Structural and Multidisciplinary Optimization (2025) 68:57
https://doi.org/10.1007/s00158-025-03976-2

RESEARCH PAPER

A t-SNE-based embedding for transfer optimisation with non-overlapping design variables

Petru-Cristian Cimpoesu[1] · David J. Toal[1] · Leran Wang[1] · Andy J. Keane[1] · Jonathan Gregory[2] · Marco Nunez[2]

Received: 9 June 2024 / Revised: 18 December 2024 / Accepted: 29 January 2025 / Published online: 29 March 2025

Abstract
The cold start problem is a chief concern in the context of surrogate-based optimisation, as it can slow down or prevent convergence towards a global minimum. Transfer optimisation (TO) has recently emerged as a promising solution, positing the reuse of historical data to improve the quality of the surrogate predictor. However, the requirement for constant design parameters across the source and target tasks severely limits the range of applicability of TO. Several strategies have been proposed to overcome this constraint. However, they typically require either linked samples or linked design variables, and thus only offer a slight extension of the aforementioned scope. This paper proposes a new transfer optimisation method that enables varying design parameters. It removes the link constraint by using simulation physics, rather than a mapping function, to represent the distribution of source and target samples. Then, it employs a t-SNE inspired optimisation routine to recreate this distribution in the target task's design variable space. Multiple-output Gaussian processes are used to model the resulting distribution of target and source samples. Results indicate significant improvements of 30-60% in optimisation performance over traditional Kriging-based approaches.

Keywords T-SNE · Transfer optimisation · Kriging · Surrogate modelling

1 Introduction

The fundamental task of engineering design optimisation involves identifying a set of design variables which define a parametric geometry such that an optimal value of some objective, f, is found. Global methods, such as genetic algorithms (GAs) or local optimisers, gradient descent, are examples of tools commonly used for the task.

However, in most practical cases, the cost of obtaining the value of the objective function makes direct evaluation infeasible. To address such scenarios, it is common to leverage machine learning models that build an approximation, $\hat{f}$, of the underlying function. The key is that the model is cheap to evaluate and thus can be used to search for optimal designs rapidly. This solution is called surrogate-based optimisation (SBO), originally introduced by Jones et al (1998), and draws its name from the idea that the model, $\hat{f}$, acts as a surrogate representation of the physical process used to determine f.

The primary limitation of SBO stems from the prohibitive cost of evaluating the objective function, which implies that surrogate models often start with a limited number of training samples X. This gives rise to the curse of dimensionality, described in detail by Bellman and Kalaba (1959), which is particularly potent in engineering problems, where, under the underlying nonlinear physics, the behaviour of the objective function is, more often than not, multi-modal.

The curse of dimensionality is most commonly addressed through adaptive sampling, also known as sequential model-based sampling. This involves first tuning the surrogate model $\hat{f}$ using the sparse initial sample. Infill points are then added to the training dataset, based on a sub-optimisation problem, where an auxiliary criterion $S(\hat{f};x)$ is either maximised or minimised. Infill strategies commonly involve balancing exploration (improving model accuracy) and exploitation (finding the optimal design). Still, it is important to note that how best to choose update points is often

Communicated by Souma Chowdhury.

✉ Petru-Cristian Cimpoesu
P-C.Cimpoesu@soton.ac.uk

[1] Faculty of Engineering & Physical Sciences, University of Southampton, Boldrewood Innovation Campus, Southampton SO16 7QF, Hampshire, UK

[2] Rolls-Royce plc, Bristol BS80 8AH, UK

problem-dependent. A taxonomy of existing surrogate modelling techniques and infill criteria are discussed and compared in works by Rojas-Gonzalez and Van Nieuwenhuyse (2020), Bartz-Beielstein et al (2020) or Zhan and Xing (2020).

While extremely useful in various applications, surrogate-based strategies are imperfect and present several limitations. The focus of this paper is the cold start problem, highlighted by Min et al (2017), which is briefly described. Since infill criteria are mathematically derived from the error of the surrogate model, the search for infill points, which involves a sub-optimisation problem, is also exposed to the curse of dimensionality. As Forrester et al (2008) pointed out, an unlucky initial sample compounded by a deceptively positioned minimum can trick the SBO framework, leading to a failure in converging to the true optimum. More commonly, however, users are faced with a slow start and exhaustion of the computational budget before the optimum can be discovered. More comprehensive infill sampling at each step can solve this problem, but this involves an undesirable use of the computational budget.

A key assertion of this work is that the slow start problem can best be mitigated through transfer optimisation (TO) Gupta et al (2018). TO consists of leveraging correlations between similar design tasks to transfer knowledge from a source (or historical) design task to a target (or new) one. This approach draws appeal from emulating how engineers think when designing innovative solutions. Previous research has highlighted the great extent to which engineers draw upon prior experience and knowledge of established systems Wills and Kolodner (1994).

The concept of knowledge transfer has been successfully applied in a range of scientific domains on problems with constant and varying parameterisation. These are reviewed in section 2. The surrogate models used in the optimisation framework are discussed in section 3. The current method is described in section 4. Section 5 showcases the application of the method on an illustrative example and discusses several practical problems and how they are addressed. Sections 6 and 7 showcase the performance of the approach on more realistic design optimisation problems. Then, section 8 offers details on the hyper-parameters of the algorithm and gives guidelines for their selection.

2 Review of knowledge transfer methods and applications

In the following section, transfer optimisation, a conceptual framework for knowledge transfer, is introduced. In addition, several difficulties associated with its efficient implementation are discussed.

Formulated by Gupta et al (2018), transfer optimisation considers a series of $\mathcal{K}$ optimisation tasks $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_\mathcal{K}$, consisting of domains $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_\mathcal{K}$. A domain $\mathcal{D}_k$ consists of the search space $\mathcal{X}_k$ and the auxiliary space $\mathcal{Y}_k$, defined as the set of all operating conditions.

A surrogate-based optimisation problem which uses knowledge transfer may be framed as

$$\begin{aligned} \min_{x_k} \quad & f_k\left(x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_k\right) \\ \text{s.t.} \quad & g_k\left(x_1, x_2, \dots, x_k, z_1, z_2, \dots, z_k\right) \leq 0 \\ \text{and} \quad & h_k\left(x_1, x_2, \dots, x_k, w_1, w_2, \dots, w_k\right) = 0 \end{aligned} \tag{1}$$

Here, f_k is the predictor of the objective function, g_k is the predictor of some inequality constraint, and h_k is the predictor of some equality constraint. The vector x_i represents the descriptor variables for task i, while the scalars y_i, z_i, and w_i represent the true objective and constraint functions.

The goal is to accelerate the optimisation of task $\mathcal{T}_k$ by building a predictor which not only makes use of the pair x_k and y_k but also uses those of previously completed tasks $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_{k-1}$. This realisation of the transfer optimisation paradigm is known as sequential transfer and is the focus of this paper. Sequential knowledge transfer problems can be further split into two types, which are described in the following two sections.

2.1 Knowledge transfer with constant parameterisation

The first type of sequential transfer involves problems, where $\mathcal{X}_1 = \mathcal{X}_2 = \cdots = \mathcal{X}_\mathcal{K}$, that is, the design variables describing the objective f_k are constant. Such problems are sometimes referred to as homogeneous knowledge transfer problems.

In such instances, it is important to note that a change in the scale or behaviour of the objective function may occur. For example, a change in the material of a truss from steel to aluminium will produce a step change in the values of an objective such as mass or deflection. In addition, subtle changes in the topology of the design space may occur due to nonlinear behaviour. Although the shapes of the two objective functions may remain qualitatively similar, such changes render traditional Gaussian processes unsuitable, calling for a more elaborate approach.

A possible solution involves normalising the objective function values through latent-ranker-based GPs, described at length by Chu and Ghahramani (2005). The algorithm assumes that there exists a set of unknown latent function values $\{f(x_i), f(x_j)\}$ which, given some preference relationship $x_i \succ x_j$, preserve the ranking of the training samples. The resulting ranking function is independent of the scale of the responses that are used to build it and thus can be used as a constant across design tasks. Bardenet et al (2013)

proposed a collaborative tuning strategy using the ranker-based GP, which successfully tuned hyper-parameters for two algorithms across a series of datasets. Here, the tasks consisted of multiple datasets for the same algorithm (e.g. 20 classification problems for an adaptive boosting algorithm).

Another way of addressing changes in the objective function values is through treed Gaussian processes (TGPs), proposed by Gramacy and Lee (2009). Although originally designed for modelling non-stationary functions, TGPs were later enhanced to handle datasets with a mixture of qualitative and quantitative variables Gramacy and Taddy (2010). However, since no explicit use of the correlations between tasks is made, the algorithm is not suited for knowledge transfer tasks. Indeed, a study by Swiler et al (2014) proved that TGPs can perform worse than building separate GPs for each task.

A more popular approach in the area of engineering design involves the fusion of source and target samples into a single model, usually by adopting a block covariance structure. Some of the more prominent approaches in the literature are the multiple-output Gaussian Processes (MOGPs) of Bonilla et al (2007) and the Co-Kriging algorithm, introduced by Kennedy and O'Hagan (2000). MOGPs have found widespread utility in problems featuring mixed continuous and discrete variables, with successful applications including meta-material design Tran et al (2019), solid rocket engine design Pelamatti et al (2019), and centrifugal pump design Renhui and Zhao (2020). On the other hand, the hierarchical nature of Co-Kriging is especially well suited for problems, where the tasks consist of varying levels of fidelity for the same geometry. Illustrative examples encompass multi-disciplinary gas turbine design Toal et al (2014), robust aerodynamic design Tao and Sun (2019), and civil building shape design Ding and Kareem (2018). Recently, Toal (2023) and Lin et al (2021) showed that a blend consisting of multi-fidelity, multiple-output Gaussian processes could also be effective for a wide range of problems, such as aerofoil design, structural design, or combustor optimisation.

In this paper, Kriging is used as a baseline, and MOGPs are used for knowledge transfer. Aside from their prominence in related literature, these approaches also provide a useful error metric, which can be used to derive powerful infill strategies. The mathematical formulation of the algorithms is presented in section 3.

2.2 Knowledge transfer with varying parameterisation

Problems involving varying parameterisations are more representative of realistic design scenarios since engineers commonly add, remove, or change variables as their understanding of a design evolves over time. Models such as MOGPs assume a homogeneous definition of the variables in each categorical level, so an additional step is required to model such cases. Most commonly, this step consists of transforming the variables or projecting them onto a common latent space.

A widely adopted algorithm that addresses this problem is space mapping (SM) Bandler et al (1994). Robinson et al (2008) employed a customised SM variant, integrating the method with trust region model management (TRMM) and a 2^{nd}-order correction. Their implementation showcased significant optimisation speed enhancements, yielding a 40% improvement for a bat flight design problem and a 70% acceleration for a wing design problem.

An alternative to Space Mapping is the Input Mapping Calibration (IMC) of Tao et al (2019). In contrast to the optimisation method used by SM, IMC optimises the linear map globally and incorporates a nominal mapping to regularise the cost function. In the context of a wing design problem, the IMC method achieved a notable result: it automatically adjusted the wing twist angles to compensate for the low-fidelity representation of the geometry.

Hebbal et al (2021) employed a two-layered deep Gaussian process to model multi-fidelity problems, where the fidelities are of varying parameterisations. In their first layer, a multiple-output Gaussian process mapped higher fidelity inputs into lower fidelities such that all samples were eventually mapped onto the lowest fidelity manifold. Then, the second layer constructed the multi-fidelity Gaussian process. The authors observed improvements in predictive accuracy across various analytical and engineering test problems, especially when the number of samples for the target task was limited.

A notable limitation of the approaches presented above is the requirement for linked samples. A set of samples x_{source} and x_{target} are linked if there exists a function that maps one onto the other: $f : x_{source} \mapsto x_{target}$. In other words, a new assumption is added, namely that the two sets of data represent the same sample, albeit with different parameters. For practical applications, this entails running a number of expensive simulations to generate the linked data $x^L_{target}, y^L_{target}$, an undesirable use of a computational budget.

An alternative is the transfer Bayesian optimisation (TBO) framework proposed by Min et al (2021), which is based upon the manifold Gaussian process regression (mGP) of Calandra et al (2016). The mGP considers a multiple-output Gaussian process consisting of a composite function: $F = G \circ M$, where G is the Gaussian process, M is a mapping function, and the operator $\circ$ is used to represent a composite function, here $G(M(x))$. In the case of the TBO framework, the mapping function is a single-layer neural network with a ReLU activation and applies the transformation to the source data. The covariance of the GP takes the form $\tilde{k}(x^{(i)}_{source}, x^{(j)}_{target}) = k(M(x^{(i)}_{source}), x^{(j)}_{target})$.

Since the weights of the neural network are given as priors to the Gaussian process, they are included in the likelihood optimisation process, thus removing the requirement for linked samples. However, the method has only been demonstrated on problems, where the target parameterisation is a subset of the source parameterisation, which ensures the presence of latent correlations between the two tasks. The method presented in this paper allows for an extension to the more general case, where no overlap between the two sets of design variables exists.

