Autoencoder-enhanced joint dimensionality reduction for constrained Bayesian optimisation

Maathuis, H.F.; De Breuker, R.; Castro, Saullo G.P.

DOI
10.1088/2632-2153/ae0efe

Publication date
2025

Document Version
Final published version

Published in
Machine Learning

PAPER • OPEN ACCESS

Autoencoder-enhanced joint dimensionality reduction for constrained Bayesian optimisation

To cite this article: Hauke Maathuis *et al* 2025 *Mach. Learn.: Sci. Technol.* **6** 045028

IOP Publishing Mach. Learn.: Sci. Technol. 6 (2025) 045028 https://doi.org/10.1088/2632-2153/ae0efe

MACHINE LEARNING
Science and Technology

PAPER

Autoencoder-enhanced joint dimensionality reduction for constrained Bayesian optimisation

Hauke Maathuis*, Roeland De Breuker and Saullo G P Castro

Aerospace Engineering, Delft University of Technology, Delft, The Netherlands
* Author to whom any correspondence should be addressed.

E-mail: h.f.maathuis@tudelft.nl, r.debreuker@tudelft.nl and s.g.p.castro@tudelft.nl

Keywords: autoencoder, Bayesian optimisation, high-dimensional, dimensionlity reduction

OPEN ACCESS

RECEIVED
6 June 2025

REVISED
3 September 2025

ACCEPTED FOR PUBLICATION
1 October 2025

PUBLISHED
30 October 2025

Abstract

Bayesian Optimisation (BO) is a sample-efficient method for optimising expensive black-box functions, making it particularly suitable for engineering problems where gradients are unavailable and evaluating the objective or constraints is computationally costly. However, such problems often involve high-dimensional inputs and a large number of constraints, posing significant challenges for standard BO frameworks. While prior research has addressed scalability with respect to high-dimensional inputs in constrained settings, efficiently handling large numbers of constraints, i.e. high-dimensional outputs, remains an open problem. This work introduces **A**utoencoder-**E**nhanced Joint Dimensionality **R**eduction for C**o**nstrained BO (AERO-BO), a framework that performs dimensionality reduction in both the input (design variable) and output (objective and constraint) spaces via autoencoders. These autoencoders are trained online, requiring no pre-training, and their respective latent representations are connected through Gaussian Processes, which serve as surrogate models during optimisation. By operating in a joint latent space, AERO-BO enables scalable and efficient optimisation in settings with hundreds of design variables and thousands of black-box constraints.

1. Introduction

Engineering problems often aim to optimise performance, cost, and other objectives. In many cases, it is not straightforward to explore the global design space to identify the best combination of parameters. These challenges arise, for example, in the design of aerospace structures, where numerous constraints must be satisfied to ensure stability (Maathuis *et al* 2025), or in drug design, where factors such as synthesisability and the compounds toxicity may impose additional restrictions (Heifetz 2024). However, many of those problems involve complex models where obtaining analytical or numerical gradients is impractical or impossible. The objective function and constraints in such problems are not only computationally expensive to evaluate but may also be noisy, further complicating the optimisation process. These challenges, often encountered in engineering disciplines, give rise to so-called black-box problems, where only the input–output relationship of the model can be observed. Bayesian optimisation (BO) has emerged as a powerful and efficient method for addressing this challenge. It leverages a probabilistic model, frequently a Gaussian Process ($\mathcal{GP}$), to approximate the objective function and constraints and intelligently guide the search for optimal solutions. By balancing exploration of the search space and exploitation of promising regions, BO is particularly well-suited for problems where function evaluations are costly. However, the aforementioned engineering problems are frequently characterised by a high number of decision variables (inputs), as well as often involving thousands of constraints (outputs), incorporating diverse disciplines to analyse and ensure feasibility. This high dimensionality of the input and output space renders traditional optimisation methods computationally prohibitive. The problem at

hand can be formulated as

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^D} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & c_i(\mathbf{x}) \leqslant 0, \quad i = 1,\ldots,G \end{aligned} \tag{1}$$

with $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ denoting a design point in the input space, $f\colon \mathcal{X} \to \mathbb{R}$ being the objective function mapping from the input space to a scalar value and $\mathbf{c}\colon \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}^G$ being a collection of constraints $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^G$. The scalability of the BO methodology for these types of problems remains limited.

Despite the fact that high-dimensional BO in unconstrained settings is already challenging to optimise, Eriksson and Poloczek (2021) propose scalable constrained BO (SCBO), a promising method for efficiently handling high-dimensional and constrained optimisation problems. However, problems with possibly hundreds of design variables and thousands of constraints remain prohibitive due to computational resources. This limitation arises because SCBO builds the probabilistic surrogate models in the full input space and requires a separate surrogate model for every constraint.

To address these computational challenges, this work aims to jointly reduce the dimensionality of both. The goal is to construct a joint latent space, or in other words a space of reduced dimensionality, in which the probabilistic models for the objective and the constraints are built and learned online directly from the acquired data. In this context, we propose a novel framework called **A**utoencoder-**E**nhanced Joint Dimensionality **R**eduction in C**o**nstrained **B**ayesian **O**ptimisation (AERO-BO), which integrates autoencoders for dimensionality reduction in both input and output spaces. Autoencoders are neural networks designed to learn low-dimensional representations of high-dimensional data in an unsupervised manner. By combining autoencoders with $\mathcal{GP}$s, we aim to develop a scalable and efficient optimisation framework that is well-suited for high-dimensional input–output problems.

Our main contributions are:

1. Introducing AERO-BO for high-dimensional input–output problems
2. Demonstrating performance and scalability on benchmark cases
3. Applying AERO-BO on a multi-disciplinary real-world design problem from aerospace engineering

Structure of this article. The remainder of this article is structured as follows. First, we introduce the theoretical fundamentals and review the relevant literature. Next, we define autoencoders for dimensionality reduction. This is followed by the presentation of AERO-BO and an evaluation of its performance against a selection of existing methods. Finally, we conclude with a summary and discussion of the findings.

