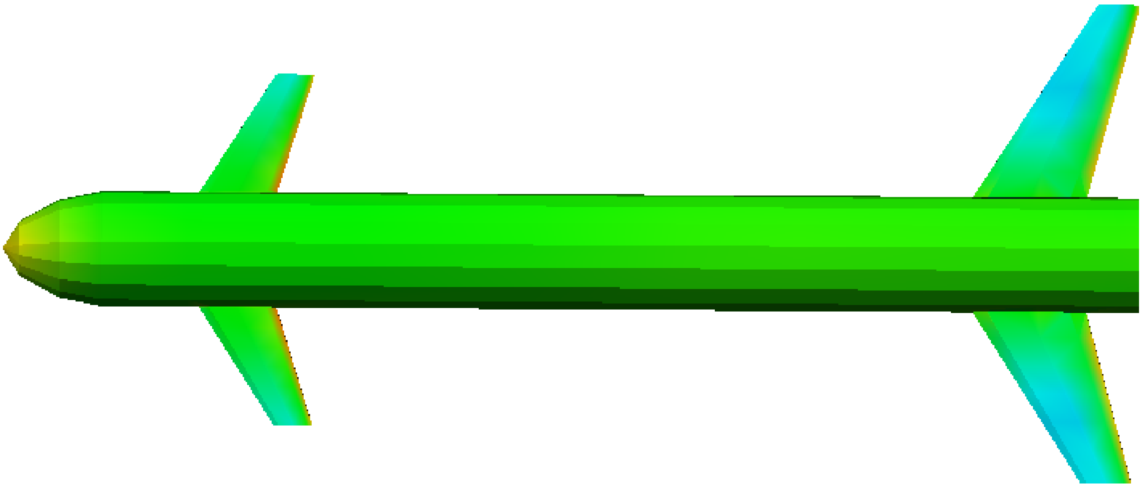


Fig. 6.22 low-fidelity winged reusable vehicle representation

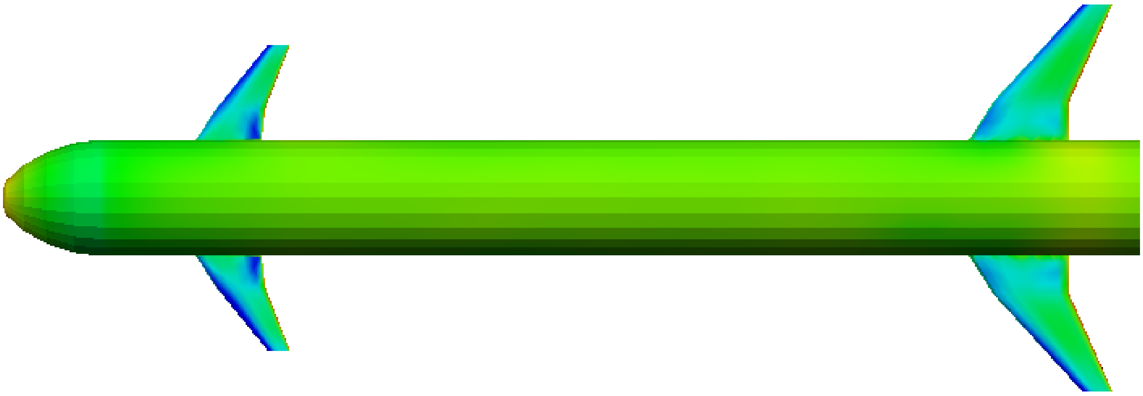


Fig. 6.23 high-fidelity winged reusable vehicle representation

- tip chord (TC_m) of the main wings,
- sweep angle (β_m) of the main wings,
- root chord (RC_c) of the canards,
- tip chord (TC_c) of the canards,
- sweep angle (β_c) of the canards.

Thus, the input-space of the LF is 6-dimensional. As mentioned previously, some LF models, even though they are less computationally expensive than the HF, they are still not computationally free. This is the case in this problem where the low-fidelity configuration requires a simplified CFD analysis for the computation of CL based

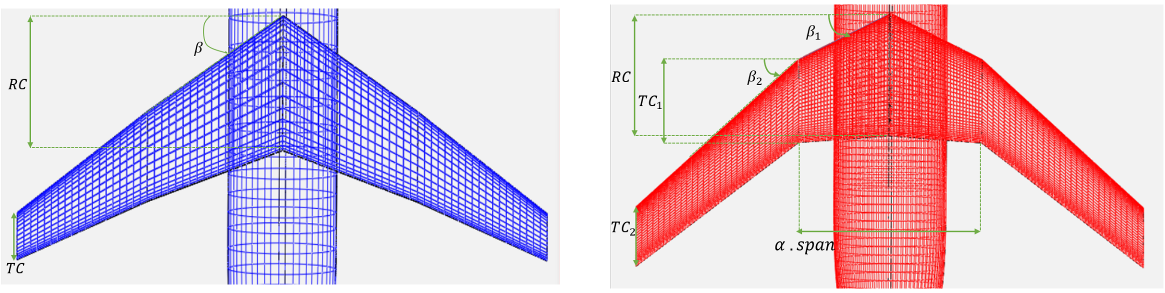


Fig. 6.24 A one-section wing characterized by 3 design variables: its root chord (RC), tip chord (TC), and the sweep angle (β) (left) can be used as a low-fidelity model of a two-section wing characterized by 6 design variables: its root chord (RC), tip chord of the first section (TC_1), tip chord of the second section (TC_2), sweep angle of the first section (β_1), sweep angle of the second section (β_2) and the relative span of the first section (α) (right).

on VLM. In the high-fidelity configuration, wings and canards with two sections are considered and meshes have been densified (number of tessellated curves has been doubled). The variables involved in this case are (Fig. 6.24):

- root chord (RC_m) of the main wings,
- tip chord (TC_{m_1}) of the first section of the main wings,
- tip chord (TC_{m_2}) of the second section of the main wings,
- sweep angle (β_{m_1}) of the first section of the main wings,
- sweep angle (β_{m_2}) of the second section of the main wings,
- relative span α_m of the first section of the main wings,
- root chord (RC_c) of the canards,
- tip chord (TC_{c_1}) of the first section of the canards,
- tip chord (TC_{c_2}) of the second section of the canards,
- sweep angle (β_{c_1}) of the first section of the canards,
- sweep angle (β_{c_2}) of the second section of the canards,
- relative span α_c of the first section of the canards.

Table 6.11 Performance of the different multi-fidelity models on the aerodynamic problem (Section 6.2.4) using 20 repetitions with different LHS generated DoE. Three scenarios on the available HF information are experimented (10, 15, and 20 input data on the HF). 120 training data points are used for the LF and 250 test data points to compute the metrics in the HF space.

Aerodynamic problem				
HF DoE size	**Algorithms**	R2 (std)	RMSE (std)	MNLL (std)
10 data points	HF model	0.0856 (0.5964)	0.0475 (0.0154)	33.92 (34.519)
	BC model	0.7284 (**0.2160**)	0.0258 (**0.0085**)	23.232 (25.911)
	MF-DGP-EM	**0.7646** (0.2796)	**0.0230** (0.0105)	**-2.0030** (**0.7456**)
15 data points	HF model	0.6358 (0.2400)	0.0300 (0.0098)	4.227 (8.0840)
	BC model	0.7522 (0.1932)	0.0248 (0.0079)	6.062 (5.6046)
	MF-DGP-EM	**0.8498 (0.1386)**	**0.0189 (0.0071)**	**-2.3094 (0.645)**
20 data points	HF model	0.8349 (**0.0769**)	0.0207 (**0.0043**)	-0.2532 (1.6970)
	BC model	0.8000 (0.1562)	0.0222 (0.0072)	1.7964 (2.3597)
	MF-DGP-EM	**0.8685** (0.1277)	**0.01764** (0.0069)	**-2.144** (**0.9646**)

Fig. 6.24 illustrates the two fidelity configurations. The input-space of the HF is 12-dimensional. Moreover, the mesh is refined in the HF with a doubled number of tessellated curves compared to the LF. This makes the computation of CL in the HF case numerically costly but more accurate than the LF configuration. This restrains the number of evaluations of the HF model, which makes multi-fidelity approaches interesting to enrich the HF with LF information. HF and LF models have different input-space dimensions (6 for the LF and 12 for the HF). MF-DGP-EM is used and compared to the BC approach and to a GP using only the HF information (GP HF). A possible nominal mapping between the input-spaces of the HF and LF is a mapping that for a set of HF design variables maps a LF design variables with the same canards and main wings surface:

$$\begin{aligned}
&RC_{bf} = RC_{hf}\\
&TC_{bf} = TC_{hf_1} + (1-\alpha_{hf})TC_{hf_2} + (\alpha_{hf}-1)RC_{hf} \qquad (6.32)\\
&\beta_{bf} = \alpha_{hf}\beta_{hf_1} + (1-\alpha_{hf})\beta_{hf_2}
\end{aligned}$$

The performance of the models is assessed on different scenarios of the HF information available. In fact, three different sizes of the HF DoE are experimented (10, 15, and 20 data points) and to evaluate the robustness with respect to the distribution of the HF data points in the input-space, the numerical experiments have been repeated on 10 different LHS for each size of the DoE. For all the scenarios the number of LF training data points is fixed to 120 training data points.

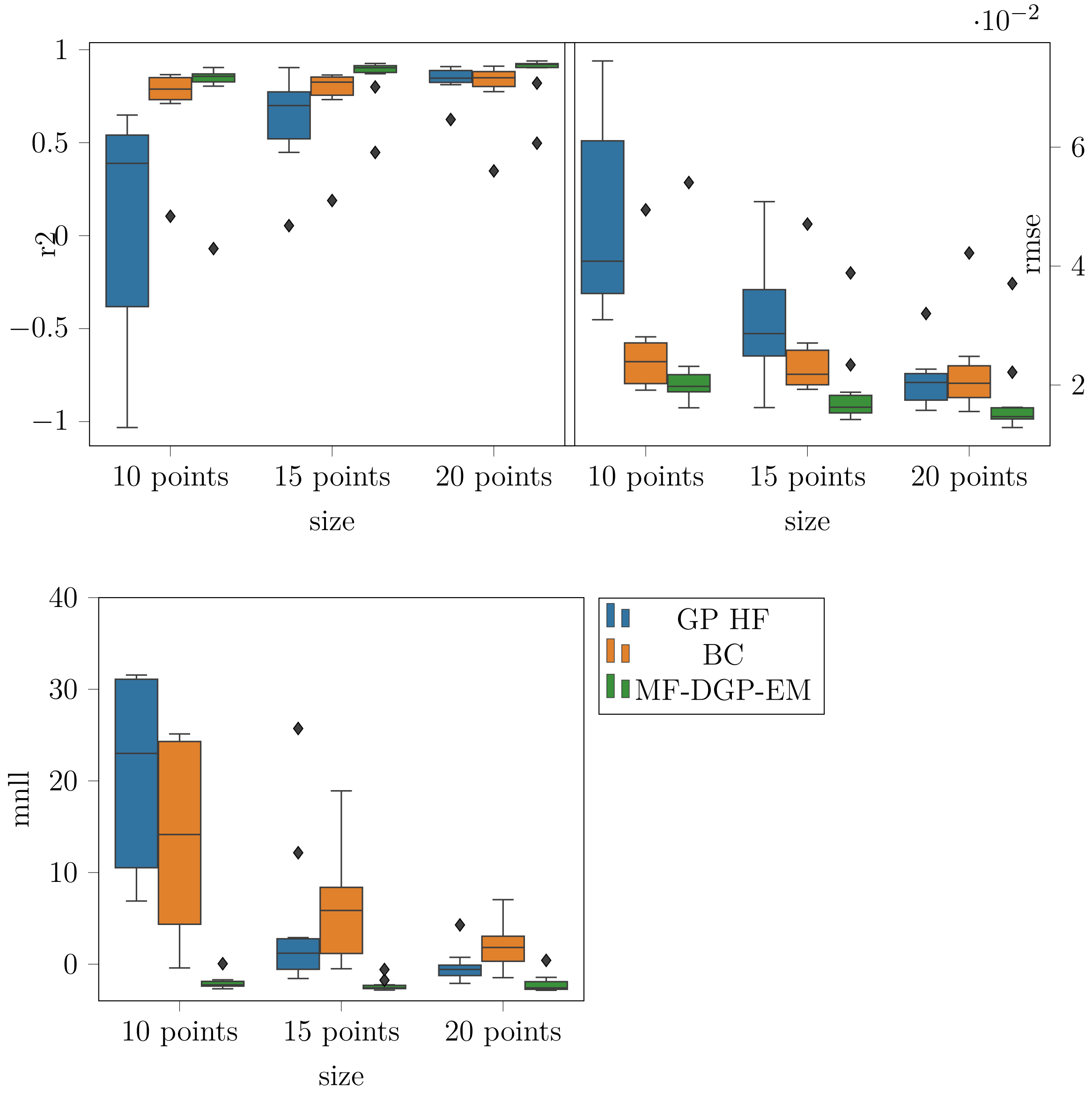


Fig. 6.25 Performance of the different models (GP HF for the high-fidelity GP, BC for the Bias correction approach, and MF-DGP-EM for the proposed model) on the aerodynamic test problem with 3 sizes of DoE (4, 6, and 8 data points on the HF and 30 input data on the LF) using 20 LHS repetitions.