3 Surrogate models

In this section, the Kriging and categorical-Kriging algorithms are briefly described. For the remainder of the paper, the two are used as regression models for the baseline and proposed sequential optimisation frameworks.

3.1 Kriging

The Kriging model is a member of the Gaussian Process (GP) family introduced into engineering design by Sacks et al (1989). Intuitively, it uses each sample in the training set as the centre of a Gaussian distribution. Mathematically, this entails representing the predictor as the addition of a mean vector and a row vector from a covariance matrix

$$\hat{f}(x_i) = \mu_i + \Sigma_i. \tag{2}$$

The choice of kernel function for the covariance matrix depends on the nature of the problem, but in the context of regression, the p-exponential formulation is common and is defined as

$$K(x_i, x_j \mid \theta, p) = \sum_{d=1}^{D} \exp\{-10^{\theta_d} \|x_i - x_j\|^{p_d}\}. \tag{3}$$

Here, D is the number of variables or features. The covariance matrix is then defined by considering all combinations of n input vectors

$$\Sigma = \begin{pmatrix} K(x_1,x_1) & \dots & K(x_1,x_n) \\ \vdots & \ddots & \vdots \\ K(x_n,x_1) & \dots & K(x_n,x_n) \end{pmatrix} + \begin{pmatrix} \lambda & \dots & 0 \\ 0 & \ddots & \vdots \\ 0 & \dots & \lambda \end{pmatrix} \tag{4}$$

The scalar λ is added to the diagonal of the covariance matrix as a regularisation term. The loss function is defined as the natural logarithm of the probability density function, also known as the log-likelihood, and can be written as

$$L(y|x,\theta,p,\lambda) = -\frac{n}{2}\ln 2 - \frac{n}{2}\ln(\sigma^2) - \dots - \frac{1}{2}\ln\det\Sigma - \frac{(y-1\mu)^T\Sigma^{-1}(y-1\mu)}{2\sigma^2}. \tag{5}$$

The σ term is the standard deviation of the vector of responses y. It, along with the mean of the distribution μ, can be directly estimated by calculating the stationary points of the loss function with respect to each hyper-parameter. This leads to the maximum likelihood estimators,

$$\hat{\mu} = \frac{1^T\Sigma^{-1}y}{1^T\Sigma^{-1}1}, \tag{6}$$

and

$$\hat{\sigma}^2 = \frac{(y-1\mu)^T\Sigma(y-1\mu)}{n}. \tag{7}$$

The quantities obtained in Eqs. 5 and 6 allow the simplification of the loss function,

$$L(y|x,\theta,p,\lambda) = -\frac{n}{2\pi}\ln(\sigma^2) - \frac{1}{2}\ln\det\Sigma. \tag{8}$$

Predictions on a new sample x_i are defined as

$$\hat{f}(x_{new}) = \mu - \psi^T\Sigma(y-1\mu). \tag{9}$$

Here, ψ is a new covariance matrix consisting of the distance between a set of new points x_{new} and the training sample X. The model and optimisation algorithm are implemented using the OPTIMAT CM02 toolbox (Toal et al (2008), Toal et al (2009), Toal et al (2011)). This uses a particle swarm optimiser (PSO) enhanced by a local terminal search. A key advantage of Kriging is that it offers an error metric for any new prediction, which can be used as an infill criterion or as a basis for other infill criteria, such as the probability of improvement or the expected improvement and is defined as

$$\hat{s}^2(x_{new}) = \sigma^2\left[1 - \psi^T\Psi^{-1}\psi + \frac{1 - 1^T\Psi^{-1}\psi}{1^T\Psi^{-1}1}\right]. \tag{10}$$

3.2 Categorical Kriging

The categorical-Kriging model is a mixed kernel method, combining traditional continuous kernels, such as the p-exponential of Eq. 4, with discrete kernels, whose purpose is to describe the correlation between categorical levels or data sources.

There is extensive literature on which combinations of discrete and continuous kernels should be used for various applications. Categorical Kriging is known in the mathematical

literature as the intrinsic model of coregionalisation (IMC) or unrestricted covariance model (UC) Bonilla et al (2007). In simple terms, it uses a matrix of correlations between discrete levels defined as

$$\tilde{K}(w^{(i)}, w^{(j)}) = \tau_{c_i,c_j} K(x_i, x_j). \tag{11}$$

Here, τ_{c_i,c_j} is a term describing the correlation between the categorical levels of the vectors x_i and x_j.

To effectively apply the correlations $\tau_{i,j}$, the hyper-sphere method of Zhou et al (2011) is used. The first step involves constructing a discrete kernel T using a lower diagonal matrix L to ensure positive, semi-definiteness. This would have the form in Eq. 11 in a case with three discrete levels.

$$T = LL^T = \begin{pmatrix} 1 & \tau_{1,2} & \tau_{1,3} \\ \tau_{1,2} & 1 & \tau_{2,3} \\ \tau_{1,3} & \tau_{2,3} & 1 \end{pmatrix}. \tag{12}$$

Here, the matrix L would have the following form:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \cos\theta_{2,1} & \sin\theta_{2,1} & 0 \\ \cos\theta_{3,1} & \sin\theta_{3,1}\cos\theta_{3,2} & \sin\theta_{3,1}\sin\theta_{3,2} \end{pmatrix} \tag{13}$$

And the terms of the matrix L (and therefore T) can be generalised using the formulation

$$L(\Theta) = \begin{cases} l_{1,1} = 1 \\ l_{2,1} = \cos(\theta_{2,1}) \\ l_{2,2} = \sin(\theta_{2,1}) \\ l_{k,1} = \cos(\theta_{2,1}) \\ l_{k,s} = \sin(\theta_{k,1}) \ldots \sin(\theta_{k,s-1})\cos(\theta_{k,s}) \\ l_{k,k} = \sin(\theta_{k,1}) \ldots \sin(\theta_{k,k-2})\cos(\theta_{k,k-1}) \end{cases} \tag{14}$$

with $k > 2$ and $s = 2, \ldots, k-1$. To expand the definition to multiple quantitative variables, a separate T_k is built for each category, then,

$$T = \prod_{k=1}^{R} T_k. \tag{15}$$

Here, R is the number of categorical features or tasks. To build the remainder of the algorithm, the same methodology as in section 3.1 can be used, where the model covariance has the following form:

$$\tilde{K} = T \otimes K. \tag{16}$$

The hyper-parameters here are optimised using the method described in Toal (2023).

4 A t-SNE-based embedding

This section describes the method proposed for aligning the heterogeneous parameterisations of a target and source task. Rather than providing an explicit link between design variables as a prior, the relationship is learned from the problem's physics. Specifically, physical measurements are collected for each sample of the target and source tasks. After some manipulation, the distribution of these samples is assessed in a latent physics-based space and then reproduced in the original variable space.

A key assumption is that the underlying physics of the source and target tasks does not change, and thus can function as a dependable bridge between two or more parameterisations.

The three-step methodology is outlined as follows:

1. **Constructing the physics matrices**; $Z^{d\times n}_{source}$ and $Z^{d\times m}_{target}$ are used to represent a consistent field of physical measurements drawn from the simulation that was used to calculate the quantity of interest. Here, d is the number of measured physical quantities, m is the number of samples in the target, and n is the number of samples in the source data.
2. **Building a graph representation of the physics space**: this step uses a representation similar to that of the t-SNE algorithm of van der Maaten and Hinton (2008) with a few modifications that are detailed in section 4.2
3. **Aligning the source data with the target data in the design variable space**: once the relationship between target and source tasks is represented using a latent physics space, the source data are embedded onto a hypercube bounded by [0,1], which corresponds to the target data manifold. The algorithm must find an embedding that positions the source data around the target data design of experiments such that the probability distribution matches that found in step 2. The mathematics of this process are described in section 4.3.

4.1 Constructing the physics matrices

Employing physics matrices draws inspiration from Perron et al (2021), where physics measurements from low-fidelity models are used to generate more accurate high-fidelity field predictions. The inconsistency of the finite volume meshes between the two levels of fidelity is addressed by separately collecting physics measurements from each set of cells and applying proper orthogonal decomposition (POD). The resulting latent spaces are aligned using an affine transformation, whose weights and biases are

determined by solving the orthogonal Procrustes problem Gower and Dijksterhuis (2004). The transformation, however, requires a set of linked observations for training, described in Sect. 2.2 as a key limitation of existing knowledge transfer methods since it involves an undesirable use of the computational budget.

The current method enforces a constant-grid constraint between tasks to address this. While this merely shifts the hindrance to the meshing process, it is argued that any significant difficulties associated with generating constant-grid measurements do not offset the cost–benefit of dropping the requirement for linked observations. This is justified by the availability of adequate interpolation methods in most modern CFD and FEA codes, where values can be easily obtained for a pre-defined constant grid. The accuracy of these interpolation methods is of secondary importance, provided their integral is well correlated to the quantity of interest. In contrast, the study of Perron et al (2021) required highly accurate measurements since their objective was predicting intricate aerodynamic flow fields.

Once a constant grid of physics measurements is achieved, the matrices $Z^{d\times n}_{source}$ and $Z^{d\times m}_{target}$ are constructed and independently normalised to have mean zero and a standard deviation of 1.

4.2 Building a graph representation of the physics space

The t-distributed stochastic neighbour embedding algorithm builds a latent, low-dimensional representation of a high-dimensional manifold. The initial step entails formulating the following model for pairwise similarities in the high-dimensional space:

$$p_{i|j} = \frac{\exp\left(-||z_i - z_j||^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-||z_i - z_k||^2/2\sigma_i^2\right)}. \tag{17}$$

In this formulation, outliers will generate very small probability values $p_{i|j}$, which will have a negligible impact on the cost function. An additional step transforms p into a joint probability, thereby ensuring that all contributions are brought to a comparable scale,

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}. \tag{18}$$

Here, n is the number of samples in z. The objective function of the t-SNE algorithm minimises the difference between the distribution in Eq. 16 and an embedding represented by

$$q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{k\neq l}\left(1 + ||y_k - y_l||^2\right)^{-1}}. \tag{19}$$

The goal of the current method is to emulate this process in a manner where the physics space is represented by p_{ij} and a design variable hypercube bounded by [0,1] is represented by q_{ij}.

The formulation in Eq. 16 is used as a starting point. Let n and m represent the number of samples in the source and target tasks, and let Z and X represent the physics and design variable matrices, respectively. The probability in the physics space can be expressed as

$$p^{physics}_{i|j} = \frac{\exp\left(-||z_i - z_j||^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-||z_i - z_k||^2/2\sigma_i^2\right)}. \tag{20}$$

The matrix $p^{physics}_{i|j}$ will have a block covariance form, where the terms can be conveniently expressed as

$$p^{physics}_{i|j} = \begin{pmatrix} p_{T|T} & p_{T|S} \\ p_{S|T} & p_{S|S} \end{pmatrix}. \tag{21}$$

Here, the indices T and S correspond to target and source samples, respectively. To find the variances σ_i, the binary search of Hinton and Roweis (2002) can be used on the whole dataset Z. The matrices $p_{T|S}$ and $p_{S|T}$ are then extracted from the cross covariance terms of $p^{physics}_{i|j}$.

The operation in Eq. 17, which had the two-fold purpose of creating a normalised and symmetric matrix of probabilities p_{ij}, is no longer viable since the matrices $p_{T|S}$ and $p_{S|T}$ are not square. To address this, a two-step solution is proposed. Firstly, it must be acknowledged that the two conditional probabilities are incompatible or, in other words, that a given probability $p_{S|T}$ is not necessarily equal to $p_{T|S}$. This situation is similar to the simplicial sets in the UMAP graph representation of McInnes et al (2020). Here, the same solution is proposed, namely, the construction of a fuzzy union based on the t-conorm

$$p_{ST} = p_{S|T} + \left(p_{T|S}\right)^T - p_{S|T} \circ \left(p_{T|S}\right)^T. \tag{22}$$

Here, $\circ$ is the pointwise product. The final step involves normalising the values in p_{ST} such that it becomes a PDF, which can be achieved by applying the softmax function as follows:

$$p_{ij} = \frac{p_{ST}}{\sum_i \sum_j p_{ST}}. \tag{23}$$

The final probability matrix p_{ij} represents the distribution between source and target tasks in the physics space. Algorithms 1 and 2 are the practical implementations used to obtain the physics space distribution.

