
O-MMGP: Optimal Mesh Morphing Gaussian Process Regression for Solving PDEs with non-Parametric Geometric Variations

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 We address the computational challenges of solving parametric PDEs with non
2 parametrized geometric variations and non-reducible problems, such as those in-
3 volving shocks and discontinuities of variable positions. Traditional dimensionality
4 reduction methods like POD struggle with these scenarios due to slowly decaying
5 Kolmogorov widths. To overcome this, we propose a novel non-linear dimension-
6 ality reduction technique to reduce the required modes for representation. The
7 non-linear reduction is obtained through a POD after applying a transformation
8 on the fields, which we call *optimal mappings*, and is a solution to an optimiza-
9 tion problem in infinite dimension. The proposed learning framework combines
10 morphing techniques, non-linear dimensionality reduction, and Gaussian Process
11 Regression (GPR). The problem is reformulated on a reference geometry before
12 applying the dimensionality reduction. Our method learns both the optimal map-
13 ping, and the solution fields, using a series of GPR models, enabling efficient and
14 accurate modeling of complex parametric PDEs posed on variable geometries that
15 share a common topology. The results obtained concur with current state-of-the-art
16 models. We mainly compare our method with the winning solution of the ML4CFD
17 NeurIPS 2024 competition.

18 1 Introduction

19 1.1 Background

20 Many scientific and engineering challenges involve solving complex boundary value problems, often
21 formulated as parametric partial differential equations (PDEs). These problems require exploring
22 the influence of varying parameters such as material properties, boundary and initial conditions, or
23 geometric configurations. Traditional numerical methods like the finite element method and finite
24 difference methods, while accurate, are computationally expensive, particularly when repeated evalu-
25 ations are necessary across a large and high-dimensional parameter set. To address this computational
26 burden, techniques in model-order reduction and machine learning have been developed, offering
27 efficient approximations without compromising accuracy.

28 Model-order reduction techniques, such as the reduced-basis method [25, 14], construct low-
29 dimensional approximation vector spaces to represent the set of solutions of the parametric PDE,
30 enabling fast computation for new parameter values. These approaches typically involve an offline
31 phase, where high-fidelity models are used to generate a reduced basis, and an online phase, where
32 this basis is employed for efficient computations. However, when the physical domain varies with
33 the parameters, standard methods like Proper Orthogonal Decomposition (POD) face challenges
34 due to the need to reconcile solutions defined on different domains. This often requires morphing

techniques to map variable domains to a common one, transforming the problem into a form suitable for reduced-order modeling [21, 28]. Linear reduced-order modeling techniques also face huge difficulties when the solution set of interest has a slowly decaying Kolmogorov width. Such problems, which we call hereafter non-reducible problems, are common for PDEs that involve shocks of variable position, discontinuities, boundary layers an so on.

On the other hand, machine learning methods, particularly those relying on deep learning, have shown promise in solving PDEs, learning solutions, and accelerating computations [13, 19]. Approaches relying on graph neural networks (GNNs) [29] have been particularly effective in handling unstructured meshes and varying geometric configurations. Despite their flexibility, these methods require substantial computational resources and large training datasets, and they often lack robust predictive uncertainty estimates.

1.2 Contribution

In this work, we propose a novel approach that integrates principles of morphing, non-linear dimensionality reduction, and classical machine learning to address the challenges posed by parametric PDEs with geometric variations for non-reducible problems. The main contribution is a novel algorithm used to maximizes the energy in the first principal modes, which directly addresses the core limitation of traditional dimensionality reduction techniques, and limits the number of modes needed to approximate non-reducible problems in moderate dimension. Such approaches are commonly called registration in the literature. Unlike most existing algorithms that aim at projecting the samples on one single mode, our approach may involve an arbitrary number of modes $r \in \mathbb{N}^*$. In addition, another major advantage with respect to similar methods is that the procedure is done automatically as we do use any feature tracking and detection method.

1.3 Related works

The challenges mentioned in this work have been extensively studied in the literature. Morphing techniques have been used for various applications in reduced order modeling to recast the problem on a reference domain [24, 1, 35]. Numerous approaches were also proposed to deal with non-reducible problems by using non-linear dimensionality reduction such as registration [30, 32], feature tracking [22], optimal transport [9], neural networks [16, 12, 3, 5], and so on. Machine learning approaches leveraging neural networks have also demonstrated remarkable success in solving numerical simulations, particularly in capturing complex patterns and dynamics. Methods such as the Fourier Neural Operator (FNO) [18] and its extension, Geo-FNO [17], efficiently learn mappings between function spaces by leveraging Fourier transforms for high-dimensional problems. Physics-Informed Neural Networks (PINNs) [26] embed physical laws directly into the loss function, enabling solutions that adhere to governing equations. Deep Operator Networks (DeepONets) [20] excel in learning operators with small data requirements, offering flexibility in various applications. Mesh Graph Networks (MGNs) [23] use graph-based representations to model simulations on irregular domains, preserving geometric and topological properties.

2 Preliminaries and notations

Let $n \in \mathbb{N}^*$ and $\Omega_1, \dots, \Omega_n \subset \mathbb{R}^d$ to be a family of n distinct domains that share a common topology, with $d = 2, 3$. We suppose the parametrization of the domains is unknown and that for all $1 \leq i \leq n$, each domain Ω_i is equipped with a (non-geometrical) parameter $\mu_i \in \mathcal{P}$ where $\mathcal{P} \subset \mathbb{R}^p$ is a set of parameter values (think of μ_i as being some material parameter value for instance). In addition, let $u_i : \Omega_i \rightarrow \mathbb{R}$ be the solution of a parametrized partial differential equation for the parameter value μ_i defined on Ω_i (think about a temperature field for instance). We assume in the following that for all $1 \leq i \leq n$, $u_i \in L^2(\Omega_i)$. We assume that for all $1 \leq i \leq n$ a finite element mesh $\mathcal{M}_i$ is chosen so that $\partial\mathcal{M}_i$ can be considered as an accurate enough approximation of the boundary of the domain $\partial\Omega_i$. We also assume the field u_i can be accurately approximated by its finite element interpolation associated to the mesh $\mathcal{M}_i$. We fix a reference domain, denoted by Ω_0 , equipped with a mesh denoted by $\mathcal{M}_0$, that shares the same topology as the other domains. For all $0 \leq i \leq n$, $\mathbf{M}_i$ be the space of bijective $W^{1,\infty}$ mappings from Ω_0 onto Ω_i , so that for all $\psi \in \mathbf{M}_i$, $\psi(\Omega_0) = \Omega_i$. We also introduce $\mathbf{M} := \mathbf{M}_1 \times \mathbf{M}_2 \times \dots \times \mathbf{M}_n$.

86 For any family of functions $\mathcal{H} = (h_i)_{1 \leq i \leq n} \in L^2(\Omega_0)^n$, we define the correlation matrix application
 87 $C_{\mathcal{H}}$ of $\mathcal{H}$ as the map

$$\begin{aligned} C_{\mathcal{H}} : \mathbf{M}_0^n &\rightarrow \mathcal{S}_n^+ \\ \Phi &\rightarrow C_{\mathcal{H}}[\Phi] := (C_{\mathcal{H},ij}[\Phi])_{1 \leq i,j \leq n}, \end{aligned} \quad (1)$$

where $\mathcal{S}_n^+$ is the set of symmetric non-negative semi-definite matrices of dimension $n \times n$, and for all $\Phi := (\phi_i)_{1 \leq i \leq n} \in \mathbf{M}_0^n$ and for all $1 \leq i, j \leq n$,

$$C_{\mathcal{H},ij}[\Phi] := \langle h_i \circ \phi_i, h_j \circ \phi_j \rangle_{L^2(\Omega_0)} = \int_{\Omega_0} h_i \circ \phi_i(x) h_j \circ \phi_j(x) dx.$$

