The state of the earth: estimating physical parameters from noisy and incomplete earth observation data

Arp, L.R.

6

eMMI: ϵ-MANIFOLDS OF POTENTIAL SOLUTIONS FOR NOISY INFERENCE

In the previous chapter, we have learned key information about noisy inversion problems through our analysis of biophysical parameter estimation using EO data and PROSAIL inversion: observation noise plays a central role in making inference problems ill-posed, resulting in optimum shifts that appear to follow the loss landscape of the inference problem. In this chapter[1], we aim to leverage this knowledge to propose the concept of ϵ-manifolds, which contain all possible solutions to a model inversion inference problem, and a method, eMMI, to approximate these ϵ-manifolds. In doing so, we address RQ4: *How can we automatically extract the set of possible solutions to a noisy inference problem?*. Although our findings in Chapter 5 suggest that pure methodological contributions are unlikely to eliminate ill-posedness from noisy inference problems altogether, the concepts and method introduced in this chapter will address Challenge 2 by enabling novel types of analyses, enabling users to, for example, make a judgement on whether to trust their parameter estimations, based on the degree of ill-posedness of their specific problem instance.

6.1. INTRODUCTION

Inferring a model parameterisation from observations is a fundamental problem in many scientific domains. Such inference problems are considered *model inver-*

[1] The contents of this chapter are based on the article: Laurens Arp, Peter van Bodegom, Nguyen Dang, Holger H. Hoos, Alistair Francis, and Mitra Baratchi. (2025). *Inference from Noisy Observations through Model Inversion: Constructing ϵ-Manifolds of Potentially Valid Solutions.* Under review.

sion problems when there is a (simulation) model available to simulate observations from a hypothesised parameter configuration, such that model parameters are the target values to infer, and the observations consist of data that can be simulated by the model (e.g., observable variables or class labels). Model inversion-based inference problems are ubiquitous in many scientific fields, including AI. Within physical sciences, model inversion is often used when inferring the unobservable values of a set of physical parameters that resulted in an observed outcome [212, 213, 214, 215], whereas in simulation-based inference (SBI), a probabilistic simulation model is used as a likelihood function in a Bayesian inference setting, where the target parameters are often inferred posterior distributions, given the observed data [216, 217, 218]. Application areas reliant on model inversion include fluid dynamics [212, 213], astronomy and astrophysics [5, 6] and Earth sciences [214, 215]; in these fields, often large amounts of observational data are available, but few labelled examples, making it challenging to apply conventional supervised machine learning approaches to map observations to labels. Instead, physical simulation models are used to estimate these parameters. Within AI, we are often using machine learning to infer a target variable based on the observed features, and the training of machine learning models is itself a model inversion-based inference problem where the model parameterisation must be learned from the observed training data.

A simulation model cannot be used directly to infer the correct parameters from real observations. However, it can be used indirectly to evaluate the quality of a possible parameter configuration by comparing its simulated observations to the real observations through a loss function. Various approximation techniques can be used to infer model parameters. These include numerical optimisation [47, 48], specialised simulation-based inference algorithms [219, 218, 217] and training a machine learning model on simulated data [50, 220, 221].

Model inversion is a non-trivial problem, shown to be NP-hard if addressed using numerical optimisation [222]. It is often *ill-posed*, meaning that the inversion of a single observation can lead into multiple different solutions with equal, or highly similar, quality [57, 223, 224]. It can also be *ill-conditioned*, meaning that small perturbations in the observations cause large shifts in the optimal parameter configuration for a problem instance [205, 206, 225]. Moreover, real-world observations are nearly always noisy, due to limitations of sensing technologies that gather observations, inaccurate scenario specifications [226], or noisy data annotation, leading to ill-posedness of the inference problem. As a result, even if these methods reliably find the global optimum of an inference problem, the resulting configuration may not correspond to the 'true' parameter values that generated the observations. The optimum would thus explain the noisy observations rather than the true state of the system.

Due to these challenges, practitioners may quantify uncertainty by inferring a distribution over parameter values rather than making point predictions [216, 46, 54, 45]. However, statistical distributions usually assume certain properties (e.g., Gaussian parametric form, independence or stationarity). These properties are often not met, or they differ per instance and cannot be known *a priori*. If the inversion problem is ill-posed with multimodal or non-continuous distributions, values will likely not be centred around the mean. For example, for a model $y = \alpha^2$ with observation $y = 100$ and a domain $\alpha \in \mathbb{R}$, the solution for the parameter α could be 10 or -10, but not any values in between. Moreover, uncertainty quantification models a distribution over parameter values, not the degree to which the parameter values fit the observations. Therefore, configurations that could effectively explain the observations, but do not naturally appear in a data set, are unlikely to be included. This confines the applicability of uncertainty quantification methods to indicating statistical confidence on a prediction, while deeper analyses of the inversion problem itself, including solutions that do not occur naturally in the observational distribution, may be desired.

In this article, for the first time, we aim to retrieve the set of all solutions that can reasonably explain the observations in model inversion problem settings, forming a manifold we refer to as the *ϵ-manifold*. The intuition behind ϵ-manifolds is that they contain configurations for which the loss function value is sufficiently close to that of the optimal configuration to be considered a potentially feasible solution, given the level of noise ϵ distorting the signal of the observations. Therefore, using ϵ-manifolds shifts the focus away from distributions over the parameters based on their posterior probability, toward the inversion loss landscape (landscapes consisting of the loss function values for all possible target value configurations). When known, the ϵ-manifold can be used to enable novel types of analysis of model inversion problems, such as ill-posedness quantification, various types of robustness analysis and classification difficulty estimation.

Our main contributions in this chapter are as follows:

- We introduce the concept of ϵ-manifolds for noisy model inversion problems and formalise the problem of capturing the set of potentially valid solutions comprising this ϵ-manifold. To our knowledge, this is the first time the problem has been defined in this manner.

- We propose a novel method, named eMMI (epsilon-manifolds for model inversion), to automatically approximate the ϵ-manifold for a given model inversion problem instance. We provide four variants of our proposed method (U-eMMI, Conv-eMMI, Seq-eMMI, and Dual-eMMI), with different sampling strategies and assumptions on the underlying loss landscape.

- We validate the concept of ϵ-manifolds through an empirical comparison to statistical uncertainty quantification. We also validate our ϵ-manifold approximation method on seven simulation models representing model inversion problem settings from physical models, dynamical systems, simulation-based inference and machine learning. We compare performance against baseline methods from uncertainty estimation methods such as Gaussian processes and Bayesian neural networks, as well as approximate Bayesian computation.

6.2. RELATED WORK

The problem setting for ϵ-manifolds and our proposed method is model inversion with noisy observations. There are three main directions of related work to this setting, which we will discuss in this section. We will start with *robust learning*, a specific type of noisy model inversion-based inference task considering machine learning with noisy labels, that has received considerable attention over the years. Next, we will discuss two methodologically related directions: *simulation-based inference* (SBI) (also referred to as *likelihood-free inference*), and *uncertainty quantification*.

Robust learning. There has long been a research interest in the machine learning community in robust learning: methods that make model training robust to noisy labels (the observations when training a model) [227, 228, 229]. More recently, Northcutt et al. [230] have proposed confident learning, where confidence applies to ground truth labels, rather than model predictions. Uma et al. [231] summarised a body of work on conflicting labels specific to natural language processing and computer vision settings, including majority voting, the source-filter model [232, 233] and the CrowdTruth aggregation approach [234]. Bernhardt et al. [235] proposed to automatically rank training instances based on the label correctness and difficulty as estimated by a prediction model, and found this to improve model performance while reducing the reliance on label-correcting experts. Jiang et al. [236] provided a dataset containing real-world (as opposed to synthetic) noisy labels and proposed the MentorMix method to overcome these noisy labels through curriculum learning and vicinal risk minimisation. Huang et al. [237] investigated the relationship between uncertainty, class imbalance and label noise, and proposed an uncertainty-aware label correction (ULC) framework, which first filters noisy labels based on epistemic uncertainty, after which the remaining corrupted labels are filtered by modelling aleotoric uncertainty as logit corruption with Gaussian noise. Kim et al. [238] proposed to use the relational structure of the data in the embedded feature space to detect noisy labels. Although classification problems tend to receive the most attention, some work has focused on

regression problems as well [239, 240, 241]. In the related predict-then-optimise problem setting [242, 243] (where unobservable parameters are imperfectly predicted using machine learning, which enables the optimisation of a second set of parameters for decision-making), smart predict then optimise (SPO) approaches [226, 244] can be used to train a machine learning model using a loss based on the regret between an optimum found using the predicted unobserved parameters and the true optimum, thereby reducing the impact of noise in the predicted unobserved parameters on the optimisation task.

Unlike the methods above, which are mainly intended to improve the model performance in terms of predictive accuracy when trained using a training data set with noisy labels, the objective of our proposed ϵ-manifolds is to find the set of parameterisations that would all fit the observations (e.g., noisy labels) up to a tolerance level specified by ϵ. These ϵ-manifolds provide deeper insight over performance-focused approaches like robust learning, and have broader applications beyond machine learning model training or robust optimisation, including ill-posedness- and robustness analyses in general model inversion settings.

Simulation-based inference. In many scientific contexts, great effort has been put into formulating models that simulate an observation from a set of input parameters. The goal of simulation-based inference (SBI) is to apply statistical methods to infer the posterior probability of the input parameters $\boldsymbol{\theta}$, which form the inference targets, from a vector of observed outcomes $\mathbf{x}$ [218, 245, 217]. The likelihood of the observations given an input parameter configuration $\boldsymbol{\theta}$ is generally intractable to compute. Following the notation by Cranmer et al. [216], the problem may be formulated (in the Bayesian case) as computing the posterior $p(\boldsymbol{\theta}|\mathbf{x})$, where $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta},\mathbf{z})$ and $z_i \sim p_i(z_i|\boldsymbol{\theta}, z_{<i})$ ($\mathbf{z}$ represents the latent internal state of the simulator). A point prediction for the inference result can be computed using, e.g., a maximal a posteriori principle $\hat{\boldsymbol{\theta}} \in \text{argmax}_\theta\, p(\boldsymbol{\theta}|\mathbf{x})$. Applications of SBI span a highly diverse set of scientific fields and topics including astrometry [5], Earth sciences [246], gravitational waves [247], astrophysics [6], and genomics [248].

Based on the taxonomy by Cranmer et al. [216], we can broadly split SBI methods into frequentist and Bayesian inference approaches. Frequentist approaches infer the probability of parameters through estimated kernel densities, while Bayesian approaches iteratively approximate the posterior probability of the parameters using observations and prior probabilities. One of the most popular methods for SBI, approximate Bayesian computation (ABC) [249], samples parameter configurations from their prior distributions. It accepts these configurations if the simulated output matches the true observations at a sufficient goodness-of-fit level determined by a threshold ϵ. Similar to our proposed methods, this results in a posterior distribution of possible configurations whose simulated output is ϵ-approximate to the observed data. Unlike our proposed methods, ϵ is a (some-

times dynamic) hyperparameter trading off accuracy for computational efficiency. The posterior distribution depends in part on the prior distribution of the parameters being inferred, making them primarily suitable for conventional inference tasks. ABC methods can often involve Markov chain Monte Carlo (MCMC) [250] and sequential Monte Carlo (SMC) sampling [251]. Some model inversion methods in, e.g., environmental biology, may use numerical optimisation techniques in a manner similar to SBI using ABC [47, 49, 48]. Given the poor scalability of inferring complex posterior probabilities using a Bayesian approach, variational methods and amortised versions thereof can also be used [252, 253, 254].

More recently, advances in machine learning have led to the use of inverse emulation models, often coupled with active learning techniques [50, 51, 52, 45, 53, 54, 255, 55, 46, 54, 45]. These approaches resemble amortised Bayesian inference methods [256, 257, 258], but the machine learning models are trained on the parameters themselves, rather than the posterior distribution parameters identified through Bayesian inference. Other methodological contributions in SBI include reducing the assumptions of models, such as pre-defined priors, targets and dimensionalities [259]. Additionally, scalability has been improved through flow matching for continuous normalising flows [260].

Although SBI research has made many valuable contributions to a wide range of (especially scientific) application areas, it adopts an inherently statistical approach to model inversion, inferring posterior distributions over the parameters instead of capturing characteristics of the loss landscape. Hermans et al. found many SBI-based methods to be overconfident in their inference results [261], indicating that unlikely solutions that could nonetheless explain the observations well would generally not be included. In contrast, our proposed ϵ-manifolds aim to provide insight into the model inversion problem for a given observation based on the model inversion loss landscape, enabling new types of analyses. Unlike methods such as ABC, which return a posterior distribution over parameters based on their goodness-of-fit to the observations and the prior probabilities of the parameters, our proposed method does not aim to make a single prediction (inference) with some degree of uncertainty for a given model inversion instance. Instead, it aims to find the set of configurations that *could* explain the observations, regardless of how likely such configurations are to naturally occur. This can be a desirable trait, for example, when considering interventions or adversarial examples (both of which involve solutions that may not naturally occur in the observational distribution). Therefore, our proposed ϵ-manifolds and our proposed method to approximate them could be considered complementary to SBI, by describing the loss landscape underlying the problem for which SBI is performing inference. Whether SBI or ϵ-manifolds are the most appropriate tool depends on the use case, as both have different types of applications (see also Section 6.4.3).

Uncertainty quantification. The topic of uncertainty quantification (UQ) has received considerable attention in recent years, and UQ is often a part of SBI. Typical approaches for UQ include ensembling [262, 263]), Monte-Carlo dropout [264, 265] and Bayesian neural networks [266, 267]). For a more comprehensive overview, see, for example, the surveys by Adbar et al. [268] or Gawlikowski et al. [269]. Recent examples for scientific machine learning in particular include UQ for physics-informed neural networks [270], physics-constrained surrogate modeling with UQ [271], Monte-Carlo UQ for the Navier-Stokes equations [272], UQ for neural networks trained on physical simulations [273], and the application of a Hamiltonian Monte Carlo algorithm for physical model inversion [195].