Algorithm 1 Building physics space probability distribution. Subscripts N and H represent new and historical samples, respectively

function SETUP($X_N, Z_N, Z_H, perplexity$)
  Initialise y using $\mathcal{N}(0,1) \times 10^{-4}$
  $X_N \leftarrow \frac{X_N - \mu}{\sigma}$
  $Z_N \leftarrow \frac{Z_N - \mu}{\sigma}$
  $Z_H \leftarrow \frac{Z_H - \mu}{\sigma}$
  $\boldsymbol{Z} \leftarrow \begin{bmatrix} \boldsymbol{Z}_H \\ \boldsymbol{Z}_N \end{bmatrix}$
  DMat $\leftarrow$ SquaredEuclideanDistance($\boldsymbol{Z}$,$\boldsymbol{Z}$)
  $P \leftarrow$ SearchVar(DMat, perplexity)
  $N_H \leftarrow$ nrows($\boldsymbol{Z}_H$)
  $p_{H|N} \leftarrow P\,[1 : N_H, N_H + 1 : end]$
  $p_{N|H} \leftarrow P\,[N_H + 1 : end, 1 : N_H]$
  $p_{NH} \leftarrow softmax(p_{H|N} + p_{N|H}^T - p_{H|N} \circ p_{N|H}^T)$
  $p_{HH} \leftarrow softmax\,(P\,[1 : N_H, 1 : N_H])$
  $Dict \leftarrow \{y_0, X_N, p_{NH}, p_{HH}, \mu_{X_N}, \sigma_{X_N}\}$
  return Dict
end function

Algorithm 2 Computing standard deviations σ. In this paper, the binary search of Hinton and Roweis (2002) is used.

function SEARCHVAR(p, perplexity)
  Search search for σ_i for each row i of p such that $perplexity = 2^{H(i)}$, where $H(i) = -\sum_j p_{j|i} \log_2 (p_{j|i})$
end function

4.3 Aligning the source and target data in the design variable space

This section describes the vectorised gradient formulation used when reconstructing the distribution of the physics-latent space in the variable space. The optimisation algorithm is left to the reader's discretion. The results of this paper are obtained using the same gradient-descent-with-momentum routine that MATLAB employs in the exact implementation of its `tsne` function.

The PDF in the variable space is identical to t-SNE, save for a slight modification which ensures its terms are equivalent to the cross covariance values of Eq. 22,

$$q_{ij}^{variable} = \frac{\left(1 + ||y_i - x_j||^2\right)^{-1}}{\sum_{k \neq l} \left(1 + ||y_k - x_l||^2\right)^{-1}}. \tag{24}$$

The terms x, associated with the target task, are fixed. The optimisation algorithm must find a way to embed the source data, y, around the target task samples.

The gradient of the Kullback–Leibler divergence with respect to the embedding is given by

$$\frac{\partial C_1}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - x_j)(1 + ||y_i - x_j||^2)^{-1}. \tag{25}$$

The physics and variable superscripts were dropped for notation convenience. The gradient calculation may be vectorised by observing that, for instance, the gradient with respect to the first point in the embedding can be written as

$$\begin{aligned} \frac{1}{4}\frac{\partial C_1}{\partial y_1} &= \frac{p_{11} - q_{11}}{1 + ||y_1 - x_1||^2}(y_1 - x_1) + \ldots \\ &+ \frac{p_{1m} - q_{1m}}{1 + ||y_1 - x_m||^2}(y_1 - x_m) = \\ &y_1\left(\frac{p_{11} - q_{11}}{1 + ||y_1 - x_1||^2} + \cdots + \frac{p_{1m} - q_{1m}}{1 + ||y_1 - x_m||^2}\right) - \ldots \\ &- x_1 \frac{p_{11} - q_{11}}{1 + ||y_1 - x_1||^2} - \cdots - x_m \frac{p_{1m} - q_{1m}}{1 + ||y_1 - x_m||^2}. \end{aligned} \tag{26}$$

This enables the gradient with respect to the first sample to be re-written as

$$\frac{1}{4}\frac{\partial C_1}{\partial y_1} = y_1 \circ \sum_m K_{1m} - K_{1m} X. \tag{27}$$

Here, K_{1m} is a vector containing the first row of a matrix K defined as

$$K = \begin{pmatrix} \frac{p_{11} - q_{11}}{1 + ||y_1 - x_1||^2} & \cdots & \frac{p_{1m} - q_{1m}}{1 + ||y_1 - x_m||^2} \\ \vdots & \cdots & \vdots \\ \frac{p_{n1} - q_{n1}}{1 + ||y_n - x_1||^2} & \cdots & \frac{p_{nm} - q_{nm}}{1 + ||y_n - x_m||^2} \end{pmatrix}. \tag{28}$$

The matrix of gradients for cross-covariance terms can thus be written in a vectorised form as

$$\frac{\partial C_1}{\partial y} = 4\left(Y \circ \sum_m K - KX\right). \tag{29}$$

There is a further complication that must be addressed in the gradient calculation. In cases where the target data are sparse, the influence of far-away target points on the source data embedding may be very small. This may lead to unwanted behaviour, such as clustering of significantly different source data samples. This problem is demonstrated in section 5.2. This implementation addresses the issue by conducting a secondary gradient calculation, which aims to

enable the source data to maintain its physics space structure in the design variable space.

Mathematically, this entails calculating the original t–SNE gradient using only source data samples. This requires first collecting the lower-right block of the matrix in Eq. 20

$$p_{i|j}^{physics'} = p_{S|S}. \tag{30}$$

The PDF in the variable space becomes

$$q_{ij}^{variable'} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{k \neq l}\left(1 + ||y_k - y_l||^2\right)^{-1}}. \tag{31}$$

The gradient has the same form as Eq. 24 and can be written as

$$\frac{\partial C_2}{\partial y_i} = 4\sum_j \left(p'_{ij} - q'_{ij}\right)(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1}. \tag{32}$$

Following the MATLAB exact implementation of t-SNE, Eq. 32 can be re-written, for the first term of y, as

$$\begin{aligned}\frac{1}{4}\frac{\partial C_2}{\partial y_1} = y_1\left(0 + \frac{p'_{12} - q'_{12}}{1 + ||y_1 - y_2||^2} + \cdots + \frac{p'_{1m} - q'_{1m}}{1 + ||y_1 - y_m||^2}\right) - \ldots \\ -y_2\frac{p'_{12} - q'_{12}}{1 + ||y_1 - y_2||^2} - \cdots - y_m\frac{p'_{1m} - q'_{1m}}{1 + ||y_1 - y_m||^2}\end{aligned}. \tag{33}$$

Letting the matrix $\tilde{K}$ be defined like Eq. 28, but with the new terms, the vectorised gradient may be written as

$$\frac{\partial C_2}{\partial y} = 4\left(D - \tilde{K}\right)Y. \tag{34}$$

Here, D is a diagonal matrix defined as

$$D = \begin{pmatrix} \sum_m \tilde{K}_{1m} & & \\ & \ddots & \\ & & \sum_m \tilde{K}_{mm} \end{pmatrix}. \tag{35}$$

The final gradient used in the embedding optimisation has the form

$$\begin{aligned}\frac{1}{4}\frac{\partial C}{\partial y} = \alpha\left(Y \circ \sum_m K - KX\right) + \ldots \\ +(1-\alpha)\left(D - \tilde{K}\right)Y.\end{aligned} \tag{36}$$

Here, the weighting parameter α depends on the number of target sample points. Generally, the more dense the sampling of the target data, the larger α should be since the information of the source data structure in the physics space is not required to ensure adequate embedding.

A practical implementation of the vectorised gradient computation is proposed in Algorithm 3.

Algorithm 3 Vectorised Computation of Embedding Gradients. The operator $\circ$ denotes element-wise multiplications, while $\times$ denotes matrix multiplication.

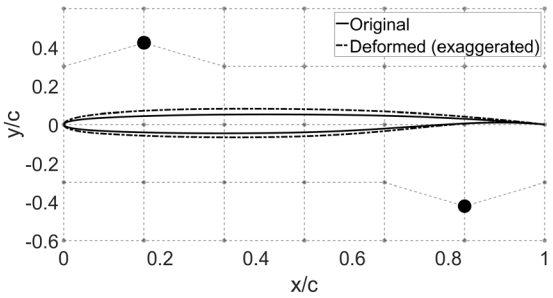
function GRAD$(p_{ij}, p_{jj}, X_N, \hat{X}, frac)$
  $P_Y \leftarrow SquaredEuclidean(\hat{X}, \hat{X})$
  $P_Y \leftarrow \frac{1}{(1+P_Y)}$
  $diag\,(P_Y) \leftarrow 0$
  $P_{Y,NORM} \leftarrow softmax\,(P_Y)$
  $\tilde{K} = P_Y \circ (p_{jj} - P_{Y,NORM})$
  $Grad_1 \leftarrow 4 * \left(diag\left(\sum_{rows} \tilde{K}\right) - \tilde{K}\right) \times \hat{X}$
  $P_{XY} \leftarrow SquaredEuclidean(\hat{X}, X_N)$
  $P_{XY} \leftarrow \frac{1}{(1+P_{XY})}$
  $P_{XY,NORM} \leftarrow softmax\,(P_{XY})$
  $K \leftarrow P_{XY} \circ (p_{ij} - P_{XY,NORM})$
  $\sigma \leftarrow \sum_{columns}(K)$
  $term_1 \leftarrow \hat{X} \circ \sigma$
  $term_2 \leftarrow K \times X_N$
  $Grad_2 \leftarrow 4 * (term_1 - term_2)$
  $Grad = frac * Grad_2 + (1 - frac) * Grad_1$
  return $Grad$
end function

5 An illustrative example

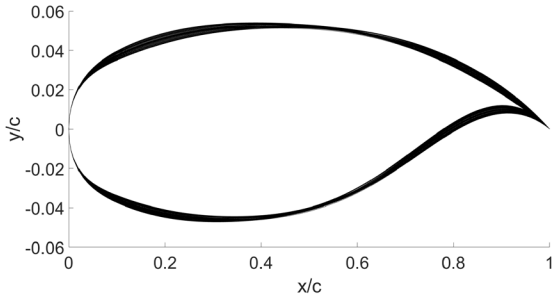
5.1 Problem description

A simple engineering case is introduced for demonstration purposes. The design parameterisation employs the Free-Form Deformation (FFD—Sederberg and Parry (1986)) scheme on one of two super-critical aerofoils: the SC(2)−0610 and SC(2)−0710, where the former is the target task and the latter is the source. The manipulation of the airfoil surface is achieved through a vertical displacement of the nodes highlighted in Fig. 1a, up to 5% of the aerofoil chord. The resulting range of changes is shown in Fig. 1b. The same parameterisation is considered across target and source tasks for simplicity.

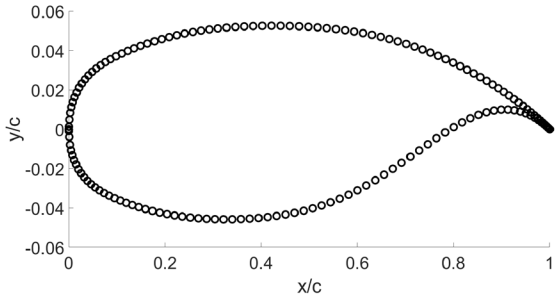
The quantity of interest is the total drag coefficient. The physics matrix is built using pressure coefficient (Cp) measurements. The spatial arrangement of pressure probes is illustrated in Fig. 1c, with an approximate uniformity maintained across all geometries. The CFD analysis uses the Viscous Garabedian and Korn (VGK) code Garabedian and Korn (1971).



(a) FFD grid example - Highlighted points can move vertically. Both vertical displacement and aerofoil coordinate change is exaggerated for easier visualisation



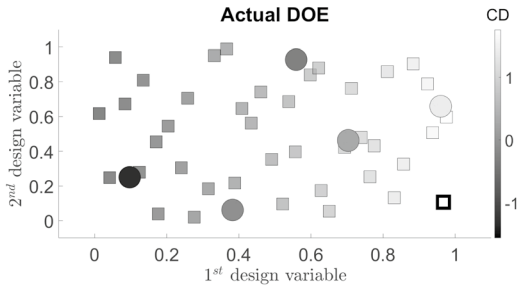
(b) Range of geometries for setup in Figure 1a; drawn from a random sample of 40 points



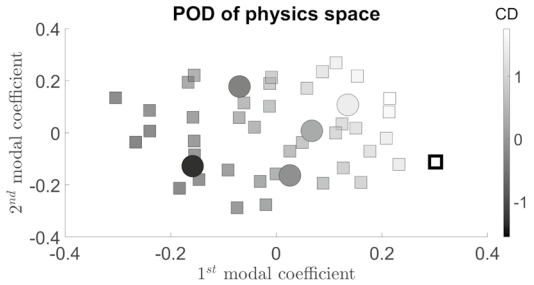
(c) Pressure probe locations. Measurements of probes are used to build a physics matrix. X-location is kept approximately constant across all geometries

Fig. 1 Illustrative problem setup

The flight conditions are characterised by a Mach number of 0.72, a Reynolds number of 6×10^6, and a lift coefficient constraint of 0.4. The latter is enforced through adjustments in the angle of attack. A spline regression model between the lift coefficient (C_l) and the angle of attack is built using angles 0 and 1. Subsequently, extrapolation techniques are employed to identify the angle of attack necessary to achieve the desired lift coefficient. In cases where the initially estimated angle does not yield the target C_l, the regression



(a) Design of experiment for the illustrative problem. Large, circular points belong to target data (SC(2)-0610). Small, square points belong to the source data (SC(2)-0710). The colouring corresponds to the drag coefficient (C_D).To visualise correlation, C_D values are normalised to have a mean of 0 and a standard deviation of 1



(b) Latent space of aerofoil Cp measurements, visualised using the first two modal coefficients of the proper orthogonal decomposition

Fig. 2 DOE and POD of illustrative problem for a random latin-hypercube design

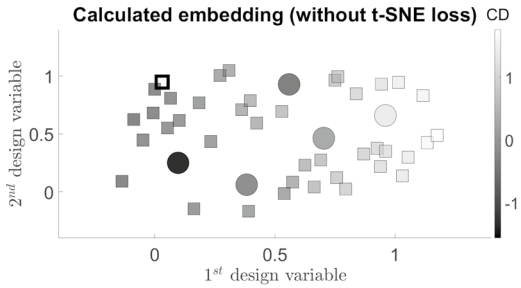
model is updated, and the process iterates until the correct value is obtained.