88 We denote by $\lambda_1^{\mathcal{H}}[\Phi] \geq \lambda_2^{\mathcal{H}}[\Phi] \geq \dots \geq \lambda_n^{\mathcal{H}}[\Phi] \geq 0$ the eigenvalues of $C_{\mathcal{H}}[\Phi]$. We also denote by
 89 $(\zeta_1^{\mathcal{H}}[\Phi], \zeta_2^{\mathcal{H}}[\Phi], \dots, \zeta_n^{\mathcal{H}}[\Phi]) \subset \mathbb{R}^n$ an orthonormal family of corresponding eigenvectors. Finally,
 90 given a positive integer $r \in \mathbb{N} \setminus \{0\}$, we define the functional

$$\begin{aligned} J_{\mathcal{H},r} : \mathbf{M}_0^n &\rightarrow \mathbb{R} \\ \Phi &\mapsto J_{\mathcal{H},r}[\Phi] := \frac{\sum_{j=1}^r \lambda_j^{\mathcal{H}}[\Phi]}{\text{Tr}(C_{\mathcal{H}}[\Phi])}. \end{aligned} \quad (2)$$

91 We now introduce the terminology of three different configurations that will be used throughout the
 92 paper.

- 93 1. First, we refer to the **physical configuration** as the collection of pairs $\{(\Omega_i, u_i)\}_{1 \leq i \leq n}$.
- 94 2. Given a reference domain Ω_0 and $\Phi^{\text{geo}} := (\phi_i^{\text{geo}})_{1 \leq i \leq n} \in \mathbf{M}$, we refer to the **reference**
 95 **configuration** as the collection of pairs $\{(\Omega_0, u_i \circ \phi_i^{\text{geo}})\}_{1 \leq i \leq n}$. In this configuration, all the
 96 fields $u_i^{\text{ref}} := u_i \circ \phi_i^{\text{geo}}$ belong to $L^2(\Omega_0)$, and classical dimensionality reduction techniques
 97 such as PCA can be applied on the family $\mathcal{U} := (u_i^{\text{ref}})_{1 \leq i \leq n} \in L^2(\Omega_0)^n$.
- 98 3. Given the reference configuration and some $r \in \mathbb{N} \setminus \{0\}$, we refer to the **r-optimal configura-**
 99 **tion** as the collection of pairs $\{(\Omega_0, u_i^{\text{ref}} \circ \phi_{i,r}^{\text{opt}})\}_{1 \leq i \leq n}$, where $\Phi_r^{\text{opt}} := (\phi_{i,r}^{\text{opt}})_{1 \leq i \leq n} \in \mathbf{M}_0^n$
 100 is a solution to the following maximization problem:

$$\text{find } \Phi_r^{\text{opt}} \in \arg \max_{\Phi \in \mathbf{M}_0^n} J_{\mathcal{U},r}[\Phi]. \quad (3)$$

101 The maximization problem (3) is considered so that the family $(u_i^{\text{opt}})_{1 \leq i \leq n} := (u_i^{\text{ref}} \circ \phi_{r,i}^{\text{opt}})$
 102 can be accurately approximated by elements of a r -dimensional vector space.

103 3 Methodology

104 3.1 Training phase

105 In the training phase, we suppose that we have access to the dataset of triplets $\{(\Omega_i, \mu_i, u_i)\}_{1 \leq i \leq n}$.
 106 The domain Ω_i (or its mesh) and the parameter μ_i are the inputs to the physical solver, and the field
 107 u_i is its output. We chose a reference domain (which can be one from the dataset) that shares the
 108 same topology.

109 3.1.1 Pretreatment

110 We perform these pretreatment steps in the training phase.

1. We pass from the physical configuration to the reference configuration (4) by computing
 for all $1 \leq i \leq n$, a mapping $\phi_i^{\text{geo}} \in \mathbf{M}_i$. We apply POD on the family $(\phi_i^{\text{geo}} - \mathbf{Id})_{1 \leq i \leq n}$
 to obtain the POD modes $\{\zeta_i^{\text{geo}}\}_{1 \leq i \leq n}$ and the generalized coordinates $\{\alpha^i\}_{1 \leq i \leq n}$ where
 $\alpha^i = (\alpha_j^i)_{1 \leq j \leq s} \in \mathbb{R}^s$, such that

$$\forall 1 \leq j \leq s, \alpha_j^i = \langle \phi_i^{\text{geo}} - \mathbf{Id}, \zeta_j^{\text{geo}} \rangle_{L^2(\Omega_0)},$$

111 and s is the number of retained modes for the geometrical mappings. Each domain Ω_i is
 112 defined now by the vector α^i .

2. We pass from the reference configuration to the optimal configuration (5) by solving problem
3. Once we obtain the functions $(\phi_i^{\text{opt}} - \mathbf{Id})_{1 \leq i \leq n}$, we apply POD to obtain the POD modes $\{\zeta_i^{\text{opt}}\}_{1 \leq i \leq n}$ and the generalized coordinates $\{\beta^i\}_{1 \leq i \leq n}$ where $\beta^i = (\beta_j^i)_{1 \leq j \leq t} \in \mathbb{R}^t$, such that

$$\forall 1 \leq j \leq t, \beta_j^i = \langle \phi_i^{\text{opt}} - \mathbf{Id}, \zeta_j^{\text{opt}} \rangle_{L^2(\Omega_0)},$$

and t is the number of retained modes for the optimal mappings.

3. Finally, after evaluating $\{u_i^{\text{opt}}\}_{1 \leq i \leq n}$, we apply POD to obtain the POD modes $\{\psi_i\}_{1 \leq i \leq n}$ and the generalized coordinates $\{\gamma^i\}_{1 \leq i \leq n}$ where $\gamma^i = (\gamma_j^i)_{1 \leq j \leq r} \in \mathbb{R}^r$, such that

$$\forall 1 \leq j \leq r, \gamma_j^i = \langle \psi_i, u_j^{\text{opt}} \rangle_{L^2(\Omega_0)},$$

After applying these dimensionality reductions, we obtain the following three approximations:

$$\phi_i^{\text{ref}} \approx \mathbf{Id} + \sum_{j=1}^s \alpha_j^i \zeta_j^{\text{ref}}, \phi_i^{\text{opt}} \approx \mathbf{Id} + \sum_{j=1}^t \beta_j^i \zeta_j^{\text{opt}}, \text{ and } u_i^{\text{opt}} \approx \sum_{j=1}^r \gamma_j^i \psi_j.$$

Notice that the passing from the physical configuration to the reference configuration is decoupled and purely geometrical, that is it does not depends on the fields $\{u_i\}_{1 \leq i \leq n}$. For example, we show in Figure 1 two forms of airfoils. ϕ^{geo} will transform one airfoil onto the other. On the other hand, the transition to the optimal configuration is coupled between all the samples in the training dataset.

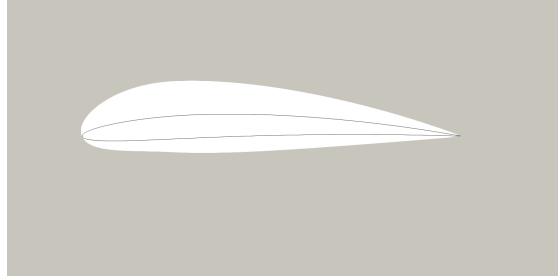


Figure 1: Two airfoils superimposed.

3.1.2 Training

After performing the dimensionality reduction step, we train two Gaussian processes regression models [33] as follows.

1. The first model is to learn the optimal mapping that transforms the reference configuration to the optimal configuration. This model takes as input the physical parameter μ_i and the geometrical mapping POD coefficient α_i , and as output the optimal mapping POD coefficient β_i . We denote this model by $\mathcal{R} : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^t$.
2. The second model is to learn the field in the optimal configuration. This model takes as input the physical parameter μ_i and the geometrical mapping POD coefficient α_i , and as output the field in the optimal configuration POD coefficient γ_i . We denote this model by $\mathcal{O} : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^r$.

3.2 Inference phase

In the inference phase, we are given a new unseen geometry $\tilde{\Omega}$ in the inference phase, with a physical parameter $\tilde{\mu}$. The goal is to predict the field of interest $\tilde{u}$, solution to the physical simulation. We proceed in the following manner.