The core idea of UQ, similar to our proposed ϵ-manifolds, is that the numerical prediction that seems to explain the observations best may not be the true solution, due to uncertainty (for example, in the form of noisy observations or probabilistic simulations). The statistical perspective of UQ enables a relatively quick computation of uncertainty in terms of, usually, a confidence interval around a (mean) point prediction, but this does not lend itself to the same type of interpretation: a relatively unlikely solution could still explain the observations well, even if it has not been observed in the training data set. Retrieving such solutions could be highly relevant to applications related to, e.g., adversarial robustness or deliberate interventions to achieve a desired outcome. Moreover, the type of relationships that can be expressed by conventional UQ is limited by the parametric form of the assumed distribution; for example, a loss landscape for a chaotic or ill-posed system would be difficult to describe parametrically, especially when the most suitable distribution type cannot be known *a priori*.

In contrast, our proposed ϵ-manifolds describe the loss landscape, identifying a set of solutions that could explain the observations. In doing so, it is possible to use ϵ-manifolds to gain insight into the nature of the inversion problem itself, beyond a statistical perspective of observations and possible outcomes.

6.3. PROBLEM DEFINITION

In this section, we briefly introduce the problems addressed in this work. First, we provide the general problem definition of model inversion, as this is the context within which our work is situated. Next we provide definitions for two problems at the core of our investigation: representing a viable solution set for model inversion (problem 1), and approximating this viable solution set (problem 2).

Context: model inversion. Suppose we are interested in a set of *parameters* P, the values of which are contained in domains $\mathcal{D}_P$. A configuration $\boldsymbol{\theta}$, which we will refer to as a solution in optimisation contexts, is a vector containing the values of the target variables P for a specific point in $\mathcal{D}_P$. The target variables can be ob-

served indirectly via known variables, whose domain is denoted as $\mathcal{X} \subseteq \mathbb{R}^d$. Given a d-dimensional vector of observations $\mathbf{x} \in \mathcal{X}$, we want to infer the (unknown) corresponding true parameter configuration of the system $\boldsymbol{\theta}^+$ via a simulation model M mapping configurations $\boldsymbol{\theta}$ to simulated observations $\mathbf{x}$.

Definition 6.1. A simulation model is a probabilistic function $M : \mathcal{D}_P \rightarrow \mathcal{X}$ that, under a target parameterisation $\boldsymbol{\theta}$, can produce (i.e., simulate) an outcome $\mathbf{x} = M(\boldsymbol{\theta})$, where $M(\boldsymbol{\theta})$ samples $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$.

In other words, we want to search for a configuration $\boldsymbol{\theta} \in \mathcal{D}_P$ such that $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ is minimised, thereby approximating the true configuration $\boldsymbol{\theta}^+$ corresponding to the real observations $\mathbf{x}$. This objective can be considered as a more general version of the RTM inversion objective of Section 2.2.1. Here $f_1(.)$ is an objective function quantifying the distance between the simulated outcome $M(\boldsymbol{\theta})$ and the observed outcome $\mathbf{x}$:

$$\hat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta} \in \mathcal{D}_P}{\operatorname{argmin}} f_1(M(\boldsymbol{\theta}), \mathbf{x}) \approx \boldsymbol{\theta}^+ \tag{6.1}$$

This optimal configuration $\boldsymbol{\theta}$ is denoted as $\hat{\boldsymbol{\theta}}$ and usually forms the point prediction within the inference results of the model inversion problem. Therefore, unlike the distributions over target values inferred by SBI (whose formulation can be found in Section 6.2), this objective is concerned with the model inversion loss landscape for the inference problem. If the search was successful, $\hat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}^+$.

Problem 1: representing the viable solution set (Section 6.4). We assume an amount of exogenous noise $\mathbf{N}$ on the observations $\mathbf{x}$ or ill-posedness in the inversion loss landscape, thereby potentially invalidating the optimal solution $\hat{\boldsymbol{\theta}}$ from Equation 6.1. Instead, we are interested a subset $\mathcal{V} \subseteq D_P$ of the domain D_P containing all solutions that could possibly be the true solution of the model inversion problem (viable solution set). This set should contain all ill-posed solutions evaluating to the same loss function value as $\hat{\boldsymbol{\theta}}$, but also all solutions that evaluate to a slightly worse loss function value than $\hat{\boldsymbol{\theta}}$, up to a factor of ϵ. The exogenous noise $\mathbf{N}$ and its distribution are often unknown, because noise-free versions of the noisy data are usually impossible to obtain. However, the impact of this noise on the loss function (quantified by ϵ) can be extracted purely from validation data for the target variables $\boldsymbol{\theta}$, without requiring noise-free observations $\mathbf{x}^+ = \mathbf{x} - \mathbf{N}$ to be available. This allows us to define the viable solution set as:

$$\mathcal{V} = \{\boldsymbol{\theta} : f_1\big(M(\boldsymbol{\theta}), \mathbf{x}\big) \leq f_1\big(M(\hat{\boldsymbol{\theta}}), \mathbf{x}\big) + \epsilon\} \tag{6.2}$$

Therefore, for problem 1, we need a framework within which we can represent this set of points $\mathcal{V}$.

Problem 2: approximating the viable solution set (Section 6.5). Computing $\mathcal{V}$ exhaustively is prohibitively expensive for large dimensions of $\boldsymbol{\theta}$, and impossible without interpolation techniques for continuous problem settings. We must therefore find a tractable approximation $\hat{\mathcal{V}}$ of $\mathcal{V}$, by maximising the accuracy on a set of validation points (H_x, H_y):

$$\hat{\mathcal{V}} \in \underset{S}{\operatorname{argmax}} \mathcal{L}(H_y, \hat{H}_y|S) \tag{6.3}$$

Here, H_y denotes the true labels (viable or non-viable solution) corresponding to a sample of points H_x in the target variable space, and $\hat{H}_y|S$ is a vector of predictions of H_y based on a candidate set of viable solutions S. The function $\mathcal{L}$ is a classification loss function of choice (e.g., accuracy), and $\hat{\mathcal{V}}$ is the set of solutions S with the best classification performance on the validation data set (H_x, H_y).

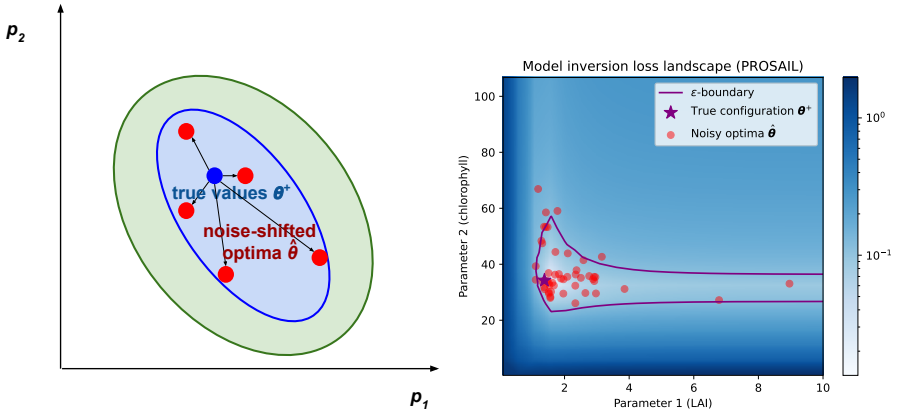
6.4. REPRESENTING THE VIABLE SOLUTION SET

In this section, we will introduce the motivation, concepts and assumptions underlying ϵ-manifolds, which form our representation of the viable solution set for model inversion problems.

6.4.1. INTRODUCING ϵ-MANIFOLDS

Recall the model inversion objective from Equation 6.1. In practice, there are two issues with this naïve search approach. Firstly, multiple configurations $\boldsymbol{\theta} \in \mathcal{D}_P$ can minimise $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ equally (ill-posedness), while only one true configuration $\boldsymbol{\theta}^+$ corresponds to the state of the real-world system. Secondly, if the observations are noisy, the vector of observations $\mathbf{x}$ has been generated from a vector of true observable values, $\mathbf{x}^+$, combined with a vector of exogenous additive noise $\mathbf{N}$: $\mathbf{x} = \mathbf{x}^+ + \mathbf{N}$. Therefore, a configuration $\hat{\boldsymbol{\theta}}$ that precisely minimises $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ is not necessarily the true configuration $\boldsymbol{\theta}^+$. $\hat{\boldsymbol{\theta}}$ would be the appropriate solution for the incorrect observed values $\mathbf{x}$, not the true state of the system (which would have had its own optimal solution $\hat{\boldsymbol{\theta}}^+ \approx \boldsymbol{\theta}^+$ minimising $f_1(M(\boldsymbol{\theta}), \mathbf{x}^+)$. For convenience, similarly to the distinction made by, e.g., Mandi et al. [226], we refer to the optimum $\hat{\boldsymbol{\theta}}$ for the noisy observations $\mathbf{x}$ as the *noisy optimum*, and the optimum $\hat{\boldsymbol{\theta}}^+$ for the original, noise-free observations $\mathbf{x}^+$ as the *noise-free optimum*.

The *optimum shift* from the noise-free optimum $\hat{\boldsymbol{\theta}}^+$ to the noisy optimum $\hat{\boldsymbol{\theta}}$, caused by observation noise, makes the inversion problem effectively ill-posed. There are many potential solutions, each corresponding to the observations with different added noise. Since the noise is unknown, any of these solutions could be the true solution to the model inversion problem. This effect is particularly strong in ill-conditioned problems, where small perturbations on the input (obser-



(a) Optimum shift illustration

(b) Example shift

Figure 6.1: Optimum shifts when noise is introduced to the observations. (a) Illustration of the principle of optimum shifts on an abstract loss landscape for target variables p_1 and p_2. When random noise is applied to the observations, the point in the space where, after simulating with those input parameter settings, the simulation output matches the observations optimally, has shifted from the true parameters (the blue dot, $\boldsymbol{\theta}^+$) to new, shifted optima (the red dots, $\hat{\boldsymbol{\theta}}$). The specific point the optimum shifts to will differ every time the random noise is applied. The blue line represents the ϵ-boundary; beyond this point, there are no possible points the optimum could shift to, at the current level of noise. The blue-shaded area represents the ϵ-manifold, containing the set of all points the optimum could potentially shift to. (b) Example of this phenomenon in practice for one instance from the physical vegetation model PROSAIL, where the application of 15% additive zero-mean Gaussian noise on the observations has caused the optimum $\hat{\boldsymbol{\theta}}$ to shift away from the true values $\boldsymbol{\theta}^+$ (repeated for 50 different samples of random noise added to the noise-free observations). The shifts are more likely for configurations close to the optimum, resulting in a cluster that could possibly be captured by conventional uncertainty quantification methods, but the shifts tend to follow the loss landscape (Assumption 1 in Section 6.4) and stay within the ϵ-boundary.

vations) result in large changes in the output (optimum) [205, 206]. We illustrate this phenomenon with a practical example in Figure 6.1.

As a result of these issues, we need a method describing a set $\mathcal{V}$ of all the potentially viable solutions $\boldsymbol{\theta}$. These solutions could, under some expected noise on the observations, reasonably be the true configuration that generated the observations $\mathbf{x}$. In simple cases matching the underlying assumptions, uncertainty quantification on the optimal configuration $\hat{\boldsymbol{\theta}}$ may be sufficient. For example, if the optimum shifts follow a Gaussian distribution, a confidence interval could be constructed, containing all values within two standard deviations σ of the predicted mean μ. In this case, any point with a probability density higher than some user-specified threshold could be added to the set $\mathcal{V}$ of possible solutions. However, these as-

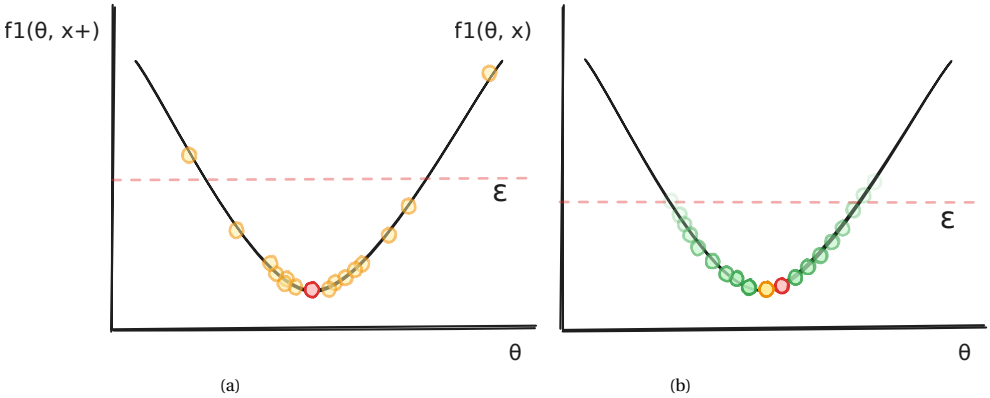


Figure 6.2: Illustration of the two key assumptions made by ϵ-manifolds. (a) In the first assumption, we assume that an optimum for noise-free observations (red dot) will shift to a new location due to observational noise (yellow dots) based on the loss landscape (parabola function) for noise-free observations; a higher loss function value results in a lower likelihood of being shifted to due to noise. (b) In the second assumption, we assume that the probability (green dots, with lower opacity signifying lower probabilities) that an optimum for noisy observations (yellow dot) was originally shifted from a noise-free optimum (red dot) is determined by the loss landscape (parabola function) for the noisy observations.

sumptions are not always met.

The inversion loss landscape for many simulation models, which are often based on complex ordinary differential equations (ODEs) and partial differential equations (PDEs), can be complex, asymmetric, biased and highly non-linear – conditions to which statistical uncertainty quantification would be ill-suited. Instead, by making two key assumptions of *loss-dependent shifts* and *loss-dependent origins* (illustrated in Figure 6.2), this set of points can be represented more accurately using a concept we dubbed ϵ-manifolds.

We formalise the loss-dependent shifts assumption in Assumption 1. According to this assumption (illustrated in Figure 6.2a), the lower the loss function value of a point for a noise-free observation, the higher its likelihood of being the optimum of a noisy version of the observations $\mathbf{x}$ (optimum shift).