The obtained results are presented in Table 6.11 and illustrated in Fig. 6.25. MF-DGP-EM presents a better prediction accuracy and uncertainty quantification even with only 10 data points in the HF dimension (RMSE: 0.023, MNLL: −2.00) compared to the GP HF (RMSE: 0.0475, MNLL: 34.) and the BC model (RMSE: 0.0258, MNLL: 23). Increasing the size of the HF training data allows the nominal mapping to be better learned in the case of MF-DGP-EM which enables a more significant difference between the MF-DGP-EM and the BC model in terms of prediction accuracy in the case of 15 and 20 HF training data points (MF-DGP-EM RMSE for 15 HF data points: 0.0189 and for 20 HF data points 0.01764; BC RMSE for 15 HF data points: 0.0248 and for 20 HF data points 0.0222). Some conclusions from the other experiments are also confirmed in this problem. For instance, the BC approach obtains a less accurate uncertainty quantification than the other approaches (MNLL for 15 HF data points 6.06 and for 20 data points 1.79) and its prediction accuracy stagnates after exceeding a threshold in the size of the training HF data (BC RMSE for 20 data points: 0.0222, GP HF RMSE for 20 data points: 0.0207). Also, the uncertainty quantification of the MF-DGP-EM is better compared to the other approaches even when the HF available information is not enough (10 data points).

6.2.5 Synthesis of the numerical experiments

These different results show the interest of using MF-DGP-EM, especially when the nominal mapping is not known for all the input space but only for the training HF data. It presents a prediction accuracy with robustness to the DoE that spares an excessive number of evaluations of the HF. Also, in the different problems, the uncertainty associated to the prediction of MF-DGP-EM is better valued than the other approaches even in the case when the HF information is scarce. This can be explained by the uncertainty quantification on the nominal mapping of MF-DGP-EM. This makes the MF-DGP-EM more interesting to use in applications where there is a trade-off between exploitation and exploration to be made such as optimization or design of experiments applications.

6.2.6 Computational aspects of MF-DGP-EM

The ELBO in eq. (6.17) is computed using Monte-Carlo sampling that propagates s samples throughout the MF-DGP-EM to compute the expectation terms. This makes the evaluation of the ELBO at each iteration computationally over-whelming compared to GP-HF and BC that are trained by optimizing analytical expressions. In fact, the computational complexity of MF-DGP-EM is $\mathcal{O}\left(s \times \left(n_1^3 + \sum_{t=2}^{n_{\mathrm{fi}}} \left(d_t \times n_t^3\right)\right)\right)$ (for a number of induced variables at each layer equal to the number of observations at the corresponding fidelity level), while GP-HF and BC are of computational complexity $\mathcal{O}n_{\mathrm{fi}}^3$. Moreover, the parameter space of MF-DGP-EM is highly-dimensional compared to GP-HF and BC. In fact, while the parameter space of GP-HF and BC contains only the kernel hyper-parameters of size $d_{n_{\text{fi}}} + 2$ (length-scale for each dimension in the case of ARD kernels, likelihood variance, and kernel variance), the parameter space of MF-DGP-EM contains additionally to the kernel hyper-parameters of the GP fidelities of size $\sum_{t=1}^{n_{\text{fi}}}(d_t + 2)$, the kernel hyper-parameters of the GP input-mappings of size $\sum_{t=2}^{n_{\text{fi}}}(d_t + 2)$, the induced inputs of the fidelity GPs of size $\sum_{t=1}^{n_{\text{fi}}} n_t \times d_t$ and of the input-mapping GPs of size $\sum_{t=2}^{n_{\text{fi}}} n_t \times d_t$, as well as the variational parameters of the GP fidelities of size $\sum_{t=1}^{n_{\text{fi}}}(n_t + n_t(n_t + 1)/2)$ and of the input-mapping GPs of size $\sum_{t=2}^{n_{\text{fi}}} d_t \times (n_t + n_t(n_t + 1)/2)$ (multi-output aspect). Table 6.12 illustrates the number of parameters of each approach in the case of a low-dimensional problem (Problem 1, Section 6.2.4) and a high-dimensional one (the aerodynamic problem, Section 6.2.4)) and the computational time needed for training the compared. While it takes less than 4 seconds to train GP-HF and BC using gradient descent in the different cases, it takes up to 580 seconds to train MF-DGP-EM using stochastic gradient (see Appendix C for details on the numerical setup) for the aerodynamic problem with 20 HF training data. This shows the heavy computational aspect of MF-DGP-EM compared to GP-HF and BC. Therefore, the interest of MF-DGP-EM is for cases where the high-fidelity is computationally intensive with evaluations that might take several hours or even days (complex computationally fluid dynamics or finite element analysis calculations for instance). Compared to alternative approaches, the computational cost overhead is large but the proposed approach offers modelling possibilities that cannot be reached by existing techniques (due to the nested training of the mapping and the multi-fidelity model).

Table 6.12 Number of parameters and time needed for training of each model for Problem 1 with 30 LF two-dimensional input data and three scenarios on the available HF information (4, 6, and 8 input data for , 4-dimensional HF) and the aerodynamic problem with 120 LF 6-dimensional input data and three scenarios on the available HF information (10, 15, and 20 input data for a 12-dimensional the HF). Details on the optimization setup for the algorithms are presented in Appendix D. The computational time is on a Tesla P100 GPU and using Tensorflow.

Analytical Problem 1			
HF DoE size	**Algorithms**	Number of parameters	Time for training (seconds)
4 data points	HF model	6	$\approx 0.5s$
	BC model	6	$\approx 0.5s$
	MF-DGP-EM	673	$\approx 232s$
6 data points	HF model	6	$\approx 0.55s$
	BC model	6	$\approx 0.55s$
	MF-DGP-EM	754	$\approx 245s$
8 data points	HF model	6	$\approx 0.62s$
	BC model	6	$\approx 0.62s$
	MF-DGP-EM	855	$\approx 280s$
Aerodynamic Problem			
HF DoE size	**Algorithms**	Number of parameters	Time for training (seconds)
10 data points	HF model	14	$\approx 1.8s$
	BC model	14	$\approx 1.8s$
	MF-DGP-EM	9221	$\approx 500s$
15 data points	HF model	14	$\approx 0.55s$
	BC model	14	$\approx 0.55s$
	MF-DGP-EM	10251	$\approx 530s$
20 data points	HF model	14	$\approx 3.4s$
	BC model	14	$\approx 3.4s$
	MF-DGP-EM	11606	$\approx 580s$

6.3 Conclusion

This chapter addressed multi-fidelity analysis using Gaussian processes with a two-fold objective, to improve a state-of-the-art approach that is MF-DGP for identically defined fidelity input spaces and also to propose a novel method for different input domain definitions.

The first contribution consisted of a training approach for MF-DGP to optimize the inducing inputs. This allowed to overcome the previous limitation of MF-DGP and increased its power of representation. Experiments on analytical problems and on an aerospace multi-fidelity benchmark have demonstrated the improvements of its prediction accuracy, uncertainty quantification and robustness to DoE. Besides dominating regular MF-DGP, the improved MF-DGP usually provides the best results among the compared approaches. In the case of insufficient HF data, improved MF-DGP still gives competitive results unlike regular MF-DGP which tends to provide limited accuracy results in these cases.

The second contribution considered the case of multi-fidelity problems with varying input-space parameterization. The different definitions of input spaces in multi-fidelity is common in physical and industrial applications. However, they are often addressed with models not specific to the problematic and appropriate models are scarce in the literature. In this chapter, a new model for this multi-fidelity problem is developed. The proposed model embeds into the existing MF-DGP, a mapping between the input-spaces using multi-output Gaussian processes. The proposed model allows a joint optimization of the input-space mapping and the multi-fidelity model, keeping the correlations in the original high-fidelity input-space, and allowing an uncertainty quantification of the input-space mapping. The efficiency of the proposed model has been assessed on analytical test problems and also on physical test problems. MF-DGP-EM outperforms the compared approaches in terms of prediction accuracy, uncertainty quantification, and robustness to the DoE in the majority of the problems and the scenarios of availability of HF data considered. Moreover, MF-DGP-EM is applicable in cases where the nominal mapping is not known for all the input-space but only for the training HF data.

The proposed model has been applied only in the case of two fidelities. However, it can be applied to more fidelities. Hence, experiments for three different fidelities with different input parameterizations may be interesting to assess the behavior of the model in more complicated configurations but may induce a computational burden during the training of the model.

The context of multi-fidelity modeling for the analysis of complex systems has been considered in this study. The natural next extension of this work is to address the multi-fidelity optimization topic with varying input-space dimensions. In this perspective, this model can be coupled to Bayesian optimization algorithms or to space mapping multi-fidelity optimization approaches.

Chapter 7

Conclusions and perspectives

7.1 Conclusions

This thesis proposed approaches to overcome three main issues related to Gaussian Processes (GPs) in the analysis and optimization of complex systems that are Bayesian optimization for non-stationary problems, multi-objective Bayesian Optimization (BO) taking into account correlations between objective functions, and multi-fidelity analysis with different input domain definitions. This has been accomplished through the hierarchical generalization of Gaussian processes, Deep Gaussian Processes (DGPs). In each proposed approach, the layers of DGPs have been used in a particular way as summarized in the following.

7.1.1 Contributions on Bayesian optimization for non-stationary problems

To address the issue of Bayesian optimization for non-stationary problems, the proposed approach is to use DGPs within a BO framework. In this case, DGPs are used classically *i.e.* the observed outputs are considered as a functional composition of GPs (Fig. 7.1). The intermediate layers play the role of Bayesian non-parametric mappings of the input space. This allows to stretch the input space in order to better capture the non-stationarity of the response. For this coupling of BO with DGPs, a specific framework has been proposed. In addition to adapting the training of DGPs to the iterative structure of BO, this framework includes a training approach based on natural gradient to obtain a better predictive uncertainty which is crucial in BO. Moreover, given the non-Gaussianity of the predictive distribution of DGPs, sampling is proposed for infill criteria such as the Expected Improvement and the Probability of Feasibility. The

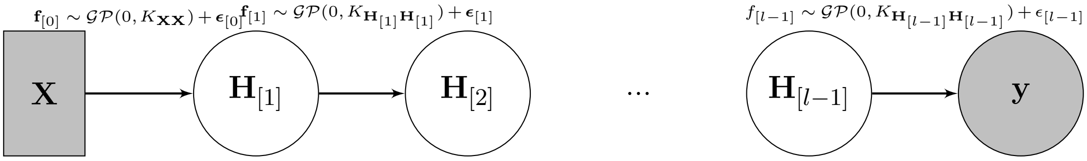

$\mathbf{f}_{[0]} \sim \mathcal{GP}(0, K_{\mathbf{XX}}) + \epsilon_{[0]}$
$\mathbf{f}_{[1]} \sim \mathcal{GP}(0, K_{\mathbf{H}_{[1]}\mathbf{H}_{[1]}}) + \epsilon_{[1]}$
$f_{[l-1]} \sim \mathcal{GP}(0, K_{\mathbf{H}_{[l-1]}\mathbf{H}_{[l-1]}}) + \epsilon_{[l-1]}$
$\mathbf{X}$
$\mathbf{H}_{[1]}$
$\mathbf{H}_{[2]}$
$\cdots$
$\mathbf{H}_{[l-1]}$
$\mathbf{y}$

Fig. 7.1 Classic DGP with hidden layers as Bayesian non-parametric mappings of the input space to handle non-stationarity.

architecture of DGPs in the context of BO has also been investigated through numerical experimentations. The proposed framework of BO with DGPs has been compared to state-of-the-art non-linear-mapping approaches for GPs to handle non-stationarity on analytical and aerospace design problems. The framework of BO with DGPs achieves the best results in terms of the final optimum obtained, the speed of convergence, and the robustness to the Design of Experiments (DoE).

7.1.2 Contributions on multi-objective Bayesian optimization with correlated objectives

To take into account a potential correlation between the objectives in a multi-objective BO algorithm, a novel model called Multi-Objective Deep Gaussian Process (MO-DGP) has been developed. In this model, each layer of the DGP is conditioned on the observed values of an objective. Therefore, each layer may be interpreted as an objective model. Moreover, the different layers constitute a clique within the DGP and are connected with undirected edges (Fig. 7.2). This allows interactions between the different layers without a specific hierarchy. However, this comes with a challenge in the inference where the expectation term of the evidence lower bound is computed with respect to the joint variational distribution of the layers. To compute this term, a Gibbs sampling approach is proposed. In addition to the model, a computational approach for the EHVI based on kernel density estimation is proposed without the assumption of independence between the objectives and of the Gaussianity of the predictive distribution. The efficiency of the developed model and the computational approach for the EHVI in a multi-objective BO algorithm is assessed on analytical and aerospace design multi-objective optimization problems with respect to multi-objective BO with independent GPs and multi-objective BO with the linear model of coregionalization. By taking advantage of the correlations between objectives, MO-DGP is able to improve multi-objective BO in terms of the final hyper-volume obtained, the speed of convergence, and the robustness to the initial DoE for multi-objective problems with correlated objectives. However, it has been demonstrated that the correlated EHVI is not decisive in the

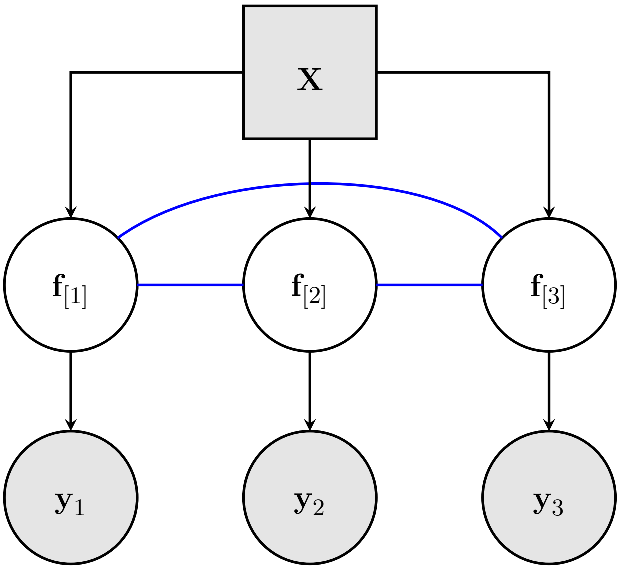


Fig. 7.2 Multi-Objective Deep Gaussian Process model. Each layer correspond to an objective and constitutes a clique with undirected edges.

improvement of the multi-objective BO algorithms. This is explained by the fact that the full predictive distribution needs more data to be well approximated.