5.2 Embedding result

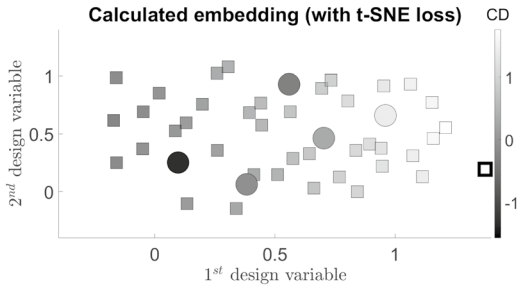
Two Latin-hypercube samples of five and 40 samples for the target and source tasks are drawn using the MATLAB `lhsdesign` function. Their distribution is shown in Fig. 2a, where the points are coloured by the drag coefficient (C_D) values obtained from VGK, which are normalised to have mean zero and a standard deviation of one.

The results of the embedding process are depicted in Fig. 3a, b. One of the points in the source data is deliberately highlighted (bold square) as it conveniently showcases the need for the secondary t-SNE error term mentioned in section 4.3. Without the source data, the location of this sample in the latent space of Fig. 2 is such that the values p_{ij} of its closest three neighbours have approximately equal values. The p_{ij} value of its fourth and fifth

neighbours are near zero. Since the first three neighbours are approximately collinear, the embedded point can be positioned either above or below the line they form with the same effect on the loss function. In this case, the result converges to the unlucky location. By including the t-SNE loss term, the issue is resolved since information about the remaining neighbours is included via the source data. An added degree of dispersion can also be observed, as fewer local clusters form in the embedding of Fig. 3b.



(a) Embedding calculated when loss depends only on cross co-variance terms. Of particular interest is the highlighted point (bold square), which is incorrectly positioned due to the sparsity of the data: the fourth and fifth closest neighbours in the target data are too far away to influence the gradient calculation of Equation [29]



(b) The calculated embedding when the additional gradient term is included. The source data structure is included in the gradient formulation ($\alpha = 0.8$); the highlighted point (bold square) now has sufficient information to reach the correct location. Less local clustering is also observed

Fig. 3 Embeddings computed by proposed method

5.3 Model build result

The benefit of the embedding may be assessed by building a Kriging model using the five target samples and then comparing its predictive accuracy with that of a categorical Kriging, which fuses the target data and the embedded points.

Figure 4a, b shows the two source functions and Fig. 4c shows the true target function from which the DOEs were sampled. Figure 5a shows the prediction generated by a Kriging model using the five available samples from the target data, which are also plotted. The model is inaccurate and misleading and may cause the cold start problem as it cannot correctly indicate the location of minimum drag. On the other hand, the categorical-Kriging prediction, which uses the embedded points of the source data (Fig. 5b, c), generates a faithful representation of the topography of the design space, correctly indicating the location of minimum drag. The former data source (SC(2)−0410) is of particular interest because the algorithm manages to fuse the data from the two samples despite the apparent lack of correlation between the original functions; this is achieved thanks to categorical-Kriging's ability to model negative correlations between responses.

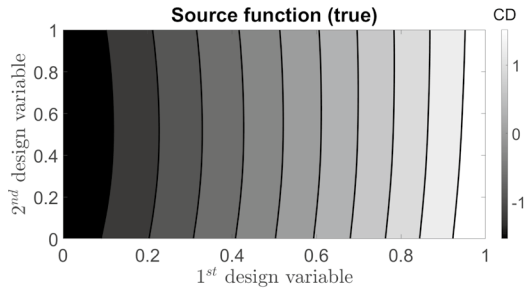
6 Aerofoil optimisation example

To more comprehensibly demonstrate the ability of the proposed method to alleviate the cold start problem, an aerofoil optimisation test case is proposed. The objective is to improve the performance of the SC(2)−0610 aerofoil at the flight conditions described in Sect. 5.1, namely a Mach number of 0.72 and a Reynolds number of 6×10^6. For this problem, however, a lift coefficient of 0.7 is used.

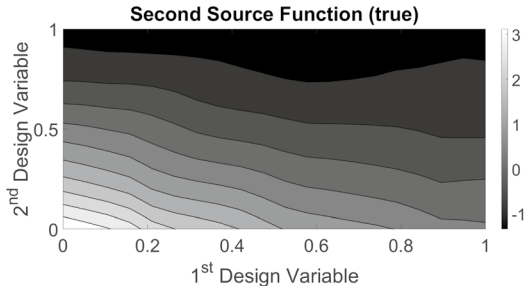
Two versions of this problem are investigated, one where the parameterisation is constant and one where it varies from source to target data. In both cases, the source data may stem from one of three aerofoils: the SC(2)−0410, SC(2)−0610, and SC(2)−0710. The two sets of descriptors are introduced in the following sections.

6.1 FFD parameterisation (10D)

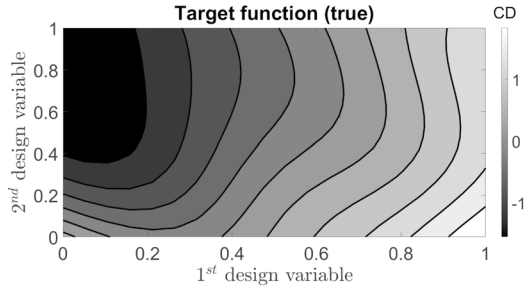
The first parameterisation is an extension of the FFD scheme described in Sect. 5.1. The number of nodes that are vertically displaced is increased to ten in this scenario, as depicted in Fig. 6.



(a) True response surface of first source drag coefficient function (SC(2)-0710)



(b) True response surface of second source drag coefficient function (SC(2)-0410)



(c) True response surface of target drag coefficient function (SC(2)-0610)

Fig. 4 True output functions corresponding to target and source tasks

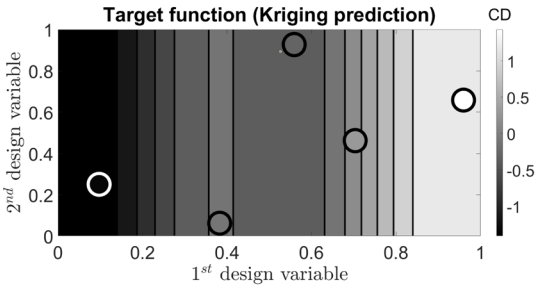
6.2 Hicks–Henne bump function parameterisation (8D)

The second parameterisation uses Hicks–Henne bump functions and is explored in depth by Masters et al (2017). A short description is provided here for convenience.

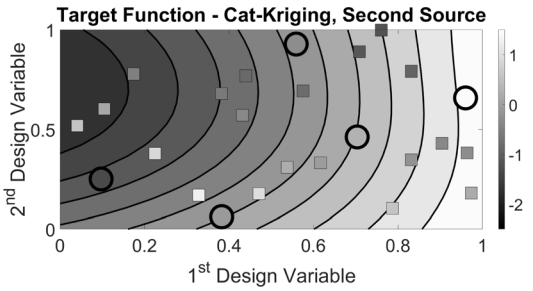
The final co-ordinates of an aerofoil deformed using the scheme are given by

$$y^{final} = y^{initial} + \sum_{i=0}^{n} a_i \sin^{t_i}\left(\pi x^{\ln 0.5/\ln h_i}\right). \tag{37}$$

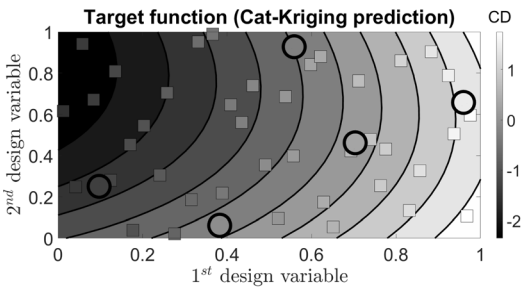
The design variables define the coefficients a_i. The terms h_i determine the locations of the bump function maxima and are given by



(a) Contour of Kriging prediction. Samples used to generate the model are also plotted. The model is an example of a surrogate that may lead to the cold start problem, as it cannot correctly place the minimum of the function. An R^2 value of 0.7849 is observed



(b) Contour of categorical Kriging prediction for first source function. Target and embedded samples used to generate the model are plotted. The prediction is greatly improved by adding the historical data. An R^2 value of 0.9778 is observed



(c) Contour of categorical Kriging prediction for second source function. Target and embedded samples used to generate the model are plotted. An R^2 value of 0.9774 is observed

Fig. 5 Predictions using Kriging and categorical Kriging

$$h_i = \frac{1}{2}\left[1 - \cos\left(\frac{i\pi}{n+1}\right)\right] \text{for} \quad i = 1, \ldots, n. \tag{38}$$

The thickness parameters t_i are defined as

$$t_i = 2\left(\frac{n-i}{n-1}\right)^3 \text{for} \quad i = 1, \ldots, n. \tag{39}$$

The suction and pressure sides are deformed separately, using six design variables a_i for each. However, since the

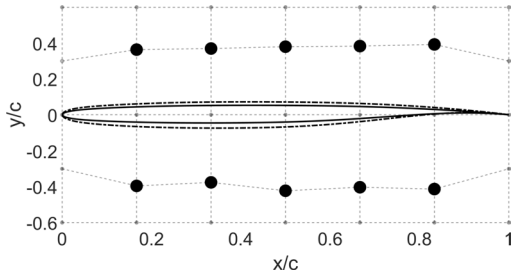


Fig. 6 Example of deformation achieved using a 10-dimensional FFD scheme with a regular grid of points. Vertical displacement and aerofoil deformation are exaggerated for visualisation purposes. The dotted line is the deformed aerofoil, and the solid is the baseline geometry

two surfaces are joined together after the deformation, the weights a_0 and a_6 must be kept at zero to ensure geometric consistency. Thus, from an optimisation perspective, four variables can be modified on each surface to reduce drag, for a total of eight. The DOE is defined inside a Latin hypercube and bounded by [0,1]. For a given sample x inside this space, the weights a_i are obtained using

$$a = 0.003(x - 0.5). \tag{40}$$

The scaling factor 0.003 controls the range of possible geometries and is arbitrarily chosen. Higher values result in a larger range of possible aerofoils.

6.3 Optimisation problem description

$$\begin{aligned} \min_x \ & C_d \\ \text{s.t.} \ & C_m(x) > C_{m0} \\ & C_L = C_{L0} \end{aligned} \tag{41}$$

The objective of this exercise (Eq. 41) is to minimise drag at M=0.7, Re $= 6 \times 10^6$, and for a lift coefficient $C_{L0} = 0.7$. A one-sided moment coefficient constraint is enforced with a tolerance of 10^{-4}. The target task geometry is the SC(2)−0610 aerofoil and its moment is $C_{m0} = -0.1107$.

The baseline method consists of a traditional sequential model-based strategy. A Kriging model, described in Sect. 3.1, acts as the surrogate. Two auxiliary criteria are optimised using the MATLAB ga routine, followed by a terminal local search using MATLAB's `fmincon`. At each update, the chosen infill points are given by the results of

1. $\min_x \hat{f}(x)$ - the exploitation step, which finds the minimum of the predictor
2. $\max_x \hat{s}^2(x)$ - the exploration step, which finds the location of the maximum error (Eq. 9) of the predictor

The Kriging-SMBO, i.e. the baseline approach, denoted KRIG, starts with a sparse sample of 10 points. Then, 15 update cycles are conducted, leading to a final set of 40 samples if all runs converge. The process is carried out 30 times with different initial DOEs to generate a suitable sample for statistical analysis.

The new method, employing historical data, is denoted TSNE-CATKRIG and is set up like the baseline. To gather the source data, a separate KRIG optimisation is carried out for the source aerofoils. The optimisation results, i.e. the original 10-point DOE plus the 30 infill points, make up the source task sample and are modelled as a second qualitative level using the categorical Kriging. The physics vector consists of the same pressure measurements described in section 5.2.