Assumption 1. *Let $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ denote two arbitrary points in the target variable space $\mathcal{D}_P$, and let $P(\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}|\mathbf{x})$ denote the probability of a point $\boldsymbol{\theta}$ being the optimum for a noisy observation $\mathbf{x}$ that was generated through exogenous additive noise N being added to the original observations $\mathbf{x}^+$. The **loss-dependent shifts** assumption states that:*

$$f_1\big(M(\boldsymbol{\theta}_1), \mathbf{x}^+\big) < f_1\big(M(\boldsymbol{\theta}_2), \mathbf{x}^+\big) \Rightarrow P(\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_1|\mathbf{x}) > P(\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_2|\mathbf{x}) \tag{6.4}$$

Using this assumption, we could, in principle, set a threshold distance ϵ between the loss of the noise-free optimum $f_1(M(\hat{\boldsymbol{\theta}}^+), \mathbf{x}^+)$ and the loss $f_1(M(\boldsymbol{\theta}), \mathbf{x}^+)$ of a new point $\boldsymbol{\theta}$: if this distance is smaller than the threshold, the point can be considered as a potential location the optimum could shift to. Although there may be applications for this set of points in, e.g., algorithm robustness [274, 275, 276], this set of solutions mainly serves as an intermediate step in our problem setting of model inversion.

Next, we formalise the loss-dependent origins assumption in Assumption 2. This assumption (illustrated in Figure 6.2b) can be considered the inverse of our first assumption. In model inversion, given a noisy observation $\mathbf{x}$, we are interested in a set of solutions in which, at a specified level of confidence, we expect the true solution $\boldsymbol{\theta}^+$ to be included. Under Assumption 2, the lower the loss function value of a point for a noisy observation, the higher its likelihood of having been the original, noise-free optimum $\hat{\boldsymbol{\theta}}^+$, where $\hat{\boldsymbol{\theta}}^+ \approx \boldsymbol{\theta}^+$.

Assumption 2. *Let $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ denote two arbitrary points in the target variable space $\mathcal{D}_P$, and let $P(\hat{\boldsymbol{\theta}}^+ = \boldsymbol{\theta}|\mathbf{x})$ denote the probability of a point $\boldsymbol{\theta}$ being the noise-free optimum $\hat{\boldsymbol{\theta}}^+$. The **loss-dependent origins** assumption states that:*

$$f_1\big(M(\boldsymbol{\theta}_1), \mathbf{x}\big) < f_1\big(M(\boldsymbol{\theta}_2), \mathbf{x}\big) \Rightarrow P(\hat{\boldsymbol{\theta}}^+ = \boldsymbol{\theta}_1|\mathbf{x}) > P(\hat{\boldsymbol{\theta}}^+ = \boldsymbol{\theta}_2|\mathbf{x}) \tag{6.5}$$

As before, we set a threshold ϵ for this distance in loss function value; with this, we can now define the concept of an ϵ-manifold.

Definition 6.2. An ϵ-manifold (eM) is a local connected set of points $\boldsymbol{\theta}$ forming a manifold, where the difference between their loss function values $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ and the loss function value of the noisy optimum $f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x})$ is equal to or smaller than the threshold parameter ϵ:

$$\begin{aligned} eM = \{\boldsymbol{\theta} : f_1\big(M(\boldsymbol{\theta}), \mathbf{x}\big) \leq f_1\big(M(\hat{\boldsymbol{\theta}}), \mathbf{x}\big) + \epsilon\} \\ \text{subject to} \\ \nexists U, V : U \neq \emptyset, V \neq \emptyset, U \cap V = \emptyset, eM = U \cup V \text{(connectedness)} \end{aligned} \tag{6.6}$$

Here eM is used to denote the ϵ-manifold for a noisy observation $\mathbf{x}$, and the constraint requires the ϵ-manifold to be connected (i.e., eM is not split into two non-empty open subsets U and V such that the union of these sets forms the ϵ-manifold). In traditional SBI algorithms, the solution $\hat{\boldsymbol{\theta}}$ will generally be a point in this ϵ-manifold [216, 249].

The ϵ variable in Equation 6.6 matches the ϵ of Equation 6.2, and its value can be interpreted as the maximum expected change from the objective function value $f_1(M(\hat{\boldsymbol{\theta}}^+), \mathbf{x}^+)$ of the noise-free optimum $\hat{\boldsymbol{\theta}}^+$ to the objective function value of the noisy optimum $f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x})$. Equivalently, it can be thought of as the maximum

amount of signal ('clean' loss values) that can be distorted or overpowered by noise on the observations. We refer to the boundary between the ϵ-manifold and the rest of the target variable space as the ϵ-boundary, and we use the term ϵ-loss to refer to ϵ added to the loss function value of the optimum $f_1(M(\hat{\boldsymbol{\theta}}),\mathbf{x})$:

$$l^{\epsilon} = f_1\big(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\big) + \epsilon \tag{6.7}$$

6.4.2. PROPERTIES OF ϵ-MANIFOLDS

We will use this section to analyse the theoretical properties of ϵ-manifolds. First, if there is any solution $\boldsymbol{\theta}$ that could be viable solution, this solution is part of an ϵ-manifold:

Lemma 6.1. Let $\boldsymbol{\theta}$ denote an arbitrary point in $\mathcal{D}_P$. Then through Equation 6.6, $f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \leq l^{\epsilon} \Rightarrow \exists eM : \boldsymbol{\theta} \in eM$.

In unimodal landscapes, the total set of potential solutions $\mathcal{V}$(Equation 6.2) exactly matches the ϵ-manifold with the same value of ϵ ($\mathcal{V} = eM$), and the ϵ-manifold for any point $\boldsymbol{\theta}$ where $f_1(M(\boldsymbol{\theta}),\mathbf{x}) \leq l^{\epsilon}$ (Lemma 6.1) will be the same ϵ-manifold eM. However, in multimodal cases (such as the ill-posed example from Section 6.1: $y = \alpha^2$, $\alpha \in \mathbb{R}$), the loss landscape contains multiple local- or approximate global optima, some with their own ϵ-manifold (it is also possible for lower-quality local optima to already be contained in the ϵ-manifold for a higher-quality optimum). In these cases, the viable solution set $\mathcal{V}$ becomes the ϵ-manifold set[2]:

Definition 6.3. An ϵ-manifold set (eMS) is a set of size m containing ϵ-manifolds for a single loss landscape, where every element is a disjoint ϵ-manifold for the ϵ-loss l^{ϵ}, and m is the number of local optima with disjoint ϵ-manifolds:

$$eMS = \{eM_1, eM_2, ..., eM_m\} \tag{6.8}$$

We can use Lemma 6.2 to iteratively construct an ϵ-manifold set:

Lemma 6.2. Let eMS denote a current, potentially incomplete ϵ-manifold set consisting of ϵ-manifolds $eM \in eMS$, and let $\boldsymbol{\theta}$ denote a point in $\mathcal{D}_P$ where $\forall eM \in eMS : \boldsymbol{\theta} \notin eM$. Then $f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \leq l^{\epsilon} \Rightarrow \exists eM' : \boldsymbol{\theta} \in eM'$ (Lemma 6.1) that is disjoint from the existing ϵ-manifolds in the ϵ-manifold set (Definition 6.3) and should be added to the ϵ-manifold set.

[2] Although it is convenient to think of ϵ-manifold sets as sets containing individual manifolds, strictly speaking, the ϵ-manifold set is itself a manifold of which $eM_1, eM_2, ..., eM_m$ are components.

Based on Lemma 6.2, if there exists any point outside the existing ϵ-manifold eM with a loss function value lower than the ϵ-loss, there must exist another ϵ-manifold eM' that, by Definition 6.3, is disjoint from eM. This leads to Theorem 6.1, which allows us to check whether an ϵ-manifold set is complete:

Theorem 6.1. Let $\hat{\boldsymbol{\theta}}^{-eMS}$ denote the optimum $\operatorname{argmin}_{\boldsymbol{\theta}\in[\mathcal{D}_P\setminus eMS]}\left[f_1(M(\boldsymbol{\theta}),\mathbf{x})\right]$ where the current ϵ-manifold set eMS is excluded from the search space $\mathcal{D}_P$. If $f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)> l^\epsilon$, there exists no further ϵ-manifold that should be added to the ϵ-manifold set eMS, and all possible ϵ-manifolds are contained in the ϵ-manifold set.

Proof.

$$\hat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta}\in[\mathcal{D}_P\setminus eMS]}{\operatorname{argmin}} \left[f_1\left(M(\boldsymbol{\theta}),\mathbf{x}\right)\right] \Rightarrow \forall \boldsymbol{\theta}' \in [\mathcal{D}_P \setminus eMS] : f_1\left(M(\boldsymbol{\theta}'),\mathbf{x}\right) \geq f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)$$

$$f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)> l^\epsilon \Rightarrow \forall \boldsymbol{\theta}' \in [\mathcal{D}_P \setminus eMS] : f_1\left(M(\boldsymbol{\theta}'),\mathbf{x}\right)> l^\epsilon$$

$$f_1\left(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\right)> l^\epsilon \Rightarrow \nexists \boldsymbol{\theta}' : f_1\left(M(\boldsymbol{\theta}'),\mathbf{x}\right)< l^\epsilon$$

Therefore, by Lemma 6.2, the ϵ-manifold is complete, and no further ϵ-manifolds should be added to it.

□

Whether the ϵ-manifold set for a multimodal landscape contains a single, larger ϵ-manifold, or multiple disjoint smaller ϵ-manifolds, will depend on the loss landscape and the setting of ϵ.

The ϵ-manifolds in the ϵ-manifold set together contain all points in the viable solution set $\mathcal{V}$:

Theorem 6.2. Let eMS denote the complete ϵ-manifold set of a model inversion problem, and $\mathcal{V}$ the set of viable solutions to the model inversion problem (Equation 6.2). Then:

$$\mathcal{V} = \bigcup_{eM\in eMS} eM \tag{6.9}$$

Proof. Let $\boldsymbol{\theta} \in \mathcal{V}$ denote an arbitrary viable solution in $\mathcal{V}$. By Equation 6.2, $f_1\left(M(\boldsymbol{\theta}),\mathbf{x}\right) \leq l^\epsilon$. Therefore, by Lemma 6.1, $\exists eM : \boldsymbol{\theta} \in eM$, and by Equation 6.8 and Theorem 6.1, this $eM \in eMS$. Therefore, $\forall \boldsymbol{\theta} \in \mathcal{V} : \boldsymbol{\theta} \in \bigcup_{eM\in eMS} eM$. Conversely, let $\boldsymbol{\theta} \in eM \in eMS$ denote an arbitrary viable solution in an arbitrary ϵ-manifold in the ϵ-manifold set. By Equation 6.6, $\forall \boldsymbol{\theta} \in eM \in eMS : f_1\left(M(\boldsymbol{\theta}),\mathbf{x}\right) \leq l^\epsilon$, and by Equation 6.2, $\boldsymbol{\theta} \in \mathcal{V}$, so $\forall \boldsymbol{\theta} \in \bigcup_{eM\in eMS} eM : \boldsymbol{\theta} \in \mathcal{V}$. Therefore, $\mathcal{V} = \bigcup_{eM\in eMS} eM$. □

In the illustration of Figure 6.1a we used to show the concept of optimum shifts, the blue-shaded region within which the optima are shifted forms a visual representation of the ϵ-manifold. The appropriate setting of ϵ, similarly to the role of the significance level α in confidence intervals, will depend on the specific problem setting, but can be approximated empirically (see Section 6.6.1).

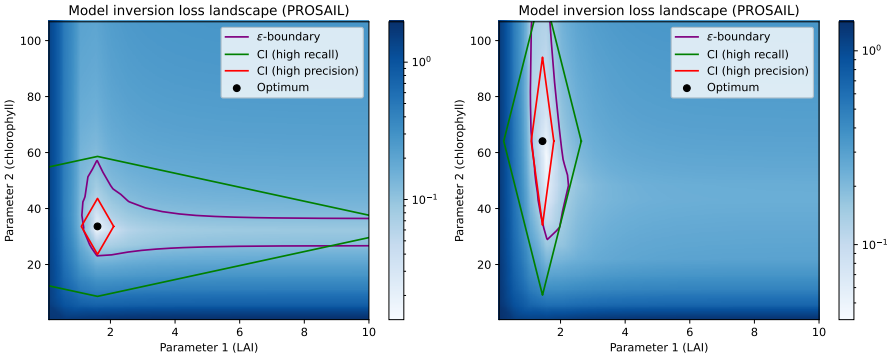
6.4.3. CONTRASTING ϵ-MANIFOLDS AND CONFIDENCE INTERVALS

Our proposed ϵ-manifolds bear some resemblance to the use of confidence intervals in statistical settings, as both concepts represent a type of uncertainty on predictions made for an observation. The key differences lie in their interpretation and application. An ϵ-manifold contains all the solutions that *could* explain the observations, whereas confidence intervals are concerned with the probability of different solutions that they *did* result in the observations.

A confidence interval provides bounds within which the true value is contained at a probability of $1-\alpha$, where α represents the significance level. This statistical quantity can be computed relatively easily. It treats the underlying processes as a black-box generating noisy outcomes, focusing on the spread of possible target values for a given observation, based on the posterior probability of the target variables given the observations.

If there are configurations that are unlikely to appear in data (far removed from the point prediction, with a low prior probability), but could explain the observations equally well (similar loss function value), confidence intervals would be unlikely to include them. This could be, because such cases were not observed in the data, or because including them (improving recall) would come at the expense of an increase in false positives (reducing precision), as illustrated in Figure 6.3. For most distribution types, the probability of target variable values monotonically decreases as distance to the point prediction (the value with the highest likelihood) increases. The range of the confidence intervals would greatly depend on the shape of the assumed distribution (usually Gaussian).

Uncertainty quantification in the form of confidence intervals and other statistical metrics can be a highly effective practical tool for indicating confidence or uncertainty for concrete prediction tasks, and can answer questions such as 'how reliable is my prediction for this specific instance?'. In contrast, an ϵ-manifold provides insight into the relationship between inputs and outputs irrespective of the probability of the inputs, functioning as a tool to analyse and gain an understanding of an inversion problem. It allows us to answer questions related to uncertainty, such as 'what is the set of possible configurations that could have satisfied these observations?', but also 'how ill-posed is this model inversion instance', 'is there a configuration that would satisfy all these observation instances simultane-



(a) Example landscape, instance 1

(b) Example landscape, instance 2

Figure 6.3: Comparing the ϵ-boundary (purple) and confidence intervals (means $\mu\pm$ two times the standard deviation σ, in green for a high recall and red for a high precision) through visualising the loss landscape for the inversion of two different instances with two parameters using a specific physical model for vegetation (PROSAIL). Deeper shades of blue on the loss landscape represent a higher loss function value (plotted at a log scale). (a) An instance where parameter 1 (LAI) is skewed. (b) An instance where parameter 2 (chlorophyll) is skewed. Relying on rigid assumptions on the shape of the distribution, the confidence intervals based on the Gaussian distribution cannot realistically capture the viable solution set for either of the instances, while the ϵ manifolds can flexibly do so in both scenarios.

ously?', 'are there configurations that do not occur naturally that would achieve a desired outcome?', or 'how accurate does my model parameterisation need to be to still achieve similar performance?' Rather than indicating confidence through a summary statistic, ϵ-manifolds enable deeper analyses of problems and problem instances. Unlike confidence intervals, where the focus is on the target variable space and distances within this space, ϵ-manifolds focus on the loss function values of solutions, regardless of how these solutions relate to one another in the target variable space.