2. Constrained BO via GPs

Consider the constrained optimisation problem in equation (1). The optimal solution $\mathbf{x}^* \in \mathcal{X}_\mathrm{f} \subseteq \mathcal{X}$ lies within the feasible space $\mathcal{X}_\mathrm{f}$ defined by the constraints:

$$\mathcal{X}_\mathrm{f} = \{\mathbf{x} \in \mathcal{X} \mid c_i(\mathbf{x}) \leqslant 0, i = 1,\ldots,G\}. \tag{2}$$

In many practical applications, evaluating $f(\mathbf{x})$ and especially $c_i(\mathbf{x})\ \forall\ i = 1,\ldots,G$ can be computationally expensive or analytically intractable. BO, firstly introduced in Kushner (1962, 1964), addresses this by using probabilistic surrogate models to approximate the objective and constraints, allowing for efficient exploration and exploitation of the design space. $\mathcal{GP}$s are frequently employed due to their flexibility and ability to provide uncertainty estimates (Frazier 2018).

Commonly, in constrained BO, separate $\mathcal{GP}$s are used to model the objective function and each constraint function. Let $\mathcal{D} = \{(\mathbf{x}_j, f_j, \mathbf{c}_j)\}_{j=1}^N$ be the dataset of N observations. Based on this data set we further define the observation vector $\mathbf{f} = [f_1,\ldots,f_N]^\top$ or for the ith constraint $\mathbf{c}_i = [c_{i,1},\ldots,c_{i,N}]^\top$ and the corresponding input matrix $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_N]^\top$. The $\mathcal{GP}$ model is defined by a mean function $\mu\colon \mathcal{X} \to \mathbb{R}$ and a covariance (kernel) function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, written as

$$f(\mathbf{x})\,|\mathcal{D} \sim \mathcal{GP}\left(\mu(\mathbf{x}), k(\mathbf{x},\mathbf{x}')\right). \tag{3}$$

Similarly, each constraint $c_i(\mathbf{x})$ is modelled using a separate $\mathcal{GP}$:

$$c_i(\mathbf{x})\,|\mathcal{D} \sim \mathcal{GP}\left(\mu_i(\mathbf{x}), k_i(\mathbf{x},\mathbf{x}')\right)\ \forall i = 1,\ldots,G. \tag{4}$$

The models are trained by optimising the marginal likelihood (Rasmussen and Williams 2006), written as

$$\log p(\mathbf{f} \mid \mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{f}^{\top}\mathbf{K}^{-1}\mathbf{f} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log 2\pi, \tag{5}$$

with $\boldsymbol{\theta}$ being some trainable hyperparameters. This expression penalises both poor data fit and high model complexity, preventing overfitting. After training the model, the posterior distributions for both the objective and each constraint at a new query point $\mathbf{x}_+ \in \mathcal{X}$ remain Gaussian:

$$\begin{aligned} f(\mathbf{x}_+) \,|\mathcal{D}, \mathbf{x}_+ &\sim \mathcal{N}\left(\mu(\mathbf{x}_+), \sigma^2(\mathbf{x}_+)\right) \\ c_i(\mathbf{x}_+) \,|\mathcal{D}, \mathbf{x}_+ &\sim \mathcal{N}\left(\mu_i(\mathbf{x}_+), \sigma_i^2(\mathbf{x}_+)\right) \;\; \forall i = 1, \ldots, G. \end{aligned} \tag{6}$$

The posterior predictive distribution of a GPs is analytically tractable due to the conjugacy of the Gaussian prior and likelihood. Common choices for the kernel function include the squared exponential, Matérn, and rational quadratic kernels, each parametrised by hyperparameters $\boldsymbol{\theta}$. Then, for any new query point $\mathbf{x}_+$, the $\mathcal{GP}$ predictive posterior mean $\mu(\mathbf{x}_+)$ and variance $\sigma^2(\mathbf{x}_+)$ are given by:

$$\begin{aligned} \mu(\mathbf{x}_+) &= k(\mathbf{x}_+, \mathbf{X})\,\mathbf{K}^{-1}\mathbf{y}, \\ \sigma^2(\mathbf{x}_+) &= k(\mathbf{x}_+, \mathbf{x}_+) - k(\mathbf{x}_+, \mathbf{X})\,\mathbf{K}^{-1}k(\mathbf{X}, \mathbf{x}_+), \end{aligned} \tag{7}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the kernel matrix with entries $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $k(\mathbf{x}_+, \mathbf{X}) \in \mathbb{R}^N$ is the covariance between the new point and the training data.

Next, an acquisition function $\alpha(\mathbf{x}; \mathcal{D}_n) : \mathcal{X} \to \mathbb{R}$ makes use of the probabilistic surrogate model by encoding a utility policy to guide the selection of the next query point $\mathbf{x}_+$, balancing exploration and exploitation. For constrained BO, the acquisition function needs to account for both the objective and the constraints, thus trying to find a next feasible query point:

$$\mathbf{x}_+ = \underset{\mathbf{x}\in\mathcal{X}_f}{\operatorname{argmax}}\, \alpha(\mathbf{x}; \mathcal{D}_n). \tag{8}$$

Examples of constrained acquisition functions, such as constrained Expected Improvement (cEI) (Gardner *et al* 2014, Gelbart *et al* 2014) and their logarithmic extension log-cEI (logCEI) (Ament *et al* 2023), predictive entropy Search with constraints (Hernández-Lobato *et al* 2016) or constrained Thompson Sampling, as used in SCBO (Eriksson and Poloczek 2021). These methods incorporate probabilistic estimates from $\mathcal{GP}$s to guide the search towards regions that improve the objective while satisfying the constraints. By iteratively updating the $\mathcal{GP}$ models, constrained BO efficiently explores the design space.