1. First, we compute the geometrical mapping $\tilde{\phi}^{\text{geo}}$ that maps Ω_0 onto $\tilde{\Omega}$, then we project $\tilde{\phi}^{\text{geo}} - \mathbf{Id}$ on the POD basis $\{\zeta_i^{\text{geo}}\}_{1 \leq i \leq n}$ to obtain the coefficient $\tilde{\alpha}$.

137 2. We use the pair $(\tilde{\alpha}, \tilde{\mu})$ to infer $\tilde{\beta} = \mathcal{R}(\tilde{\alpha}, \tilde{\mu})$ and $\tilde{\gamma} = \mathcal{O}(\tilde{\alpha}, \tilde{\mu})$. Then we have $\tilde{\phi}^{\text{opt}} :=$
138 $\text{Id} + \sum_{j=1}^t \tilde{\beta}_j \zeta_j^{\text{opt}}$ and $\tilde{u}^{\text{opt}} := \sum_{j=1}^r \tilde{\gamma}_j \psi_j$.
139 3. Finally, the quantity of interest $\tilde{u}$ is obtained as $\tilde{u} := \tilde{u}^{\text{opt}} \circ (\tilde{\phi}^{\text{opt}})^{-1} \circ (\tilde{\phi}^{\text{geo}})^{-1}$.

140 4 Geometrical mapping

141 Numerous techniques exists in the literature to construct a mapping between domains [2, 4, 31]. In
142 this work, we use RBF morphing [10] to construct the mapping from the reference domain Ω_0 to
143 each target domain Ω_i . When using this technique, we suppose that the deformation on a subset of
144 points in Ω_0 , called the control points, is known. These points are usually on the boundary of Ω_0 .
145 Then, we can leverage the knowledge of $\phi_i(\partial\Omega_0) = \partial\Omega_i$ to compute the deformation in the bulk of
146 the domain $\phi_i(\Omega_0)$. In this work, we suppose that the geometries are not parametrized and $\phi_i(\partial\Omega_0)$
147 is not given. Thus, we start by computing $\phi_i(\partial\Omega_0)$, for all $1 \leq i \leq n$. When computing $\phi_i(\partial\Omega_0)$ is
148 not feasible, methods proposed in [11, 15] provides solutions to automatically finds a mapping from
149 the reference onto the target domain.

150 5 Optimal mapping

151 The second major building block we introduce is the optimal mapping algorithm to compute $\Phi^{\text{opt}} =$
152 $(\phi_i^{\text{opt}})_{1 \leq i \leq n}$ that maximize the energy in the first r modes.

153 In order to pass from the reference configuration to the optimal configuration, we solve (3):

$$\text{find } \Phi^* \in \arg \max_{\Phi \in \mathbf{M}_0^n} J_{\mathcal{U},r}[\Phi].$$

154 The optimization problem presented here is non-convex with possibly infinitely many solutions. We
155 present here briefly the optimization strategy employed to solve this problem. Refer to the appendixes
156 for a detailed discussion.

157 5.1 Gradient algorithm in infinite dimension

We can show that the differential of $J_{\mathcal{U},r}$ with respect to Φ at a point Ψ has the following expression
(see appendix A):

$$DJ_{\mathcal{U},r}[\Phi][\Psi] = \sum_{i=1}^n \int_{\Omega_0} \mathbf{f}_i[\Phi] \cdot \psi_i dx.$$

158 with

$$\mathbf{f}_i[\Phi] := \sum_{j=1}^n \sum_{k=1}^r \left(\frac{2\zeta_{k,i}^{\mathcal{U}}[\Phi] \zeta_{k,j}^{\mathcal{U}}[\Phi]}{\text{Tr}(C_{\mathcal{U}}[\Phi])} - \frac{2\lambda_k^{\mathcal{U}}[\Phi]}{\text{Tr}(C_{\mathcal{U}}[\Phi])^2} \delta_{ij} \right) u_j^{\text{ref}} \circ \phi_j \vec{\nabla} u_i^{\text{ref}} \circ \phi_i. \quad (4)$$

159 A standard gradient algorithm in infinite dimension consists of (i) choosing an appropriate inner
160 product, denoted here by a , (ii) computing the Riesz representation of each $\mathbf{f}_i[\Phi]$ with respect to a ,
161 which will be denoted by $\mathbf{u}_i[\Phi]$ and finally (iii) updating, at each iteration m , $\phi_i^{(m)}$ by

$$\phi_i^{(m+1)} = \phi_i^{(m)} + \epsilon \mathbf{u}_i[\Phi^{(m)}] \quad (5)$$

162 with $\epsilon > 0$ in the gradient step. The simplest example of a is the $L_2(\Omega_0)$ inner product which gives
163 the following iterative scheme:

$$\phi_i^{(m+1)} = \phi_i^{(m)} + \epsilon \mathbf{f}_i[\Phi^{(m)}]. \quad (6)$$

164 However, this may suffer from one of the following problems:

- 165 • $\phi_i^{(m+1)}$ might not be bijective.
- 166 • $\phi_i^{(m+1)}$ might not map Ω_0 onto itself.
- 167 • If the function $\Phi^{(m)}$ is far from a maximum, the algorithm may fail to converge properly.

168 We address these issues in the following subsections.

5.2 Bijectivity

To address bijectivity, we consider the following optimization statement:

$$\text{find } \Phi^* \in \arg \max_{\Phi \in \mathcal{M}_0^n} I_{\mathcal{U},r}[\Phi] := J_{\mathcal{U},r}[\Phi] - c_1 \sum_{i=1}^n E[\phi_i], \quad (7)$$

where E is an energy term to enforce bijectivity and $c_1 > 0$ is a penalization parameter. Ideally, the term E should diverge to $+\infty$ if the mapping ϕ_i becomes non-bijective. In this case, the objective function I diverges to $-\infty$. In this work, we use linear elastic energy for the term E , (we note that it does not diverges to $+\infty$ for non-bijective mappings, however, it gives acceptable results). Finally, we solve the problem 7 using the continuation method, which consists of iteratively decrease the parameter c_1 in order to obtain a solution to problem 3. We give more details about the continuation method and the term E in appendix C.

5.3 Mapping condition

A necessary condition to map Ω_0 onto itself is that the boundary of Ω_0 is mapped also onto itself. If $\partial\Omega_0$ consists only of straight lines (faces in 3d), we impose that points on the boundary deform only in the tangential direction, so that they stay on the boundary. When $\partial\Omega_0$ presents also curves, we can fix points that lies on the curved part of $\partial\Omega_0$. Thus, we partition the boundary of Ω_0 to $\partial\Omega_0 := \partial\Omega_0^p \cup \partial\Omega_0^c$ with $\partial\Omega_0^p$ is the union of all the straight lines (faces in 3d) of $\partial\Omega_0$ and $\partial\Omega_0^c$ is the curved part of $\partial\Omega_0$. We define the space $H_n^1(\Omega_0) := \{\mathbf{u} \in H^1(\Omega_0) : \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega_0^p, \mathbf{u} = 0 \text{ on } \partial\Omega_0^c\}$ and the inner product a on $H_n^1(\Omega_0)$ as

$$(\mathbf{u}, \mathbf{v}) \mapsto a(\mathbf{u}, \mathbf{v}) := \int_{\Omega_0} \sigma(\mathbf{u}) : \varepsilon(\mathbf{v}) d\mathbf{x}, \quad (8)$$

with σ and ε are respectively the classical elasticity stress and strain tensors (we fix the Young modulus $E = 1$ and Poisson ratio $\mu = 0.3$ in this work). We compute, for all $1 \leq i \leq n$, and for each iteration m , the Riesz representation $\mathbf{u}_i^{(m)}$ of $\mathbf{f}_i$ with respect to this inner product, which is the unique solution to

$$\forall \mathbf{v} \in H_n^1(\Omega_0; \mathbb{R}^d), \quad a(\mathbf{u}_i^{(m)}, \mathbf{v}) = \int_{\Omega_0} \mathbf{f}_i[\Phi^{(m)}] \mathbf{v} d\mathbf{x}, \quad (9)$$

to obtain the iterative scheme:

$$\phi_i^{(m+1)} = \phi_i^{(m)} + \epsilon \mathbf{u}_i[\Phi^{(m)}]. \quad (10)$$

We can also obtain a similar expression when solving problem (7) and using linear elastic energy (see more in appendix C):

$$\phi_i^{(m+1)} = \phi_i^{(m)} + \epsilon \left(\mathbf{u}_i[\Phi^{(m)}] - c_1 \left(\phi_i^{(m)} - Id \right) \right). \quad (11)$$

By using the above procedure, we guarantee that, if $\phi_i^{(m)} + \epsilon \mathbf{u}_i[\Phi^{(m)}]$ is bijective, then necessary we have that the points on $\partial\Omega_0^p$ stay on $\partial\Omega_0^p$, thus, preserving the boundary of Ω_0 . We note however that this is suboptimal as we do not allow $\partial\Omega_0^c$ to deform in this case. A full treatment of curved boundaries is the subject of future work.