Consider the model inversion problems in Figure 6.3. The optimum for Figure 6.3a (which would be the point prediction of a statistical model) for the leaf area index (LAI) variable is around 1.8, but its ϵ-manifold ($\epsilon = 0.1$) covers LAI values between 1.5 and its maximal value, 10. This is a known phenomenon in the domain [277], where the signal from LAI gets 'saturated' after a certain point, after which the observed light spectrum is no longer affected by further increases. As a result, the ϵ-manifold is asymmetric, which a standard Gaussian distribution could not represent, as reducing under- or overinclusivity in one direction would increase it in the other. We show this in Figure 6.3, where 'high recall' refers to a confidence

interval designed to include as many of the points in the ϵ-manifold as possible, and 'high precision' refers to a confidence interval designed to exclude as many of the points outside the ϵ-manifold as possible. Neither objective can be achieved without sacrificing the other, illustrating that confidence intervals often cannot represent the same set of points as ϵ-manifolds. Additionally, phenomena such as the saturation of LAI values would be difficult to identify if only values close to the mean (with high prior probabilities) were included.

Finally, even if all possible viable solutions were represented in the data, and a suitable long-tailed LAI distribution type were found for the problem in Figure 6.3a, the next instance in Figure 6.3b would require a completely different type of distribution to characterise the shape of the loss landscape for the viable solutions. Given the high variability of distribution properties between instances, which cannot be known *a priori*, it is unlikely that similar analyses to those enabled by ϵ-manifolds could be achieved through existing frameworks, such as confidence intervals.

For this reason, we consider the use of ϵ-manifolds to be the appropriate choice when aiming to gain insight into the loss landscape of the inversion problem itself, rather than treating the inversion as a noisy black box that causes uncertainty on the inference results. However, their use may come at the expense of a higher computational cost.

6.5. APPROXIMATING THE VIABLE SOLUTION SET

In this section, we will describe our proposed method, called ϵMMI or, more conveniently, eMMI (ϵ/epsilon-Manifolds for Model Inversion), for approximating the ϵ-manifold in practice[3]. If fully committed to existing statistical frameworks, in some cases it may be possible to approximate a set of points similar to the ϵ-manifold as the non-parametric posterior distribution approximated by an ABC algorithm with a uniform prior over the entire search space, and an acceptance condition based on the ϵ-loss. However, such an approach would quickly become computationally expensive, as the number of simulations required to accurately approximate the posterior for the large search space would quickly become intractable in higher dimensions. If amortised approaches were used to improve computational efficiency, the reliance on summary statistics would reduce the applicability of such approaches to ϵ-manifold approximation (see Section 6.4.3).

In contrast, our proposed method, eMMI, aims to efficiently sample points based on the loss function landscape. In most problem settings, the loss landscape will be intractable to compute fully, particularly in high dimensions where

[3]All code for our proposed method and experiments is publicly available at https://github.com/ADA-research/eMMI

the required number of samples increases exponentially. However, the curse of dimensionality can become a blessing in this problem setting. For example, if 50% of a target variable range is viable per dimension, and dimensionality $d = 10$, only $(0.5)^{10} \times 100\% \approx 0.1\%$ of the space would be viable, enabling the use of efficient local sampling strategies. By Theorem 6.2, an effective approximation of the *local* ϵ-manifold or ϵ-manifold set, through such an efficient sampling approach, approximates the viable solution set $\mathcal{V}$ as $\hat{\mathcal{V}}$ (Equation 6.3). Therefore, eMMI aims to exploit the sparsity of the search space through heuristics and assumptions about the loss landscape. As this will be the first time a method is proposed to explicitly approximate the ϵ-manifold, we strove to keep the design of eMMI modular by splitting its execution into different steps.

6.5.1. eMMI HIGH-LEVEL OVERVIEW

We propose a general three-step approach for approximating the ϵ-manifold in the loss function landscape for a model inversion problem instance. We consider that eMMI is given a finite budget of function evaluations (simulations with loss function value computation) B, which can be split freely between the different steps and whose division is a tunable hyperparameter of the method. The general steps of the method are described below:

1. Searching over the target variable space $\mathcal{D}_P$ for a configuration $\hat{\boldsymbol{\theta}}$ such that $f_1\big(M(\hat{\boldsymbol{\theta}}), \mathbf{x}\big)$ is minimised.

2. Given $\hat{\boldsymbol{\theta}}$ (found in step 1) and ϵ, conducting a search over $\mathcal{D}_P$ to find a diverse set of configurations $\boldsymbol{\Theta}$ around the ϵ-boundary.

3. Approximating the ϵ-manifold using the points sampled in step 2.

We have further provided an overview of eMMI in Algorithm 6.1. We will be providing additional details to the operations contained in Algorithm 6.1 over the rest of this section. In Algorithm 6.1, lines 6 through 9 correspond to step 1 as described above (showing random search as an example for simplicity), lines 11 through 28 correspond to step 2, and lines 29 through 33 correspond to step 3.

Algorithm 6.1 eMMI algorithm overview

Input: observation $\mathbf{x}$; simulator M; maximal shift ϵ; eMMI variant var; maximal ϵ-manifold set size m; optimisation budgets B_1, B_2; #iterations/population size n_{iter}
Output: ϵ-manifold set eMS
$eMS \leftarrow \emptyset$, $l^{\epsilon} \leftarrow \infty$
for $n = 1, ..., m$ **do**
Begin step 1
  $\hat{\boldsymbol{\theta}} \leftarrow$ a random solution sampled from $D_P \setminus eMS$ ▷ initialise; can be warm-started
  for $i = 1, \ldots, B_1$ **do**
    $\boldsymbol{\theta} \leftarrow$ a random solution sampled from $D_P \setminus eMS$ ▷ sample a new solution
    if $f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x})$ **then** ▷ f_1 from Eq. 6.10
      $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$ ▷ update optimum
  if $f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) > l^{\epsilon}$ **then break loop** ▷ break loop to return eMS
Begin step 2
  $l^{\epsilon} \leftarrow f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) + \epsilon$ ▷ store the ϵ-loss (Eq. 6.7)
  if var = Seq-eMMI **then**
    $\boldsymbol{\theta}' \leftarrow \hat{\boldsymbol{\theta}}$ ▷ search for a point $\boldsymbol{\theta}'$ on ϵ-boundary
    for $i = 1, \ldots, \lfloor B_2 / n_{iter} \rfloor$ **do**
      $\boldsymbol{\theta} \leftarrow$ a random solution sampled from $D_P \setminus eMS$
      if $|f_1(M(\boldsymbol{\theta}), \mathbf{x}) - l^{\epsilon}| \leq |f_1(M(\boldsymbol{\theta}'), \mathbf{x}) - l^{\epsilon}|$ **then**
        $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$ ▷ update solution if new point is closer to ϵ-loss
  $\boldsymbol{\Theta} \leftarrow initialise\_population(var, \hat{\boldsymbol{\theta}})$ ▷ initialise population
  $H \leftarrow []$ ▷ list to store function evaluations
  for $i = 1, \ldots, \lfloor B_2 / n_{iter} \rfloor$ **do** ▷ assuming synchronous updates (Seq/Dual-eMMI)
    for $j = 1, \ldots, n_{iter}$ **do** ▷ flip loops for asynchronous updates
      $\boldsymbol{\theta} \leftarrow$ a random solution sampled from $D_P \setminus eMS$
      if $f_2(M(\boldsymbol{\theta}), \mathbf{x}) \leq f_2(M(\boldsymbol{\Theta}^j), \mathbf{x})$ **then** ▷ f_2 from Eq. 6.13, 6.14, Eq. 6.15, 6.16
        $\boldsymbol{\Theta}^j \leftarrow \boldsymbol{\theta}$ ▷ update j-th solution in population
      $y \leftarrow 0$ ▷ original label is false
      if $f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq l^{\epsilon}$ **then**
        $y \leftarrow 1$ ▷ change label to true if in ϵ-manifold
      append $(\boldsymbol{\theta}, y)$ **to** H
Begin step 3
  if var = Conv-eMMI **then**
    $eM \leftarrow convex\_hull(\boldsymbol{\Theta})$ ▷ ϵ-manifold becomes convex hull of solutions
  if $var \in$ { U-eMMI, Seq-eMMI, Dual-eMMI} **then**
    $eM \leftarrow train\_classifier(H)$ ▷ ϵ-manifold becomes trained classifier
  append eM **to** eMS ▷ Add eM to ϵ-manifold set
return eMS

6.5.2. STEP 1: FINDING AN OPTIMUM

In step 1, the system performs a search with an evaluation budget B_1 to find the solution $\hat{\boldsymbol{\theta}}$ minimising the main objective function f_1. This search can be performed by any black-box optimisation algorithm; our current implementation supports random search, greedy local search, CMA-ES [278, 279] and gradient descent (using finite-difference gradient approximation). Although the objective may vary per domain, we used the proportional absolute difference between the observation $\mathbf{x}$ and the simulated output $M(\boldsymbol{\theta})$, to avoid observed variables with different ranges from dominating the loss function:

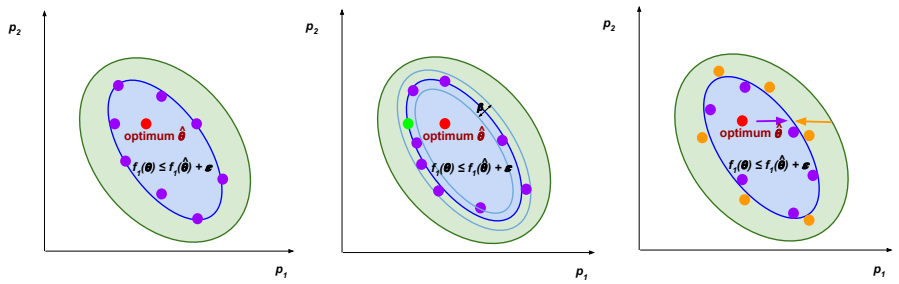
$$f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big)=\frac{1}{d}\cdot\sum_{j=0}^{d}\frac{|x_j-M(\theta)_j|}{|x_j|} \tag{6.10}$$

Here $M(\boldsymbol{\theta})_j$ is the jth element of the model output (simulated observations) for a sampled configuration $\boldsymbol{\theta}$. The optimum $\hat{\boldsymbol{\theta}}$ can now be searched for as the solution $\boldsymbol{\theta}$ minimising f_1:

$$\hat{\boldsymbol{\theta}}\in\underset{\boldsymbol{\theta}\in D_P}{\operatorname{argmin}}\, f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \tag{6.11}$$

Finding the optimum of a function through black-box optimisation can become a challenging and computationally intensive problem in higher dimensions. To alleviate this problem, we take advantage of the special characteristics of model inversion. Since forward simulation models are available, it is possible to follow a 'hybrid modeling' approach (see, e.g., Verrelst et al. [35], Binh et al. [45] and Ranghetti et al. [55]) to warm-start the optimisation algorithm. In hybrid modeling, a machine learning model is trained on a look-up table (LUT) of simulated data to predict the original inputs from the simulated outputs. Since these models will have their own inaccuracies, we opted to use their output to warm-start the optimisation for step 1 in a part of the search space that is likely closer to the optimum than a random- or mean initialisation would be.

Step 1 will converge to a single (generally global, depending on the landscape and the choice of optimisation algorithm) optimum in the loss landscape of f_1. If the loss landscape is known to be multimodal and globally non-convex, there may be solutions of similar quality to the identified $\hat{\boldsymbol{\theta}}$ in other parts of the target variable space, with their own ϵ-manifold. In this case, it is possible to repeat the three steps for additional optima to approximate the ϵ-manifold set. After obtaining the ϵ-manifold and adding these points to the ϵ-manifold set eMS, we exclude those points in subsequent searches in lines 5, 7, 15 and 22 of Algorithm 6.1. This prevents the algorithm from entering parts of the search space that are already



(a) Conv-eMMI (b) Seq-eMMI (c) Dual-eMMI

Figure 6.4: Illustrations of the sampling strategies of the three optimisation-based variants of eMMI (U-eMMI would simply uniformly sample the space) in an abstracted 2-dimensional loss landscape. The green shaded area represents the parameter space D_P, the red dot represents the optimum $\hat{\boldsymbol{\theta}}$, and the blue shaded area represents the ϵ-manifold with its blue border representing the ϵ-boundary. In Conv-eMMI (Figure 6.4a), the points (purple dots) are optimising for diversity, constrained to not exceed the ϵ-boundary. In Seq-eMMI (Figure 6.4b), the method first finds any point on the ϵ-boundary (the green dot), after which it uses diversity optimisation (purple dots), constrained to not deviate from the ϵ-boundary further than a threshold controlled by a hyperparameter β. In Dual-eMMI (Figure 6.4c), half of the population (purple dots) is maximising its diversity as well as the distance from $\hat{\boldsymbol{\theta}}$, constrained to not move outside the ϵ-boundary, while the other half of the population (orange dots) is maximising diversity and minimising its distance to $\hat{\boldsymbol{\theta}}$, constrained to not move inside the ϵ-boundary.

part of an ϵ-manifold. By Definition 6.3, any point $\boldsymbol{\theta}$ already in an ϵ-manifold $eM_j \in eMS$ could not be contained in any other ϵ-manifold eM_j, enabling the deletion of such points from the search space. By Theorom 6.1, if the objective function value $l^\epsilon = f_1(M(\hat{\boldsymbol{\theta}}_t), \mathbf{x})$ for the optimum $\hat{\boldsymbol{\theta}}_t$ identified at the tth iteration is greater than the ϵ-loss $f_1(M(\hat{\boldsymbol{\theta}}_0), \mathbf{x}) + \epsilon$ for the first optimum $\hat{\boldsymbol{\theta}}_0$, we consider all optima and their ϵ-manifolds that should be within the ϵ-manifold set to have been found.