7.1.3 Contributions on multi-fidelity analysis

For multi-fidelity analysis, firstly the case of identically defined input spaces for the different fidelities has been considered. The goal was to improve the existing model Multi-Fidelity Deep Gaussian Process (MF-DGP) in which the inducing inputs were set to arbitrary values during the optimization. This has been accomplished by proposing a training approach for the inducing inputs. This method takes into account the augmented input space in which the inducing inputs lie. For that, the last dimension of the inducing inputs is not considered in the optimization. In fact, it is inferred by propagating the first dimensions, which are freely optimized, through the previous layers. This improvement of MF-DGP is assessed with respect to regular MF-DGP and Gaussian-based multi-fidelity approaches on a benchmark of analytical and aerospace design problems in different scenarios of data availability. The improved MF-DGP provides the best results in terms of prediction accuracy, uncertainty quantification, and robustness to the data-set. For the second part of the contributions on multi-fidelity analysis, the case of a different input-space domain definition for each fidelity is considered. For this, a novel model is developed. This model is a two-level DGP, in which the first level maps between the different input spaces and the second level

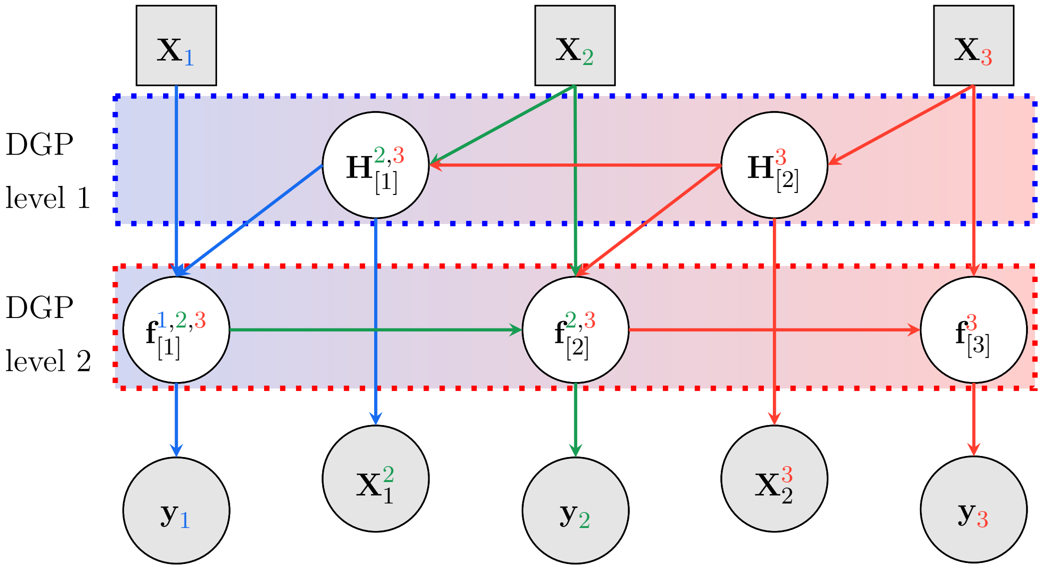


Fig. 7.3 Multi-Fidelity Deep Gaussian Process Embedded Mapping model. Two DGP levels are used where the first level maps between the input spaces of the different fidelities and the second level propagates the input through the fidelity layers.

propagates the input through the different fidelity levels (Fig 7.3). This enables a joint optimization of the input mappings and the multi-fidelity model, thus, the name of the model Multi-Fidelity Deep Gaussian Process Embedded Mapping (MF-DGP-EM). Moreover, using GPs as input mappings allows a non-parametric and Bayesian mapping which is more flexible than classical parametric mapping optimized disjointly from the multi-fidelity model. This proposed model is assessed on analytical and engineering design problems in different scenarios of data availability with respect to classic approaches. MF-DGP-EM shows promising results in terms of prediction accuracy and uncertainty quantification especially in the case of scarce data.

7.2 Perspectives

Different improvements and extensions of the approaches developed in this thesis can be identified.

7.2.1 Improvements and extensions of the framework BO & DGPs

As discussed in Chapter 4, one of the crucial aspects of the integration of DGP in BO is the predictive uncertainty. In the proposed framework, using natural gradient proves to improve the obtained predictive uncertainty quantification. However, using variational inference in the inference of DGP may still yield to an under-estimation of the uncertainty. To overcome this issue, recently some inference approaches for DGP have emerged using Hamiltonian Monte-Carlo [Havasi et al., 2018] and also using implicit posterior variational inference [Haibin et al., 2019] that may obtain a better-calibrated inference for DGPs.

One of the limits identified in the natural gradient optimization of the DGP variational parameters is the ill-conditioning of the Fisher information matrices of the inner layers. In the proposed framework, it has been addressed by small steps in the optimization procedure. However, more sophisticated approaches may be used based on approximate Fisher information methods [Ly et al., 2017] or classic conditioning techniques used for kernel machines [Cutajar et al., 2016].

The framework BO-DGP is used in the case of real valued design variables. However, the design of complex systems may include discrete technological and architectural choices. BO with GPs has been used in the case of categorical variables through formulation of discrete kernels expressed as the product between one-dimensional kernels [Pelamatti et al., 2019]. An extension of BO-DGP to non-stationary mixed variable optimization problems can be developed by using discrete kernels at each layer of the DGP. The optimization of the infill criteria has also to be adapted to the mixed variable design space.

7.2.2 Improvements and extensions of the proposed algorithm for MO-BO with correlated objectives

The numerical experiments obtained in Chapter 5 proved that multi-objective BO with MO-DGP outperforms the compared algorithms when there is correlation between the objectives over all the objective space. However, in the case when the correlation is only around the approximated Pareto front (Kursawe problem), MO-DGP gives comparable results to independent GPs. One way to overcome this issue would be to take into account only the correlations between the approximated Pareto front at each iteration when training MO-DGP. Therefore, the structure of the model has to be changed since the edges between the layers are active only for a subset of training points.

Another limit identified in the numerical experiments is that the correlated EHVI is not decisive for the improvement of multi-objective BO. This may be explained by the fact that the full predictive distribution given by the model needs more data-points to be well approximated. Numerical experiments with initial DoE with more data-points can be used to identify the improvement given by the correlated EHVI in this case.

Since a wide range of problems in the design of complex systems can be formulated as two-objective problems, in this thesis, only this case has been considered. While MO-DGP can be directly used for more than two objectives, the expression of the EHVI considered throughout this thesis is valid only for the two objective case. However, methods to compute the EHVI for more objectives have been developed in the literature [Hupkens et al., 2015]. Due to the computational complexity of the inference of MO-DGP, it is actually challenging to apply it in a many-objective context. A simplification of the model may be considered in this case. For instance, to use a configuration where each layer i is connected to only $i-1$ and $i+1$ using undirected edges which yields to a circular graph. The transfer of information from layer j to i is done through the GP propagation throughout the different layers.

MO-DGP was used for the objective functions in multi-objective optimization. However, in optimization, there may also be inter-correlations between the constraints. For instance, in the design of an aerospace launch vehicle, the constraint on the minimum payload carried by the vehicle and the constraint on the minimum attitude to reach are negatively correlated. In fact, if one of the two constraints is feasible and largely non-saturated, there is a high probability that the other constraint is not feasible. Therefore, MO-DGP can be used in this context to take into account this correlation between the different functions involved (either objective functions or constraints) in a single or multi-objective problem.

To handle non-stationarity in a multi-objective context, MO-DGP may be coupled with regular DGPs that stretch the input space using hidden layers as achieved in Chapter 4. For that, instead of considering for each objective a GP layer, a DGP is used. The connection between the different objectives is done according to the last layer of each DGP objective. Therefore, while the hidden layers handle non-stationarity, the last layers encode the correlation between the objectives.

MO-DGP model may be extended to other problems involving correlated functions with an unknown hierarchy between them. In fact, it can be used as a multi-task model for the analysis of complex systems. For instance in reliability analysis, to estimate the failure probability of different failure modes, MO-DGP may be used where each layer corresponds to a failure mode, therefore, taking into account the dependencies between each pair of failure modes.

7.2.3 Improvements and extensions for multi-fidelity analysis

The first level of the developed model MO-DGP-EM is conditioned on nominal mapping values between the input spaces. However, for some computationally expensive legacy codes, these nominal mapping values may be not possible to obtain. A first-level non-conditioned in MO-DGP-EM would be difficult to train using only the conditioning on the second level. Another configuration of the model or another inference approach is necessary to avoid that the posterior distribution of the first level GPs collapses to its mean function.

For now, the model MO-DGP-EM has been applied only to multi-fidelity problems with two fidelities. The configuration of the model still holds in the case of multiple fidelities. Therefore, it would be interesting to confirm its efficiency for three different fidelities with different input space parameterizations and to evaluate the computational burden induced by increasing the number of fidelities.

In the presented work on multi-fidelity analysis, only the modeling aspects have been investigated. The next extension would be to use MO-DGP-EM within a BO algorithm for optimization purposes in varying input-space dimensions. This may present some challenges for the optimization of the infill criterion. In fact, unlike classical BO with multi-fidelity where the criterion is evaluated for different fidelities but within the same design space, in this case for each fidelity the infill criterion lies in its own input space. Therefore, a mapping may be needed within the optimization process.

The considered multi-fidelity organization relies on a hierarchic decomposition in which each fidelity level corresponds to a physical model. However, in some cases, it may be more nuanced. For instance, there might be different physical models at the same level of fidelity and each physical model may be more adapted in a specific region of the design space (e.g., aerodynamic models dedicated to subsonic or hypersonic regimes). For instance, different physical models at the low-fidelity each adapted to its own design space region and one high-fidelity model may be considered. To address this multi-fidelity formulation, a three-layer DGP may be considered. The first layer would be a multiple-unit layer where each unit is conditioned on the observations of a specific low-fidelity physical model. The relevance of each output of the first layer depends on a specific region of the input space, inducing non-stationarity behavior. Therefore, the second layer is a hidden layer that plays the role of a non-linear mapping of the input space as in Chapter 4. The last layer is a one-unit layer augmented with the outputs of the previous layer and conditioned on the high-fidelity observations.

7.2.4 Extensions of deep Gaussian processes to other problems in the design of complex systems

In this thesis, DGPs have been applied through different methods to BO for non-stationary problems, multi-objective BO with correlated objectives, and multi-fidelity analysis. Other problems that occur in the design of complex systems might take advantage of the deep and Bayesian structure of DGPs. For instance, for reliability analysis, MO-DGP in which each layer corresponds to a failure mode might be used. Moreover, the predictive distribution of DGPs proves to be better calibrated than GPs for complex models making it interesting to use for sensitivity analysis and uncertainty quantification problems. In fact, the DGP predictive distribution can be propagated to the sensitivity index estimates. Therefore, using DGPs for design strategies adapted to sensitivity analysis may be interesting.

A computational limitation of DGPs occurs in the handling of high-dimensional problems, due to the necessity to increase the size of the DoE, which results in a more complex inference. In this thesis, a constrained optimization problem with 15 design variables remains the problem with the highest dimension considered. However, in the design of complex systems,problems with higher dimensions may be considered. Therefore, the size of the DoE for analysis and the number of added points in a BO context are larger. To overcome this issue, it would be interesting to couple DGPs with dimension reduction approaches such as partial least squares for GPs [Bouhlel et al., 2016].

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

Ankit Agrawal, Parijat D Deshpande, Ahmet Cecen, Gautham P Basavarsu, Alok N Choudhary, and Surya R Kalidindi. Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. *Integrating Materials and Manufacturing Innovation*, 3(1):8, 2014.

Devanshu Agrawal, Theodore Papamarkou, and Jacob Hinkle. Wide neural networks with bottlenecks are deep Gaussian processes. *arXiv preprint arXiv:2001.00921*, 2020.

Ahmed M Alaa and Mihaela van der Schaar. Deep multi-task Gaussian processes for survival analysis with competing risks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2326–2334, 2017.

Greg M Allenby, Peter E Rossi, and Robert E McCulloch. Hierarchical Bayes models: A practitioners guide. ssrn scholarly paper id 655541. *Social Science Research Network, Rochester, NY*, 2005.

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

Mauricio A Alvarez, Lorenzo Rosasco, and Neil D Lawrence. Kernels for vector-valued functions: A review. *arXiv preprint arXiv:1106.6251*, 2011.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

John D Anderson. *Fundamentals of Aerodynamics*. McGraw–Hill, 1991.

John D Anderson and John F Wendt. *Computational fluid dynamics*, volume 206. Springer, 1995.

Mark J Anderson and Patrick J Whitcomb. Design of experiments. *Kirk-Othmer Encyclopedia of Chemical Technology*, pages 1–22, 2000.

Ken Anjyo and John P Lewis. RBF interpolation and Gaussian process regression through an RKHS formulation. *Journal of Math-for-Industry*, 3(6):63–71, 2011.

Francesco Archetti and Antonio Candelieri. The surrogate model. In *Bayesian Optimization and Data Science*, pages 37–56. Springer, 2019.

Alfredo Arias-Montano, Carlos A Coello Coello, and Efrén Mezura-Montes. Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Transactions on Evolutionary Computation*, 16(5):662–694, 2012.

Yves F Atchadé. An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. *Methodology and Computing in applied Probability*, 8(2): 235–254, 2006.

Peter M Atkinson and Christopher D Lloyd. Non-stationary variogram models for geostatistical sampling optimisation: An empirical investigation using elevation data. *Computers & Geosciences*, 33(10):1285–1300, 2007.

Charles Audet, John Denni, Douglas Moore, Andrew Booker, and Paul Frank. A surrogate-model-based method for constrained optimization. In *8th Symposium on Multidisciplinary Analysis and Optimization*, page 4891, 2000.

Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75–103, 2012.

Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76, 2011.

Joëlle Bailly and Didier Bailly. Multifidelity aerodynamic optimization of a helicopter rotor blade. *AIAA Journal*, 57(8):3132–3144, 2019.

M. Balesdent, N. Bérend, and P. Dépincé. Stagewise multidisciplinary design optimization formulation for optimal design of expendable launch vehicles. *Journal of Spacecraft and Rockets*, 49:720–730, 2012a.

Mathieu Balesdent, Nicolas Bérend, Philippe Dépincé, and Abdelhamid Chriette. A survey of multidisciplinary design optimization methods in launch vehicle design. *Structural and Multidisciplinary optimization*, 45(5):619–642, 2012b.

John W Bandler, Qingsha S Cheng, Sameh A Dakroury, Ahmed S Mohamed, Mohamed H Bakr, Kaj Madsen, and Jacob Sondergaard. Space mapping: the state of the art. *IEEE Transactions on Microwave theory and techniques*, 52(1):337–361, 2004.

John W Bandler, Slawomir Koziel, and Kaj Madsen. Space mapping for engineering optimization. *SIAG/Optimization Views-and-News Special Issue on Surrogate/Derivative-free Optimization*, 17(1):19–26, 2006.

Nathalie Bartoli, Thierry Lefebvre, Sylvain Dubreuil, Romain Olivanti, Nicolas Bons, Joaquim Martins, Mohamed A Bouhlel, and Joseph Morlier. An adaptive optimization strategy based on mixture of experts for wing aerodynamic design optimization. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4433, 2017.

Nathalie Bartoli, Thierry Lefebvre, Sylvain Dubreuil, Romain Olivanti, Rémy Priem, Nicolas Bons, Joaquim RRA Martins, and Joseph Morlier. Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerospace Science and Technology*, 90:85–102, 2019.

Imad A Basheer and Maha Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1):3–31, 2000.

Kinjal Basu and Souvik Ghosh. Analysis of thompson sampling for Gaussian process optimization in the bandit setting. *arXiv preprint arXiv:1705.06808*, 2017.

Oliver A Bauchau and James I Craig. Euler-Bernoulli beam theory. In *Structural analysis*, pages 173–221. Springer, 2009.

Adrian Bekasiewicz and Slawomir Koziel. Efficient multi-fidelity design optimization of microwave filters using adjoint sensitivity. *International Journal of Radio Frequency and Microwave Computer-Aided Engineering*, 25(2):178–183, 2015.

Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

James O. Berger, Robert L. Wolpert, M. J. Bayarri, M. H. DeGroot, Bruce M. Hill, David A. Lane, and Lucien LeCam. The likelihood principle. *Lecture Notes-Monograph Series*, 6:iii–199, 1988.

Dimitri Bettebghor, Nathalie Bartoli, Stéphane Grihon, Joseph Morlier, and Manuel Samuelides. Surrogate modeling approximation using a mixture of experts based on em joint estimation. *Structural and multidisciplinary optimization*, 43(2):243–259, 2011.

Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

Christopher M Bishop. *Pattern recognition and machine learning.* Springer, 2006.

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

Kenneth Blomqvist, Samuel Kaski, and Markus Heinonen. Deep convolutional Gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 582–597. Springer, 2019.

Frederic E Bock, Roland C Aydin, Christian J Cyron, Norbert Huber, Surya R Kalidindi, and Benjamin Klusemann. A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials*, 6:110, 2019.

Mohamed A Bouhlel, Nathalie Bartoli, Abdelkader Otsmane, and Joseph Morlier. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53(5): 935–952, 2016.

Mohamed A Bouhlel, Nathalie Bartoli, Rommel G Regis, Abdelkader Otsmane, and Joseph Morlier. Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. *Engineering Optimization*, 50(12):2038–2053, 2018.

Lucas Bradstreet. *The hypervolume indicator for multi-objective optimisation: calculation and use.* PhD thesis, University of Western Australia Perth, 2011.

Loic Brevault, Mathieu Balesdent, and Ali Hebbal. Multi-objective multidisciplinary design optimization approach for partially reusable launch vehicle design. *Journal of Spacecraft and Rockets*, pages 1–17, 2020a.

Loïc Brevault, Mathieu Balesdent, and Jérôme Morio. Aerospace system analysis and optimization in uncertainty. *Springer Optimization and Its Applications*, 2020b.

Christopher J Brooks, Alexander I.J Forrester, Andy J Keane, and Shahrokh Shahpar. Multi-fidelity design optimisation of a transonic compressor rotor. In *9th European Conf. Turbomachinery Fluid Dynamics and Thermodynamics*, March 2011.

Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press, 2019.

Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep Gaussian processes for regression using approximate expectation propagation. In *International conference on machine learning*, pages 1472–1481, 2016.

Claudio Carmeli, Ernesto De Vito, and Alessandro Toigo. Reproducing kernel Hilbert spaces and mercer theorem. *arXiv preprint math/0504071*, 2005.

Francesco Castellini, Annalisa Riccardi, Michèle Lavagna, and Christof Büskens. Global and local multidisciplinary design optimization of expendable launch vehicles. In *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2011.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International conference on machine learning*, pages 1683–1691, 2014.

Clément Chevalier, Julien Bect, David Ginsbourger, Emmanuel Vazquez, Victor Picheny, and Yann Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56 (4):455–465, 2014.

Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

Michael E Cholette, Pietro Borghesani, Egidio Di Gialleonardo, and Francesco Braghin. Using support vector machines for the computationally efficient identification of acceptable design parameters in computer-aided engineering applications. *Expert Systems with Applications*, 81:39–52, 2017.

Andreas Christmann and Ingo Steinwart. Consistency of kernel-based quantile regression. *Applied Stochastic Models in Business and Industry*, 24(2):171–183, 2008.

Ian Cordery and S L.Yao. Non stationarity of phenomena related to drought. *Extreme hydrological events. Proc. international symposium, Yokohama, 1993*, 01 1993.

Dennis D Cox and Susan John. SDO: A statistical method for global optimization. In *in Multidisciplinary Design Optimization: State-of-the-Art*, pages 315–329, 1997.

Kurt Cutajar, Michael Osborne, John Cunningham, and Maurizio Filippone. Preconditioning kernel matrices. In *International Conference on Machine Learning*, pages 2529–2538, 2016.

Kurt Cutajar, Edwin V Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep Gaussian processes. In *International Conference on Machine Learning*, pages 884–893. PMLR, 2017.

Kurt Cutajar, Mark Pullin, Andreas Damianou, Neil Lawrence, and Javier González. Deep Gaussian processes for multi-fidelity modeling. *arXiv preprint arXiv:1903.07320*, 2019.

Zhenwen Dai, Andreas Damianou, Javier González, and Neil Lawrence. Variational auto-encoded deep Gaussian processes. *arXiv preprint arXiv:1511.06455*, 2015.

Andreas Damianou. *Deep Gaussian processes and variational propagation of uncertainty.* PhD thesis, University of Sheffield, 2015.

Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

Nicolaas Govert De Bruijn. *Asymptotic methods in analysis*, volume 4. Courier Corporation, 1981.

Alexander G De G. Matthews, Mark Van Der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. Gpflow: A Gaussian process library using tensorflow. *The Journal of Machine Learning Research*, 18(1):1299–1304, 2017.

Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature*, pages 849–858. Springer, 2000.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer, 2005.

Guido Dhondt. Calculix crunchix user's manual version 2.12. 2017.

Dennis M Dimiduk, Elizabeth A Holm, and Stephen R Niezgoda. Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering. *Integrating Materials and Manufacturing Innovation*, 7(3):157–172, 2018.

Francesco Dinuzzo and Bernhard Schölkopf. The representer theorem for Hilbert spaces: a necessary and sufficient condition. In *Advances in neural information processing systems*, pages 189–196, 2012.

Rémi Domingues, Pietro Michiardi, Jihane Zouaoui, and Maurizio Filippone. Deep Gaussian process autoencoders for novelty detection. *Machine Learning*, 107(8-10): 1363–1383, 2018.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

Vincent Dubourg, Bruno Sudret, and Jean-Marc Bourinet. Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):673–690, 2011.

Vincent Dutordoir, Nicolas Knudde, Joachim van der Herten, Ivo Couckuyt, and Tom Dhaene. Deep Gaussian process metamodeling of sequentially sampled non-stationary response surfaces. In *2017 Winter Simulation Conference*, pages 1728–1739. IEEE, 2017.

Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse Gaussian processes with spherical harmonic features. *arXiv preprint arXiv:2006.16649*, 2020.

Michael Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of Pareto front approximations. *Rapport technique, Leiden University*, 34:7–3, 2008.

Michael Emmerich, Kyriakos C Giannakoglou, and Boris Naujoks. Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

Michael Emmerich, Kaifeng Yang, André Deutz, Hao Wang, and Carlos M Fonseca. A multicriteria generalization of Bayesian global optimization. In *Advances in Stochastic and Deterministic Global Optimization*, pages 229–242. Springer, 2016.

Robert D Falck and Justin S Gray. Optimal control within the context of multidisciplinary design, analysis, and optimization. In *AIAA Scitech 2019 Forum*, page 0976, 2019.

Guanchao Feng, J Gerald Quirk, and Petar M Djurić. Supervised and unsupervised learning of fetal heart rate tracings with deep Gaussian processes. In *14th Symposium on Neural Networks and Applications*, pages 1–6. IEEE, 2018.

Giselle M Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T Haftka. Review of multi-fidelity models. *arXiv preprint arXiv:1609.07196*, 2016.

Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1): 176–190, 2008.

Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide.* John Wiley & Sons, 2008.

Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

David Freedman et al. Wald lecture: On the Bernstein-von Mises theorem with infinite-dimensional parameters. *The Annals of Statistics*, 27(4):1119–1141, 1999.

Mark Fuge, Bud Peters, and Alice Agogino. Machine learning algorithms for recommending design methods. *Journal of Mechanical Design*, 136(10), 2014.

Charles Gadd, Sara Wade, and Alexis Boukouvalas. Enriched mixtures of generalised Gaussian process experts. In *International Conference on Artificial Intelligence and Statistics*, pages 3144–3154, 2020.

Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 1(3), 2016.

Sahil Garg, Amarjeet Singh, and Fabio Ramos. Learning non-stationary space-time models for environmental monitoring. In *Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto*, 2012.

Alan E Gelfand, Susan E Hills, Amy Racine-Poon, and Adrian FM Smith. Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association*, 85(412):972–985, 1990.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis.* CRC press, 2013.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-6(6):721–741, 1984.

Iain Murray Zoubin Ghahramani. A note on the evidence and bayesian occam's razor. Technical report, Gatsby Unit Technical Report GCNU-TR 2005–003, 2005.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

Roger Ghanem, David Higdon, and Houman Owhadi. *Handbook of uncertainty quantification*, volume 6. Springer, 2017.

Robert Gibbons. *A primer in game theory.* Prentice Hall Books, 1994.

Mark N Gibbs. *Bayesian Gaussian processes for regression and classification.* PhD thesis, University of Cambridge, 1998.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In *Computational intelligence in expensive optimization problems*, pages 131–162. Springer, 2010.

James Gloudemans, Paul Davis, and Paul Gelhausen. A rapid geometry modeler for conceptual aircraft. In *34th Aerospace Sciences Meeting and Exhibit*, page 52, 1996.

Paul W Goldberg, Christopher KI Williams, and Christopher M Bishop. Regression with input-dependent noise: A Gaussian process treatment. In *Advances in neural information processing systems*, pages 493–499, 1998.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

Robert B Gramacy and Daniel W Apley. Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24 (2):561–578, 2015.

Robert B Gramacy and Herbert K H Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

Justin S Gray, John T Hwang, Joaquim RRA Martins, Kenneth T Moore, and Bret A Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4): 1075–1104, 2019.

Timothy C Haas. Kriging and automated variogram modeling within a moving window. *Atmospheric Environment. Part A. General Topics*, 24(7):1759–1769, 1990.

David Hadka. Platypus-multiobjective optimization in python, 2015.

Yu Haibin, Yizhou Chen, Bryan K.H Low, Patrick Jaillet, and Zhongxiang Dai. Implicit posterior variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 14502–14513, 2019.

Keith W Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

Marton Havasi, José Miguel Hernández-Lobato, and Juan José Murillo-Fuentes. Inference in deep Gaussian processes using stochastic gradient Hamiltonian Monte Carlo. In *Advances in neural information processing systems*, pages 7506–7516, 2018.

Ali Hebbal, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Nouredine Melab. Multi-objective optimization using deep Gaussian processes: Application to aerospace vehicle design. In *AIAA Scitech 2019 Forum*, page 1973, 2019.

Ali Hebbal, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Nouredine Melab. Bayesian optimization using deep Gaussian processes with applications to aerospace system design. *Optimization and Engineering*, pages 1–41, 2020.

Ravi S Hegde. Accelerating optics design optimizations with deep learning. *Optical Engineering*, 58(6):065103, 2019.

Markus Heinonen, Henrik Mannerström, Juho Rousu, Samuel Kaski, and Harri Lähdesmäki. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. In *Artificial Intelligence and Statistics*, pages 732–740, 2016.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. MCMC for variationally sparse Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1648–1656, 2015.

James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *The Journal of Machine Learning Research*, 18(1):5537–5588, 2017.

Albert L Herman and Bruce A Conway. Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules. *Journal of Guidance, Control, and Dynamics*, 19(3):592–599, 1996.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

Dave Higdon, Jenise Swall, and John Kern. Non-stationary spatial modeling. *Bayesian statistics*, 6(1):761–768, 1999.

Arthur Hobson and Bin-Kang Cheng. A comparison of the Shannon and Kullback information measures. *Journal of Statistical Physics*, 7(4):301–310, 1973.

Matthew D Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for Bayesian optimization. In *27th Conference on Uncertainty in Artificial Intelligence*, pages 327–336, 2011.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, pages 1171–1220, 2008.