6.4 Case 1 - Using FFD as source data

In the first example, the SC(2)−0610 aerofoil is optimised using only the variables of the FFD scheme. The source data come from one of two aerofoils, as indicated in Fig. 7, where the title of the figures has the following form: source task aerofoil (source parameterisation) → target task aerofoil (target parameterisation).

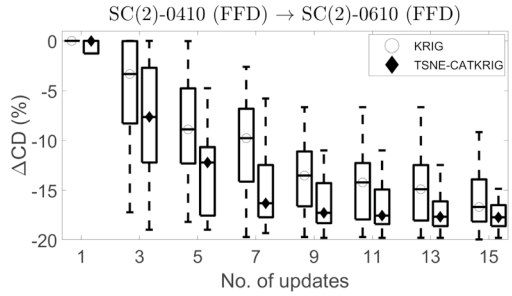
The performance of the methods is indicated using boxplots. For this graphical representation, the line's location (where circles/diamonds are also plotted) indicates the median value. The edges of the boxes correspond to the 25^{th} and 75^{th} percentiles. The whiskers extend to values of $\pm 1.5 \times IQR$ (inter-quartile range), and points located beyond are classed as outliers; no outliers were observed in this example.

Across all three source tasks, a significant improvement (indicated in the captions) is observed when applying the proposed method. The median values are significantly lower across all updates, and the *IQR*s and outlier bounds are shrunk to a substantial degree.

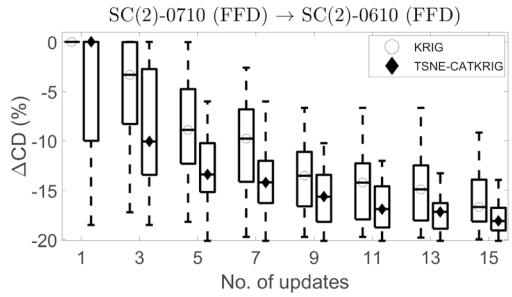
6.5 Case 2 - Using HHB as source data

In the second example, the HHB parameterisation is employed as the source task. The same title convention and visualisation tool is used in Fig. 8.

Similar conclusions can be drawn from the HHB → FFD case, namely that the median is lower, with a tighter distribution of discovered minimum values. However, there is a slight reduction in the benefit of reusing historical data. This is to be expected, of course, since the latent correlations between tasks will not be as strong when the parameterisation changes. Still, the optimisation performance enhancement given by the proposed method is considerable. The embedding strategy shows promise as a solution to the cold start problem.

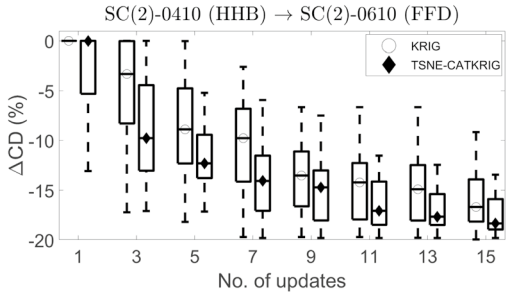


(a) TSNE-CATKRIG achieves the same performance as KRIG at approximately the 8^{th} update (if variance is considered). A 47% reduction in cost

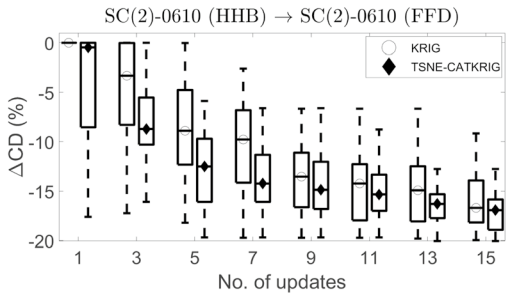


(b) TSNE-CATKRIG achieves the same performance as KRIG at approximately the 10^{th} update (if variance is considered). A 33% reduction in cost

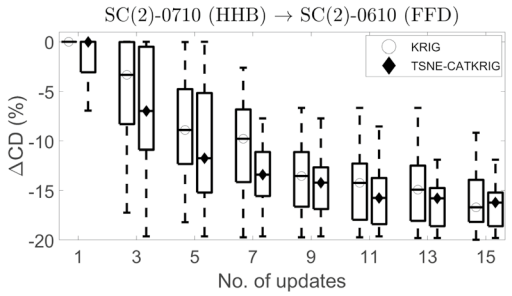
Fig. 7 Optimisation performance when target and source tasks have the same descriptors, namely those defined by the FFD scheme. KRIG achieves a 16.692% drag reduction at the final update



(a) TSNE-CATKRIG achieves the same performance as KRIG at approximately the 10^{th} update. A 33% reduction in cost



(b) TSNE-CATKRIG achieves the same performance at the 11^{th} update. A 27% reduction in cost



(c) TSNE-CATKRIG achieves the same performance at the 10^{th} update (variance is considered as well). A 33% reduction in cost

Fig. 8 Optimisation performance when the target task consists of the FFD predictors and the source task consists of the HHB predictors. KRIG achieves 16.692% drag reduction at the final update

6.6 Assessment of predictive accuracy

Plots of the average validation error at various stages in the optimisation process are shown in Fig. 9. These errors are obtained by predicting the values of a test set consisting of 2000 samples built using MATLAB's `lhsdesign`. The 16^{th} update corresponds to a model built after the final update. The legend indicates the parameterisation and aerofoil of the source data. The marker style is used to indicate the source parameterisation, while the choice of line style corresponds to the choice of aerofoil.

A great degree of fluctuation is observed due to the stochasticity of the optimisation process. However, the methods that utilise historical data have a clear advantage in predictive accuracy in the early stages and tend to decay relative to the baseline, the *KRIG* approach, as more infill points are added. In the case of the objective function, C_d, this advantage is maintained up to the 9^{th} update. For the constraint, the advantage is lost after the 5^{th} update. This, the authors believe, is the source of the performance advantage of the t-SNE-based method. The initial improvement is sufficient to guide the initial search correctly, indicating that the method is a valid solution to the cold start problem.

7 Combustor optimisation example

This section examines the performance of the proposed method on a combustor representative of those present on modern commercial airliners. The geometry of the model is shown in Fig. 10. The fluid volume is built using a parametric geometry modelled in Siemens NX.

7.1 Design variable description

The source data parameterisation consists of the arrangement described by Toal et al (2021), where the ports on the

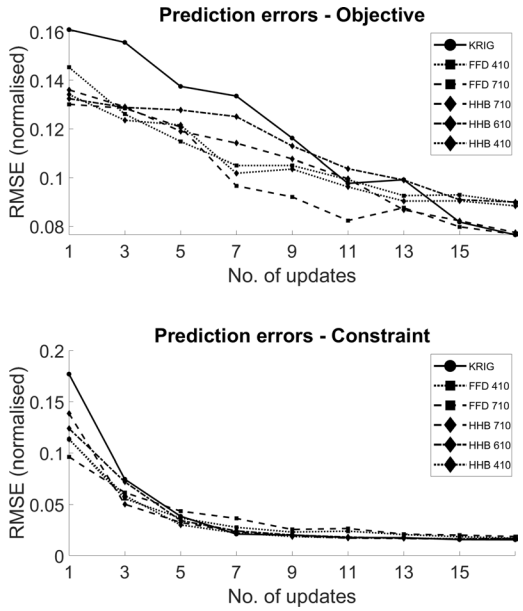


Fig. 9 Average validation error of models during optimisation - aerofoil case

annuli are grouped into four rows. Eight design variables are used, four of which describe the axial location of each row, while the other four define their radius.

In the target data, twelve design variables are used. These consist of the x-y displacements of six control points, which define the centres of radial basis functions that are used to morph the mesh of the inner and outer annuli walls. Rather than morphing the walls and re-meshing the geometry each time, the mesh itself is morphed, an equivalent operation. Three basis functions are used for each wall.

7.2 CFD

The mesh consists of an unstructured tetrahedral design with hexahedral core conversion. The baseline geometry contains 16.7M elements and was constructed using the Ansys ICEM CFD package.

The CFD solver runs in the proprietary Rolls-Royce package PRECISE-UNS Anand et al (2013). Its setup is identical to the high-fidelity case in Toal et al (2021). Turbulence is simulated using the realisable $K-\epsilon$ turbulence model. Fuel and ignition are introduced after the $500th$ and 600^{th} iterations, respectively, and the code stops after 5000 iterations. Combustion is modelled using the flamelet-generated manifold technique. Boundary conditions in the form of velocity profiles are placed at the inlets of both annuli and the exits of each swirler passage. These remain unchanged irrespective of the design geometry. The modelled condition is representative of a large commercial airliner running at full throttle.

NO_x emissions are obtained directly from PRECISE-UNS, while wall pressure losses are calculated using Paraview.

7.3 Optimisation problem description

The goal of the exercise is to reduce the NO_x emissions index ($EINO_x$). The physics matrix for this quantity is constructed using NO_x measurements from a uniform grid of 250×100 points at the exit wall of the combustor. This grid does not consist of cells of the fluid volume mesh. As a result, the NO_x quantities it contains are interpolated.

Constraints are placed upon the wall pressure loss, which is the drop in total pressure observed between the inner and outer annuli entrances and the exit of the combustor. These values are constrained to be within 10% of the baseline. Unlike NO_x, the physics matrix, built using total pressure values, uses the measurements in the volume cells at the combustor exit, as these could more conveniently be calculated without the interpolation step. This does not pose any problems for the target data, where the mesh is morphed, and thus, the grid on the combustor exit is constant across designs. However, in the case of the source data, i.e. the port parameterisation, the combustor was re-meshed each time. Thus, the vector of measurements differed in size for the different geometries. To ensure consistency with the constant mesh of the target data, a truncation operation was performed: only the first 16,259 values were kept. This caused some inconsistencies, as not all measurements represented the same cells. However, because the cells corresponding to the measurements were still relatively close in the fluid volume, it was found that the inconsistencies were not sufficiently major to affect the correlation between the integral of the vector and the pressure drop scalar.

Designs that did not converge after 5000 iterations are treated as infeasible. The structure of MATLAB's optimisers requires that constraint functions return values below or equal to zero if a function is feasible and values above zero if the opposite is true. To ensure that failed runs are avoided, a value of one is ascribed to the failed runs.

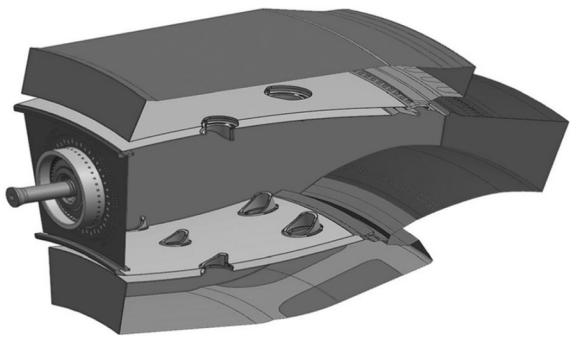


Fig. 10 Combustor fluid volume - reproduced from Toal et al (2021)

Due to the restrictive cost of the CFD code, historical data do not originate from a previous optimisation. Rather, the source data are selected from a database of 424 CFD runs. For each repetition of the experiment, 100 points are sampled using a routine detailed by Forrester et al (2008) that utilises the Morris criterion to choose points such that an approximation of a Latin hypercube is selected. Because the routine uses a greedy algorithm, a different initialisation will result in different sub-samples. Thus, by varying the random seed, different source datasets can be picked each time the experiment is repeated.

The target data consist of a 60-sample Latin-hypercube design generated using the MATLAB `lhsdesign` function. The same seed as per the source data is provided for this DOE.

The search for the auxiliary criteria is conducted using the built-in MATLAB `ga` routine. At each update, four infill points are added, two consisting of exploring and two of exploiting the design space:

1. $\min_x \hat{f}(x)$ - the first exploitation step, which finds the minimum of the predictor
2. $\max_x EI[\hat{f}(x)]$ - the second exploitation step, which finds the highest value of the expected improvement of the predictor
3. $\max_x \hat{s}^2(x)$ - the first exploration step, which finds the location of the maximum error
4. $\max_x \left(||x - X_{target}||^2\right)$ - the second exploration step, which finds a space-filling point, defined as furthest away from all other points in the target dataset

As in the previous example, the baseline is a Kriging-SMBO, denoted *KRIG*, which is compared to the proposed method, consisting of the embedding and Categorical Kriging, denoted *TSNE-CATKRIG*. The initial sample of 60 points is updated ten times, with four infill points per update, such that a 100-point sample is eventually reached, provided all runs converge. The process is repeated ten times rather than the 30 aerofoil cases, again due to the restrictive cost of the CFD simulation.