We note that there may be more efficient solutions for multimodal globally non-convex landscapes possible, especially when there are many viable local optima, by using optimisation algorithms directly converging to multiple local optima in step 1, instead of iterating the entire algorithm. However, in this article, we focus primarily on unimodal and multimodal globally convex landscapes; further extensions to improve the efficiency of eMMI for multimodal globally non-convex landscapes are beyond the scope of this work.

6.5.3. STEP 2: FINDING A DIVERSE SET OF SOLUTIONS AROUND THE ϵ-BOUNDARY

For step 2, we perform diversity optimisation to efficiently obtain a set of samples, usually along the ϵ-boundary. We sample along the ϵ-boundary, because this will allow us in step 3 (see Section 6.5.4) to either directly approximate the ϵ-manifold from the final solution set, or take advantage of the efficient sampling strategy to greatly reduce the number of samples required for training a classification approach.

There are four different variants of eMMI, which differ mainly in their sampling strategy in step 2 (uniform without heuristics, constrained diversity using Equation 6.13, along the ϵ-boundary using Equation 6.14, and mutually opposite objectives and constraints using Equations 6.15 and 6.16). The selection of the appropriate eMMI-variant for a given problem can be automated using hyperparameter optimisation (HPO). The first and simplest variant of our method, U-eMMI, does not leverage any additional optimisation or heuristics, and instead uniformly samples the target variable space (sample size B_2). This variant may have advantages over the other variants in low-dimensional problem settings with many disjoint ϵ-manifolds, as it does not attempt to exploit the locality of viable solutions, but scales poorly to high-dimensional problems, where it would be strongly affected by the curse of dimensionality (requiring an exponentially growing number of samples).

The remaining variants of eMMI are founded on diversity optimisation techniques. The appeal of diversity optimisation is that solutions push each other to the edges of the constrained manifold, where distances are larger, while also maintaining maximum distance to each other to span the entire manifold. We illustrate the ideas of these three variants via the examples in Figure 6.4. If we denote $\boldsymbol{\Theta}$ as a population of points $\boldsymbol{\theta}$ obtained in different ways by the different eMMI variants, every individual $\boldsymbol{\theta}$ in the population $\boldsymbol{\Theta}$ can be thought of as a single point moving through the search space. Based on this population, a new objective function, f_2 (line 23 in Algorithm 6.1), can then be used to optimise for diversity within $\boldsymbol{\Theta}$.

The f_2 function, therefore, requires a metric to quantify diversity within a population. It can be desirable to keep control over how many neighbours to consider when computing the diversity metric value. Therefore, when computing the diversity of a new candidate solution $\boldsymbol{\theta}$, we sort $\boldsymbol{\Theta}$ based on the distance of its elements to $\boldsymbol{\theta}$. If j indexes a target variable as the jth element of the vector $\boldsymbol{\theta}$, the generic form of our diversity term $div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta})$ can be written as:

$$div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}) = \frac{1}{k} \cdot \sum_{s=0}^{k} \frac{1}{|P|} \cdot \sum_{j=1}^{|P|} |\boldsymbol{\Theta}_j^s - \boldsymbol{\theta}_j| \tag{6.12}$$

Here, we select the k-nearest neighbours to $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$, indexed by s. For example, $\boldsymbol{\Theta}_3^2$ would refer to the third variable of the second-closest configuration vector $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$, after being sorted. This results in a diversity scalar term representing the average distance between the values of a new solution $\boldsymbol{\theta}$ and its k-nearest neighbours in $|P|$-dimensional space.

In the following, we will describe the different heuristics enabled by Equation 6.12 to efficiently explore the search space for step 2 of the eMMI algorithm.

CONV-EMMI

In the first variant of our method, we use diversity optimization to push the solutions in the population toward the ϵ-boundary, maximizing diversity to split the population as equally as possible along the ϵ-boundary, forming an outline of the boundary through its final solution set.

In the visual example of Figure 6.4a, the solutions $\boldsymbol{\theta}$ of the population Θ are visualised as purple dots, spread around the ϵ-boundary (though some later iterations may start placing points to the centre of the ϵ-manifold, once the diversity pressure from points on the ϵ-boundary becomes stronger). To achieve this, we perform an iterative constrained diversity optimisation procedure for n_{iter} iterations. The total budget for this step, B_2, is split evenly between the n_{iter} iterations (resulting in an individual budget of $\frac{B_2}{n_{iter}}$). In each iteration of this variant, the algorithm searches for a new solution that maximises a new objective function, f_2, which quantifies the diversity of a candidate solution given the current set of solutions $\boldsymbol{\Theta}$ obtained from previous iterations. Motivated by how diversity is measured in quality diversity (QD) evolutionary algorithms [280], we can define the new objective f_2 (used in line 23 of Algorithm 6.1) through the diversity term from Equation 6.12, and add a constraint to achieve the desired behaviour. In Conv-eMMI, the search is constrained to not exceed the ϵ-boundary, only allowing solutions $\boldsymbol{\theta}$ within a distance of ϵ from the f_1 value of the optimum $\hat{\boldsymbol{\theta}}$. This can be formalised as:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_2(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}) \\ \text{subject to} \quad & f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) + \epsilon \end{aligned} \tag{6.13}$$

The population $\boldsymbol{\Theta}$ is initialised to the optimum found in step 1, encouraging initial solutions to move to the points in the ϵ-manifold at the furthest distance from the optimum, and each individual in the population updates sequentially. By optimising for diversity while constraining the points not to exceed the ϵ-boundary, the final set of points $\boldsymbol{\Theta}$ will create an outline of the shape of the ϵ-manifold (line 30 in Algorithm 6.1).

Conv-eMMI is conceptually intuitive, can easily integrate with arbitrary optimisation frameworks because every iteration essentially searches for a single new optimum in a new (f_2) loss landscape, and does not rely on an additional approximation step that may introduce inaccuracies to the algorithm. On the other hand, this variant will 'waste' some computation on filling the space between the optimum and the ϵ-boundary, making it less efficient for large ϵ-manifolds. Moreover, when representing the ϵ-manifold in step 3, Conv-eMMI can also only use the convex hull-based approach described in Section 6.5.4, because its optimisation only samples points within the manifold. This can limit its applicability in non-convex use cases, while the computation of the convex hull can also be impossible for some solution sets, or become intractable in higher dimensions.

SEQ-EMMI

The next two variants of our proposed method approximate the ϵ-manifold using classifiers, trained such that their decision boundary corresponds to the ϵ-boundary, as opposed to the convex hull of the solution set. Although the details of this procedure will be discussed for step 3 in Section 6.5.4, the efficient training of such a classifier requires a change in the sampling approach for step 2, with a new focus on sampling points that contribute most toward training such a classifier. The history of function evaluations for the points sampled during the optimisation procedure can later form a training set for a classifier.

The intuition behind the second variant of our proposed method, Seq-eMMI, is that it aims to sample along the ϵ-boundary. To do this, it first identifies any point on the ϵ-boundary, and pushes its solutions along the boundary, at a specified tolerance level.

Visually, in Figure 6.4b, after finding a point on the ϵ-boundary (the green dot), the solutions in the population move along the ϵ-boundary (the thick blue line) within some tolerance level (indicated by the thin blue lines), until the budget is exhausted. To this end, it uses a synchronous population-based update rule for its diversity optimisation, rather than the asynchronous iterated approach employed by Conv-eMMI. In Seq-eMMI, the size of this population n_{pop} is analogous to the number of iterations n_{iter} in Conv-eMMI, and likewise, the individual budget for every individual in the population is $\frac{B_2}{n_{pop}}$. In this sequential (Seq) version of eMMI, the optimisation budget of one invdividual in the population is dedicated to finding any point $\boldsymbol{\theta}^\epsilon \in \text{argmin}_{\boldsymbol{\theta}} \left[|f_1(M(\boldsymbol{\theta}), \mathbf{x}) - l^\epsilon| \right]$ on the ϵ-boundary, which will usually be relatively close to the optimum (lines 13-17 in Algorithm 6.1).

Once $\boldsymbol{\theta}^\epsilon$ has been found, we initialise the rest of the population as copies of $\boldsymbol{\theta}^\epsilon$, and optimise for diversity within the population $\boldsymbol{\Theta}$, while constraining the values to remain close to the ϵ-boundary (parameterised by a tolerance hyperparameter

β). The new objective function f_2 (used in line 23 of Algorithm 6.1) for this variant then becomes:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_2(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = div(\boldsymbol{\Theta}, \mathbf{x}, \boldsymbol{\theta}) \\ \text{subject to} \quad & l^{\epsilon} - \beta \leq f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq l^{\epsilon} + \beta \end{aligned} \tag{6.14}$$

In this variant, the set of solutions $\boldsymbol{\Theta}$ contains all the current positions of the population, and the final set of points, like Conv-eMMI, shows an outline of the ϵ-boundary (but its function evaluations will contain both points barely inside the ϵ-manifold and points barely outside of it). Seq-eMMI will not waste function evaluations on sampling far away from the ϵ-boundary, or be pushed away from the boundary by an overpowering push from the diversity objective. It also supports the use of classifiers as the ϵ-manifold representation approach for step 3 as described in Section 6.5.4. However, it can be an inefficient way of exploring the ϵ-boundary, because its steps must be made in a precise direction that does not violate its constraints, and because only the outer-most points in the population can have a large impact on the diversity objective. For example, when imagining a population spread out over a line, only the two outermost individuals could increase diversity by moving away from the rest of the population, while diversity improvements in one direction by individuals in the centre of the line would come at the cost of a reduction of diversity in the other direction, thereby potentially wasting computation on these function evaluations.

DUAL-EMMI

The intuition behind the last variant of our method is that the population is split into two 'competing' halves, with both sub-populations pushing from opposite sides against, but unable to exceed, the ϵ-boundary, thus achieving a balanced data set of in-samples and out-samples in the process.

In Figure 6.4c, the purple dots are trying to 'push' the ϵ-boundary outward from their origin point of the optimum, while the orange dots are trying to push the ϵ-boundary inward from their origin point outside of the ϵ-manifold. Like Seq-eMMI, Dual-eMMI uses a population-based approach. The population is split into sub-population $\boldsymbol{\Theta}_a$, contained inside the ϵ-manifold (initialised to the optimum), and sub-population $\boldsymbol{\Theta}_b$, situated outside of it (initialised randomly). The individuals in $\boldsymbol{\Theta}_a$ will try to maximise their distance from the optimum $\hat{\boldsymbol{\theta}}$, as well as the diversity within the population $\boldsymbol{\Theta}_a$, constrained to not move outside of the ϵ-boundary. Meanwhile, individuals in $\boldsymbol{\Theta}_b$ will still optimise for diversity within their sub-population $\boldsymbol{\Theta}_b$, but will also aim to minimise the distance to the optimum $\hat{\boldsymbol{\theta}}$, constrained to not move within the ϵ-boundary. The balance between the

diversity and distance objectives can be tuned via the hyperparameter α, following a scalarisation approach to multi-objective optimisation [281, 282]. Formally, individuals in $\mathbf{\Theta}_a$ solve:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_{2_a}(\mathbf{\Theta}_a, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = \alpha \cdot div(\mathbf{\Theta}_a, \mathbf{x}, \boldsymbol{\theta}) + (1-\alpha) \cdot \sum_{j=0}^{|P|} |\boldsymbol{\theta}_j - \hat{\boldsymbol{\theta}}_j| \\ \text{subject to} \quad & f_1(M(\boldsymbol{\theta}), \mathbf{x}) \leq l^{\epsilon} \end{aligned} \tag{6.15}$$

Here l^{ϵ} refers to the ϵ-loss. Meanwhile, individuals in $\mathbf{\Theta}_b$ solve:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximise}} \quad & f_{2_b}(\mathbf{\Theta}_b, \mathbf{x}, \boldsymbol{\theta}, \epsilon) = \alpha \cdot div(\mathbf{\Theta}_b, \mathbf{x}, \boldsymbol{\theta}) - (1-\alpha) \cdot \sum_{j=0}^{|P|} |\boldsymbol{\theta}_j - \hat{\boldsymbol{\theta}}_j| \\ \text{subject to} \quad & f_1(M(\boldsymbol{\theta}), \mathbf{x}) \geq l^{\epsilon} \end{aligned} \tag{6.16}$$

Dual-eMMI is more efficient than Seq-eMMI, obtains more diverse samples away from the ϵ-boundary and ensures that there is a reasonable balance between in-samples and out-samples. On the other hand, its two objectives must be combined, with benefits to one objective potentially coming at the expense of the other. Unlike in typical multi-objective settings, where a Pareto-front of non-dominated solutions is the desired outcome, our reliance on diversity and need for training data set samples made this a non-trivial extension to add; however, future work may look further into the use of other types of multi-objective optimisation approaches.

6.5.4. STEP 3: APPROXIMATING THE ϵ-MANIFOLD

Finally, in step 3, we use the points sampled in step 2 to approximate the ϵ-manifold. We required methods applicable to problems of arbitrary dimensionality, because, unlike the two-dimensional examples used in our visualisations, these boundaries cannot be trivially drawn using, e.g., contour lines. In our current approach, there are two options for extracting a representation of the ϵ-manifold from these points.

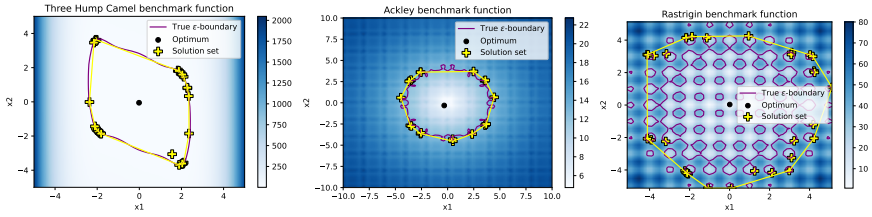
The first approach is to compute the convex hull of the final set of points $\mathbf{\Theta}$. The optimisation procedure would have encouraged points to span the ϵ-boundary, making the convex hull of the resulting point cloud an intuitive representation of the ϵ-manifold that follows directly from the final points without further approximation steps. We used Delaunay triangulation (see, e.g., Lee and Schachter [283]) to check whether a new point is contained in this convex hull. This approach assumes convexity and may fail to compute a convex hull if the assumption is violated, may be costly in higher dimensions, and does not take advantage of the many points sampled during the optimisation procedure.