However, high-dimensional input and output spaces introduce significant challenges, primarily due to the curse of dimensionality and storage limitations. As noted in equation (6), the objective and all constraints needs to be modelled via a separate or correlated $\mathcal{GP}$. When dealing with potentially thousands of outputs, this becomes computationally infeasible, as $\mathcal{GP}$s scale cubically with the number of sample points $\mathcal{O}(N^3)$ and storage requirements of $\mathcal{O}(N^2)$. Since high-dimensional problems normally require hundreds up to thousands of samples, the development of alternative approaches becomes necessary, which are discussed in the following section.

3. BO in high dimensions

This section reviews recent developments in BO for tackling problems with high-dimensional inputs and outputs, for unconstrained and constrained settings.

3.1. BO with high-dimensional inputs

Scaling BO to high-dimensional input problems poses three main challenges: increased predictive uncertainty, more model hyperparameters, and computational difficulty in optimising the acquisition function (Binois and Wycoff 2022).

A prominent line of research to mitigate these problems focuses on reducing the dimensionality of the input space. Linear projection methods such as REMBO (Wang *et al* 2016), ALEBO (Letham *et al* 2020), and HeSBO (Nayebi *et al* 2019) assume the objective varies in a lower-dimensional subspace. These methods rely on random or adaptive linear projections to restrict the search domain. Given a random matrix $\mathbf{A} \in \mathbb{R}^{d\times D}$, optimisation is performed in the subspace $\tilde{\mathcal{X}} \subseteq \mathbb{R}^d$, where $d \ll D$, with the function approximated as $g(\tilde{\mathbf{x}}) \approx f(\mathbf{A}\mathbf{x})$.

More recent work has explored non-linear dimensionality reduction techniques in the input space. Notably, autoencoder-based methods (Gómez-Bombarelli *et al* 2018, Tripp *et al* 2020, Grosnit *et al* 2021, Maus *et al* 2023) have been proposed, leveraging their ability to learn more flexible, data-adaptive representations of the optimisation space. For example, Gómez-Bombarelli *et al* (2018) pioneered latent space optimisation for molecular design, using a variational autoencoder to encode discrete molecular graphs into a continuous latent manifold. Tripp *et al* (2020) and Grosnit *et al* (2021) extended this idea to more general structured and high-dimensional inputs, demonstrating improved sample efficiency and optimisation in learned latent spaces. Maus *et al* (2023) further propose the use of encoders to reduce the dimensionality of high-dimensional intermediate components in composite functions, improving the efficiency in grey-box BO. However, these methods have primarily addressed unconstrained optimisation tasks and focus exclusively on input space compression. They do not account for high-dimensional constraint outputs or incorporate feasibility modelling into the latent optimisation loop.

SAASBO (Eriksson and Jankowiak 2021) takes a different approach, using a sparsity-inducing prior over $\mathcal{GP}$ lengthscales to identify relevant dimensions, gradually expanding the subspace as more data becomes available. Although effective, this approach incurs significant overhead. Trust region-based methods such as TuRBO (Eriksson *et al* 2019) and BAxUS (Papenmeier *et al* 2023) limit the search to a local region, balancing exploration and exploitation. BAxUS combines this with gradually expanding subspaces. More recently, Hvarfner *et al* (2024) demonstrated that standard GPs with scaled lengthscale priors can perform well in high-dimensional settings without embedding.

In constrained settings, SCBO (Eriksson and Poloczek 2021) extends TuRBO by using trust regions and constrained Thompson sampling, training one GP per constraint. Similarly, vanilla BO with logCEI (Ament *et al* 2023) provides a scalable baseline but also requires a separate GP for each constraint, limiting applicability in problems with many constraints. Section 4 details how AERO-BO addresses these issues through joint input–output dimensionality reduction.

3.2. BO with high-dimensional outputs

When multiple outputs (e.g. objectives and constraints) are involved, a straightforward approach is to model each independently using batched GPs, as in SCBO. However, modelling cross-output correlations can yield performance improvements.

Multi-task GPs (MTGPs) (Bonilla *et al* 2007) model output dependencies via structured covariance matrices using the Intrinsic or linear co-regionalisation model. When all outputs are observed at all points, the full covariance adopts a Kronecker structure $\mathbf{K}_{XX} \otimes \mathbf{K}_{\mathrm{f}} \in \mathbb{R}^{NG \times NG}$, enabling some computational savings. Nevertheless, their inference and memory complexity $\mathcal{O}(N^3G^3)$ and $\mathcal{O}(N^2G^2)$, respectively remains prohibitive for large-scale problems. High-Order GPs (Zhe *et al* 2019) extend GPs to matrix and tensor outputs. Maddox *et al* (2021) improved sampling efficiency using Matheron's rule, reducing complexity to $\mathcal{O}(N^3 + G^3)$ in the MTGP case, but scalability remains limited.

Another direction is output space dimensionality reduction. Higdon *et al* (2008) proposed using principal component analysis (PCA) to project outputs onto a lower-dimensional space, where GPs are trained. Variants include kPCA-GP and IsoMap-GP (Xing *et al* 2015, 2016). These reduce inference costs, but rely on linear (or fixed non-linear) embeddings. Maathuis *et al* (2025) combine PCA-GP with SCBO to handle many constraints efficiently. However, these methods assume fixed embeddings and are not learned adaptively during BO.

AERO-BO builds upon this line of work by jointly learning input and output embeddings via autoencoders during optimisation. This enables tractable modelling of high-dimensional constraints and scalable acquisition in joint latent spaces.