Iris Hupkens, André Deutz, Kaifeng Yang, and Michael Emmerich. Faster exact algorithms for computing expected hypervolume improvement. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 65–79. Springer, 2015.

Joseph G Ibrahim and Purushottam W Laud. On Bayesian analysis of generalized linear models using Jeffreys's prior. *Journal of the American Statistical Association*, 86(416):981–986, 1991.

Gordana Ivosev, Lyle Burton, and Ron Bonner. Dimensionality reduction and visualization in principal component analysis. *Analytical chemistry*, 80(13):4933–4944, 2008.

Praveen Iyappan and Ranjan Ganguli. Multi-fidelity analysis and uncertainty quantification of beam vibration using correction response surfaces. *International Journal for Computational Methods in Engineering Science and Mechanics*, 21(1):26–42, 2020.

Tommi S Jaakkola and Michael I Jordan. Variational probabilistic inference and the QMR-DT network. *Journal of artificial intelligence research*, 10:291–322, 1999.

Tahira Jamil and Cajo JF ter Braak. Selection properties of type II maximum likelihood (empirical Bayes) in linear models with individual variance components for predictors. *Pattern Recognition Letters*, 33(9):1205–1212, 2012.

Mitja Jančič, Juš Kocijan, and Boštjan Grašič. Identification of atmospheric variable using deep Gaussian processes. *1st IFAC Workshop on Integrated Assessment Modelling for Environmental Systems IAMES*, 2018.

John P. Jasa, John T. Hwang, and Joaquim R. R. A. Martins. Open-source coupled aerostructural optimization using Python. *Structural and Multidisciplinary Optimization*, 57(4):1815–1827, 2018.

Edwin T Jaynes. Prior probabilities. *IEEE Transactions on systems science and cybernetics*, 4(3):227–241, 1968.

Edwin T Jaynes. *Probability theory: The logic of science.* Cambridge university press, 2003.

William H Jefferys and James O Berger. Ockham's razor and Bayesian analysis. *American Scientist*, 80(1):64–72, 1992.

Ming Jin, Andreas Damianou, Pieter Abbeel, and Costas Spanos. Inverse reinforcement learning via deep Gaussian process. *arXiv preprint arXiv:1512.08065*, 2015.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

M Chris Jones, James S Marron, and Simon J Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical association*, 91 (433):401–407, 1996.

Ingi M Jonsson, Leifur Leifsson, Slawomir Koziel, Yonatan A Tesfahunegn, and Adrian Bekasiewicz. Shape optimization of trawl-doors using variable-fidelity models and space mapping. In *International Conference On Computational Science*, pages 905–913, 2015.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2): 183–233, 1999.

Melih Kandemir. Asymmetric transfer learning with deep Gaussian processes. In *International Conference on Machine Learning*, pages 730–738, 2015.

Andy J Keane. Cokriging for robust design optimization. *AIAA journal*, 50(11): 2351–2364, 2012.

Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

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

Hong Seok Kim, Muammer Koc, and Jun Ni. A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network. *International Journal of Machine Tools and Manufacture*, 47(2):211–222, 2007.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Jack PC Kleijnen. Regression and kriging metamodels with their experimental designs in simulation: a review. *European Journal of Operational Research*, 256(1):1–16, 2017.

Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

Sreenivas Konda. *Fitting Models of Nonstationary Time Series: An Application to EEG Data.* PhD thesis, Case Western Reserve University, 2006.

Tomoki Koriyama and Takao Kobayashi. Statistical parametric speech synthesis using deep Gaussian processes. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(5):948–959, 2019.

Slawomir Koziel. Computationally efficient multi-fidelity multi-grid design optimization of microwave structures. *ACES Journal-Applied Computational Electromagnetics Society*, 25(7):578, 2010.

Tipaluck Krityakierne and David Ginsbourger. Global optimization with sparse and local Gaussian process models. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 185–196. Springer, 2015.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Vinayak Kumar, Vaibhav Singh, PK Srijith, and Andreas Damianou. Deep Gaussian processes with convolutional kernels. *arXiv preprint arXiv:1806.01655*, 2018.

Frank Kursawe. A variant of evolution strategies for vector optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 193–197. Springer, 1990.

Yuichi Kuya, Kenji Takeda, Xin Zhang, and Alexander IJ Forrester. Multifidelity surrogate modeling of experimental and computational aerodynamic data sets. *AIAA journal*, 49(2):289–298, 2011.

Jaroslaw Sobieszczanski-Sobieski Langley, Jeremy S Agte, et al. Bi-level integrated system synthesis (BLISS). *NASA/TM-1998-208715*, 1998.

Julien Laurenceau and P Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA journal*, 46(2):498–507, 2008.

Miguel Lázaro-Gredilla and Anibal Figueiras-Vidal. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, pages 1087–1095, 2009.

Miguel Lázaro-Gredilla, Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.

Loic Le Gratiet and Josselin Garnier. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5), 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Jay H Lee, Joohyun Shin, and Matthew J Realff. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, 114:111–121, 2018.

Xian Yeow Lee, Aditya Balu, Daniel Stoecklein, Baskar Ganapathysubramanian, and Soumik Sarkar. A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting. *Journal of Mechanical Design*, 141(11), 2019.

Wei Li, Shishi Chen, Zhen Jiang, Daniel W Apley, Zhenzhou Lu, and Wei Chen. Integrating Bayesian calibration, bias correction, and machine learning for the 2014 Sandia verification and validation challenge problem. *Journal of Verification, Validation and Uncertainty Quantification*, 1(1), 2016.

Yang Li, Wanchun Chen, Hao Zhou, and Liang Yang. Conjugate gradient method with pseudospectral collocation scheme for optimal rocket landing guidance. *Aerospace Science and Technology*, page 105999, 2020.

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation propagation. In *Advances in neural information processing systems*, pages 2323–2331, 2015.

Dennis V Lindley. *Bayesian statistics: A review.* SIAM, 1972.

Dennis V Lindley. The future of statistics: A Bayesian 21st century. *Advances in Applied Probability*, 7:106–115, 1975.

James M Longuski, José J Guzmán, and John E Prussing. *Optimal control with aerospace applications.* Springer, 2014.

Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55, 2017.

Sébastien Marmin and Maurizio Filippone. Variational calibration of computer models. *arXiv preprint arXiv:1810.12177*, 2018.

Sébastien Marmin, David Ginsbourger, Jean Baccou, and Jacques Liandrat. Warped Gaussian processes and derivative-based sequential designs for functions with heterogeneous variations. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3): 991–1018, 2018.

G Matheron. Kriging or polynomial interpolation procedures. *CIMM Transactions*, 70: 240–244, 1967.

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

Edward Meeds and Simon Osindero. An alternative infinite mixture of Gaussian process experts. In *Advances in neural information processing systems*, pages 883–890, 2006.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning*, pages 2391–2400, 2017.

Paul CD Milly, Julio Betancourt, Malin Falkenmark, Robert M Hirsch, Zbigniew W Kundzewicz, Dennis P Lettenmaier, and Ronald J Stouffer. Stationarity is dead: Whither water management? *Science*, 319(5863):573–574, 2008.

Edmondo Minisci and Massimiliano Vasile. Robust design of a reentry unmanned space vehicle by multifidelity evolution control. *AIAA journal*, 51(6):1284–1295, 2013.

Thomas P Minka. Expectation propagation for approximate Bayesian inference. *arXiv preprint arXiv:1301.2294*, 2013.

Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Jonas Močkus. On Bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pages 400–404. Springer, 1975.

Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

Bryan A Moore, Esteban Rougier, Daniel O'Malley, Gowri Srinivasan, Abigail Hunter, and Hari Viswanathan. Predictive modeling of dynamic fracture growth in brittle materials with machine learning. *Computational Materials Science*, 148:46–53, 2018.

Amir Mosavi, Timon Rabczuk, and Annamária R Varkonyi-Koczy. Reviewing the novel machine learning tools for materials design. In *International Conference on Global Research and Education*, pages 50–58. Springer, 2017.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Simon Nanty, Céline Helbert, Amandine Marrel, Nadia Pérot, and Clémentine Prieur. Sampling, metamodeling, and sensitivity analysis of numerical simulators with functional stochastic inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 4 (1):636–659, 2016.

Radford M Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, Ontario, Canada, 1993.

Radford M Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

Antonio J Nebro, Juan José Durillo, Jose Garcia-Nieto, CA Coello Coello, Francisco Luna, and Enrique Alba. SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pages 66–73. IEEE, 2009.

Margaret A Oliver and Richard Webster. Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information System*, 4(3):313–332, 1990.

Eva Ostertagová. Modelling using polynomial regression. *Procedia Engineering*, 48: 500–506, 2012.

Christopher J Paciorek and Mark J Schervish. Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics*, 17(5):483–506, 2006.

John W Paisley, David M Blei, and Michael I Jordan. Bayesian nonnegative matrix factorization with stochastic variational inference., 2014.

Andrei Paleyes, Mark Pullin, Maren Mahsereci, Neil Lawrence, and Javier González. Emulation of physical processes with emukit. In *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS*, 2019.

Young-Jin Park, Piyush M Tagade, and Han-Lim Choi. Deep Gaussian process-based Bayesian inference for contaminant source localization. *IEEE Access*, 6:49432–49449, 2018.

JM Parr, AJ Keane, Alexander IJ Forrester, and CME Holden. Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization*, 44(10):1147–1166, 2012.

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60 (3):550–591, 2018.

Julien Pelamatti, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Yannick Guerin. Efficient global optimization of constrained mixed variable problems. *Journal of Global Optimization*, 73(3):583–613, 2019.

Paris Perdikaris, Maziar Raissi, Andreas Damianou, ND Lawrence, and George Em Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751, 2017.

KB Petersen, MS Pedersen, et al. The matrix cookbook, vol. 7. *Technical University of Denmark*, 15, 2008.

Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3):607–626, 2013.

Victor Picheny, Robert B Gramacy, Stefan Wild, and Sebastien Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian. In *Advances in neural information processing systems*, pages 1435–1443, 2016.

Christian Plagemann, Kristian Kersting, and Wolfram Burgard. Nonstationary Gaussian process regression using point estimates of local smoothness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 204–219. Springer, 2008.

Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

Rémy Priem, Nathalie Bartoli, and Youssef Diouane. On the use of upper trust bounds in constrained Bayesian optimization infill criteria. In *AIAA Aviation 2019 Forum*, page 2986, 2019.

Majdi I Radaideh and Tomasz Kozlowski. Surrogate modeling of advanced computer simulations using deep Gaussian processes. *Reliability Engineering & System Safety*, 195:106731, 2020.

MY Rafiq, G Bugmann, and DJ Easterbrook. Neural network design for engineering applications. *Computers & Structures*, 79(17):1541–1552, 2001.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

Maziar Raissi and George Karniadakis. Deep multi-fidelity Gaussian processes. *arXiv preprint arXiv:1604.07484*, 2016.

Dushhyanth Rajaram, Tejas G Puranik, Ashwin Renganathan, Woong Je Sung, Olivia J Pinon-Fischer, Dimitri N Mavris, and Arun Ramamurthy. Deep Gaussian process enabled surrogate models for aerodynamic flows. In *AIAA Scitech 2020 Forum*, page 1640, 2020.

Ananth Ranganathan. Assumed density filtering. *Nov*, 23:2, 2004.

Carl E Rasmussen and Zoubin Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in neural information processing systems*, pages 881–888, 2002.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. In *Advances in neural information processing systems*, pages 294–300, 2001.

José E Rayas-Sánchez. EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art. *IEEE Transactions on Microwave Theory and Techniques*, 52(1):420–435, 2004.

Jose E Rayas-Sanchez. Power in simplicity with ASM: tracing the aggressive space mapping algorithm over two decades of development and engineering applications. *IEEE Microwave Magazine*, 17(4):64–76, 2016.

Junuthula Narasimha Reddy. An introduction to the finite element method. *New York*, 27, 1993.

Samuel Temple Reeve and Alejandro Strachan. Error correction in multi-fidelity molecular dynamics simulations using functional uncertainty quantification. *Journal of Computational Physics*, 334:207–220, 2017.

David T Robinson, Michael S Eldred, Karen E Willcox, and Robert Haimes. Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *Aiaa Journal*, 46(11):2814–2822, 2008.

Simone Rossi, Markus Heinonen, Edwin Bonilla, Zheyang Shen, and Maurizio Filippone. Rethinking sparse gaussian processes: Bayesian approaches to inducing-variable approximations. *arXiv preprint arXiv:2003.03080*, 2020.

Tim Roughgarden. Algorithmic game theory. *Communications of the ACM*, 53(7): 78–86, 2010.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural gradients in practice: Non-conjugate variational inference in Gaussian process models. *arXiv preprint arXiv:1803.09151*, 2018.

Davod K Salkuyeh. Generalized jacobi and gauss-seidel methods for solving linear system of equations. *Numerical mathematics - English series* -, 16(2):164, 2007.

Paul D Sampson and Peter Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, 87 (417):108–119, 1992.

Michael J Sasena. *Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations.* PhD thesis, University of Michigan Ann Arbor, MI, 2002.

Michael J Sasena, Panos Y Papalambros, and Pierre Goovaerts. The use of surrogate modeling algorithms to exploit disparities in function computation time within simulation-based optimization. *Constraints*, 2:5, 2001.

Matthias Schonlau, William J Welch, and D Jones. Global optimization with nonparametric function fitting. *Proceedings of the ASA, section on physical and engineering sciences*, pages 183–186, 1996.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

David W Scott. *Multivariate density estimation: theory, practice, and visualization.* John Wiley & Sons, 2015.

Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *9th Workshop on AI and Statistics*, 2003.

Larry J Segerlind. *Applied finite element analysis*, volume 316. Wiley New York, 1976.