The second approach is to store the results of the function evaluations for the points sampled during step 2 in a data set H, consisting of the points H_x sampled during step 2 and their labels H_y indicating whether their loss is lower or higher than the ϵ-loss, and train a classifier on these points. Intuitively, the ϵ-boundary can be viewed as an ideal decision boundary for a binary classification problem, separating points inside the ϵ-manifold (positive labels) from points outside of it (negative labels). Therefore, the ϵ-boundary can be approximated by the decision boundary of a binary classifier trained on positive and negative examples in H, that had already been sampled during optimisation. The ϵ-manifold them becomes the space bounded by this decision boundary, or, if finite sets are preferred, the points in H themselves can be used.

The advantages of using a classifier include reduced assumptions on the loss landscape (as different types of classifier can model different types of decision boundaries), improved computational efficiency, and a convenient method of checking whether new points are in the ϵ-manifold. On the other hand, representing ϵ-manifolds as a classifier may be less intuitive than using a convex hull. There can also be technical downsides to such an approach: classifiers usually perform best when presented with balanced data (roughly equal numbers of in- and out-samples), some classifiers may be unable to extrapolate beyond the neighbourhood of the sampled training points (requiring training samples spread throughout $\mathcal{D}_P$), and using classifiers introduces an additional step of imperfect approximation.

To make our method more robust to two specific scenarios it may be weak to, we added two additional options to eMMI. First, we allow users to filter points in low-density regions out of H for probabilistic simulators or loss functions. The rationale for this option is that, in probabilistic cases, parts of the search space with a low sampling density may be misrepresented by one or a few points that returned an unlikely, non-representative result for a single point. By filtering points based on their density, this problem can be alleviated (if computational resources are available for it, it is also possible to sample every point multiple times in these cases).

The second option is to allow users to spend a proportion of the budget of step 2 (B_2) on pure exploration, as in U-eMMI; we refer to this proportion as B_{exp}. Depending on the type of classifier, there may be a benefit of H containing points that sparsely cover the entire search space, to enable extrapolation beyond the area in the direct vicinity of the decision boundary. Both options can be controlled through hyperparameters, which can be tuned automatically through hyperparameter optimisation.



(a) Three hump camel function (b) Ackley function (c) Rastrigin function

Figure 6.5: Examples of Conv-eMMI applied to two-dimensional non-convex optimisation benchmark functions with increasing difficulty. The three hump camel function (Figure 6.5a) is globally non-convex, but locally convex around its three local optima with no sub-maxima within the neighbourhood of the optima. In this case, the optima were close together, so all optima were contained within the ϵ-manifold. The Ackley function (Figure 6.5b) is globally non-convex and locally convex for the level of ϵ, but with multiple sub-optima between the optimum and the ϵ-boundary. In this case, the sub-maxima did not exceed the ϵ-loss, allowing eMMI to effectively extract most of the ϵ-manifold, but with some lower detail at the border. The Rastrigin function (Figure 6.5c) is globally and locally non-convex, with the sub-maxima between the optimum and the ϵ-boundary exceeding the ϵ-loss. In this case, the ϵ-manifold extracted by Conv-eMMI contained many false positives for the parts of the parameter space where sub-maxima exceeded the ϵ-loss, marking the limits of Conv-eMMI in its current form.

6.5.5. LIMITATIONS OF THE CURRENT HEURISTICS

The current heuristics for eMMI (excluding U-eMMI, which is generally applicable but scales poorly) are applicable to loss landscapes where viable solutions are centred around an optimum. This makes our method directly applicable to unimodal, globally convex inversion loss landscapes. Even if a full loss landscape is globally non-convex, our method is still applicable if there is local convexity around the optimum. We describe how the method could be applied to this type of multimodal, locally convex landscape in Section 6.5.2.

Within local convexity, we can further differentiate between monotonic locally convex landscapes and non-monotonic locally convex landscapes. If the landscape is (locally) monotonic, it is also locally convex: if its loss values only strictly increase or stay equal with distance to the optimum, it is impossible for another optimum to exist within this local neighbourhood. If the local landscape is non-monotonic, there may be 'sub'-minima and maxima within the local neighbourhood of the optimum. If these sub-maxima do not exceed the ϵ-loss, eMMI can still be applied to such non-monotonic local landscapes (because even all of its highest values lie between the minimum and the ϵ-loss).

Finally, the current heuristics of eMMI would be less well-suited to landscapes where the sub-maxima of a non-monotonic local landscape exceed the ϵ-loss, in

which case the ϵ-manifold could contain false positives, while the approximated ϵ-boundary is likely to have stopped expanding too early. For this type of problem, we recommend using U-eMMI to avoid the assumption of local convexity, although the lack of efficient sampling heuristics from the other variants may result in poor scalability.

We have visualised example runs of Conv-eMMI (for details, see Section 6.5.3) for three non-convex benchmark functions in Figure 6.5: the three-hump-camel function, which is globally non-convex but monotonically locally convex, the Ackley function [284], which is globally convex, but locally non-monotonically convex, and the Rastrigin function [285], which is globally and locally non-monotonically non-convex. As the Figure shows, Conv-eMMI approximated the ϵ-manifold well for the three hump camel function, performed reasonably well (but showed some inaccuracies along the ϵ-boundary) for the Ackley function, but contained many false positives for the sub-maxima in the Rastrigin function. Therefore, this type of landscape can be considered the limit of the type of ϵ-manifold that can efficiently be approximated by eMMI with the current heuristics (although U-eMMI could still approximate it inefficiently, or other, novel heuristics may perform better).

6.6. EXPERIMENTS

In the following, we explain the details of our computational experiments aimed at answering the following chapter research questions (CRQs):

1. What is the effectiveness of ϵ-manifolds at representing the set of viable solutions $\mathcal{D}_P^{'}$ compared to uncertainty quantification approaches?

2. How does the ϵ-manifold approximation performance of eMMI compare to statistical baseline methods in terms of classification performance on validation points spread around the target variable space?

3. How large does the eMMI budget need to be to converge to its best performance, and how is this impacted by the dimensionality of the problem?

We will first expand on the research questions and the experimental setup we used to answer them in Section 6.6.1, after which we will introduce the simulators used in our experiments in Section 6.6.2, and the baseline methods in Section 6.6.3.

6.6.1. EXPERIMENTAL SETUP

We will explain the motivation behind the research questions, and the experiments we created to answer them, individually per research question.

CRQ1: EFFECTIVENESS OF ϵ-MANIFOLDS

For this research question, we were interested whether a 'perfect' ϵ-manifold would result in a performance increase over generalising existing statistical uncertainty quantification techniques to this purpose. If these ideal ϵ-manifolds significantly outperform uncertainty quantification techniques, it suggests that approximating them is a worthwhile exercise. Conversely, if they do not outperform these methods, it indicates that the assumptions underlying ϵ-manifolds (loss-dependent shifts and loss-dependent origins) may not hold. This would imply that a strong approximation performance by eMMI might not necessarily lead to more robust analyses.

To test for this, we introduce two 'oracle' style methods. These methods enable a direct comparison between ϵ-manifolds and confidence intervals, independent of their approximation efficacy. Given a noisy observation vector $\mathbf{x}$, both methods return a set of potential solutions $\mathcal{V}$ for this model inversion problem. Intuitively, the solution set of a perfect method should always contain the optimum for the (unavailable) noise-free observations $\mathbf{x}^+$, while excluding any solutions that could not have been the true solution if the random noise had been different. In this experiment, we iterated over instances, and performed classification for two points: first, the true values $\boldsymbol{\theta}^+ \approx \hat{\boldsymbol{\theta}}^+$, which should always be contained in the viable solution set, and second, a negative example point that should be excluded.

The first oracle-style method concerns ϵ-manifolds. Classifying the noise-free optimum and any negative example point is straightforward: the pre-computed noisy optimum $\hat{\boldsymbol{\theta}}$ and its loss function value are already known to the oracle method. Therefore, for any ϵ, we can compute the ϵ-loss as $l^\epsilon = f_1(M(\hat{\boldsymbol{\theta}}), \mathbf{x}) + \epsilon$. We classify new points $\boldsymbol{\theta}$ by computing their loss function value $f_1(M(\boldsymbol{\theta}), \mathbf{x})$ and comparing it to the ϵ-loss l^ϵ following Equation 6.6. If the loss-dependent origins assumption holds, we would expect this oracle-based method to classify these points nearly perfectly, bounded only by the suitability of ϵ to the current instance and the accuracy of the optimum $\hat{\boldsymbol{\theta}}$. If the performance is weaker, it could indicate either a large variability of the appropriate setting for ϵ between instances, or that the loss-dependent origins assumption is less strongly satisfied. In other words, a lower score suggests that the loss landscape for a noisy observation corresponds less closely to the probability of being the original, noise-free optimum.

As a baseline, we compared our approach to an oracle-based uncertainty quantification approach using a Gaussian distribution parameterised by mean μ and standard deviation σ. We perform classification using the confidence interval $[\mu - 2\sigma, \mu + 2\sigma]$ as described in Section 6.6.3. The parameters μ and σ were set using oracle knowledge, with the mean $\mu = \boldsymbol{\theta}^+$, and the standard deviation σ derived directly from the evaluation points labelled as being in the ϵ-manifold (see Section 6.6.2 for details). This setup, using unknown true values and computing sample statistics directly from the validation points used to evaluate perfor-

mance, ensured that the Gaussian distribution parameterisation had the strongest possible performance. We expected this baseline method to perform well on the simulation-based inference tasks that are based on distribution parameterisation, but perform worse than ϵ-manifolds for complex loss landscapes, such as those of physical models or dynamical systems.

CRQ2: EMMI PERFORMANCE FOR APPROXIMATING THE ϵ-MANIFOLD

Having shown that ϵ-manifolds can entail substantial performance improvements (CRQ1), the next step is to empirically validate the performance of our proposed approximation method, eMMI.

In all experiments, we set the total function evaluation budget B available to eMMI at 20 000. For the baseline methods (see Section 6.6.3), ABCSMC shared this budget, while the uncertainty quantification baselines do not rely on sampling or optimisation at inference time. Instead, these baseline methods were trained on 20 000 training instances. Prior to running the methods on our main experiments, we performed hyperparameter optimisation for all methods using SMAC3 [286] for 48 hours to ensure that the methods were properly configured. For a single hyperparameter configuration, we evaluated the performance in batches of 10 instances to make the procedure more robust to noisy simulations and f_1 function evaluations.

We measured the performance of the different methods based on the classification performance (notably accuracy) on a balanced set of validation points, as described in Section 6.6.2. A high classification performance indicates that a large proportion of the validation points were correctly classified to be either inside or outside of an oracle ϵ-manifold (as used in CRQ1), thereby indicating effective approximation.

CRQ3: EMMI BUDGET AND SCALABILITY

The eMMI method requires an optimisation procedure to identify the optimum $\hat{\boldsymbol{\theta}}$ (step 1), after which it must spend more function evaluations to sample points around the search space (step 2). As a result, running eMMI will often be more computationally intensive than running the baseline methods at inference time, and the applications for eMMI may differ from those of uncertainty quantification (see Section 6.4.3). For CRQ3, we were interested in how much budget B eMMI needs to converge to a strong classification performance, and how this budget is affected by the dimensionality of the problem.

To test this, we ran eMMI on 50 instances for all versions of the multi-dimensional linear regression simulator (see Section 6.6.2), which we configured for 2, 5, 10, 20 and 50 dimensions. We ran eMMI with a budget of 50 000 function evaluations, and trained a classifier on subsets of the sampled points in steps of 500 additional

evaluations (i.e., $0-500$, $0-1000$, $0-1500$, etc). This simulated different step 2 sampling budgets B_2. We then plotted the average classification performance over the instances as a function of the budget, for all 5 dimensionalities. This plot will show how many function evaluations are needed to converge to a stable performance, as well as showing whether larger-dimensional problems require a larger budget to converge.

GENERAL EXPERIMENTAL WORKFLOW

For all experiments described above, we first needed a suitable value for ϵ for every simulator, based on the expected optimum shift. Although the desired level of confidence is up to the user, we designed the following process to set the value of ϵ to correspond to the 95% confidence intervals of UQ, thereby allowing a comparison between the two approaches. To automatically determine this ϵ value, we loaded the original true target variable configuration $\boldsymbol{\theta}^+$ for the validation instances, along with the pre-computed optimum $\hat{\boldsymbol{\theta}}$ for the noisy observations $\mathbf{x}$ of the same instance, and computed the increase in loss value between the simulated output for the true configuration $M(\boldsymbol{\theta}^+)$ and the simulated output for the pre-computed optimum $M(\hat{\boldsymbol{\theta}})$. Finally, we derived the appropriate value for ϵ as the 95th percentile of the differences in f_1 values between the pre-computed optima and true configurations. Here we note that, despite this statistical approach to setting ϵ, the relevant statistics are still loss function values, and not distances in the target variable space.

Every experiment was run on a compute cluster with an Intel Xeon E5-2683 v4 CPU and 128GB RAM per node, of which we reserved 16GB per experiment.

In the following, we will introduce the data sets (simulators), baselines and performance metrics used in our experiments.

6.6.2. SIMULATION MODELS

Our workflow for the simulators is highly modular, and adding new simulators to our implementation consists only of adding a description of its input parameters (target variables) and a function call for the forward simulation, allowing users to incorporate their own simulators if they wish to. We performed empirical experiments on the following 7 simulator models representing physical models (Earth science), dynamical systems, simulation-based inference and machine learning problem settings. We will introduce the simulation models grouped by the type of model.

Physical models (1 model, 2 versions). As our representative physical model we used PROSAIL, [190, 43, 44] a radiative transfer model (RTM) that is extensively used in real-world applications estimating vegetation properties from remotely

sensed spectral data. We selected the 4 most impactful variables (leaf area index (LAI), chlorophyll $a+b$ content, average leaf angle, and dry matter content) of this model to perform our experiments, while fixing the remaining variables to static values. We also included experiments with only two of the most important variables, to show the impact of the dimensionality of the problem for physical models, and enable the visualisations of Figures 6.1b and 6.3.