7.4 Results and discussion

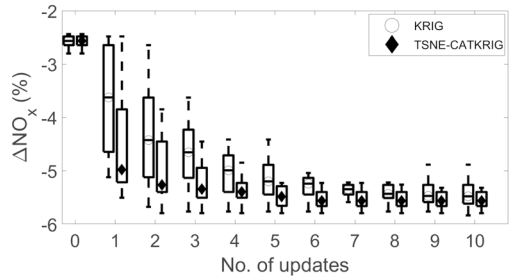
The results of the optimisation process are shown in Fig. 11a. The 0^{th} update corresponds to the minimum feasible design in the original 60-point DOE. On this occasion, a reduction in predictive error cannot be used to explain the improvement in optimisation performance. The comparison between the two methods is shown in Fig. 11b, where it is visually obvious that the proposed method does not perform as well as the baseline. Granted, the scale of the validation errors is small relative to the output range, but this would only mean that the two methods should at least perform equally in the optimisation process of Fig. 11a; however, this is clearly not the case. The question arises as to what the TSNE-CATKRIG learns from the historical data.

The answer can be found by analysing the physics-based latent space of the NO_x measurements. The latent space is visualised by reducing the physics matrix, which has dimensions $n \times 25000$, where n is the number of samples, to two dimensions. The tool of choice is the proper orthogonal decomposition on the full physics matrix, which is defined as a matrix, where Z_{source} and Z_{target} are vertically stacked. The result of this operation, in the form of two sets of samples, is showcased in Fig. 12. The samples are drawn from a randomly selected run of the ten repeats, and the DOEs correspond to the 0^{th} update, i.e. the initial sample. The plots are similar to those in Sect. 5.2. The colour is once again given by the objective, NO_x, which is normalised with mean zero and standard deviation one.

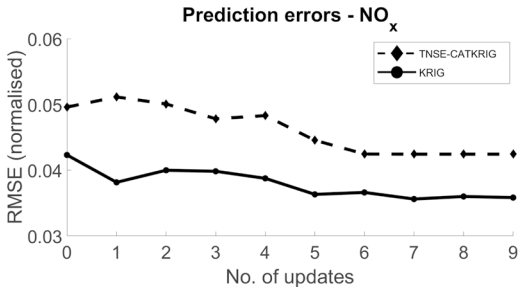
The most evident difference between the target and source data is the multi-modality of the former. Since the integral of the physics vector is tightly correlated to the output function, $EINO_x$, it may be assumed that this multi-modality will be reflected in the design variable space. On the other hand, the source data appear to have a relatively simple objective function distribution, almost linear in nature. As previously shown, categorical Kriging yields a model akin to a weighted average, where the number of available samples decides the weights. The source data will thus be preferred. Naturally, the predictive accuracy of the model on the target data will suffer. However, the search for a global minimum will be greatly simplified since, in the case of the source data, the location of the minimum can easily be identified (top right in the source physics-latent space).

In Fig. 13, the subspace is rebuilt after three updates. The distribution of the target and source data has remained almost unchanged, except that additional points are now present in the top right. This is, in essence, what the model learns from the historical sample; that is, the global minimum is located at the indicated edge of the design space.

Further confirmation is offered in Fig. 14. The best update selected for each experiment repeat, up to the third update (to match Fig. 13), is plotted as a parallel axis chart. The first twelve axes represent the design variables, while the last axis is the NO_x objective normalised to have a value in the interval [0,1]. The plot in Fig. 14b consists of the proposed method, which, although noisy to a degree, shows a clear pattern in choosing a specific edge of the design space. On the other hand, the baseline Kriging only shows a pattern for a few of the variables, most likely as a consequence of the multi-modality of the design space.

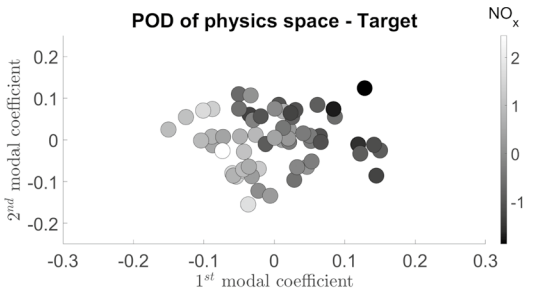


(a) Combustor optimisation results - KRIG reaches a 5.48% NO_x reduction. TSNE-CATKRIG reaches the same performance after four iterations, a 60% decrease in required cost

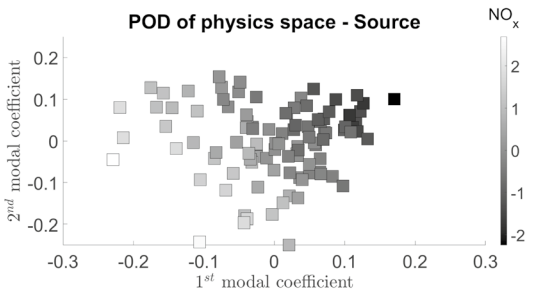


(b) Combustor validation error - unlike the aerofoil case, the prediction of the proposed method is less accurate than the baseline Kriging

Fig. 11 Optimisation result in combustor case and validation error on the prediction of the objective function, $EINO_x$



(a) The target data in the physics-latent-space obtained using POD. The design landscape is highly multi-modal, which is likely reflected in the design variable space, posing a significant challenge to a Kriging-based SMBO



(b) The source data in the physics-latent-space obtained using POD. The landscape of the source data is, by comparison, simple and almost linear in nature. Thus, finding the minimum of this function is a straightforward exercise, as it can be visually tracked to the top right corner

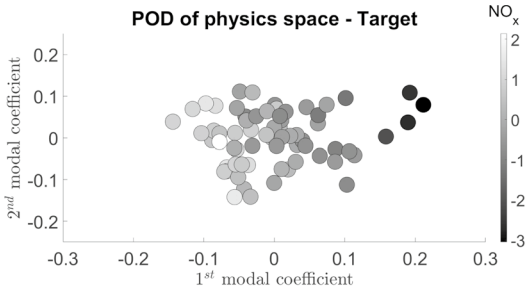
Fig. 12 Latent, physics-based sub-spaces of NO_x

8 Hyper-parameter Optimisation

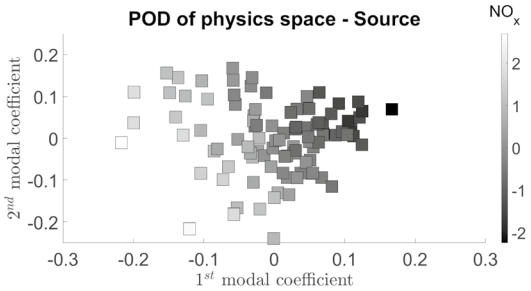
Several hyper-parameters control the shape and accuracy of the embedding. This section investigates the sensitivity of the embedding to these parameters and proposes guidelines for identifying an optimal set. Of particular concern are the three most important parameters inherited from t-SNE: perplexity, exaggeration, and learning rate. The weighting factor, α, introduced by the current method, is also studied.

Assessing embedding correctness is difficult. Using the distance between the embedding and actual locations, known a priori, is not feasible for two reasons. Firstly, the ground truth, i.e. where the points should be, will usually not be known or available to practitioners at inference time (there would be no need for an embedding otherwise). Secondly, there may be significant differences between the physics and design space distribution. This will also entail that two DOE locations may generate identical physics, meaning multiple embedding solutions may exist for the same point. It has also been noticed that even if the actual location is known a priori, the embedding will often generate a significantly different set of points. This is not unexpected and is partly a function of the difference between the design space and physics distributions and partly caused by the stochasticity of the method.

An essential contention is that this is not problematic because the critical information passed from historical data is the relationship between the design variables and the response. Given the importance of accurately representing the response, the validation metric of choice is the R^2 correlation between the response values of the embedded points and the response values of the actual DOE locations. Unfortunately, assessing responses for the embedded points was computationally infeasible, so Kriging models for the two responses were used. The final issue is that the R^2 metric is also unavailable at inference time. Gove et al (2022) provide a feasible solution involving a consistency check between the distribution in the physics space and the distribution in the design variable space. The mean accuracy metric (MAM) proposed by Gove et al (2022) consists of the average of four related metrics:

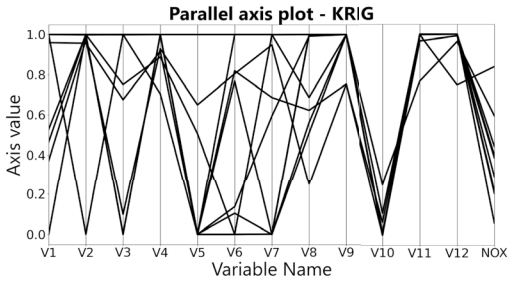


(a) The target data in the physics-latent space after three update iterations. Informed by the source data, multiple updates have been added to the top right corner

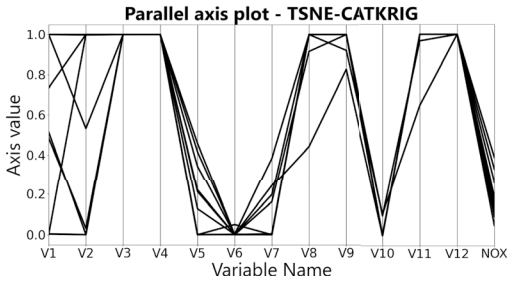


(b) The source data in the physics-latent. The landscape of the source data has remained mostly unchanged

Fig. 13 Latent, physics-based sub-spaces of NO_x



(a) Parallel axis plot of Kriging updates. There is no clear pattern to the selected by the infill strategy, likely due to the multi-modality of the function. The objective function values are in the range $[0.1, 0.8]$



(b) Parallel axis plot of embedding-based cat-Krig updates. In this instance, informed by the historical data, a clear pattern can be seen regarding which edge of the design space is most often selected by the infill strategy. Objective function values are in the range $[0.1, 0.4]$

Fig. 14 Parallel axis plots of updates for each of the strategies. Only the best design from each update is plotted

- The Spearman correlation: correlation between the terms of p_{ij} and q_{ij}
- The 1-Knn accuracy: For each point p_i, this metric is one of the nearest neighbour in the latent physics space and the variable space match, and 0 otherwise
- The 10-Knn accuracy: For each point p_i, this metric is the fraction of the k nearest neighbours that are neighbours of p_i in both the latent physics space and the variable space
- The rank accuracy metric: to balance the Knn accuracy metrics, this metric penalises large changes in rank more than small changes in rank and is detailed in Gove et al (2022).

The MAM metric can be used to tune the hyper-parameters, and its correspondence with the R^2 validation metric is showcased in the following section. In the case studies of Sects. 6 and 7, a grid search was used to maximise the MAM metric.

8.1 Experimental Setup

The experiment will involve conducting grid searches for optimal hyper-parameters and demonstrating that general rules of thumb can be derived without requiring advanced optimisation. As the problems of previous sections are reused, information on the physics and parameterisations can be found in Sects. 6 and 7.

To obtain a ground truth for validation, the same parameterisations are used (FFD in the aerofoil case with the SC(2)−0610 baseline and the wall morph parameters for the combustor). In both cases, datasets of varying total sample numbers are collected and split as 30:70 into new and historical samples. The procedure thus involves the following:

- Randomly splitting a sample into 30% new and 70% historical points,
- Embedding the historical sample around the new points using the proposed method,
- Predicting the response at the embedded locations, and
- Assessing the correlation between the prediction at embedded locations and response values at ground truth locations.

Table 1 Leave-one-out cross-validation metrics for Kriging predictors used to assess correlation

Metric	Aerofoil	Combustor
R^2	0.9098	0.99
Mean absolute error	0.36%	1.3%
Root mean squared error	6%	5.5%

Percentages represent the error value divided by the mean response in the training set

The predictions are provided by Kriging models built using large DOEs; 2000 samples of the FFD parameterisation are used to build the aerofoil model and 1455 samples are used for the combustor. Leave-one-out cross-validation metrics for the two models are shown in Table 1.

8.2 Sensitivity to Hyper-Parameters

The first task involves understanding which parameters most influence the validation metric. A Latin-hypercube sample of 100 points is generated for the four hyper-parameters of interest. The bounds are set up as follows:

- Perplexity - [< *spanclass* =′ *convertEndash*′ > 1 − 1000 < /*span* >],
- Exaggeration - [< *spanclass* =′ *convertEndash*′ > 1 − 5 < /*span* >],
- Learning Rate - [$10^{-4} - 2$],
- Weighting Factor - [< *spanclass* =′ *convertEndash*′ > 0 −1 < /*span* >].

The sensitivity of the metric to the hyper-parameters is assessed through the correlation plots of Fig. 15. The associated values in the Latin-hypercube sample are plotted against the correlation metric for each hyper-parameter. Only two of the parameters, perplexity and weighting factor, generate a significant effect and, as such, will be investigated further. The remaining hyper-parameters appear less significant, provided reasonable values (e.g. values of one for both) are used.

Having narrowed down the list of relevant parameters, a grid of 11 weight factors × 8 perplexities is generated for the remaining two. Figure 16 shows contour plots of the two variables, coloured by the correlation metric. The analysis consists of trying the 88 combinations of hyper-parameters on two datasets of varying sizes. The small DOE consists of 100 and the large DOE consists of 500 samples, again split 30:70. Figures are shown for both the combustor and aerofoil problems to demonstrate that the conclusions are problem-independent.