5.4 Uncorrelated samples

One major reason for the poor convergence of the gradient algorithm is when the samples are heavily not correlated. This happens when the non-zero values of u_i^{ref} are compactly supported in Ω_0 . More precisely, if, for some $1 \leq j \leq n$, $\text{supp}(u_j^{\text{ref}} \circ \phi_j) \cap \text{supp}(\vec{\nabla} u_i^{\text{ref}} \circ \phi_i) = \emptyset$, then the contribution of $u_j^{\text{ref}} \circ \phi_j \vec{\nabla} u_i^{\text{ref}} \circ \phi_i$ in $\mathbf{f}_i$ is null. To this end, we transform the fields $\{u_i\}_{1 \leq i \leq n}$ to the family of fields $\hat{\mathcal{U}}(c_2) := \{\hat{u}_i\}_{1 \leq i \leq n}$, where each is defined as the solution to

$$\begin{aligned} -\Delta \hat{u}_i + c_2 \hat{u}_i &= c_2 u_i, \\ \partial_n \hat{u}_i &= 0 \end{aligned} \quad (12)$$

where Δ is the Laplacian operator and $c_2 > 0$. This equation has a unique solution. The transformed fields will diffuse the value of u_i throughout the domain Ω_0 for small values of c_2 (see Figures 3-4). On the other hand, $\hat{u}_i$ converges to u_i as c_2 goes to $+\infty$. We use the fields $\hat{\mathcal{U}}(c_2)$ in order to solve (3)-(7). The value of c_2 is also changed throughout the iterations using the continuation method.

6 Numerical example

In this section, we illustrate the method on two example of non-reducible problems.

6.1 Example 1: advection-reaction equation

This example is taken from [30], and it illustrates a non-reducible problem. Here, we show the effect of the optimal mapping algorithm on a fixed geometry. Let the following advection-reaction equation:

$$\begin{cases} \nabla \cdot (c_\mu u_\mu) + \sigma_\mu u_\mu = f_\mu & \text{in } \Omega = (0, 1)^2, \\ u_\mu = u_{D,\mu} & \text{on } \Gamma_{\text{in},\mu} := \{x \in \partial\Omega : c_\mu \cdot \mathbf{n} < 0\}, \end{cases}$$

where $\mathbf{n}$ denotes the outward normal to $\partial\Omega$, and

$$c_\mu = \begin{bmatrix} \cos(\mu_1) \\ \sin(\mu_1) \end{bmatrix}, \quad \sigma_\mu = 1 + \mu_2 e^{x_1 + x_2}, \quad f_\mu = 1 + x_1 x_2,$$

$$u_{D,\mu} = 4 \arctan \left(\mu_3 \left(x_2 - \frac{1}{2} \right) \right) (x_2 - x_2^2),$$

$$\mu = [\mu_1, \mu_2, \mu_3] \in \mathcal{P} := \left[-\frac{\pi}{10}, \frac{\pi}{10} \right] \times [0.3, 0.7] \times [60, 100].$$

We consider $n = 250$ samples. In Figure 2, we plot u_μ for three different values of the parameter before and after computing the optimal mappings that maximize $J_{\mathcal{U},r}$, for $r = 1$. The purpose of this first example is to demonstrate the effect of the optimal mapping algorithm on the fields. We can clearly see how they are aligned and can be approximated using a low dimensional space. In this scenario, the regression task becomes much easier. The optimal mappings algorithm can be

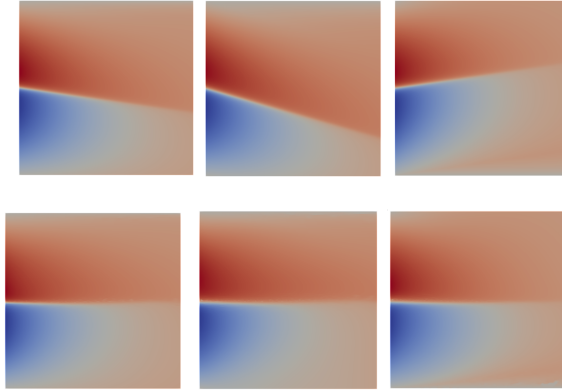


Figure 2: Top: three samples before the optimization. Bottom: three samples after the optimization.

seen as a multi-modal generalization to the registration methods, where the case $r = 1$ gives similar results to aligning all the samples on one mode. However, the advantage of our method is that we can go beyond a single mode. Furthermore, the alignment is automatic and does not any of the feature detection and tracking methods.

6.2 ML4CFD NeurIPS 2024 competition

In the second example, we apply the method to the airfoil design case considered in the ML4CFD NeurIPS 2024 competition [34], and we compare it with the winning solution. The dataset adopted for the competition is the AirfRANS dataset from [6]. Each sample have two scalar inputs, the inlet velocity and the angle of attack, and three output fields, the velocity, pressure and the turbulent viscosity. The dataset is composed of three splits.

1. Training set: composed of 103 samples.
2. Testing set: composed of 200 samples.

232 3. OOD testing set: composed of 496 samples, where the Reynold number considered for
 233 samples in this split is taken out of distribution.

234 In this work, we focus on the turbulent viscosity field as it represents a non-reducible field. We
 illustrate this field in Figure 3.

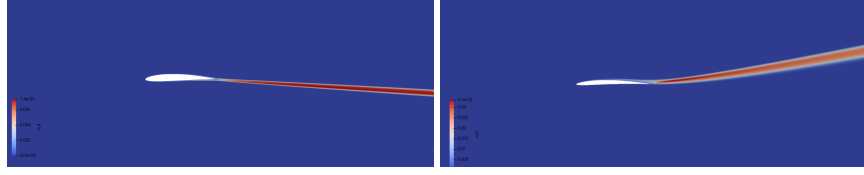


Figure 3: The turbulent viscosity field illustrated for two of the samples.

235

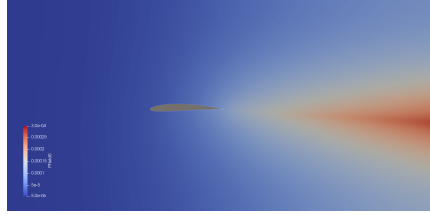


Figure 4: The diffused field $\hat{u}$ for $c_2 = 1$.

236 6.3 NeurIPS solution: MMGP + wake line prediction

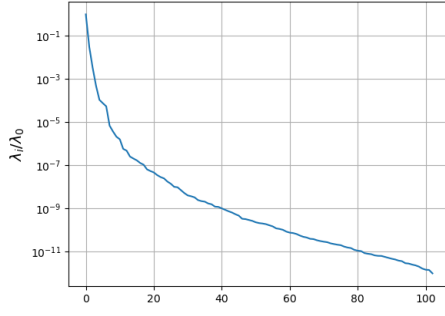
237 The winning solution to the ML4CFD competition [7] relies on the application of the MMGP
 238 method[8], with the addition of aligning the wake line behind the airfoil for all the samples at same
 239 position. The latter step being necessary to obtain accurate prediction of the turbulent viscosity.
 240 While this correction step gives accurate results, it remains case-dependent and needs to be done
 241 manually. Moreover, it suffers from the same problem as any other registration technique, which is
 242 the restriction to one mode.