Dynamical systems (2 models). We used the implementation of dynamical systems simulation models provided by DAPPER [287], from which we selected the Two Pendulums (TP) and Lorenz63 models. In dynamical systems, the parameters of the simulator represent the state of some system, that is updated over multiple time steps. The observations, in this case, consists of the state of the system after updating for a user-defined number of time-steps. Inverting this type of model is often ill-posed, and the system is often considered chaotic (dynamic systems with a strong sensitivity to the initial conditions) [288]. Finding a set of potential solutions through uncertainty quantification in such ill-posed landscapes would be challenging, because it violates the assumptions of the distributions, while the current versions of eMMI will likewise not be well suited to these landscapes due to the convexity assumptions (though ϵ-manifolds themselves may still be highly effective, if they are approximated well). For both simulators, we limited our inversion to a single time step, as the compound of uncertainty over multiple time steps of inversion may quickly get computationally infeasible.

Simulation-based inference (2 models). We used the Gaussian mixture (GM) [251] and Two Moons (TM) tasks from the simulation-based inference benchmark by Lueckmann et al. [219] to evaluate our proposed method. Both were selected for their tractable execution time, with GM representing a relatively simple problem conforming to the assumptions made by eMMI and the baseline methods, while TM represented a more complex, bimodal and often non-convex landscape.

Machine learning parameterisation (1 model, 5 versions). As explained in Section 6.1, machine learning training is a special kind of model inversion problem, where its parameterisation (usually model weights) must be set in such a way that, when combined with a data set of features, its simulated output (predictions) has a minimal distance to the observations (ground truth data). Therefore, although the predictions and their quality-of-fit are generally the variables of interest in machine learning, the weight parameters are the variables being inferred during the training procedure. Although it is unlikely that ϵ-manifolds can be computed for deep neural networks with numbers of parameters orders of magnitude higher than typical problem cases, we consider the analyses on traditional machine learning algorithms enabled by ϵ-manifolds (for example, certain types of robustness analysis, loss landscape analysis, ill-posedness analysis, and data set difficulty) to have a high potential for impact.

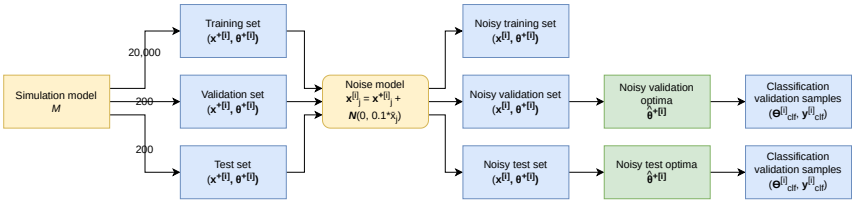


Figure 6.6: Visualisation of our data generation pipeline.

The simulation model we used consisted of an d-dimensional linear regression (LR) model, with $d+1$ trainable parameters (the weights and a bias term). We included problems with a dimensionality d of 2, 5, 10, 20, and 50 weights. For every dimensionality, we pre-generated d independent feature vectors for 100 examples (we refer to the instances of the machine learning task as examples to avoid confusion with the model inversion instances), which we kept constant for all instances. We simulated ground truth data using randomly generated model parameterisations, after which we perturbed the simulated ground truth to emulate noisy training data – a realistic scenario, for which ϵ-manifolds could be used in the same manner as for the other model inversion tasks.

EXPERIMENTAL DATA GENERATION

We visualised the general workflow for data generation in Figure 6.6. We have organised the data generation steps as *phases*. The data required by our experiments consisted of, for every instance of every simulator, i) a true target variable configuration $\boldsymbol{\theta}^+$ and its simulated output (observations) $\mathbf{x}^+ = M(\boldsymbol{\theta}^+)$, ii) a noisy version $\mathbf{x}$ of the observations $\mathbf{x}^+$ and the pre-computed true noisy optimum $\hat{\boldsymbol{\theta}}$, and iii), a balanced set of validation points $\boldsymbol{\theta}_{clf}$ with associated labels y_{clf}, such that $y_{clf} = 1$ if and only if $f_1\big(M(\boldsymbol{\theta}),\mathbf{x}\big) \leq f_1\big(M(\hat{\boldsymbol{\theta}}),\mathbf{x}\big) + \epsilon$, and $y_{clf} = 0$ otherwise.

Phase 1: generating noise-free and noisy data. For every simulator M, we generated a training-, validation- and test set. The training data set was used by eMMI (if warm-starting the optimisation) and the uncertainty quantification baselines (for details, see Section 6.6.3), and consisted of 20000 instances. The validation- and test sets contained 200 instances each. To create these data sets, we generated individual instances i by randomly sampling a true input vector $\boldsymbol{\theta}^{+[i]}$ from the target variable space $\mathcal{D}_P$, for which we created a noise-free simulated observation vector $\mathbf{x}^{+[i]}$ by performing a simulation on $\boldsymbol{\theta}^{[i]}$: $\mathbf{x}^{+[i]} = M(\boldsymbol{\theta}^{+[i]})$. We then saved the pair of noise-free observations and true configurations $(\mathbf{x}^{+[i]}, \boldsymbol{\theta}^{+[i]})$ for all instances i. After this, we created a noisy version $\mathbf{x}^{[i]}$ of the noise-free observations $\mathbf{x}^{+[i]}$ by adding 10% additive Gaussian noise to the elements j of $\mathbf{x}^{+[i]}$:

$\mathbf{x}_j^{[i]} = \mathbf{x}_j^{+[i]} + \mathcal{N}(0, 0.1 \cdot \overline{x}_j)$. Here $\overline{x}_j$ is the mean value of target variable P_j in the sample of noise-free observation instances.

Phase 2: pre-computing the noisy optimum. Every instance in the validation- and test sets required a 'ground truth' global optimum $\hat{\boldsymbol{\theta}}$ to the loss landscape of $f_1\big(M(\boldsymbol{\theta}), \mathbf{x}\big)$, allowing us to compute the true ϵ-loss and evaluate methods later. To this end, we performed 50 000 iterations of black-box optimisation to pre-compute the optimum $\hat{\boldsymbol{\theta}}^{[i]}$ of the instance for the noisy observations $\mathbf{x}^{[i]}$ (using random search with a very high budget to avoid confounding through the choice of optimisation algorithm). Because this budget is much larger than the budget used by eMMI in practice, pre-computing the optima in this manner enabled reliable and efficient evaluation of method performance later, although the high computational cost limited our experiments on simulation models with tractable computational costs. In the interest of conserving computational resources, we did not pre-compute optima for the training set, where the ground truth values $\boldsymbol{\theta}^+$ could be used for training.

Phase 3: generating validation points. In addition to the simulation instances themselves, every instance in the validation- and test sets required a sample of points $\boldsymbol{\Theta}_{clf}^{[i]}$, consisting of individual points $\boldsymbol{\theta}_{clf}^{[i]}$, allowing us to evaluate the classification performance, as explained in Section 6.5.4, for the ϵ-manifold of that instance. These points were associated with the label vector $\mathbf{y}_{clf}^{[i]}$, whose labels denote if the instance is inside or outside of the ϵ-manifold.

To ensure that we obtained balanced validation data sets (especially in higher dimensions), we performed random sampling with a large budget of 100 000 function evaluations, and performed post-hoc rejection sampling to obtain a balanced number of true and false evaluation examples. Computing these evaluation samples, along with the pre-computed optima mentioned above, is highly computationally intensive, and these elements together form the largest computational bottleneck of our experiments. We note that this computational load was used only to ensure a *fair empirical evaluation* of the methods, and it would not be a factor when applying our proposed method or baseline methods to real model inversion problems.

A summary of the data set characteristics can be found in Table 6.1.

6.6.3. BASELINES AND PERFORMANCE METRICS

In our experiments for CRQ1 we compared ϵ-manifolds to an oracle-based uncertainty quantification (UQ) baseline in the form of parameterised Gaussian distributions; we explain this baseline in Sections 6.6.1, since it does not necessarily represent any particular method. In contrast, when evaluating eMMI, we compared the empirical performance of our method to those of concrete baseline methods.

Property	Data split		
	Training set	**Validation set**	**Test set**
# noise-free instances	20 000	200	200
# noisy instances	20 000	200	200
# validation points / instance	0	2 – 100 000	2 – 100 000
contains noisy optimum	✗	✓	✓

Table 6.1: Summary of the data generated for every simulator. Because we used rejection sampling to generate validation points for every instance, the number of points can vary between 2 and 100 000 points.

These methods can be highly effective at their intended purpose of inference and uncertainty quantification; our empirical comparison will, therefore, not evaluate the value of the methods themselves, but rather gauge whether such existing methods can be generalised such that the resulting statistical confidence intervals can be interpreted as an effective approximation of ϵ-manifolds. We compared against the following baselines:

- **Gaussian processes (GP).** We used GPs as an UQ baseline because these are the primary prediction model preferred by domain users of physical models [46, 54], where GPs are preferred in part because of their inherent uncertainty estimation. It is, therefore, useful to test whether the uncertainty estimation of GPs approximates the actual ϵ-manifold of a problem instance.
- **Random forests (RF).** RFs are a frequently used ensemble method, allowing for uncertainty quantification through the standard deviation of the predictions of individual trees following an ensembling approach to uncertainty quantification [262, 263].
- **Bayesian neural networks (BNN).** BNNs represent advances in neural network-based approaches boasting impressive performance. To perform inference with the BNN, we computed the standard deviation of the predictions of the model, which, due to the weights being a distribution, are not deterministic. The model itself was based on the implementation provided by Lee et al. [289], although we automated the selection of the architecture using hyperparameter optimisation.
- **Approximate Bayesian computation – sequential Monte Carlo (ABCSMC) [290].** We used this recent variant of ABC as a representative method for conventional SBI methods. Although SBI methods and ϵ-manifolds are not strictly competitive (see Section 6.2), in principle, the inherent statistical nature of SBI could be interpreted as an equivalent to ϵ-manifolds, making this

a meaningful comparison. Since ABC usually infers a (non-parametric) posterior distribution over the target variables, a 95% confidence interval can be constructed by including points whose weights lie between the 2.5th and 97.5th percentiles.

- **Tabular prior-fitted network (TabPFN)** [291, 292]. TabPFN is a state-of-the-art foundation model for tabular data, able to perform zero-shot inference on tabular data sets, often with a strong performance comparable to specialised models. When performing inference, the model can include percentiles in its predictions, enabling the construction of 95% confidence intervals. To prevent out-of-memory issues, we increased the memory available to this method fourfold to 64GB.

For all machine learning-based baselines we trained (GP, RF, BNN) or fine-tuned (TabPFN) the model on 20 000 simulated training instances (with noise), and we also trained the machine learning model used to warm-start the eMMI optimisation from step 1 (see Section 6.5.2) on this training set. We quantified uncertainty for the UQ methods as the 95% confidence interval, given by two standard deviations σ from the mean unless otherwise specified. This confidence interval matches the confidence interval used to derive ϵ, as described in Section 6.6.1. Given a model inversion instance i, for which the UQ methods predict a mean $\mu^{[i]}$ and a standard deviation $\sigma^{[i]}$, and a sample of pre-computed evaluation points for instance i, the points within the ϵ-manifold should lie within the interval $[\mu^{[i]} - 2\sigma^{[i]}, \mu^{[i]} + 2\sigma^{[i]}]$.

To evaluate the performance of the different methods numerically, we computed the accuracy (suitable because we strictly enforced data set balance using our post-hoc rejection sampling approach) for the different methods on all model inversion problems. We determine the significance of a performance difference using Wilcoxon signed-rank tests at a significance level $\alpha = 0.05$. We note that the empirical results for these experiments are not designed to test the quality of the predictions of these methods themselves, but rather the suitability of the uncertainty quantification components of existing methods for approximating an ϵ-manifold.

6.7. RESULTS

The empirical results presented in the following are organised per chapter research question (CRQ).

Dataset	ϵ-manifold	Uncertainty quantification
PROSAIL	$\mathbf{1.0 \pm 0.0}$	0.84 ± 0.37
PROSAIL 2D	0.68 ± 0.47	0.74 ± 0.44
TP	$\mathbf{0.8 \pm 0.4}$	0.62 ± 0.49
Lorenz63	$\mathbf{0.9 \pm 0.3}$	0.62 ± 0.48
GM	0.86 ± 0.35	0.89 ± 0.31
TM	$\mathbf{0.97 \pm 0.17}$	0.74 ± 0.44
LR	$\mathbf{0.86 \pm 0.34}$	0.76 ± 0.43

Table 6.2: Results for the oracle-based ϵ-manifold validation experiment for RQ2, showing accuracy scores for a classification task where, for every instance, the true values $\boldsymbol{\theta}^+$ had to be predicted along with a negative sample from the validation points. A bold column in the table represents a significantly better result, determined by a Wilcoxon signed-rank test at a significance level $\alpha = 0.05$, where the samples consisted of correct or incorrect classification (thereby enabling the computation of a standard deviation and rank-based comparisons).

6.7.1. CRQ1: EFFECTIVENESS OF ϵ-MANIFOLDS

The results for CRQ1 can be found in Table 6.2. As the table shows, a perfect ϵ-manifold performed significantly better than a perfect Gaussian distribution parameterisation as uncertainty quantification on nearly all tested simulation models, only tying on the Gaussian mixture and PROSAIL 2D simulators (both of which have loss landscapes that adhere relatively well to the assumptions of a Gaussian statistical kernel). This means that an ϵ-manifold for a noisy model inversion instance is more likely to contain the true solution $\boldsymbol{\theta}^+$, without needlessly including infeasible points.

Despite the strong advantage over perfectly parameterised uncertainty quantification, the performance of ϵ-manifolds appears to vary between simulation models, especially between the full version of PROSAIL (perfect scores) and the two-dimensional version of PROSAIL (lowest scores out of the tested simulators). This counterintuitive result implies that higher-dimensional problems are not necessarily more difficult to deal with, in terms of uncertainty quantification, than higher-dimensional manifolds.

In the case of PROSAIL, this behaviour might be explained through domain knowledge by examining the properties of the physical model, in which the leaf area index (LAI) parameter is known to have an impact on the behaviour of other parameters. In lower-dimensional settings, the relative impact of the complexities introduced by the LAI parameter is higher than it is in higher-dimensional settings, where the other model parameters may have a more modest and independent impact on the loss landscape. It is also possible that the appropriate setting of ϵ, which we set constant for all instances, is more variable for low-dimensional

problems than for high-dimensional settings, in which case future work aimed at further improving the heuristic by which ϵ is set might hold significant promise.