The weight factor has, as expected, a decisive influence on the correctness of the embedding. Too low values can break the process because the embedding will ignore new data and, as such, collapse into a standard t-SNE of the historical data physics space; values above 0.8−0.85 are recommended. Perplexity also has an impact, especially in the edge cases. However, Fig. 16 shows that, apart from edge cases, changes in R^2 are relatively small for a constant weight factor of 0.9. This entails that perplexity is not critical to a good embedding if conservative values are used. As such, perplexity values between 40–50% of the total sample size are recommended. Figure 17 shows an analogue experiment, but this time, colouring the contours by the MAM. A similar pattern is noticed, and although the MAM is more generous regarding feasible hyper-parameter combinations, the recommendation of a weight factor around 0.85 and perplexities in the range of 40–50% of the total sample size still yield the best results. It is also important to note that the feasible region appears to be convex in both problems. Practitioners may thus use the current recommendation as

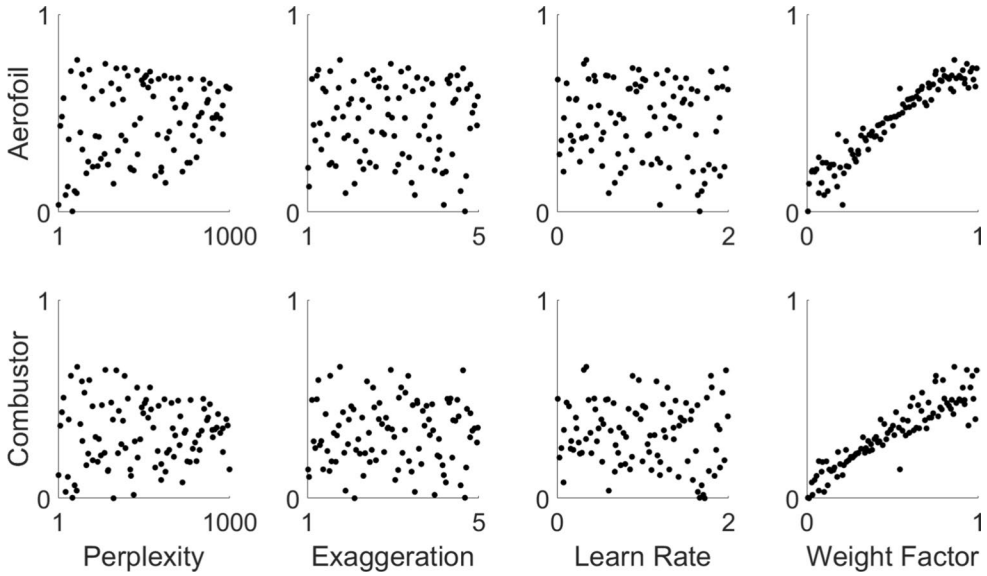


Fig. 15 Correlations between embedding accuracy metric (R^2 correlation between responses) and hyper-parameter values for the test cases. The weighting factor, α, is the most important parameter; perplexity is a distant second. Exaggeration and learn rate appear unimportant (provided sensible values are used) and will not be investigated further

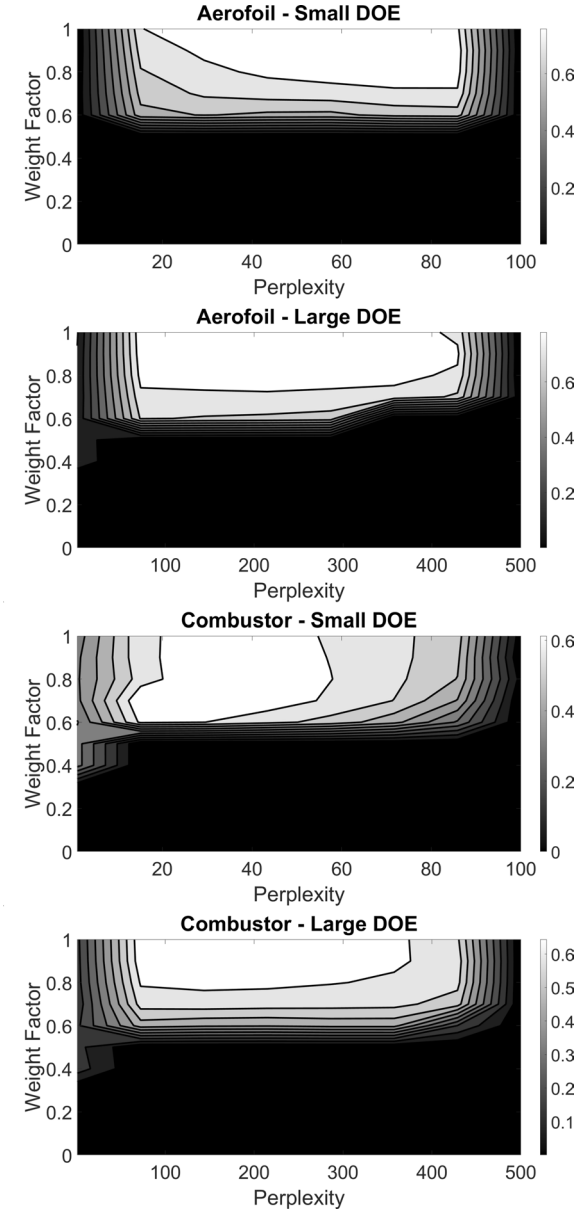


Fig. 16 Contours of response R^2 for the two key hyper-parameters

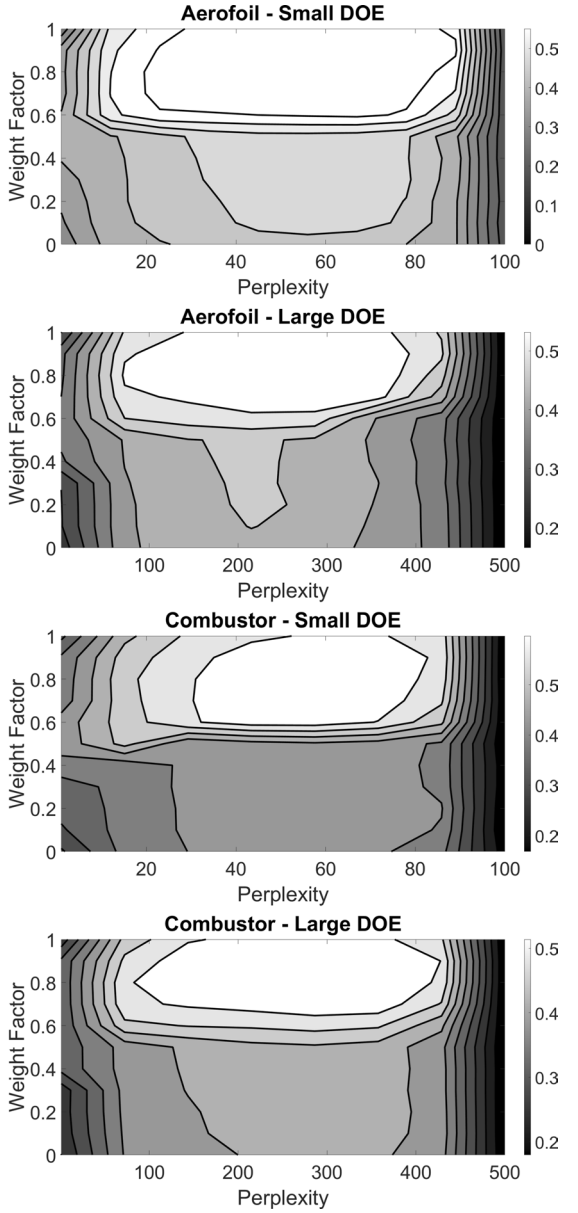


Fig. 17 Contours of response MAM for the two key hyper-parameters

a starting point, followed by a greedy binary search, if the computational budget allows.

Further practical considerations are explored in the appendix.

9 Conclusions and future work

A new method was proposed to address the cold start problem associated with sequential model-based optimisation (SMBO). The approach uses categorical Kriging as the surrogate model, and its key advantage is its ability to combine data sources with varying parameterisation. Unlike existing methods, no overlap is required between the two sets of parameters.

The method utilised simulation measurements from the code used to calculate the quantity of interest. For each sample, measurements were drawn from a constant grid and flattened into a vector. A crucial requirement was that the integral of the resulting vectors was correlated with the objective function. The vectors were then stacked in a physics matrix and treated as a latent, physics-based manifold. The distribution of samples in the manifold was used to build a t-SNE-based embedding, where the source data were embedded onto the manifold defined by the target data DOE in variable space.

The method shows promise in alleviating the cold start problem. For an aerofoil design optimisation problem, it achieved a reduction of 27–47% in the budget required to reach the same design as a standard Kriging. This improvement was attributed to an increase in the predictive accuracy of the underlying surrogate. For a more realistic combustor design scenario, the method achieved a reduction of more than 60% of samples needed to reach the same design as a Kriging-based SMBO. In the latter case, the predictive accuracy of the model was reduced due to the relatively simple design space of the source task. However, because of the linear nature of the historical objective function, the infill strategy could more efficiently identify the edge of the design space that housed the optimal set of parameters.

An element of the knowledge transfer problem that has remained unaddressed is how best to decide whether two physics matrices will lead to an embedding that ensures positive knowledge transfer. In the examples provided above, this was a given. Qualitatively, the extent of the similarity of the tasks can be ascertained from the latent spaces of the two tasks. For instance, in the combustor case (Fig. 12), it was easy to see that the objective functions were well correlated. Since the latent space is reproduced in the variable space using the embedding, this correlation likely carried over, ensuring positive transfer. However, the correlation may not always exist, and negative transfer, i.e. the transfer of false information from source to target tasks, may occur. In the context of the large databases of historical data available for most engineering companies, some quantitative measures, likely based on the aforementioned correlation, should be derived.

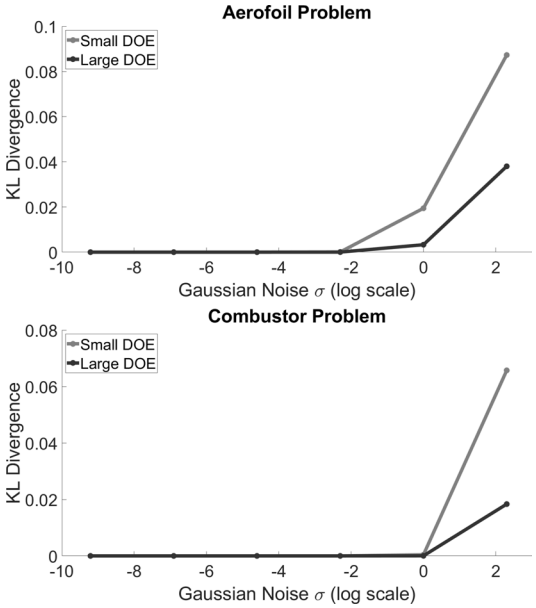
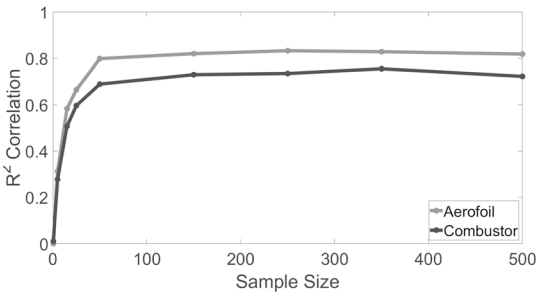
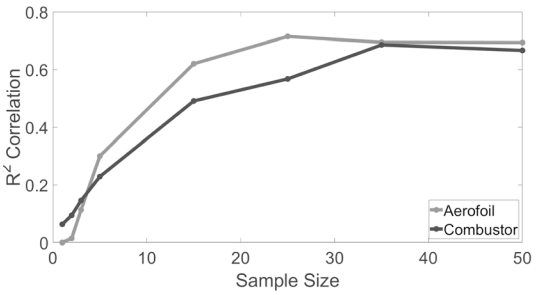


Fig. 18 Difference between noisy and standard embedding. A Gaussian distribution with varying standard deviation is added to physics matrix. Noise only becomes an issue for standard deviations equal to or above 1



(a) First case — 500 source samples



(b) Second case — 50 source samples

Fig. 19 Decay of embedding correctness as target sample size decreases. The x-axis indicates the number of points available in the target sample

Appendix A practical considerations

A.1 Sensitivity to noise

See Figs. 18, 19.

Noise is a common feature in engineering applications and, as such, studying the sensitivity of the embedding to random fluctuations is critical. It must be noted that the provided case studies will naturally contain some level of noise due to varying degrees of residual convergence. This section also adds random Gaussian noise of varying degree, to test how the method performs in more extreme cases.