243 6.4 O-MMGP: optimal mesh morphing Gaussian process

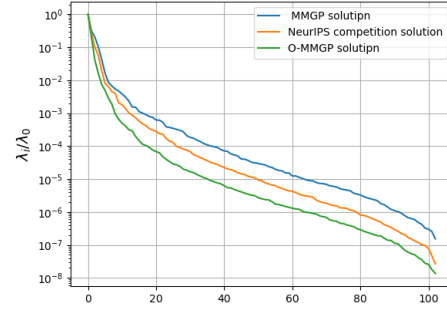
244 The method we present in this paper is founded on the MMGP method [8]. The main differences
 245 being:

- 246 1. First, in MMGP, the mapping ϕ_i^{geo} is computed from each geometry onto the reference
 247 domain Ω_0 . In this case, the input to the GP model is the POD coefficients of the inverse
 248 mapping $(\phi_i^{\text{geo}})^{-1}$. Thus, evaluating $(\phi_i^{\text{geo}})^{-1}$ would introduce extra computations to the
 249 procedure. In addition, when using RBF morphing, such as in this work, computing ϕ_i^{geo}
 250 from the same geometry proves to be much efficient numerically, as the RBF interpolation
 251 matrix $\mathbf{K}$ can be assembled and factorized once and for all (see appendix E for the definition
 252 of $\mathbf{K}$).
- 253 2. Secondly, the MMGP mappings are not field-optimized. This is equivalent to omitting the
 254 transition to the optimal configuration. So, when it comes to non-reducible problems, even
 255 if a large number of modes are retained, the error introduced by POD truncation and the
 256 discretization error would remain large to obtain reliable predictions for new samples.

257 We run the optimal mapping algorithm in order to maximize the compression of the turbulent
 258 viscosity field. We choose the reference geometry Ω_0 to be one of the samples in the training set. We
 259 then proceed to compute ϕ_i^{geo} using RBF morphing. Once computed, we solve problem 7 for the
 260 family $\hat{\mathcal{U}}(c_2)$. We choose $r := 1$, $c_1^{(0)} := 0.1$ and $c_2^{(0)} := 1$. The last two parameters are changed
 261 throughout the iterations as mentioned in appendix C. In Figure 5a, we show that the eigenvalues of
 262 the correlation matrix of $(\phi_i^{\text{opt}})_{1 \leq i \leq n}$ decay rapidly, proving numerically that this family is reducible,



(a) Decay of the eigenvalues of the correlation matrix of the family $(\phi_i^{\text{opt}})_{1 \leq i \leq n}$.



(b) Decay of the eigenvalues of the correlation matrix for the turbulent viscosity fields.

Figure 5: Decay of the eigenvalues.

and justifying the regression on the optimal mapping POD coordinates. The cost of the optimal mapping computation is discussed in appendix D

After computing the optimal mappings, we train two Gaussian process regression models, one to learn the optimal mapping $\tilde{\phi}$, and the other to learn the field $\tilde{u}$, which is the turbulent viscosity in this case.

In order to show the efficiency of the method, we compare the results of the following three tests:

1. First, we apply the method without solving the optimal mappings problem, similar to the original MMGP method. Thus, we predict the turbulent viscosity field in the reference configuration.
2. Second, we run the winning solution of the NeurIPS 2024 challenge, aligning the wake line behind the airfoil manually.
3. Finally, we apply the full O-MMGP procedure described in this paper.

In table 1, we report the mean square error (MSE) of the three tests on the turbulent viscosity field, for the testing and OOD splits. While the original MMGP performs poorly for this field, the O-MMGP method produces very accurate predictions, slightly surpassing the NeurIPS 2024 competition solution. The major advantage of the method is that the alignment of the snapshot was done automatically, without using the specificity of the case. In Figure 5b, we report the decay of the

	MMGP	NeurIPS solution	O-MMGP
Test	0.143	0.025	0.024
OOD	0.171	0.048	0.046

Table 1: MSE errors for the different tests.

eigenvalues of the correlation matrix for the three tests. For the MMGP solution, no aligning of the solution was performed, which explains the slow decay of the eigenvalues for this case. As expected, the eigenvalues decay the most rapidly when using the optimal mappings algorithm.

7 Conclusion

In this article, we have presented a new algorithm that aims to maximize the energy in the first r principal modes of a family of functions. The novelty of the algorithm lies in the fact that it automatically aligns the functions on an arbitrary number of modes, unlike other methods in the literature which may require feature tracking of the solution or assume that the family of functions can be compressed onto a single mode. We have shown how the proposed method can be integrated into the MMGP workflow to learn and predict the solutions of PDEs with geometric variabilities. The quality of the prediction was found to be on a par with state-of-the-art methods. Current work

291 aims to solve the problem in the general case of curved boundaries and provide a more thorough
 292 mathematical analysis of the method, as well as applying the method in the presence of multiple
 293 shocks.

294 References

- 295 [1] Owe Axelsson and Stanislav Sysala. Continuation newton methods. *Computers & Mathematics*
 296 *with Applications*, 70(11):2621–2637, 2015.
- 297 [2] Timothy J. Baker. Mesh movement and metamorphosis. *Engineering with Computers*, 18(3):188–
 298 198, 2002.
- 299 [3] Joshua Barnett, Charbel Farhat, and Yvon Maday. Neural-network-augmented projection-
 300 based model order reduction for mitigating the kolmogorov barrier to reducibility. *Journal of*
 301 *Computational Physics*, 492:112420, 2023.
- 302 [4] M. Faisal Beg, Michael I. Miller, Alain Trouné, and Laurent Younes. Computing large de-
 303 formation metric mappings via geodesic flows of diffeomorphisms. *International Journal of*
 304 *Computer Vision*, 61:139–157, 2005.
- 305 [5] Jules Berman and Benjamin Peherstorfer. Colora: Continuous low-rank adaptation for re-
 306 duced implicit neural modeling of parameterized partial differential equations. *arXiv preprint*
 307 *arXiv:2402.14646*, 2024.
- 308 [6] Florent Bonnet, Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. Airfrans: High fi-
 309 delity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes
 310 solutions. *Advances in Neural Information Processing Systems*, 35:23463–23478, 2022.
- 311 [7] Fabien Casenave. Ml4cfd neurisp2024 solution. [https://gitlab.com/drti/airfrans_](https://gitlab.com/drti/airfrans_competitions/-/tree/main/ML4CFD_Neurisp2024)
 312 [competitions/-/tree/main/ML4CFD_Neurisp2024](https://gitlab.com/drti/airfrans_competitions/-/tree/main/ML4CFD_Neurisp2024), since 2025 (accessed 30 January
 313 2025).
- 314 [8] Fabien Casenave, Brian Staber, and Xavier Roynard. Mmgp: a mesh morphing gaussian process-
 315 based machine learning method for regression of physical problems under nonparametrized
 316 geometrical variability. *Advances in Neural Information Processing Systems*, 36, 2024.
- 317 [9] Simona Cucchiara, Angelo Iollo, Tommaso Taddei, and Haysam Telib. Model order reduction
 318 by convex displacement interpolation. *Journal of Computational Physics*, 514:113230, 2024.
- 319 [10] Aukje De Boer, Martijn S. Van der Schoot, and Hester Bijl. Mesh deformation based on radial
 320 basis function interpolation. *Computers & Structures*, 85(11-14):784–795, 2007.
- 321 [11] Maya De Buhan, Charles Dapogny, Pascal Frey, and Chiara Nardoni. An optimization method
 322 for elastic shape matching. *Comptes Rendus. Mathématique*, 354(8):783–787, 2016.
- 323 [12] Stefania Fresca and Andrea Manzoni. Pod-dl-rom: Enhancing deep learning-based reduced
 324 order models for nonlinear parametrized pdes by proper orthogonal decomposition. *Computer*
 325 *Methods in Applied Mechanics and Engineering*, 388:114181, 2022.
- 326 [13] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow ap-
 327 proximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge*
 328 *discovery and data mining*, pages 481–490, 2016.
- 329 [14] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for*
 330 *parametrized partial differential equations*, volume 590. Springer, 2016.
- 331 [15] Abbas Kabalan, Fabien Casenave, Felipe Bordeu, Virginie Ehrlicher, and Alexandre Ern. Elasticity-based morphing technique and application to reduced-order modeling. *Applied*
 332 *Mathematical Modelling*, page 115929, 2025.
- 333 [16] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear mani-
 334 folds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973,
 335 2020.
- 336