4. AERO-BO: constrained BO in a joint input–output latent space

As outlined earlier, only a few methods have been proposed for high-dimensional constrained BO. These include SCBO and standard BO with scaled lengthscale priors and logCEI (Hvarfner *et al* 2024), here referred to as vanilla BO. Both approaches construct one surrogate model per constraint in addition to the objective. However, most high-dimensional BO methods reviewed in section 3 target unconstrained problems. When constraints are considered, they are often handled using soft formulations such as penalty terms, as in BAxUS (Papenmeier *et al* 2023) and SAASBO (Eriksson and Jankowiak 2021). BAxUS, for instance, shows that combining subspace modelling with trust region strategies improves scalability. Nevertheless, extending random subspace methods to constrained settings remains challenging: as the number of constraints and the dimensionality grow, the likelihood of a randomly projected subspace containing feasible points declines significantly, limiting their applicability. In such cases, additional supervision during subspace construction may be required.

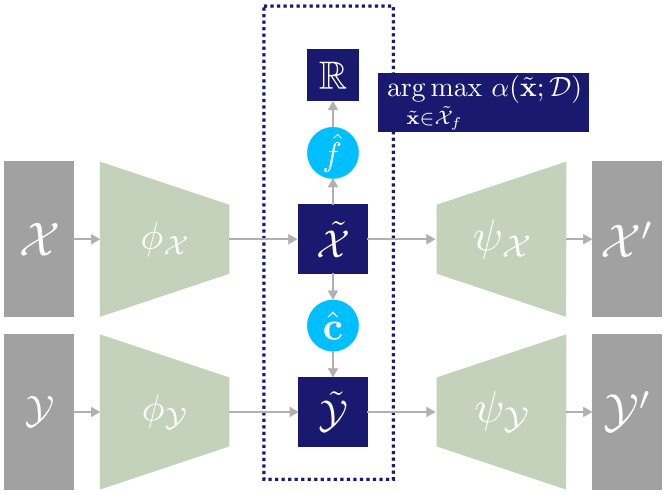

Figure 1. AERO-BO Architecture: Two autoencoders map the high-dimensional input and constraint output spaces to corresponding latent spaces which are connected via $\mathcal{GP}$s. These latent space surrogates are then used within the acquisition of the next points.

In parallel, dimensionality reduction using autoencoders has gained traction for addressing the curse of dimensionality in BO. Yet, prior work has focused exclusively on unconstrained problems and compressed only the input space. Reducing the dimensionality of the output space, particularly in the presence of numerous constraints, remains underexplored, despite its potential to yield substantial computational savings. We address this gap by employing autoencoders to reduce both the input and output dimensions, enabling BO for problems with hundreds of design variables and thousands of constraints.

This is achieved using two autoencoders (section 4.1): one for the inputs and one for the outputs. Each maps the high-dimensional data into a lower-dimensional latent representation that retains essential structural information. $\mathcal{GP}s$ are trained to model the relationships between the latent input and output spaces, as illustrated in figure 1. Following Maus *et al* (2024), the autoencoders and GPs are trained jointly to ensure that the latent representations remain coherent and aligned with the optimisation task. This joint training allows the latent spaces to adapt dynamically based on surrogate model feedback. The decoders are then used to project the optimised latent variables back into the original high-dimensional space for training purposes.

Additionally, AERO-BO trains both autoencoders in an online manner during the optimisation process, relying exclusively on data gathered within the BO loop. This eliminates the need for a costly offline pre-training phase, which may be impractical when function evaluations are expensive or data is limited. The online training scheme is a further novelty of our approach, enabling the latent representations to adapt in response to the evolving optimisation landscape. While autoencoders have been used previously in unconstrained BO, AERO-BO is, to our knowledge, the first to extend this concept to constrained high-dimensional problems by jointly learning latent representations for both design variables and constraint responses. This allows BO to scale to problem sizes that are otherwise intractable with existing methods.

In the following, we briefly introduce autoencoders before presenting our proposed method in detail.

4.1. Manifold-learning via autoencoders

This section briefly introduces autoencoders (Rumelhart *et al* 1986) as a manifold learning technique for non-linear dimensionality reduction. Unlike PCA, which is limited to linear transformations via eigendecomposition, autoencoders are capable of capturing non-linear data structures by learning mappings from the high-dimensional input space to a lower-dimensional latent space. This enables the representation of complex data manifolds that PCA may fail to model.

An autoencoder $\mathcal{A}(\mathbf{x}) = \psi \circ \phi(\mathbf{x})$ consists of two components: an encoder $\phi : \mathbb{R}^K \to \mathbb{R}^k$ that maps the input $\mathbf{x} \in \mathbb{R}^K$ to a lower-dimensional latent representation $\tilde{\mathbf{x}} \in \mathbb{R}^k$, and a decoder $\psi : \mathbb{R}^k \to \mathbb{R}^K$ that reconstructs an approximation of the original input. The input dimensionality K is determined by the data, while the latent dimension $k \ll K$ is a user-defined parameter. Throughout this work, both encoder and decoder are implemented as single-layer feedforward neural networks. The encoder applies a linear

transformation followed by a rectified linear unit (ReLU) activation:

$$\phi(\mathbf{x};\boldsymbol{\theta}_\phi) = \mathrm{ReLU}(\mathbf{W}_\phi \mathbf{x} + \mathbf{b}_\phi), \tag{9}$$

where $\boldsymbol{\theta}_\phi = \{\mathbf{W}_\phi, \mathbf{b}_\phi\}$ with $\mathbf{W}_\phi \in \mathbb{R}^{k\times K}$, $\mathbf{b}_\phi \in \mathbb{R}^k$ and $\mathrm{ReLU}(\cdot): \mathbb{R} \to \mathbb{R}$ denotes the element-wise ReLU defined by $\mathrm{ReLU}(z) = \max(0, z)$. The decoder performs a similar transformation using a sigmoid activation to constrain outputs to $[0,1]^K$:

$$\psi(\tilde{\mathbf{x}};\boldsymbol{\theta}_\psi) = \sigma(\mathbf{W}_\psi \tilde{\mathbf{x}} + \mathbf{b}_\psi), \tag{10}$$

with $\boldsymbol{\theta}_\psi = \{\mathbf{W}_\psi, \mathbf{b}_\psi\}$, where $\mathbf{W}_\psi \in \mathbb{R}^{K\times k}$, $\mathbf{b}_\psi \in \mathbb{R}^K$ and $\sigma(\cdot): \mathbb{R} \to \mathbb{R}$ is the sigmoid activation function applied element-wise, defined as $\sigma(z) = [1 + \exp(-z)]^{-1}$. Together, these network components define a parametrised non-linear mapping $\mathcal{A}(\mathbf{x}) = \psi \circ \phi(\mathbf{x})$ from the input space to an approximation of itself. The autoencoder is trained to minimise the reconstruction loss:

$$\mathcal{L}(\mathbf{x};\boldsymbol{\theta}_\phi,\boldsymbol{\theta}_\psi) = ||\mathbf{x} - \psi(\phi(\mathbf{x};\boldsymbol{\theta}_\phi);\boldsymbol{\theta}_\psi)||^2, \tag{11}$$

using stochastic gradient-based optimisation. This results in a compact representation that preserves the essential structure of the input data. Once trained, the encoder ϕ serves as a dimensionality reduction tool, efficiently mapping new samples $\mathbf{x}_*$ to their corresponding latent representations $\tilde{\mathbf{x}}_* = \phi(\mathbf{x}_*;\boldsymbol{\theta}_\phi)$.

4.2. AERO-BO: architecture

As previously mentioned, we employ two autoencoders to efficiently reduce the dimensionality of the input $\mathcal{X}$ and output $\mathcal{Y}$ spaces, respectively, defined as $\mathcal{A}_\mathcal{X}(\mathbf{x}) = \psi_\mathcal{X} \circ \phi_\mathcal{X}(\mathbf{x})$ and $\mathcal{A}_\mathcal{Y}(\mathbf{y}) = \psi_\mathcal{Y} \circ \phi_\mathcal{Y}(\mathbf{y})$. The encoders map data to lower-dimensional latent spaces, $\phi_\mathcal{X}: \mathcal{X} \subseteq \mathbb{R}^D \to \tilde{\mathcal{X}} \subseteq \mathbb{R}^d$ and $\phi_\mathcal{Y}: \mathcal{Y} \subseteq \mathbb{R}^G \to \tilde{\mathcal{Y}} \subseteq \mathbb{R}^g$, while the decoders reconstruct approximations of the original data:

$$\begin{aligned} \tilde{\mathbf{x}} &= \phi_\mathcal{X}(\mathbf{x};\boldsymbol{\theta}_{\phi,\mathcal{X}}), \quad \tilde{\mathbf{y}} = \phi_\mathcal{Y}(\mathbf{y};\boldsymbol{\theta}_{\phi,\mathcal{Y}}) \\ \mathbf{x}' &= \psi_\mathcal{X}(\tilde{\mathbf{x}};\boldsymbol{\theta}_{\psi,\mathcal{X}}), \quad \mathbf{y}' = \psi_\mathcal{Y}(\tilde{\mathbf{y}};\boldsymbol{\theta}_{\psi,\mathcal{Y}}). \end{aligned} \tag{12}$$

We denote the reconstructed data as $\mathbf{x}'$ and $\mathbf{y}'$, respectively, acknowledging that reconstruction may introduce an error, which the training process aims to minimise. The trainable parameters, $\boldsymbol{\theta}_{\phi,\mathcal{X}}, \boldsymbol{\theta}_{\phi,\mathcal{Y}}, \boldsymbol{\theta}_{\psi,\mathcal{X}}, \boldsymbol{\theta}_{\psi,\mathcal{Y}}$, govern the encoding and decoding transformations, ensuring optimal representation learning. When trained independently, these models remain decoupled. To address this, we adopt the approach proposed by Maus *et al* (2024) to couple them through variational $\mathcal{GP}$s, modelling the constraints in this joint latent space and train all models together. Specifically, we construct an approximate $\mathcal{GP}$ (Hensman *et al* 2014) for the objective function f, mapping from the latent input space $\tilde{\mathcal{X}}$ to a scalar value $\tilde{f}: \tilde{\mathcal{X}} \to \mathbb{R}$. Additionally, we construct independent approximate $\mathcal{GP}$s for the constraints $\mathbf{c}$, mapping from the latent input space to the latent output space $\tilde{\mathbf{c}}: \tilde{\mathcal{X}} \to \tilde{\mathcal{Y}}$. In total, this results in $n_m = 2 + 1 + g$ interconnected models: two autoencoders, one objective $\mathcal{GP}$ and g $\mathcal{GP}$s for the latent space constraints.