The results for the dynamic systems simulators (Two Pendulums and Lorenz63) were quite favourable for ϵ-manifolds compared to UQ. This follows expectations, considering the chaotic nature of these simulators, which a Gaussian distribution may be ill-suited to represent. The results for the probabilistic simulation-based inference simulators (GM and TM) also both favoured ϵ-manifolds over uncertainty quantification, as did those for the linear regression (LR) machine learning model parameterisation task.

We found that the lower accuracy of uncertainty quantification often stemmed from a combination of near-perfect precision with low recall or, more rarely, high recall with poor precision (these results can be found in Appendix B.2). This frequent overconfidence may be explained in part by a low prior probability of some viable solutions, where a solution that is far removed from the point prediction (and therefore has a low probability) would be considered unlikely, regardless of whether there is a large difference in the loss function value. Therefore, these results support the intuition we described in Section 6.4.3 on why statistical uncertainty quantification may not always be an appropriate choice for finding viable solution sets. Alternatively, the loss landscape may simply have not adhered to a Gaussian distribution form, such as in the chaotic landscapes of the dynamical systems simulators (as the particularly low accuracy scores for TP and Lorenz63 also indicate).

In conclusion, regarding CRQ1, perfect ϵ-manifolds appear to be more effective than perfect uncertainty quantification for identifying the set of potentially valid solutions in most noisy model inversion problems. This indicates that UQ methods, while highly valuable when applied to their intended purpose, cannot be directly generalised to approximate ϵ-manifolds.

6.7.2. CRQ2: EMMI PERFORMANCE

The results for our experiments validating eMMI as a method to approximate ϵ-manifolds can be found in Tables 6.3 (comparing eMMI to baseline methods) and 6.4 (comparing eMMI variants).

The results in Table 6.3 show that eMMI performed significantly better than the baseline methods we considered on all tested simulators, with a generally high accuracy that only dropped for Lorenz63 (a chaotic system) and TM (a bimodal problem), both of which give rise to loss landscape types for which the current heuristics were not designed. The baseline methods, as might be expected based on the results from Table 6.2, often achieved an accuracy close to 0.5, indicating overconfidence with a low recall (precision and recall results can be found in Ap-

Method	PROSAIL	PROSAIL 2D	TP	Lorenz63	GM	TM	LR
RF	0.54 ± 0.06	0.55 ± 0.08	0.5 ± 0.0	0.5 ± 0.01	0.88 ± 0.12	0.57 ± 0.19	0.5 ± 0.0
GP	0.74 ± 0.1	0.61 ± 0.13	0.51 ± 0.03	0.5 ± 0.0	0.57 ± 0.16	0.5 ± 0.0	0.5 ± 0.0
BNN	0.53 ± 0.09	0.73 ± 0.16	0.51 ± 0.03	0.5 ± 0.0	0.58 ± 0.13	0.5 ± 0.0	0.5 ± 0.0
ABCSMC	0.52 ± 0.16	0.53 ± 0.2	0.52 ± 0.17	0.49 ± 0.1	0.45 ± 0.18	0.53 ± 0.17	0.5 ± 0.0
TabPFN	0.5 ± 0.0	0.5 ± 0.0	0.5 ± 0.0	0.5 ± 0.0	0.42 ± 0.05	0.5 ± 0.0	0.5 ± 0.0
eMMI	$\mathbf{0.87 \pm 0.1}$	$\mathbf{0.89 \pm 0.09}$	$\mathbf{0.88 \pm 0.12}$	$\mathbf{0.57 \pm 0.1}$	$\mathbf{0.97 \pm 0.07}$	$\mathbf{0.73 \pm 0.14}$	$\mathbf{0.89 \pm 0.12}$

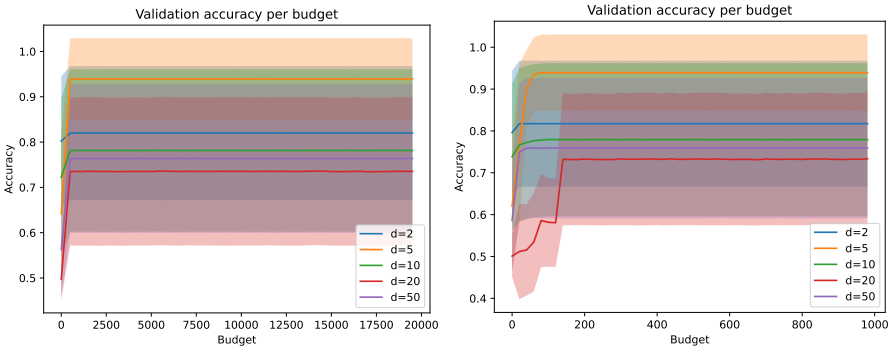
Table 6.3: Accuracy of the different methods approximating the ϵ-manifolds, with all hyperparameters for all methods (including the appropriate eMMI variant) automatically determined through hyperparameter optimisation. For every simulator, the best performance has been marked in **boldface**, with statistical significance determined by a Wilcoxon signed-rank test at significance level $\alpha = 0.05$. These results suggest that existing statistical methods cannot be generalised to effectively approximate ϵ-manifolds, necessitating new, specialised frameworks (i.e., eMMI).

Method	PROSAIL	PROSAIL 2D	TP	Lorenz63	GM	TM	LR
U-eMMI	0.55 ± 0.09	0.72 ± 0.19	0.56 ± 0.07	0.5 ± 0.01	$\mathbf{0.99 \pm 0.03}$	$\mathbf{0.76 \pm 0.17}$	$\mathbf{0.92 \pm 0.15}$
Conv-eMMI	$\mathbf{0.68 \pm 0.15}$	0.82 ± 0.1	0.54 ± 0.03	$\mathbf{0.5 \pm 0.02}$	$\mathbf{0.99 \pm 0.04}$	0.75 ± 0.11	0.51 ± 0.01
Seq-eMMI	$\mathbf{0.69 \pm 0.17}$	0.77 ± 0.15	0.55 ± 0.06	0.5 ± 0.01	0.98 ± 0.06	0.68 ± 0.12	0.89 ± 0.11
Dual-eMMI	$\mathbf{0.72 \pm 0.18}$	$\mathbf{0.88 \pm 0.1}$	$\mathbf{0.58 \pm 0.08}$	0.5 ± 0.01	0.97 ± 0.06	0.73 ± 0.14	0.91 ± 0.11

Table 6.4: Accuracy of approximating the ϵ-manifolds by individual eMMI variants. For every simulator, the best performance has been marked in **boldface**, with statistical significance determined by a Wilcoxon signed-rank test at significance level $\alpha = 0.05$.

pendix B.2). This pattern was likely caused by many solutions that could satisfy observations with low prior probabilities not being represented in the training data set. In some cases, such as PROSAIL 2D, the results for eMMI were better than those for the oracle-based ϵ-manifolds in Table 6.2. This is possible, because the results in Table 6.2 measure the efficacy of ϵ-manifolds themselves at including the true solution, while the results in Table 6.3 measure the efficacy of eMMI in approximating the ϵ-manifold. Therefore, the accuracy would still be expected to be relatively low for PROSAIL 2D, despite eMMI approximating the ϵ-manifold for this simulator well.

The results in Table 6.4 indicate that the different variants of eMMI perform well on different types of simulators, supporting our view that having multiple eMMI variants (whose selection can be automatically tuned using hyperparameter optimisation, as we did) can be beneficial. U-eMMI, Conv-eMMI and Dual-eMMI all performed significantly best on 3 simulators (subject to possible ties in performance); however, out of the four variants we tested, we consider Dual-eMMI to have achieved strong results most reliably. Even in cases where U-eMMI performed significantly better than Dual-eMMI (GM, TM and LR), the differences in performance were very small, while in cases where U-eMMI performed poorly (PROSAIL), Dual-eMMI performed substantially better.



(a) $B_2 = 20\,000$ (b) $B_2 = 1\,000$

Figure 6.7: Average classifier accuracy over 50 instances of the linear regression simulator (dimensionality d of 2, 5, 10, 20, and 50 parameters) when trained on a dataset consisting of points sampled by eMMI, as a function of the eMMI sampling budget B_2, for a maximum budget of 20 000 (Figure 6.7a) and 1 000 (Figure 6.7b). In all cases, the classifier converged to a stable performance very early on (< 200 samples), even for the 50-dimensional linear regression model, although the variability between instances was high.

In conclusion, regarding CRQ2, the current form of eMMI appears to be able to capture ϵ-manifolds better than UQ-based methods, indicating that even sophisticated UQ methods cannot be generalised to ϵ-manifold approximation. However, when faced with chaotic systems (such as the dynamical systems simulators), the performance improvement over UQ can be small. In those cases, none of the methods performed particularly well. Based on these results, eMMI appears effective at approximating ϵ-manifolds compared to UQ-based approximations, in a broad range of applications (physical model inversion, simulation-based inference and machine learning model parameterisation), but is not yet universally applicable to all possible types of loss landscapes.

6.7.3. CRQ3: BUDGET AND SCALABILITY

The results for our experiment aimed at answering CRQ3 on scalability can be found in Figure 6.7. As the figure shows, the classifier tend to converge to a stable performance in fewer than 200 samples, indicating that the budget B_2 for step 2 does not need to be overly large for eMMI to achieve good performance on the simulators we tested. Our results further indicate that the dimensionality of the problem did not affect this pattern much: although performance on problems of higher dimensionality tended to be lower than lower dimensional models, this was not always consistent (e.g., performance was slightly worse for $d = 20$ than $d = 50$).

This result implies that eMMI may be applicable to larger-scale model inversion problems than expected, because the budget necessary to converge for the largest problem we tested (up to 50-dimensional linear regression) was less than 1% of the total budget B we made available in our experiments (20 000). Although part of this budget needs to be spent on finding the optimum $\hat{\boldsymbol{\theta}}$, this cost may be alleviated using, e.g., the hybrid model warm-starting approach we deployed in our experiments.

In conclusion, regarding CRQ3, it appears that eMMI requires a surprisingly low amount of sampling budget to converge to its final performance level, and that its convergence speed is not heavily affected by the dimensionality of the problems under consideration. Efficiently sampling around the optimum $\hat{\boldsymbol{\theta}}$ to train a classifier, thereby approximating the ϵ-manifold, appears to only require a fraction of the function evaluation budget that would, in any case, need to be spent on finding the optimum $\hat{\boldsymbol{\theta}}$ in the first place. Therefore, eMMI would be applicable to many problem settings any SBI or black-box optimisation algorithm is applicable to, since the marginal cost of approximating the ϵ-manifold would be small.

6.8. CONCLUSIONS AND FUTURE WORK

In this chapter, we addressed the problem of finding viable solution sets to model inversion problems. Firstly, we introduced ϵ-manifolds, i.e. manifold of potentially valid solutions to a model inversion problem instance. We formalised these ϵ-manifolds and their core assumptions, as well as ϵ-manifold sets, and derived several theoretical properties. Secondly, we proposed eMMI, a method to automatically approximate ϵ-manifolds. We introduced four variants of our method to approximate ϵ-manifolds: U-eMMI, Conv-eMMI, Seq-eMMI and Dual-eMMI, each of which relies on different sampling heuristics, mostly based on diversity optimisation techniques, to extract the ϵ-manifold. We performed computational experiments evaluating the advantages of perfect ϵ-manifolds using an oracle-based approach, and compared the practical ϵ-manifold approximation performance of eMMI compared to statistical baseline methods. The results from these experiments demonstrate that ϵ-manifolds are much more effective than an oracle-based uncertainty quantification approach at including the true solution, indicating that existing statistical frameworks may not be sufficient to address the problem, thereby necessitating the use of ϵ-manifolds. The eMMI heuristics were effective at approximating the ϵ-manifold compared to statistical uncertainty quantification baseline methods.

Access to these ϵ-manifolds can improve the interpretability of model inversion and inference results, contribute to a greater understanding of the scientific processes underlying the simulation models, and enables novel types of analyses.

These analyses can extend beyond the traditional domain of simulation-based inference, such as the training of machine learning models and robustness analyses. A large ϵ-manifold may suggest ill-posedness, while the degree to which an ϵ-manifold shape conforms to statistical priors (e.g., Gaussian) enables practitioners to decide whether to trust statistical uncertainty quantification on their inference predictions. In the context of biophysical parameter estimation, we can use ϵ-manifolds to describe the behaviour we observed in Chapter 5 and determine its impact on parameter estimation results. They may also contribute to the discovery of new scientific patterns; for example, if the "LAI saturation" of Section 6.4.3 had not been a known phenomenon, the long-tailed ϵ-manifold for this parameter dimension in Figure 6.1b would have shown it.

Future work building upon our research presented here could focus on fundamental extensions of our proposed framework, as well as exploring novel applications of ϵ-manifolds. Examples of further explorations we would recommend include dynamic settings for ϵ, as opposed to the static simulator-wide settings we have been using, extending the heuristics of eMMI by introducing new objective functions f_2 or constraints, or incorporating surrogate model-based approaches. Amortised approaches, similar to the amortisation of Bayesian inference methods, could also be explored, as could approaches optimising in a latent space with reduced dimensionality (we explore this idea further in Section 7.2.2). As further applications, we believe it would be interesting to explore machine learning data set difficulty analyses through a comparison of viable parameterisations between instances, identifying manifolds of possible counterfactual predictions for interventions, or performing adversarial robustness analyses.

This concludes the technical contributions of this thesis. We have now covered every step of the parameter estimation pipeline shown in Figure 1.1. Chapters 3 and 4 have addressed Challenge 1 by answering RQs 1 and 2, thereby improving data consistency, while Chapters 5 and 6 addressed Challenge 2 by answering RQs 3 and 4. Although eliminating the ill-posedness of Challenge 2 altogether is likely infeasible through methodological contributions alone, our work in Chapter 5 resulted in concrete recommendations to alleviate the ill-posedness in a data-centric manner, while the concepts and method introduced in this chapter enable users to judge whether their specific parameter estimation results are reliable through an inspection of the ϵ-manifold.

In the next chapter, we will reflect further on the findings throughout Chapters 3–6, and answer the research questions from Section 1.1.