- [17] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.
- [18] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [19] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.
- [20] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.
- [21] Andrea Manzoni and Federico Negri. Efficient reduction of pdes defined on domains with variable shape. *Model Reduction of Parametrized Systems*, pages 183–199, 2017.
- [22] Marzieh Alireza Mirhoseini and Matthew J Zahr. Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking. *Journal of Computational Physics*, 473:111739, 2023.
- [23] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.
- [24] Stefano Porziani, Corrado Groth, Witold Waldman, and Marco Evangelos Biancolini. Automatic shape optimisation of structural parts driven by bgm and rbf mesh morphing. *International Journal of Mechanical Sciences*, 189:105976, 2021.
- [25] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.
- [26] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [27] Werner C Rheinboldt. Numerical continuation methods: a perspective. *Journal of computational and applied mathematics*, 124(1-2):229–244, 2000.
- [28] Filippo Salmoiraghi, Angela Scardigli, Haysam Telib, and Gianluigi Rozza. Free-form deformation, mesh morphing and reduced-order methods: Enablers for efficient aerodynamic shape optimisation. *International Journal of Computational Fluid Dynamics*, 32(4-5):233–247, 2018.
- [29] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.
- [30] Tommaso Taddei. A registration method for model order reduction: Data compression and geometry reduction. *SIAM Journal on Scientific Computing*, 42(2):A997–A1027, 2020.
- [31] Tommaso Taddei. An optimization-based registration approach to geometry reduction. *arXiv:2211.10275*, 2022.
- [32] Tommaso Taddei. Compositional maps for registration in complex geometries. *arXiv:2308.15307*, 2023.
- [33] Christopher Ki Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [34] Mouadh Yagoubi, David Danan, Milad Leyli-Abadi, Jean-Patrick Brunet, Jocelyn Ahmed Mazari, Florent Bonnet, Asma Farjallah, Paola Cinnella, Patrick Gallinari, Marc Schoenauer, et al. Neurips 2024 ml4cfd competition: Harnessing machine learning for computational fluid dynamics in airfoil design. *arXiv preprint arXiv:2407.01641*, 2024.
- [35] Dongwei Ye, Valeria Krzhizhanovskaya, and Alfons G Hoekstra. Data-driven reduced-order modelling for blood flow simulations with geometry-informed snapshots. *Journal of Computational Physics*, 497:112639, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction state the problem setting and challenges. In addition, we have a contribution subsection in the introduction that summarize the method.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations are mentioned in sections 5.2,5.3, the common topology assumption multiple times in the paper. Some appendices discuss the computational cost and the robustness of the method with respect to some parameters.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We give a rigorous description of the optimization problem, mainly for the gradient algorithm in the infinite dimension settings and the use of continuation method.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: we describe the equations to reproduce the first dataset, and reference the second dataset in the references. The algorithm is described in all the necessary details, including a derivation of the gradient. We also proved values for the multiple parameters used.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: the datasets are reproducible/public. However, the code for the main method is not readily available to be published at the moment.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper presents numerical values to all the required parameters to solve the optimization problem.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper shows the effect of the optimal mappings presented on the decay of the eigenvalues of the correlation matrix, which is the core element of the paper. In addition, we provide an error comparison with respect to other methods.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: This is discussed in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research respects every aspect of the NeurIPS code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The paper discuss a generic algorithm to solve partial differential equations using data with no specific applications in consideration.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not pose such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper mention the dataset used that was not generated by the authors.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not release any new asset.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: The paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

699 • We recognize that the procedures for this may vary significantly between institutions
700 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
701 guidelines for their institution.
702 • For initial submissions, do not include any information that would break anonymity (if
703 applicable), such as the institution conducting the review.

704 **16. Declaration of LLM usage**

705 Question: Does the paper describe the usage of LLMs if it is an important, original, or
706 non-standard component of the core methods in this research? Note that if the LLM is used
707 only for writing, editing, or formatting purposes and does not impact the core methodology,
708 scientific rigorousness, or originality of the research, declaration is not required.

709 Answer: [NA]

710 Justification: The development of the methods in the paper did not make use of any LLM.

711 Guidelines:

712 • The answer NA means that the core method development in this research does not
713 involve LLMs as any important, original, or non-standard components.

714 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
715 for what should or should not be described.

716 **A Differential of $J_{\mathcal{H},r}$**

717 Let $\Phi = (\phi_i)_{1 \leq i \leq n} \in \mathbf{M}$, $\Psi = (\psi_i)_{1 \leq i \leq n}$ a "small" variation around Φ and $\epsilon \in \mathbb{R}, \epsilon \ll 1$. To
 718 evaluate the differential of $J_{\mathcal{H},r}$ at a point Φ , we evaluate $J_{\mathcal{H},r}$ at $\bar{\Phi} := \Phi + \epsilon\Psi$ and calculate:

$$DJ_{\mathcal{H},r}[\Phi][\Psi] = \lim_{\epsilon \rightarrow 0} \frac{J_{\mathcal{H},r}(\Phi + \epsilon\Psi) - J_{\mathcal{H},r}(\Phi)}{\epsilon} \quad (13)$$

719 We have:

$$\begin{aligned} \forall i, \quad h_i \circ (\phi_i + \epsilon\psi_i)(x) &= h_i(\phi_i(x) + \epsilon\psi_i(x)) \\ &\simeq h_i(\phi_i(x)) + \epsilon \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) \end{aligned}$$

720 where we neglect higher order terms. Next we evaluate :

$$\begin{aligned} C_{\mathcal{H},ij}[\Phi + \epsilon\Psi] &= \langle h_i \circ (\phi_i + \epsilon\psi_i), h_j \circ (\phi_j + \epsilon\psi_j) \rangle_{L^2(\Omega_0)} \\ &= \int_{\Omega_0} h_i \circ (\phi_i + \epsilon\psi_i)(x) h_j \circ (\phi_j + \epsilon\psi_j)(x) dx \\ &= \int_{\Omega_0} h_i \circ \phi_i(x) h_j \circ \phi_j(x) dx + \epsilon \int_{\Omega_0} h_i \circ \phi_i(x) \vec{\nabla} h_j(\phi_j(x)) \cdot \psi_j(x) dx \\ &\quad + \epsilon \int_{\Omega_0} \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) h_j \circ \phi_j(x) dx + \epsilon^2 \int_{\Omega_0} \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) \vec{\nabla} h_j(\phi_j(x)) \cdot \psi_j(x) dx \\ &:= C_{\mathcal{H},ij}[\Phi] + \epsilon DC_{\mathcal{H},ij}[\Phi][\Psi] + \epsilon^2 D^2 C_{\mathcal{H},ij}[\Phi][\Psi] \end{aligned} \quad (14)$$

721 and

$$\begin{aligned} Tr(C_{\mathcal{H}}[\Phi + \epsilon\Psi]) &= \sum C_{ii}[\Phi + \epsilon\Psi] \\ &= Tr(C_{\mathcal{H}}[\Phi]) + 2\epsilon \sum_{i=1}^n \int_{\Omega_0} h_i \circ \phi_i(x) \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) dx + \epsilon^2 \sum_{i=1}^n \int_{\Omega_0} [\vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x)]^2 dx \\ &= Tr(C_{\mathcal{H}}[\Phi]) + \epsilon Tr(DC_{\mathcal{H}}[\Phi][\Psi]) + \epsilon^2 D^2 C_{\mathcal{H}}[\Phi][\Psi] \end{aligned}$$

722 with $DC_{\mathcal{H},ij}[\Phi][\Psi] := \int_{\Omega_0} h_i \circ \phi_i(x) \vec{\nabla} h_j(\phi_j(x)) \cdot \psi_j(x) dx + \int_{\Omega_0} \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) h_j \circ \phi_j(x) dx$

723 and $DC_{\mathcal{H}}[\Phi][\Psi] = (DC_{\mathcal{H},ij}[\Phi][\Psi])_{1 \leq i,j \leq n}$.