These relationships can be expressed as:

$$\begin{aligned} \tilde{f} \mid \tilde{\mathbf{X}} &\sim \mathcal{GP}\left(\mu(\tilde{\mathbf{X}}), k(\tilde{\mathbf{X}},\tilde{\mathbf{X}})\right) \\ \tilde{c}_i \mid \tilde{\mathbf{X}} &\sim \mathcal{GP}\left(\mu_i(\tilde{\mathbf{X}}), k_i(\tilde{\mathbf{X}},\tilde{\mathbf{X}})\right) \ \ \forall i = 1,\ldots,g. \end{aligned} \tag{13}$$

The n_m models are trained by minimising the joint loss:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) = &\log p\left(f \mid \phi_\mathcal{X}(\mathbf{x};\boldsymbol{\theta}_{\phi,\mathcal{X}}), \boldsymbol{\theta}_f\right) + \log p\left(\phi_\mathcal{Y}(\mathbf{c};\boldsymbol{\theta}_{\phi,\mathcal{Y}}) \mid \phi_\mathcal{X}(\mathbf{x};\boldsymbol{\theta}_{\phi,\mathcal{X}}), \boldsymbol{\theta}_{c_i}\right) \\ &+ ||\mathbf{x} - \psi_\mathcal{X}(\phi_\mathcal{X}(\mathbf{x};\boldsymbol{\theta}_{\phi,\mathcal{X}});\boldsymbol{\theta}_{\psi,\mathcal{X}})||^2 + ||\mathbf{c} - \psi_\mathcal{Y}(\phi_\mathcal{Y}(\mathbf{c};\boldsymbol{\theta}_{\phi,\mathcal{Y}});\boldsymbol{\theta}_{\psi,\mathcal{Y}})||^2 \end{aligned} \tag{14}$$

where the terms ensure the accurate reconstruction of inputs and outputs while maintaining the coherence of $\mathcal{GP}$ predictions. Here, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_f, \boldsymbol{\theta}_{c_i}, \boldsymbol{\theta}_{\phi,\mathcal{X}}, \boldsymbol{\theta}_{\psi,\mathcal{X}}, \boldsymbol{\theta}_{\phi,\mathcal{Y}}, \boldsymbol{\theta}_{\psi,\mathcal{Y}}\}$ represents not only the aforementioned hyperparameters of the autoencoders but also the ones of the $\mathcal{GP}$s. By leveraging automatic differentiation alongside the Adam optimiser (Kingma and Ba 2017), we efficiently train these interconnected models. We embed this modelling strategy into a trust region heuristic, akin to SCBO (Eriksson and Poloczek 2021). The proposed method is summarised in algorithm 1.

During the acquisition strategy we first sample from each latent space model's posterior to obtain $\hat{\tilde{f}}, \hat{\tilde{c}}_1, \ldots, \hat{\tilde{c}}_g$. Subsequently, a set of randomly sampled candidate points $\mathbf{X}_\mathrm{c}$ is mapped to the latent space $\tilde{\mathbf{X}}_\mathrm{c} \in \tilde{\mathcal{X}}$ using the encoder $\phi_\mathcal{X}$ which allows us to conduct constrained Thompson Sampling within the

Algorithm 1. AERO-BO.

Require: Input space $\mathcal{X}$, Number of initial samples N, Number of candidates N_c, batch size q_c, SCBO hyperparameters
1: Compute initial DoE $\mathcal{D}_0 = \{x_i, f(\mathbf{x}_i), \mathbf{c}(\mathbf{x}_i)\}_{i=1:N}$
2: Initialise SCBO state
3: Initialise models $\tilde{f}, \tilde{\mathbf{c}}, \mathcal{A}_\mathcal{X}$ and $\mathcal{A}_\mathcal{Y}$
4: $k = 0$
5: **while** Computational budget is not exhausted **do**
6: $\mathbf{x}_+ \leftarrow$ AcquisitionStrategy (see algorithm 2)
7: Evaluate $\mathbf{x}_+$ and observe $f(\mathbf{x}_+)$, $\mathbf{c}(\mathbf{x}_+)$
8: $\mathcal{D}_{k+1} = \mathcal{D}_k \cup \{x_+, f(\mathbf{x}_+), \mathbf{c}(\mathbf{x}_+)\}$
9: Update SCBO state
10: Update models $\hat{f}, \hat{\mathbf{c}}, \mathcal{A}_\mathcal{X}, \mathcal{A}_\mathcal{Y}$ jointly (see equation (14))
11: $k \leftarrow k + 1$
12: **end while**

Algorithm 2. AcquisitionStrategy in AERO-BO.

Require: Input space $\mathcal{X}$, Number of candidates N_c, batch size q_c, acquisition function $\alpha(\bullet)$, Samples form the $\mathcal{GP}$ posteriors in latent space $(\hat{\tilde{f}}, \hat{\tilde{c}}_1, \cdots, \hat{\tilde{c}}_g)$, autoencoder models $\mathcal{A}_\mathcal{X}, \mathcal{A}_\mathcal{Y}$
1: Generate N_c candidate $\mathbf{X}_c \in \mathbb{R}^{D \times N_c}$ with $\mathbf{x}_c^i \in \mathcal{X}_{tr}$
2: Map candidates into latent space $\tilde{\mathbf{X}}_c = \phi_\mathcal{X}(\mathbf{X}_c)$
3: Construct acquisition function $\tilde{\mathbf{A}} = \alpha(\tilde{\mathbf{X}}_c; \mathcal{D}_k)$ (see equation (15))
4: Choose the q next points $\mathbf{X}_+ \leftarrow \mathbf{X}_c[\operatorname{argmin} \tilde{\mathbf{A}}]$

joint latent space, employing the aforementioned latent space $\mathcal{GP}$s (equation (13)) for the objective and constraints. Therein, we use an utility function $\alpha(\tilde{\mathbf{x}})$, written as

$$\alpha(\tilde{\mathbf{x}}; \mathcal{D}) = \begin{cases} \operatorname{argmin}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}} \hat{\tilde{f}}(\tilde{\mathbf{x}}) & \text{if } \mathcal{F} \neq \emptyset \\ \operatorname{argmin}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}} \sum_{j=1}^{g} \max\left\{\hat{\tilde{c}}_j(\tilde{\mathbf{x}}), 0\right\} & \text{else} \end{cases} \tag{15}$$

with the set of feasible points, defined as $\mathcal{F} = \{\tilde{\mathbf{x}}_i \mid \hat{\tilde{c}}_j(\tilde{\mathbf{x}}_i) \leqslant 0,\ j = 1, \ldots, g\}$. By solving

$$\tilde{\mathbf{x}}_+ = \underset{\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}_f}{\operatorname{argmin}}\ \alpha(\tilde{\mathbf{x}}, \mathcal{D}) \tag{16}$$

a batch of q points in the latent space $\tilde{\mathcal{X}}$ can be obtained. Instead of using the decoder to map the points back, the indices are used to select the corresponding points in the original space to ensure they lie within the bounds (Maus *et al* 2024).