The experiment in Sect. 8 is repeated, but this time the hyper-parameters are kept constant, at values, perplexity: 40% of sample size, weight factor: 0.85, exaggeration: 1, learn rate: 1. The small DOE consists of 100 samples, while the Large DOE is built using 1000 samples. A random sample of the standard normal distribution is collected using MATLAB's `randn` function and multiplied by increasingly larger scalars to simulate random normal distributions with increasing standard deviation.

The KL divergence metric is used to show the difference between the embedding generated when no noise is applied and the embedding obtained when the random Gaussian noise is added. Figure 18 shows that for both cases and regardless of DOE size noise only becomes a significant issue when the standard deviation of the noise Gaussian approaches a value of one. This is an encouraging sign as such values are relatively large and, the authors contend, could only be the result of a problematic finite volume code. Therefore, while the method would fail in such cases, they are unlikely to represent realistic scenarios.

Scaling and time complexity

As evidenced by van der Maaten (2013) the exact t-SNE implementation scales as $\mathcal{O}(n^2)$. This makes the method difficult to scale and intractable for massive datasets. For problems encountered in engineering design, however, this may not be an issue. The method was developed for cases where finite volume simulations require significant resources (e.g. large CFD codes taking upwards of hours or days to compute). The contention is that for such problems, the cost of the finite volume code is so high that practitioners will always be forced to work with a sparse sample. Furthermore, for engineering problems such as the combustor case described in section 7, studies such as Alizadeh et al (2020) recommend the use of Kriging, which has even worse scaling properties, with a time complexity of $\mathcal{O}(n^3)$.

Still, future publications will aim to alleviate the computational cost using proven solutions, such as the Barnes–Hut implementation of van der Maaten (2013), which reduces the complexity to $\mathcal{O}(n \log n)$.

Sensitivity to target data sparsity

The embedding was designed with sparse target sampling in mind. Still, it is important to quantify whether the method can produce useful embeddings under increased target data sparsity. The experiment of Section A.1 is repeated with slight modifications. Three of the hyper-parameters are constant: weight factor: 0.85, exaggeration: 1, learn rate: 1. The perplexity is varied so that it is always at 40% of the total sample size which, as previously discussed, usually generates a solution close to optimal. In Fig. 19, the correlation between predictions generated by an embedding and predictions generated by the known locations is used. This time, however, the embedding has to rely on smaller and smaller source datasets, as indicated by the x-axis. When a large historical dataset (500 samples) is available, the embedding appears to suffer a drastic drop-off at around the 15-25 sample mark. When a smaller historical sample is available (50 samples), there is no sudden drop-off, but rather a constant decay with embeddings becoming unusable around the 5-15 sample mark. This suggests that the method is prolific at working with sparse samples, as previously demonstrated by the test problems. However, practitioners should be aware that an all-encompassing verdict of how small is too small is impossible. It will depend on the problem, the number of variables, and the number of total samples. Therefore, the embeddings produced by the method should still be checked for consistency.

Acknowledgements Not applicable.

Author contributions Petru-Cristian Cimpoesu developed and tested the method presented in this paper. David J. Toal developed the Kriging toolset and pre- and post-processing scripts for the combustor CFD code and the VGK code. In addition, they had an indispensable supervisory role, helping conceive and develop the method presented in the paper. Leran Wang and Andy J. Keane had significant supervisory roles. They suggested better testing methods and assisted in improving the presentation of the method and results. Jonathan Gregory and Marco Nunez provided the proprietary CFD code used in the combustor case.

Conflict of interest On behalf of all authors, the corresponding author states that there is no Conflict of interest.

Replication of results The results presented within the paper have been created by a number of proprietary Rolls-Royce software tools. The raw results are, therefore, sensitive and cannot be published.

References

Alizadeh R, Allen J, Mistree F (2020) Managing computational complexity using surrogate models: a critical review. Res Eng Design. https://doi.org/10.1007/s00163-020-00336-7

Anand M, Eggels R, Staufer M, (2013) An advanced unstructured-grid finite-volume design system for gas turbine combustion analysis. ASME, et al (2013) Gas Turbine India Conference. GTINDIA 2013 https://doi.org/10.1115/GTINDIA2013-3537

Bandler JW, Biernacki RM, Chen SH, Grobelny PA, Hemmers RH (1994) Space mapping technique for electromagnetic optimization. IEEE Trans Microw Theory Tech 42(12):2536–2544

Bardenet R, Brendel M, Kégl B, Sebag M (2013) Collaborative hyperparameter tuning. In: Dasgupta S, McAllester D (eds) Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 28. PMLR, Atlanta, pp 199–207. https://proceedings.mlr.press/v28/bardenet13.html

Bartz-Beielstein T, Filipič B, Korošec P, Talbi, EG (eds) (2020) Infill Criteria for Multiobjective Bayesian Optimization, Springer International Publishing, Cham, pp 3–16. https://doi.org/10.1007/978-3-030-18764-4_1

Bellman R, Kalaba R (1959) On adaptive control processes. IRE Trans. Automatic Control 4(2):1–9. https://doi.org/10.1109/TAC.1959.1104847

Bonilla EV, Chai K, Williams C (2007) Multi-task gaussian process prediction. In: Platt J, Koller D, Singer Y, Roweis, ST (eds) Advances in Neural Information Processing Systems, vol 20. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2007/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf

Calandra R, Peters J, Rasmussen CE, Deisenroth MP (2016) Manifold gaussian processes for regression. 1402.5876

Chu W, Ghahramani Z (2005) Preference learning with gaussian processes. In: Proceedings of the 22nd International Conference on Machine Learning. Association for Computing Machinery, New York, ICML '05, p 137-144, https://doi.org/10.1145/1102351.1102369

Ding F, Kareem A (2018) A multi-fidelity shape optimization via surrogate modeling for civil structures. J Wind Eng Indus Aerod 178:49–56. https://doi.org/10.1016/j.jweia.2018.04.022

Forrester AIJ, Sóbester A, Keane AJ (2008) Engineering Design via Surrogate Modelling: A Practical Guide, 1st edn. Wilehttps://doi.org/10.1002/9780470770801

Garabedian PR, Korn D (1971) Analysis of transonic airfoils. Commun Pure Appl Math 24:841–851

Gove R, Cadalzo L, Leiby N, Singer JM, Zaitzeff A (2022) New guidance for using t-sne: alternative defaults, hyperparameter selection automation, and comparative evaluation. Visual Inform 6(2):87–97. https://doi.org/10.1016/j.visinf.2022.04.003

Gower JC, Dijksterhuis GB (2004) Procrustes Problems. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780198510581.001.0001

Gramacy R, Taddy M (2010) Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an r package for treed gaussian process models. Journal of Statistical Software 33. https://doi.org/10.18637/jss.v033.i06

Gramacy RB, Lee HKH (2009) Bayesian treed gaussian process models with an application to computer modeling. 0710.4536

Gupta A, Ong YS, Feng L (2018) Insights on transfer optimization: because experience is the best teacher. IEEE Trans Emerg Top Comput Intell 2(1):51–64. https://doi.org/10.1109/TETCI.2017.2769104

Hebbal A, Brevault L, Balesdent M, Talbi EG, Melab N (2021) Multi-fidelity modeling with different input domain definitions using deep gaussian processes. Struct Multidiscip Optim. https://doi.org/10.1007/s00158-020-02802-1

Hinton GE, Roweis ST (2002) Stochastic neighbor embedding. In: Neural Information Processing Systems, https://api.semanticscholar.org/CorpusID:20240

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Global Optim 13(4):455–492. https://doi.org/10.1023/A:1008306431147

Kennedy MC, O'Hagan A (2000) Predicting the output from a complex computer code when fast approximations are available. Biometrika 87(1):1–13. http://www.jstor.org/stable/2673557

Lin Q, Hu J, Zhou Q, Cheng Y, Hu Z, Couckuyt I, Dhaene T (2021) Multi-output gaussian process prediction for computationally expensive problems with multiple levels of fidelity. Knowl Based Syst 227:107151. https://doi.org/10.1016/j.knosys.2021.107151

Masters DA, Taylor NJ, Rendall TC, Allen CB, Poole DJ (2017) Geometric comparison of aerofoil shape parameterization methods. AIAA J 55(5):1575–1589. https://doi.org/10.2514/1.J054943

McInnes L, Healy J, Melville J (2020) Umap: Uniform manifold approximation and projection for dimension reduction. 1802.03426

Min AT, Sagarna R, Gupta A, Ong YS, Goh CK (2017) Knowledge transfer through machine learning in aircraft design. IEEE Comput Intell Mag 12(4):48–60. https://doi.org/10.1109/MCI.2017.2742781

Min ATW, Gupta A, Ong YS (2021) Generalizing transfer bayesian optimization to source-target heterogeneity. IEEE Trans Autom Sci Eng 18(4):1754–1765. https://doi.org/10.1109/TASE.2020.3017644

Pelamatti J, Brevault L, Balesdent M, Talbi EG, Guerin Y (2019) Efficient global optimization of constrained mixed variable problems. J Global Optim 73:583–613

Perron C, Rajaram D, Mavris D (2021) Multi-fidelity non-intrusive reduced-order modelling based on manifold alignment. Proc Royal Soc A: Math Phys Eng Sci 477:20210495. https://doi.org/10.1098/rspa.2021.0495

Renhui Z, Zhao X (2020) Inverse method of centrifugal pump blade based on gaussian process regression. Math Probl Eng 2020:1–10. https://doi.org/10.1155/2020/4605625

Robinson TD, Eldred MS, Willcox KE, Haimes R (2008) Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. AIAA J 46(11):2814–2822. https://doi.org/10.2514/1.36043

Rojas-Gonzalez S, Van Nieuwenhuyse I (2020) A survey on kriging-based infill algorithms for multiobjective simulation optimization. Comput Operat Res 116:104869. https://doi.org/10.1016/j.cor.2019.104869

Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 4(4):409–423. https://doi.org/10.1214/ss/1177012413

Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. In: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques. Association for Computing Machinery, New York, SIGGRAPH '86, p 151-160, https://doi.org/10.1145/15922.15903

Swiler LP, Hough PD, Qian P, Xu X, Storlie C, Lee H (2014) Surrogate Models for Mixed Discrete-Continuous Variables, Springer International Publishing, Cham, pp 181–202. https://doi.org/10.1007/978-3-319-04280-0_21

Tao J, Sun G (2019) Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. Aerosp Sci Technol 92:722–737. https://doi.org/10.1016/j.ast.2019.07.002

Tao S, Apley DW, Chen W, Garbo A, Pate DJ, German BJ (2019) Input mapping for model calibration with application to wing aerodynamics. AIAA J. doi 10(2514/1):J057711

Toal D (2023) Applications of multi-fidelity multi-output kriging to engineering design optimization. Struct Multidiscip Optim. https://doi.org/10.1007/s00158-023-03567-z

Toal DJ, Bressloff NW, Keane AJ, Holden CM (2011) The development of a hybridized particle swarm for kriging hyperparameter tuning. Eng Optim 43(6):675–699. https://doi.org/10.1080/0305215X.2010.508524

Toal DJ, Forrester AI, Bressloff NW, Keane AJ, Holden C (2009) An adjoint for likelihood maximization. Proc Royal Soc A: Math Phys Eng Sci 465(2111):3267–3287. https://doi.org/10.1098/rspa.2009.0096

Toal DJ, Keane AJ, Benito D, Dixon JA, Yang J, Price M, Robinson T, Remouchamps A, Kill N (2014) Multifidelity multidisciplinary whole-engine thermomechanical design optimization. J Propul Power 30(6):1654–1666. https://doi.org/10.2514/1.B35128

Toal DJJ, Bressloff NW, Keane AJ (2008) Kriging hyperparameter tuning strategies. AIAA J 46(5):1240–1252. https://doi.org/10.2514/1.34822

Toal DJJ, Zhang X, Keane AJ et al (2021) The potential of a multifidelity approach to gas turbine combustor design optimization. J Eng Gas Turbines Power 143(5):051002. https://doi.org/10.1115/1.4048654

Tran A, Tran M, Wang Y (2019) Constrained mixed-integer gaussian mixture bayesian optimization and its applications in designing fractal and auxetic metamaterials. Struct Multidiscip Optim. https://doi.org/10.1007/s00158-018-2182-1

van der Maaten L (2013) Barnes-hut-sne. https://arxiv.org/abs/1301.3342, 1301.3342

van der Maaten L, Hinton G (2008) Viualizing data using t-sne. J Mach Learn Res 9:2579–2605

Wills LM, Kolodner JL (1994) Towards more creative case-based design systems. In: Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence, AAAI'94, vol 12. AAAI Press, p 50-55

Zhan D, Xing H (2020) Expected improvement for expensive optimization: a review. J Global Optim 78(3):507–544. https://doi.org/10.1007/s10898-020-00923-x

Zhou Q, Qian PZG, Zhou S (2011) A simple approach to emulation for computer models With qualitative and quantitative factors. Technometrics 53(3):266–273. https://doi.org/10.1198/TECH.2011.10025