Amar Shah and Zoubin Ghahramani. Pareto frontier learning with expensive correlated objectives. In *International Conference on Machine Learning*, pages 1919–1927, 2016.

Harsheel Shah, Serhat Hosder, Slawomir Koziel, Yonatan A Tesfahunegn, and Leifur Leifsson. Multi-fidelity robust aerodynamic design optimization under mixed uncertainty. *Aerospace Science and Technology*, 45:17–29, 2015.

Bobak Shahriari, Ziyu Wang, Matthew W Hoffman, Alexandre Bouchard-Côté, and Nando de Freitas. An entropy search portfolio for Bayesian optimization. *arXiv preprint arXiv:1406.4625*, 2014.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Simon J Sheather. Density estimation. *Statistical science*, pages 588–597, 2004.

Maolin Shi, Shuo Wang, Wei Sun, Liye Lv, and Xueguan Song. A support vector regression-based multi-fidelity surrogate model. *arXiv preprint arXiv:1906.09439*, 2019.

Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Jeffrey S Simonoff. *Smoothing methods in statistics*. Springer Science & Business Media, 2012.

Timothy W Simpson, JD Poplinski, Patrick N Koch, and Janet K Allen. Metamodels for computer-based engineering design: survey and recommendations. *Engineering with computers*, 17(2):129–150, 2001.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan P Adams. Input warping for Bayesian optimization of non-stationary functions. In *International Conference on Machine Learning*, pages 1674–1682, 2014.

Bruno Sudret. Uncertainty propagation and sensitivity analysis in mechanical models–contributions to structural reliability and stochastic spectral methods. *Habilitation a diriger des recherches, Université Blaise Pascal, Clermont-Ferrand, France*, 2007.

Bruno Sudret. Meta-models for structural reliability and uncertainty quantification. *arXiv preprint arXiv:1203.2062*, 2012.

Timothy John Sullivan. *Introduction to uncertainty quantification*, volume 63. Springer, 2015.

Joshua D Svenson and Thomas J Santner. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis*, 94(C):250–264, 2016.

El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.

El-Ghazali Talbi, Matthieu Basseur, Antonio J Nebro, and Enrique Alba. Multi-objective optimization using metaheuristics: non-standard algorithms. *International Transactions in Operational Research*, 19(1-2):283–305, 2012.

Siyu Tao, Daniel W Apley, Wei Chen, Andrea Garbo, David J Pate, and Brian J German. Input mapping for model calibration with application to wing aerodynamics. *AIAA Journal*, pages 2734–2745, 2019.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393): 82–86, 1986.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

Michalis Titsias and Neil D Lawrence. Bayesian Gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *International conference on machine learning*, pages 1971–1979, 2014.

David JJ Toal and Andy J Keane. Efficient multipoint aerodynamic design optimization via cokriging. *Journal of Aircraft*, 48(5):1685–1695, 2011.

David J.J Toal and Andy J Keane. Non-stationary kriging for design optimization. *Engineering Optimization*, 44(6):741–765, 2012.

David JJ Toal, Andy J Keane, Diego Benito, Jeffery A Dixon, Jingbin Yang, Matthew Price, Trevor Robinson, Alain Remouchamps, and Norbert Kill. Multifidelity multidisciplinary whole-engine thermomechanical design optimization. *Journal of Propulsion and Power*, 30(6):1654–1666, 2014.

Martin Trapp, Robert Peharz, Franz Pernkopf, and Carl Edward Rasmussen. Deep structured mixtures of Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 2251–2261, 2020.

Volker Tresp. Mixtures of Gaussian processes. In *Advances in neural information processing systems*, pages 654–660, 2001.

Frank Tuyl, Richard Gerlach, and Kerrie Mengersen. A comparison of Bayes–Laplace, Jeffreys, and other priors: the case of zero events. *The American Statistician*, 62(1): 40–44, 2008.

Nalika Ulapane, Karthick Thiyagarajan, et al. Hyper-parameter initialization for squared exponential kernel-based Gaussian process regression. In *15th IEEE Conference on Industrial Electronics and Applications*, pages 1154–1159, 2020.

Michele Vallisneri. Use and abuse of the Fisher information matrix in the assessment of gravitational-wave parameter-estimation prospects. *Physical Review D*, 77(4): 042001, 2008.

Imco van Gent and Gianfranco La Rocca. Formulation and integration of mdao systems for collaborative design: A graph-based methodological approach. *Aerospace Science and Technology*, 90:410–433, 2019.

Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research*, 15(1): 3483–3512, 2014.

Jarno Vanhatalo, Pasi Jylänki, and Aki Vehtari. Gaussian process regression with student-t likelihood. In *Advances in neural information processing systems*, pages 1910–1918, 2009.

Felipe AC Viana, Raphael T Haftka, and Layne T Watson. Efficient global optimization algorithm assisted by multiple surrogate techniques. *Journal of Global Optimization*, 56(2):669–689, 2013.

Roberto Vitali, Raphael T Haftka, and Bhavani V Sankar. Multi-fidelity design of stiffened composite panel with a crack. *Structural and Multidisciplinary Optimization*, 23(5):347–356, 2002.

Abraham Wald. Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, 20(4):595–601, 1949.

Matt P Wand and James F.C Yu. Density estimation via Bayesian inference engines. *arXiv preprint arXiv:2009.06182*, 2020.

Gary Wang and Songqing Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design*, 129(4):370–380, 2007.

Liping Wang, Don Beeson, Srikanth Akkaram, and Gene Wiggs. Gaussian process meta-models for efficient probabilistic design in complex engineering design spaces. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 4739, pages 785–798, 2005.

Yali Wang, Marcus Brubaker, Brahim Chaib-Draa, and Raquel Urtasun. Sequential inference for deep Gaussian process. In *Artificial Intelligence and Statistics*, pages 694–703, 2016.

Alan G Watson and Randal J Barnes. Infill sampling criteria to locate extremes. *Mathematical Geology*, 27(5):589–608, 1995.

Christopher KI Williams and Carl E Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784. PMLR, 2015.

David H Wolpert et al. On the Bayesian" Occam factors" argument for Occam's razor. *Computational learning theory and natural learning systems III, T. Petsche et al.(eds)*, 1995.

Max A Woodbury. Inverting modified matrices. *Memorandum report*, 42(106):336, 1950.

Manyu Xiao, Guohua Zhang, Piotr Breitkopf, Pierre Villon, and Weihong Zhang. Extended co-kriging interpolation method based on multi-fidelity data. *Applied Mathematics and Computation*, 323:120–131, 2018.

Shifeng Xiong, Peter ZG Qian, and Jeff C.F Wu. Sequential design and analysis of high-accuracy and low-accuracy computer codes. *Technometrics*, 55(1):37–46, 2013.

Ying Xiong, Wei Chen, Daniel Apley, and Xuru Ding. A non-stationary covariance-based kriging method for metamodelling in engineering design. *International Journal for Numerical Methods in Engineering*, 71(6):733–756, 2007.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 9951–9960, 2019.

Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 75(1):3–34, 2019.

Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21:1881–1888, 2008.

Alexey Zaytsev and Evgeny Burnaev. Large scale variable fidelity surrogate modeling. *Annals of Mathematics and Artificial Intelligence*, 81(1-2):167–186, 2017.

Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2010.

Jingying Zhao, Hai Guo, Likun Wang, and Min Han. Computer modeling of the eddy current losses of metal fasteners in rotor slots of a large nuclear steam turbine generator based on finite-element method and deep Gaussian process regression. *IEEE Transactions on Industrial Electronics*, 67(7):5349–5359, 2019.

Eckart Zitzler, Marco Laumanns, Lothar Thiele, Carlos M Fonseca, and Viviane Grunert da Fonseca. Why quality assessment of multiobjective optimizers is difficult. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pages 666–674, 2002.

Appendix A

Publications and communications

A.1 Journal Articles

- **Hebbal, A.**, Brevault, L., Balesdent, M., Talbi, E. G., & Melab, N. Bayesian optimization using deep Gaussian processes with applications to aerospace system design. In *Optimization and Engineering*, 1-41. Springer, 2020.

- **Hebbal, A.**, Brevault, L., Balesdent, M., Talbi, E. G., & Melab, N. Multi-fidelity modeling with different input domain definitions using deep Gaussian processes. In *Structural and Multidisciplinary Optimization Journal.* (Accepted).

- Brevault, L., Balesdent, M., & **Hebbal, A**. Multi-objective multidisciplinary design optimization approach for partially reusable launch vehicle design. In *Journal of Spacecraft and Rockets*, 57(2), 373-390, 2020.

- Brevault, L., Balesdent, M., & **Hebbal, A**. Overview of Gaussian process based multi-fidelity techniques with variable relationship between fidelities, application to aerospace systems. In *Aerospace Science and Technology*, 106339. Vol.107, Elsevier, 2020.

A.2 Book Chapters

- Brevault, L., Pelamatti, J., **Hebbal, A.**, Balesdent, M., Talbi, E. G., & Melab, N. MDO Related Issues : Multi-objective and mixed continuous/discrete optimization. In *Aerospace System Analysis and Optimization in Uncertainty* (pp. 321-358). Springer, Cham, 2020.

- Brevault, L., Balesdent, M., & **Hebbal, A**. Expendable and reusable launch vehicle design. In *Aerospace System Analysis and Optimization in Uncertainty* (pp. 421-476). Springer, Cham, 2020.

A.3 Communications

- **Hebbal, A.**, Brevault, L., Balesdent, M., Talbi, E., & Melab, N. Multi-fidelity modeling using DGPs : Improvements and a generalization to varying input space dimensions. *4th workshop on Bayesian Deep Learning (NeurIPS 2019)*, Vancouver, Canada.

- **Hebbal, A.**, Brevault, L., Balesdent, M., Talbi, E., & Melab, N. A deep Gaussian Process based model for multi-objective optimization. *The 13th International Conference on Multiple Objective Programming and Goal Programming (MOPGP)*, 2019, Marrakesh, Morocco.

- **Hebbal, A.**, Brevault, L., Balesdent, M., Talbi, E. G., & Melab, N. Multi-objective optimization using deep Gaussian processes: application to aerospace vehicle design. In *AIAA Scitech 2019 Forum* (p. 1973), San Diego, USA.

- Brevault, L., Balesdent, M., **Hebbal, A.**, & Patureau De Mirand, A. Surrogate model-based multi-objective MDO approach for partially reusable launch vehicle design. In *AIAA Scitech 2019 Forum* (p. 0704), San Diego, USA.

- **Hebbal, A.**, Brevault, L., Balesdent, M., Talbi, E., and Melab, N., Bayesian Optimization using deep Gaussian processes for non-stationary problems. *Program Gaspard Monge (PGMO) days 2018*, Paris, France.

- **Hebbal, A.**, Brevault, L., Balesdent, M., Taibi, E. G., & Melab, N. Efficient global optimization using deep Gaussian processes. In *IEEE Congress on evolutionary computation (CEC)* 2018 (pp. 1-8). Rio de Janeiro, Brazil

- **Hebbal, A.**, Brevault, L., Balesdent, M., Taibi, E. G., & Melab, N. Multidisciplinary design multi-objective optimization of aerospace vehicles using surrogate models. *International workshop on Optimization and Learning: Challenges and Applications* 2018. Alicante, Spain.

Appendix B

Multivariate Gaussian Identities

This appendix summarizes some of the most used multivariate Gaussian equations in this thesis. Details on the demonstrations of these relations can be found in [Petersen et al., 2008; Murphy, 2012].

B.1 Marginals and conditionals of a multivariate Gaussian

Given a joint Gaussian distribution $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2)$ with $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where:

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \tag{B.1}$$

Then, the marginals are given by:

$$\boxed{\begin{aligned} p(\mathbf{f}_1) &= \mathcal{N}(\mathbf{f}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\mathbf{f}_2) &= \mathcal{N}(\mathbf{f}_2 | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned}} \tag{B.2}$$

and the conditional is given by:

$$\boxed{\begin{aligned} &p(\mathbf{f}_1 | \mathbf{f}_2) = \mathcal{N}(\mathbf{f}_1 | \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ &\quad \boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_{\mathbf{1}} + \boldsymbol{\Sigma}_{12} + \boldsymbol{\Sigma}_{22}^{-1}(\mathbf{f}_2 - \boldsymbol{\mu}_2) \\ &\quad \boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \end{aligned}} \tag{B.3}$$

B.2 Bayes rule for linear Gaussian systems

For a given system $\mathbf{y} = \mathbf{A}\mathbf{w} + \mathbf{b}$, and the following prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}})$ and likelihood $p(\mathbf{y}|\mathbf{w}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{w} + \mathbf{b}, \boldsymbol{\Sigma}_{\mathbf{y}})$, the posterior is given by:

$$\boxed{\begin{aligned} p(\mathbf{w}|\mathbf{y}) &= \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_{\mathbf{w}|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{y}}) \\ \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{y}}^{-1} &= \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} + \mathbf{A}^{\intercal}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{A} \\ \boldsymbol{\mu}_{\mathbf{w}|\mathbf{y}} &= \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{y}}\left(\mathbf{A}^{\intercal}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Sigma}_{\mathbf{w}}^{-1}\boldsymbol{\mu}_{\mathbf{w}}\right) \end{aligned}} \tag{B.4}$$

and the marginal likelihood is given by:

$$\boxed{p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}_{\mathbf{x}} + \mathbf{b}, \boldsymbol{\Sigma}_{\mathbf{y}} + \mathbf{A}\boldsymbol{\Sigma}_{\mathbf{w}}\mathbf{A}^{\intercal}\right)} \tag{B.5}$$

B.3 Product of two multivariate Gaussians

Given two multivariate Gaussian densities $p_1(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $p_2(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, then, the product of these densities comes back to a scaled multi-variate Gaussian density:

$$\boxed{\begin{aligned} p_1(\mathbf{x}).p_2(\mathbf{x}) &= c_{\times}.\mathcal{N}(\mathbf{x}|(\mathbf{x}|\boldsymbol{\mu}_{\times}, \boldsymbol{\Sigma}_{\times}) \\ c_{\times} &= \mathcal{N}(\boldsymbol{\mu}_1|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \\ \boldsymbol{\mu}_{\times} &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) \\ \boldsymbol{\Sigma}_{\times} &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} \end{aligned}} \tag{B.6}$$

B.4 Kullback-Liebler divergence between two multivariate Gaussians

Given two multivariate Gaussian densities $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{m}, \mathbf{S})$ and $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of dimension d, the Kullback–Leibler divergence between the distributions is as follows:

$$\boxed{\begin{aligned} \mathbb{KL}[q||p] &= \int q(\mathbf{x})\log\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right)\mathrm{d}\mathbf{x} \\ &= \frac{1}{2}\left(\mathrm{tr}(\mathbf{S}^{-1}\boldsymbol{\Sigma})\right) + \frac{1}{2}\left((\mathbf{m} - \boldsymbol{\mu})^{\intercal}\mathbf{S}^{-1}(\mathbf{m} - \boldsymbol{\mu})\right) - \frac{d}{2} + \frac{1}{2}\log\left(\frac{|\mathbf{S}|}{|\boldsymbol{\Sigma}|}\right) \end{aligned}} \tag{B.7}$$

where $\mathrm{tr}(\cdot)$ stands for the trace of a matrix, and $|\cdot|$ its determinant.