Summarising, from equations (15) and (16) it can be seen that the constrained Thompson sampling (Thompson 1933, Eriksson and Poloczek 2021) is performed in the latent space, making use of the encoded inputs $\tilde{\mathbf{x}}_i$ and outputs $\tilde{\mathbf{c}}_i$. This heuristic is summarised in algorithm 2.

During the first iteration of the BO algorithm, when $k = 0$, all four models are initialised and trained, then continuously updated after each batch of q new points is acquired. The trust region in this process acts to restrain the N_c candidate points locally by using a hyperrectangle. Thus, if we denote $\mathcal{X}_{tr}$ as the trust region-confined subspace of the original design space $\mathcal{X}$, we can write $\mathbf{X}_c \in \mathcal{X}_{tr} \subseteq \mathcal{X}$. This trust region is centred around the current best point in the original space $\mathbf{x}^* \in \mathcal{X}$. Initialised with an initial length $L = L_0$, the length is increased or decreased according to the progress of the optimisation. Therefore, the algorithm counts the number of successes and failures which record whether a better point has been found or not. In doing so, the centre of the trust region moves with success. Once a defined number of failures τ_f or successes τ_s is exceeded, the trust region length is either decreased in size $L \leftarrow \frac{L}{2}$ or increased $L \leftarrow \min\{2L, L_{max}\}$. If $L < L_{min}$, a new trust region is initialised.

5. Experiments

In the following, we evaluate AERO-BO on a range of high-dimensional input–output benchmark problems, as well as a real-world aircraft design task. As described in section 4.1, both encoder and decoder

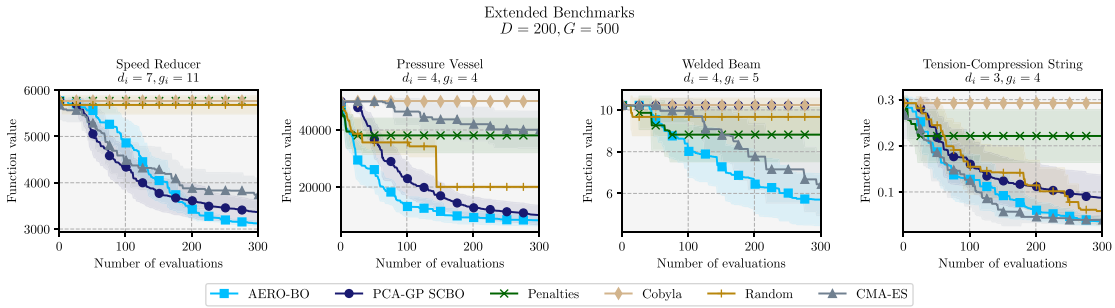

Figure 2. Collection of five physics-based and synthetic benchmarks all embedded in a higher dimensional space and augmented with synthetic constraints.

are implemented as single-layer feedforward neural networks. To assess performance, AERO-BO is compared against several baseline methods, including PCA-GP SCBO (Maathuis *et al* 2025), soft constraint handling via quadratic penalty terms (Eriksson and Jankowiak 2021), constrained optimisation by linear approximation (COBYLA) (Powell 1994), the covariance matrix adaptation evolution strategy (CMA-ES) (Hansen 2006), and a random search heuristic. Since SCBO and vanilla BO require training a separate surrogate model for each constraint, quickly exceeding memory limitations in large-scale problems, comparisons to these methods are discussed separately in section 5.3. AERO-BO is implemented using GPyTorch (Gardner *et al* 2018) and BoTorch (Balandat *et al* 2020). The code for reproducibility is available at: https://github.com/haukemaa/aerobo.

In line with Hernández-Lobato *et al* (2016), we adopt the principle that a feasible solution is always preferable over an infeasible one when comparing optimisation outcomes. Therefore, infeasible solutions are assigned the value of the worst feasible objective value found. Additionally, since PCA-GP SCBO also operates with a user-defined output latent dimension, we select the same latent dimension for PCA-GP SCBO as we do for AERO-BO, ensuring a fair comparison between the two methods.

5.1. Benchmarks

We adopt four physics-based test problems: the Speed Reducer (Lemonge *et al* 2010), Pressure Vessel (Coello Coello and Mezura Montes 2002), Welded Beam Design (Hedar and Fukushima 2006) and Tension Compression String (Hedar and Fukushima 2006). Additionally, we consider the 128D MOPTA08 problem with 68 black-box constraints (Anjos 2008). For more information, please refer to appendix A.1. To emulate the characteristics of many engineering problems having high-dimensional inputs and outputs, all benchmark problems are embedded in a $D = 200$ space and the number of constraints is artificially increase to $G = 500$ without altering the feasible optimal value of the optimisation problems (see appendix A.2). Thus, the original dimensionality (reduced space) and number of constraints of these problems are denoted by d_i and g_i, respectively, while D and G represent the new, extended dimensionality (original space) and number of constraints.