724

725 Next we evaluate $D\lambda_i^{\mathcal{H}}[\Phi][\Psi]$. We have $\forall i$:

$$\|\zeta_i^{\mathcal{H}}[\Phi]\|^2 = 1$$

726 Now taking the differential on both sides, we get

$$\langle D\zeta_i^{\mathcal{H}}[\Phi][\Psi], \zeta_i^{\mathcal{H}}[\Phi] \rangle = 0$$

727 Using the fact that $C_{\mathcal{H}}[\Phi]\zeta_i^{\mathcal{H}}[\Phi] = \lambda_i^{\mathcal{H}}[\Phi]\zeta_i^{\mathcal{H}}[\Phi]$, we get again by taking the differential on both sides

$$DC_{\mathcal{H}}[\Phi][\Psi]\zeta_i^{\mathcal{H}}[\Phi] + C_{\mathcal{H}}[\Phi]D\zeta_i^{\mathcal{H}}[\Phi][\Psi] = D\lambda_i^{\mathcal{H}}[\Phi][\Psi]\zeta_i^{\mathcal{H}}[\Phi] + \lambda_i^{\mathcal{H}}[\Phi]D\zeta_i^{\mathcal{H}}[\Phi][\Psi]$$

728 We multiply the last equation by $\zeta_i^{\mathcal{H}}[\Phi]$ to get

$$(\zeta_i^{\mathcal{H}}[\Phi])^T DC_{\mathcal{H}}[\Phi][\Psi]\zeta_i^{\mathcal{H}}[\Phi] + 0 = D\lambda_i^{\mathcal{H}}[\Phi][\Psi] + 0$$

729 Thus we have finally

$$\lambda_i^{\mathcal{H}}[\Phi + \epsilon\Psi] = \lambda_i^{\mathcal{H}}[\Phi] + (\zeta_i^{\mathcal{H}}[\Phi])^T DC_{\mathcal{H}}[\Phi][\Psi]\zeta_i^{\mathcal{H}}[\Phi] \quad , \forall 1 \leq i \leq n \quad (15)$$

730 Taking the sum over the first r eigenvalues, we obtain:

$$\sum_{j=1}^r \lambda_j^{\mathcal{H}}[\Phi + \epsilon\Psi] = \sum_{j=1}^r \lambda_j^{\mathcal{H}}[\Phi] + \epsilon Tr((Z_r^{\Phi})^T DC_{\mathcal{H}}[\Phi][\Psi] Z_r^{\Phi}) \quad (16)$$

731 with $Z_r^\Phi = (\zeta_1^\Phi, \zeta_2^\Phi, \dots, \zeta_r^\Phi)^T$. Now we can evaluate (13) to obtain:

$$DJ_{\mathcal{H},r}[\Phi][\Psi] = \frac{Tr((Z_r^\Phi)^T DC_{\mathcal{H}}[\Phi][\Psi] Z_r^\Phi)}{Tr(C_{\mathcal{H}}[\Phi])} - \frac{\sum_{k=1}^r \lambda_k^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])^2} \times Tr(DC_{\mathcal{H}}[\Phi][\Psi]) \quad (17)$$

732 which can be written explicitly as

$$\begin{aligned} DJ_{\mathcal{H},r}[\Phi][\Psi] &= \frac{2}{Tr(C_{\mathcal{H}}[\Phi])} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^r \zeta_{k,i}^{\mathcal{H}}[\Phi] \zeta_{k,j}^{\mathcal{H}}[\Phi] \int_{\Omega_0} h_j \circ \phi_j(x) \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) dx \\ &\quad - \frac{2 \sum_{k=1}^r \lambda_k^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])^2} \sum_{i=1}^n \int_{\Omega_0} h_j \circ \phi_j(x) \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) dx \\ &= \sum_{i=1}^n DJ_i[\Phi][\psi_i] \end{aligned}$$

733 with

$$\begin{aligned} DJ_i[\Phi][\psi_i] &:= \sum_{k=1}^r \left(\frac{2\zeta_{k,i}^{\mathcal{H}}[\Phi]^2}{Tr(C_{\mathcal{H}}[\Phi])} - \frac{2\lambda_k^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])^2} \right) \int_{\Omega_0} h_i \circ \phi_i(x) \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) dx \\ &\quad + \sum_{\substack{j=1 \\ j \neq i}}^n \sum_{k=1}^r \frac{2\zeta_{k,i}^{\mathcal{H}}[\Phi] \zeta_{k,j}^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])} \int_{\Omega_0} h_j \circ \phi_j(x) \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) dx \\ &= \sum_{j=1}^n \sum_{k=1}^r \left(\frac{2\zeta_{k,i}^{\mathcal{H}}[\Phi] \zeta_{k,j}^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])} - \frac{2\lambda_k^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])^2} \delta_{ij} \right) \int_{\Omega_0} h_j \circ \phi_j(x) \vec{\nabla} h_i(\phi_i(x)) \cdot \psi_i(x) dx, \end{aligned} \quad (18)$$

734 and δ_{ij} is the Kronecker delta. For simplicity, we note the term

$$\mathbf{f}_i[\Phi] := \sum_{j=1}^n \sum_{k=1}^r \left(\frac{2\zeta_{k,i}^{\mathcal{H}}[\Phi] \zeta_{k,j}^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])} - \frac{2\lambda_k^{\mathcal{H}}[\Phi]}{Tr(C_{\mathcal{H}}[\Phi])^2} \delta_{ij} \right) h_j \circ \phi_j(x) \vec{\nabla} h_i(\phi_i(x)), \quad (19)$$

$\mathbf{F}[\Phi] := (\mathbf{f}_1[\Phi], \dots, \mathbf{f}_n[\Phi])$, and we write

$$DJ_{\mathcal{H},r}[\Phi][\Psi] = \sum_{i=1}^n \int_{\Omega_0} \mathbf{f}_i[\Phi] \cdot \psi_i dx.$$

735 B Differential of $I_{\mathcal{H},r}$

We define the objective function $I_{\mathcal{H},r}$ on M_0^n as

$$I_{\mathcal{H},r}[\Phi] := J_{\mathcal{H},r}[\Phi] - c_1 \sum_{i=1}^n E[\phi_i].$$

As mentioned in section 5, the energy term used in this work in the linear elastic energy defined as

$$E[\phi_i] := \frac{1}{2} a(\phi_i - \mathbf{Id}, \phi_i - \mathbf{Id}),$$

where we recall the definition of a in (8). We can easily show that

$$DE[\phi_i][\psi] = a(\phi_i - \mathbf{Id}, \psi).$$

736 Using the linearity of the differential, we obtain the differential of $I_{\mathcal{H},r}$ as

$$\begin{aligned}
DI_{\mathcal{H},r}[\Phi][\Psi] &= DJ[\Phi][\Psi] - c_1 \sum_{i=1}^n DE[\phi_i][\psi_i] \\
&= \sum_{i=1}^n \int_{\Omega_0} \mathbf{f}_i[\Phi] \cdot \psi_i dx - c_1 \sum_{i=1}^n DE[\phi_i][\psi_i] \\
&= \sum_{i=1}^n \left(\int_{\Omega_0} \mathbf{f}_i[\Phi] \cdot \psi_i dx - c_1 DE[\phi_i][\psi_i] \right) \\
&:= \sum_{i=1}^n DI_i[\Phi][\psi_i].
\end{aligned}$$

In order to obtain equation (11), we compute the Riesz representation of $DI_i[\Phi]$, denoted as $\bar{\mathbf{u}}_i$, with respect to the inner product a , the unique solution to

$$a(\bar{\mathbf{u}}_i, \psi) = \int_{\Omega_0} \mathbf{f}_i[\Phi] \cdot \psi_i dx - c_1 a(\phi_i - \mathbf{Id}, \psi),$$

for all test function ψ . Since we also have $\mathbf{u}_i$ as the Riesz representation of $DJ_i[\Phi]$ ($a(\mathbf{u}_i, \psi) = DJ_i[\Phi][\psi]$), we obtain

$$a(\bar{\mathbf{u}}_i, \psi) = a(\mathbf{u}_i, \psi) - c_1 a(\phi_i - \mathbf{Id}, \psi).$$

737 which is true for all test function ψ . Thus, by the linearity of a , we obtain $\bar{\mathbf{u}}_i = \mathbf{u}_i - c_1(\phi_i - \mathbf{Id})$,
738 and hence equation (11).