B.5 Information form of multivariate Gaussians

Given a multivariate Gaussian distribution $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The natural parameters are given by:

$$\boxed{\begin{pmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ -\frac{1}{2}\boldsymbol{\Sigma}^{-1} \end{pmatrix}} \tag{B.8}$$

The expectation parameters are given by:

$$\boxed{\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}} \end{pmatrix}} \tag{B.9}$$

Appendix C

Analytical problems

In this appendix, the different analytical problems used throughout this thesis are described.

C.1 Analytical problems in Chapter 4

Modified Xiong function:

$$\begin{aligned} f(x) = & \quad -0.5\left(\sin\left(40(x-0.85)^4\right)\cos\left(2.5(x-0.95)\right)+0.5(x-0.9)+1\right) \\ \text{s.t.} & \quad x \in [0,1] \end{aligned} \tag{C.1}$$

Modified TNK constraint function:

$$\begin{aligned} f(\mathbf{x}) = & \quad 1.6(x_0-0.6)^2+1.6(x_1-0.6)^2-0.2\cos\left(20\arctan\left(\frac{0.3x_0}{(x_1+10^{-8})}\right)\right)-0.4 \\ \text{s.t.} & \quad \mathbf{x} \in [0,1]\times[0,1] \end{aligned} \tag{C.2}$$

10d Trid function:

$$\begin{aligned} f(\mathbf{x}) & = \textstyle\sum_{i=1}^{10}(x_i-1)^2-\sum_{i=2}^{10}x_i x_{i-1} \\ \text{s.t.} & \quad x_i \in [-100,100], \forall i=1,\dots,10 \end{aligned} \tag{C.3}$$

Hartmann-6d function:

$$\begin{aligned} f(\mathbf{x}) = & \quad \textstyle\sum_{i=1}^{4}\alpha_i \exp\left(-\sum_{j=1}^{6}A_{ij}(x_j-P_{ij})^2\right) \\ \text{s.t.} & \quad x_i \in [0,1], \forall i=1,\dots,6 \end{aligned} \tag{C.4}$$

with:

$$\alpha = [1,1.2,3,3.2]^\top$$

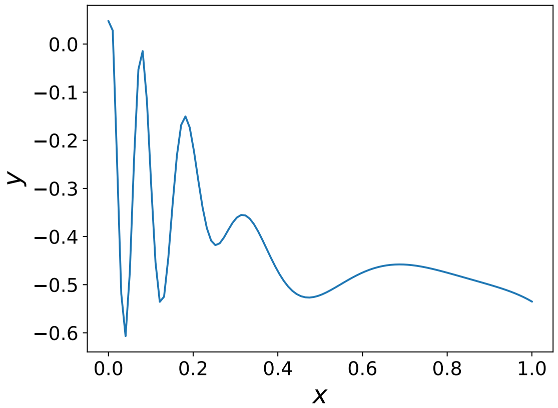


Fig. C.1 Modified Xiong function

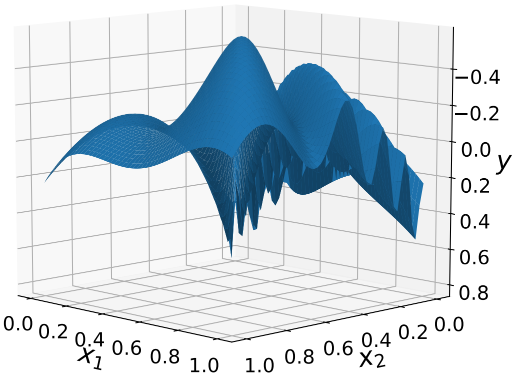


Fig. C.2 Modified TNK constraint

and

$$P = 10^{-4}\begin{bmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{bmatrix}$$

and

$$A = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}$$

C.2 Analytical problems in Chapter 5

1-d two-objective problem:

$$\begin{aligned} \text{min} \quad & [f_1(x), f_2(x)], \\ \text{s.t.} \quad & 0 \le x \le 1, \\ \text{with} \quad & f_1(x) = \exp\left(\cos\left(15(2x-0.2)\right)\right) - 1, \\ \text{and} \quad & f_2(x) = -x\exp\left(\cos\left(15(2x-0.2)\right)\right) - 1; \end{aligned} \tag{C.5}$$

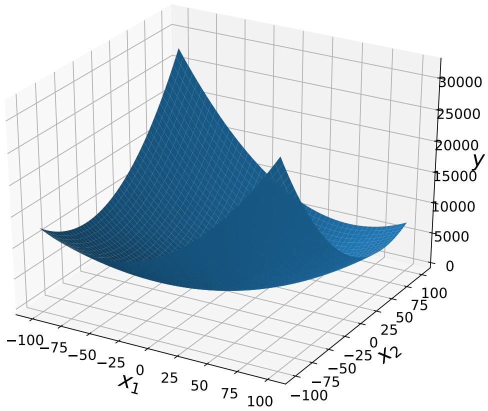


Fig. C.3 Sectional $2d$ view of the Trid function showing where the global minimum lies

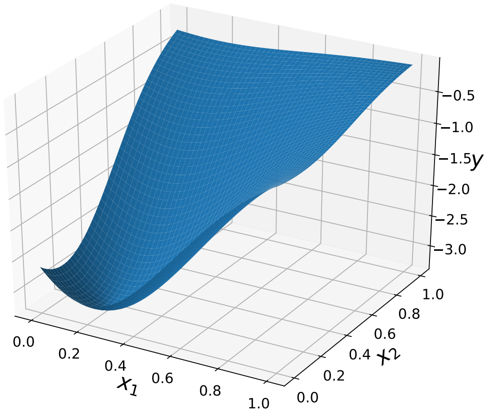


Fig. C.4 Sectional $2d$ view of the Hartmann-6d function showing where the global minimum lies

Kursawe problem:

$$\begin{aligned}
&\min && [f_1(\mathbf{x}), f_2(\mathbf{x})], \\
&\text{s.t.} && -5 \le x_i \le 5 && i = 1,\ldots,3, \\
&\text{with} && f_1(\mathbf{x}) = \textstyle\sum_{i=1}^{2}\left(-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right), \\
&\text{and} && f_2(\mathbf{x}) = \textstyle\sum_{i=1}^{3}\left(|x_i|^{0.8} + 5\sin\left(x_i^3\right)\right);
\end{aligned} \tag{C.6}$$

Modified DTLZ1 problem:

$$\begin{aligned}
&\min && [f_1(\mathbf{x}), f_2(\mathbf{x})], \\
&\text{s.t.} && 0 \le x_i \le 1 && i = 1,\ldots,5, \\
&\text{with} && f_1(\mathbf{x}) = -0.5x_1\left(1 + h(\mathbf{x})\right), \\
& && f_2(\mathbf{x}) = -0.5(1 - x_1)\left(1 + h(\mathbf{x})\right), \\
&\text{and} && h(\mathbf{x}) = 100\left(5 + \textstyle\sum_{i=1}^{5}\left((x_1 - 0.5)^2 - \cos\left(2\pi(x_i - 0.5)\right)\right)\right);
\end{aligned} \tag{C.7}$$

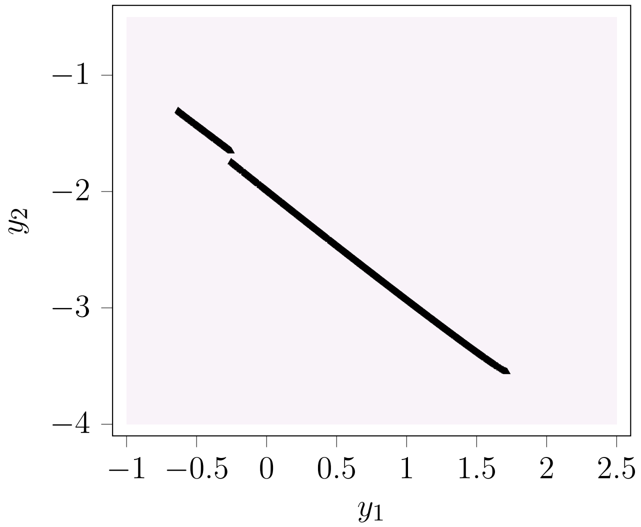


Fig. C.5 Exact Pareto front of the 1-d two-objective problem

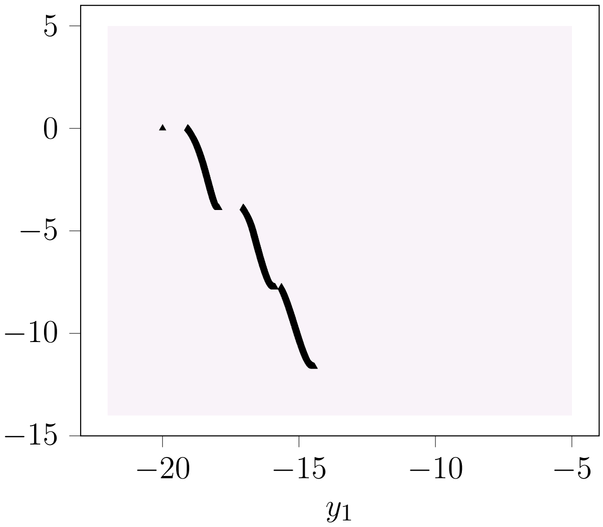


Fig. C.6 Exact Pareto front of the Kursawe problem

ZDT6 problem:

$$\begin{aligned}
&\min && [f_1(\mathbf{x}), f_2(\mathbf{x})], \\
&\text{s.t.} && 0 \le x_i \le 1 && i = 1,\ldots,10, \\
&\text{with} && f_1(\mathbf{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1), \\
& && f_2(\mathbf{x}) = \varphi(\mathbf{x}) h\left(f_1(\mathbf{x}), \varphi(\mathbf{x})\right), \\
& && \varphi(\mathbf{x}) = 1 + 9\left(\frac{\sum_{i=2}^{10} x_i}{9}\right)^{0.25}, \\
&\text{and} && h(f_1(\mathbf{x}), \varphi(\mathbf{x})) = 1 - \sqrt{\frac{f_1(\mathbf{x})}{\varphi(\mathbf{x})}};
\end{aligned} \tag{C.8}$$

C.3 Analytical problems in Chapter 6

Currin function is a 2-d multi-fidelity function defined by two-levels of fidelity, a high-fidelity $f_{\text{hf}}(\cdot)$ and a low fidelity $f_{\text{lf}}(\cdot)$:

$$\begin{aligned}
f_{\text{hf}}(\mathbf{x}) &= \left(1 - \exp\left(-\frac{1}{2x_2}\right)\right) \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}, \\
f_{\text{lf}}(\mathbf{x}) &= \frac{1}{4}\left(f_{\text{hf}}(x_1 + 0.005, x_2 + 0.05) + f_{\text{hf}}(x_1 + 0.05, \max(0, x_2 - 0.05))\right) \\
&\quad + \frac{1}{4}\left(f_{\text{hf}}(x_1 - 0.05, x_2 + 0.05) + f_{\text{hf}}(x_1 - 0.05, \max(0, x_2 - 0.05))\right), \\
\text{s.t.} &\quad 0 \le x_i \le 1 \qquad i = 1,2;
\end{aligned} \tag{C.9}$$

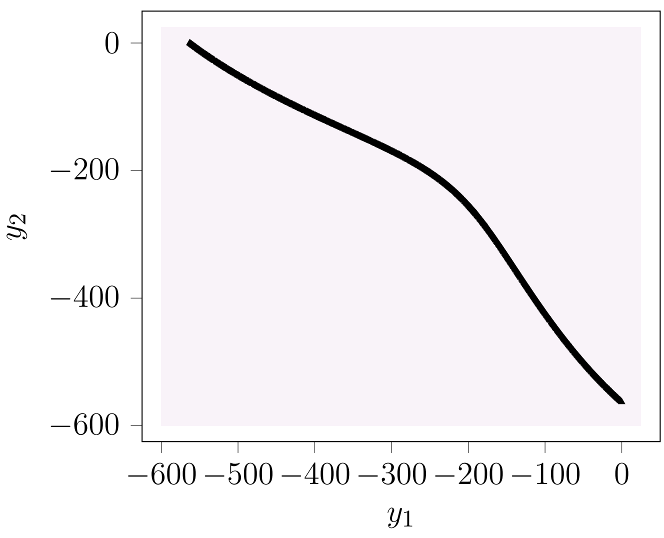


Fig. C.7 Exact Pareto front of the modified DTLZ1 problem

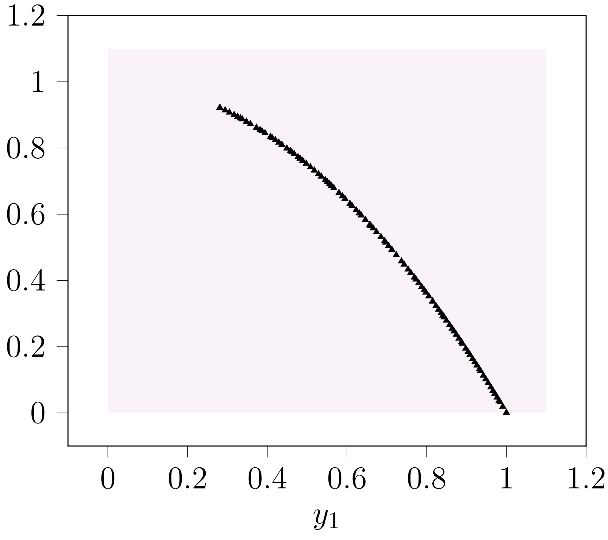


Fig. C.8 Exact Pareto front of the ZDT6 problem

Park function is a 4-d multi-fidelity function defined by two-levels of fidelity, a high-fidelity $f_{\text{hf}}(\cdot)$ and a low fidelity $f_{\text{lf}}(\cdot)$:

$$\begin{aligned} f_{\text{hf}}(\mathbf{x}) &= \tfrac{x_1}{2}\left(\sqrt{1+(x_2+x_3^2)\tfrac{x_4}{x_1^2}}-1\right)+(x_1+3x_4)\exp\left(1+\sin(x_3)\right), \\ f_{\text{lf}}(\mathbf{x}) &= \left(1+\tfrac{\sin(x_1)}{10}\right) f_{\text{hf}}(\mathbf{x})-2x_1+x_2^2+x_3^2+0.5, \\ \text{s.t.} \quad & 0 \le x_i \le 1 \qquad i=1,\dots,4; \end{aligned} \tag{C.10}$$

Borehole function is a 8-d multi-fidelity problems that represents water flow through a borehole. It is defined by two-levels of fidelity, a high-fidelity $f_{\text{hf}}(\cdot)$ and a low fidelity

$f_{\text{lf}}(\cdot)$:

$$\begin{aligned}
f_{\text{hf}}(\mathbf{x}) &= \frac{2\pi x_3(x_4-x_6)}{\log\left(\frac{x_2}{x_1}\right)\left(1+\frac{2x_7x_3}{\log(x_2/x_1)x_1^2x_8}\right)+\frac{x_3}{x_5}},\\
f_{\text{lf}}(\mathbf{x}) &= \frac{5x_3(x_4-x_6)}{\log\left(\frac{x_2}{x_1}\right)\left(1+\frac{2x_7x_3}{\log(x_2/x_1)x_1^2x_8}\right)+\frac{x_3}{x_5}},\\
\text{s.t.} \quad & 0.05 \leq x_1 \leq 0.15,\\
& 100 \leq x_2 \leq 50000,\\
& 63070 \leq x_3 \leq 115600,\\
& 990 \leq x_4 \leq 1110,\\
& 63.1 \leq x_5 \leq 115,\\
& 700 \leq x_6 \leq 820,\\
& 1120 \leq x_7 \leq 1680,\\
& 9855 \leq x_8 \leq 12045;
\end{aligned} \tag{C.11}$$

The three-levels Branin function is a 2-d multi-fidelity function defined by three-levels of fidelity, a high-fidelity $f_{\text{hf}}(\cdot)$, a medium fidelity $f_{\text{m}}(\cdot)$, and a low fidelity $f_{\text{lf}}(\cdot)$:

$$\begin{aligned}
f_{\text{hf}}(\mathbf{x}) &= \left(\frac{-1.275x_1^2}{\pi^2}+\frac{5x_1}{\pi}+x_2-6\right)^2+\left(10-\frac{5}{4\pi}\right)\cos(x_1)+10,\\
f_{\text{m}}(\mathbf{x}) &= 10\sqrt{f_{\text{hf}}(\mathbf{x}-2)}+2(x_1-0.5)-3(3x_2-1)-1,\\
f_{\text{lf}}(\mathbf{x}) &= f_{\text{m}}(1.2(\mathbf{x}+2))-3x_2+1,\\
\text{s.t.} \quad & -5 \leq x_1 \leq 10,\\
& 0 \leq x_2 \leq 15;
\end{aligned} \tag{C.12}$$

Appendix D

Numerical setup

D.1 General numerical setup

- The Experiments presented in this manuscript were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

- All experiments were executed on Grid'5000 using a Tesla P100 GPU.

- The codes involving GPs and DGPs are based on Tensorflow [Abadi et al., 2016], GPflow [De G. Matthews et al., 2017] (https://github.com/GPflow/GPflow), and Doubly-Stochastic-DGP [Salimbeni and Deisenroth, 2017] (https://github.com/ICL-SML/Doubly-Stochastic-DGP) in Python 3.

- The data is always normalized and standardized (zero mean and a variance equal to 1).

- For all DGPs (in BO with DGPs, MO-DGP, and MF-DGP), ARD RBF kernels are used with a length-scale and variance initialized to 1 if it does not get an initialization from a previous DGP.

- The optimization of DGPs is performed following Algorithm 1. Adam optimizer is set with $\beta_1 = 0.8$ and $\beta_2 = 0.9$ and a step size $\gamma^{adam} = 0.01$. The natural gradient step size is initialized for the last layer at $\gamma^{nat} = 0.1$ and the inner layers at $\gamma^{nat} = 0.01$

D.2 Specific numerical setup to Chapter 4

- For BO with DGP, the number of successive updates before optimizing from scratch is 5.

- The infill criteria are optimized using a parallel differential evolution algorithm with a population of 400 and 100 generations (https://www.tensorflow.org/probability/api_docs/python/tfp/optimizer/differential_evolution_minimize).

- BO with Bayesian non-linear mapping is set using the numerical setup proposed in [Snoek et al., 2014].

- For DGP with BO, the inducing inputs at the different layers are initialized to $[\mathbf{X}, \mathbf{0}]^\intercal$, where $\mathbf{0}$ is the null matrix of size $n_{\text{final}} - n \times d$, n the current size of the DoE, n_{final} is the final size of the data-set at the end of the BO algorithm.

- DGP with BO is optimized in two stages. In the first one, 5000 Adam optimization steps are performed while fixing the variational parameters. Then, 20000 iterations of Algorithm 1 are performed.

D.3 Specific numerical setup to Chapter 5

- The number of Gibbs sampling iterations used is 4 and in each iteration 1000 samples are drawn.

- LMC is used with a coregionalization matrix of rank 2.

- The Python package Platypus [Hadka, 2015] (https://github.com/Project-Platypus/Platypus) was used for NSGA-II. It is used with its default parameters proposed in [Deb et al., 2000] with a population of 5 candidates. The population evolves until reaching $15 \times d$ evaluations of the objective functions, where d is the dimension of the input space.

- For MO-DGP, the inducing inputs at the different layers are initialized to $\mathbf{X}$.

- The mean of the variational distribution of the inducing variables for the layer i is initialized at $\mathbf{y}_i$.

- MO-DGP is optimized in three stages. In the first one, 3000 Adam optimization steps are performed while fixing the variational parameters and the inducing inputs. In the second one, the inducing inputs are also optimized using 3000 Adam optimization steps. Then, 15000 iterations of Algorithm 1 are performed.

D.4 Specific numerical setup to Chapter 6

- The python package Emukit [Paleyes et al., 2019] is used for the multi-fidelity models AR1 and NARGP (https://github.com/EmuKit/emukit).
- For MF-DGP-EM, the inducing inputs of the fidelity GP at layer t are initialized to $\mathbf{X}_t$, and for the input mapping GP at layer t they are initialized at $\mathbf{X}_{t+1}$.
- The mean of the variational distribution of the inducing variables for layer t is initialized at $\mathbf{y}_t$, and for the input mapping GP at layer t at X_t^{t+1}.
- MF-DGP-EM is optimized in three stages. In the first one, 3000 Adam optimization steps are performed while fixing the variational parameters and the inducing inputs. In the second stage, the inducing inputs are also optimized using 3000 Adam optimization steps. Then, 15000 iterations of Algorithm 1 are performed.

Abstract

In engineering, the design of complex systems, such as aerospace launch vehicles, involves the analysis and optimization of problems presenting diverse challenges. Actually, the designer has to take into account different aspects in the design of complex systems, such as the presence of black-box computationally expensive functions, the complex behavior of the optimized performance (*e.g.*, abrupt change of a physical property here referred as non-stationarity), the multiple objectives and constraints involved, the multi-source information handling in a multi-fidelity framework, and the epistemic and aleatory uncertainties affecting the physical models. A wide range of machine learning methods are used to address these various challenges. Among these approaches, Gaussian Processes (GPs), benefiting from their Bayesian and non-parametric formulation, are popular in the literature and diverse state-of-the-art algorithms for the design of complex systems are based on these models.

Despite being widely used for the analysis and optimization of complex systems, GPs, still present some limitations. For the optimization of computationally expensive functions, GPs are used within the Bayesian optimization framework as regression models. However, for the optimization of non-stationary problems, they are not suitable due to the use of a prior stationary covariance function. Furthermore, in Bayesian optimization of multiple objectives, a GP is used for each involved objective independently, which prevents the exhibition of a potential correlation between the objectives. Another limitation occurs in multi-fidelity analysis where GP-based models are used to improve high-fidelity models using low-fidelity information. However, these models usually assume that the different fidelity input spaces are identically defined, which is not the case in some design problems.

In this thesis, approaches are developed to overcome the limits of GPs in the analysis and optimization of complex systems. These approaches are based on Deep Gaussian Processes (DGPs), the hierarchical generalization of Gaussian processes.

To handle non-stationarity in Bayesian optimization, a framework is developed that couples Bayesian optimization with DGPs. The inner layers allow a non-parametric Bayesian mapping of the input space to better represent non-stationary functions. For multi-objective Bayesian optimization, a multi-objective DGP model is developed. Each layer of this model corresponds to an objective and the different layers are connected with undirected edges to encode the potential correlation between objectives. Moreover, a computational approach for the expected hyper-volume improvement is proposed to take into account this correlation at the infill criterion level as well. Finally, to address multi-fidelity analysis for different input space definitions, a two-level DGP model is developed. This model allows a joint optimization of the multi-fidelity model and the input space mapping between fidelities.

The different approaches developed are assessed on analytical problems as well as on representative aerospace vehicle design problems with respect to state-of-the-art approaches.

Résumé

En ingénierie, la conception de systèmes complexes, tels que les lanceurs aérospatiaux, implique l'analyse et l'optimisation de problèmes présentant diverses problématiques. En effet, le concepteur doit prendre en compte différents aspects dans la conception de systèmes complexes, tels que la présence de fonctions coûteuses en temps de calcul et en boîte noire , la non-stationnarité des performances optimisées, les multiples objectifs et contraintes impliqués, le traitement de multiples sources d'information dans le cadre de la multi-fidélité, et les incertitudes épistémiques et aléatoires affectant les modèles physiques. Un large éventail de méthodes d'apprentissage automatique est utilisé pour relever ces différents défis. Dans le cadre de ces approches, les Processus Gaussiens (PGs), bénéficiant de leur formulation Bayésienne et non paramétrique, sont populaires dans la littérature et divers algorithmes d'état de l'art pour la conception de systèmes complexes sont basés sur ces modèles.

Les PGs, bien qu'ils soient largement utilisés pour l'analyse et l'optimisation de systèmes complexes, présentent encore certaines limites. Pour l'optimisation de fonctions coûteuses en temps de calcul et en boite noire, les PGs sont utilisés dans le cadre de l'optimisation Bayésienne comme modèles de régression. Cependant, pour l'optimisation de problèmes non stationnaires, les PGs ne sont pas adaptés en raison de l'utilisation d'une fonction de covariance stationnaire. En outre, dans l'optimisation Bayésienne multi-objectif, un PG est utilisé pour chaque objectif indépendamment des autres objectifs, ce qui empêche de prendre en considération une corrélation potentielle entre les objectifs. Une autre limitation existe dans l'analyse multi-fidélité où des modèles basés sur les PGs sont utilisés pour améliorer les modèles haute fidélité en utilisant l'information basse fidélité, cependant, ces modèles supposent généralement que les différents espaces d'entrée de fidélité sont définis de manière identique, ce qui n'est pas le cas dans certains problèmes de conception.

Dans cette thèse, des approches sont développées pour dépasser les limites des PGs dans l'analyse et l'optimisation de systèmes complexes. Ces approches sont basées sur les Processus Gaussiens Profonds (PGPs), la généralisation hiérarchique des PGs.

Pour gérer la non-stationnarité dans l'optimisation bayésienne, un algorithme est développé qui couple l'optimisation bayésienne avec les PGPs. Les couches internes permettent une projection Bayésienne non paramétrique de l'espace d'entrée pour mieux représenter les fonctions non stationnaires. Pour l'optimisation Bayésienne multiobjectif, un modèle de PGPs multiobjectif est développé. Chaque couche de ce modèle correspond à un objectif et les différentes couches sont reliées par des arrêtes non orientés pour coder la corrélation potentielle entre objectifs. De plus, une approche de calcul de l'expected hyper-volume improvement est proposée pour prendre également en compte cette corrélation au niveau du critère d'ajout de point. Enfin, pour aborder l'analyse multi-fidélité pour différentes définitions d'espace d'entrée, un PGP à deux niveaux est développé. Ce modèle permet une optimisation conjointe du modèle multi-fidélité et du mapping entre les espaces d'entrée des différentes fidélités.

Les différentes approches développées sont évaluées sur des problèmes analytiques ainsi que sur des problèmes de conception de véhicules aérospatiaux et comparées aux approches de l'état de l'art.