739 C Continuation method

We give here a quick description on the continuation methods. Readers can refer to [1, 27] and references within for more details.

Continuation methods are a variety of methods used to solve equations of the type

$$G(x) = 0$$

where G is a non-linear function, and x is the unknown. Continuation methods rely on solving a succession of problems of the form

$$H(x, \lambda) = 0$$

740 where H is another non-linear function and $H(x, \lambda^*) = G(x)$ for some λ^* . λ is called the continua-
741 tion parameter. In the simplest form of a continuation method, we start by choosing a set a values
742 $\{\lambda^0, \lambda^1 \dots \lambda^*\}$ for λ . We then solve $H(x^i, \lambda^i) = 0$, and initialize $H(x^{i+1}, \lambda^{i+1}) = 0$ with x^i .

743 C.1 Continuation for c_1

744 In order to solve equation (7), we proceed by using the continuation method as follows:

- 745 1. We start by solving equation the optimization problem, using the gradient algorithm, for
746 $c_1 = c_1^0$, where c_1^0 should be sufficiently large in order to converge properly. This will
747 produce the function Φ^0 .
- 748 2. At each iteration k , we set $c_1^k := \frac{c_1^{k-1}}{2}$, and we resolve the problem for Φ^k by initializing
749 $\Phi^{(0),k} = \Phi^{k-1}$.
- 750 3. We iterate over k as long as there are no elements of the morphed meshes are inverted.

751 We give a few comments about the above procedure. First, we note that there are two loops for the
752 algorithm: the outer loop over k that updates c_1 , and the inner loop that solves the non-linear equation
753 using the gradient algorithm. Second, ideally we want to solve (7) for $c_1 = 0$. If, while iterating over
754 the outer loop, c_1^k gets sufficiently small, we can set $c_1^{k+1} = 0$. Finally, we note that full convergence
755 for intermediate results of Φ^k is generally not necessary.

756 C.2 Continuation for c_2

757 Applying continuation with two or more parameters proves to be more complicated for the case of
 758 one parameter. To this end, we propose in this paper to alternate changing the values of the two
 759 parameters c_1 and c_2 . We proceed in the same manner as above, expect that:

- 760 1. At iteration $2k$, we set $c_1^{2k} := \frac{c_1^{2k-1}}{2}$.
- 761 2. We start with a small value for c_2^0 . At iteration $2k + 1$, we set $c_2^{2k+1} := 10 \times c_2^{2k}$.

762 C.3 Initialization and scheduling strategy

763 In our numerical examples, we find that the method is not sensitive for the initial choices for both
 764 parameters as expected. In fact, choosing a large value for c_1 would force the mapping to be close to
 765 the identity map. In this case, the mesh would not deform by much and the algorithm would converge
 766 quickly, and in return we change the value of c_1 . On the other hand, choosing a small value of c_2
 767 would have a similar effect in terms of convergence. A small value would diffuse very much the
 768 values of the field in the domain, and they approach a constant field allowing to converge rapidly and
 769 to change the values of c_2 in return.

770 The scheduling strategy used to change to values of c_1 and c_2 is a simple heuristic that we observed
 771 to work well with the tests provided. Other strategies would be to change both values at the same
 772 time, which also worked well for the same datasets. However, we do mention that other continuation
 773 methods exists when dealing with more than one parameter as in the references provide. These
 774 methods are developed for non linear problems in finite dimension and could be potentially harder to
 775 implement in functional settings as in this work.

776 D Computational cost of the optimal mapping

777 The training phase of O-MMGP consists of i) computing ϕ_i^{geo} , ii) computing ϕ_i^{opt} , iii) performing
 778 POD and iv) training of the gaussian processes to learn ϕ^{opt} and u^{opt} . With respect to MMGP, this
 779 would add step iii, and one GP training (for ϕ^{opt}). For the AirFRANS dataset, steps i, iii, and iv take
 780 about 40 seconds. Step ii consists also of multiples steps (remeshing the reference mesh to tackle
 781 computational bottlenecks (see below), the optimization process, computing $\tilde{u}(\alpha)$ and $\nabla \tilde{u}(\alpha)$...). It
 782 takes about 35 minutes (we do mention however that some of these steps could be implemented in
 783 parallel which would greatly decrease the optimal mappings computational time).

784 Inference time: the inference takes about 81 seconds for 696 samples (200 test and 496 OOD). Both
 785 MMGP and NeurIPS 2024 solutions are faster to train (no optimization problem in offline), however
 786 they are both more expensive than O-MMGP in online. All the simulations are performed using
 787 128 CPU cores and no GPU. In comparison, the full order solver takes about 25 min for each full
 788 simulation.

789 Acceleration strategies

790 Solving the optimal mappings optimization problem requires evaluating $F_i[\Phi]$ (thus u_i and its gradi-
 791 ent) at the integration points of the (deformed) reference mesh, the assembly of the stiffness matrix
 792 (once and for all for all the iterations), and solving the linear systems for the Riesz representation.
 793 Potential computational bottlenecks come from having a very fine reference mesh, which increases
 794 the degrees of freedom and the number of integration points. In this paper, we used an acceleration
 795 strategy consisting of using a coarse reference mesh to compute the optimal mappings, then the
 796 mappings are evaluated on the fine mesh using interpolation. This greatly decrease optimization
 797 time while giving accurate results. To further accelerate computation, when needed, other techniques
 798 can be used like reduced quadrature formulae or randomized linear algebra. However, this was not
 799 needed for the provided examples.

800 E RBF morphing

801 In radial basis function (RBF) interpolation, we aim to approximate a scalar-valued function $g(x)$
 802 using a linear combination of radial basis functions as follows:

$$g(x) \approx \sum_{i=1}^{n_c} a_i \xi(\|x - x_i\|), \quad (20)$$

803 where $a_i \in \mathbb{R}$ and ξ is a radial basis function that depends only on the Euclidean distance between x
 804 and a control point x_i . The control points $\{x_i \mid i = 1, \dots, n_c\}$ are specific locations where the exact
 805 values of $g(x)$, denoted g_i , are known. To determine the coefficients a_i , we enforce the interpolation
 806 condition at all control points, yielding the system:

$$\sum_{i=1}^{n_c} a_i \xi(\|x_j - x_i\|) = g_j, \quad j = 1, \dots, n_c. \quad (21)$$

807 This can be expressed in matrix form as:

$$\mathbf{K}\mathbf{a} = \mathbf{g}, \quad (22)$$

808 where $\mathbf{K}$ is an $n_c \times n_c$ interpolation matrix with entries $K_{ij} = \xi(\|x_j - x_i\|)$, $\mathbf{a} = [a_1, \dots, a_{n_c}]^T$ is
 809 the vector of coefficients, and $\mathbf{g} = [g_1, \dots, g_{n_c}]^T$ is the vector of known values. Solving this linear
 810 system yields the coefficients a_i , which can then be used to compute $g(x)$ at any point x .

811 In the context of radial basis function-based mesh morphing, RBF interpolation is applied separately
 812 to each component of the morphing function $\phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x))$ in 3D (or similarly in
 813 2D). The control points are usually chosen as boundary points. The interpolated morphing function is
 814 then used to smoothly deform the interior of the mesh according to the control point displacements.