Physics-based and synthetic benchmarks. The results for the four physics-based test problems and the synthetic Ackley function are summarised in figure 2. We use a total budged of 300 evaluations, a batch size of $q = 5$, 10 initial samples and perform 20 experiments per method and benchmark. For the sake of generality, we apply the same hyperparameters for all benchmarks, namely learning rate $\alpha = 0.01$, number of epochs $N_e = 10$, dimension of the input latent space $\dim(\tilde{\mathcal{X}}) = 50$, dimension of the output latent space $\dim(\tilde{\mathcal{Y}}) = 10$. For the Speed Reducer problem, we note that while PCA-GP SCBO and CMA-ES initially converge slightly faster, AERO-BO ultimately identifies a superior solution compared to the other two methods. All other competing methods either fail to find a feasible point or struggle to make meaningful progress. In the pressure vessel problem, AERO-BO and PCA-GP SCBO deliver the best performance, closely followed by the Random Search heuristic. By contrast, constraints handled via Penalties, Cobyla, and CMA-ES either fail to identify a feasible point or remain stuck, unable to improve further. The Welded Beam benchmark, on the other hand, appears challenging for all algorithms, exhibiting large variances across methods. While PCA-GP SCBO and Cobyla fail to find a feasible point and Penalties stagnates, CMA-ES is ultimately outperformed by AERO-BO. In the Tension-Compression String benchmark, AERO-BO and CMA-ES demonstrate strong performances, dominating the results. PCA-GP SCBO and Random Search perform slightly worse, while Cobyla and Penalties show limited progress. Lastly, the synthetic Ackley function is dominated by PCA-GP SCBO, followed by AERO-BO and Penalties, while all other methods fail to locate a feasible solution. Additionally, we observe that

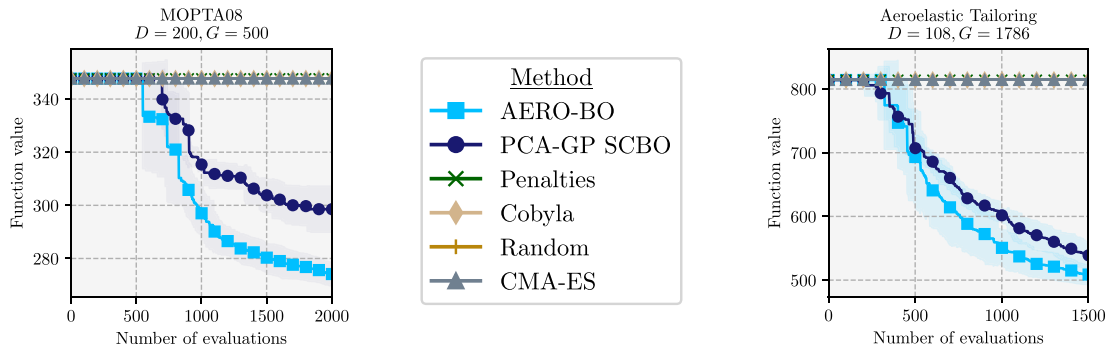

Figure 3. (left) 124*D* MOPTA08 benchmark with 68 black-box constraints. (right) 108*D* Aeroelastic Tailoring problem with 1786 black-box constraints arising form two different cases. The constraints ensure static strength, buckling, aeroelastic stability, aileron effectiveness and local angle of attack of the system.

the Cobyla algorithm struggles with optimising the extended benchmark problems (we verified this by testing on the original benchmarks, where it could identify feasible points). However, it is important to emphasise that the same hyperparameters and architectures, such as latent input and output dimensionalities, were applied across all methods. As demonstrated in the ablation study in section A.3, further improvements can be achieved by varying these parameters.

MOPTA08 benchmark. The MOPTA08 benchmark, proposed by Anjos (2008), originates from the automotive industry and is widely used for testing optimisation algorithms. The problem entails 124 dimensions and 68 black-box constraints. Again, we embed this benchmark into a 200D space and increase the number of constraints to 500. We utilise a total evaluation budget of 2000 samples, a batch size of $q = 15$, 100 initial samples and run 10 experiments. Furthermore, we set the latent dimension of the input autoencoder to $\dim(\tilde{\mathcal{X}}) = 60$ and for the latent output dimension to $\dim(\tilde{\mathcal{Y}}) = 20$. For training of AERO-BO we use $\alpha = 0.01$ and $N_e = 10$. The results of the MOPTA08 benchmark are shown in figure 3(left). Most methods fail to locate a feasible point under these challenging conditions, with the exception of AERO-BO and PCA-GP SCBO. Notably, AERO-BO outperforms PCA-GP SCBO, further reinforcing the findings from the previously discussed benchmarks.

5.2. Aeroelastic tailoring: a multi-disciplinary design optimisation problem

Finally, we test AERO-BO and the aforementioned methods on a real-world aircraft wingbox design problem. The objective is to optimise the weight of a wingbox while satisfying thousands of constraints. The wingbox is divided into multiple design regions, with its stiffness and thickness adjustable through the use of composite materials. The thousands of constraints that need to be satisfied, arise from multidisciplinary analyses, namely constraining the minimum strength of the structure, ensuring structural stability via a buckling analysis, as well as ensuring dynamic stability by analysing the dynamic aeroelastic capabilities of the structure, leading in total to $G = 1786$ constraints while $D = 108$ design variables describe the aforementioned structural properties. More on this problem can be found in Maathuis *et al* (2024).

In figure 3(right) the corresponding results are presented. AERO-BO and PCA-GP SCBO successfully identify a feasible solution after approximately 200 evaluations, even in this highly constrained design space. In contrast, all other methods fail to locate a feasible point, highlighting the robustness and effectiveness of AERO-BO. We use a total evaluation budget of 1500 samples, a batch size of $q = 15$, 100 initial samples and run 10 experiments. Furthermore, we set the latent dimension of the input autoencoder to $\dim(\tilde{\mathcal{X}}) = 60$ and the latent output dimension to $\dim(\tilde{\mathcal{Y}}) = 30$.

5.3. Comparison with SCBO and Vanilla BO

To evaluate the capabilities and limitations of AERO-BO, we compare its performance with two baseline methods: (i) SCBO (Eriksson and Poloczek 2021), which constructs a GP for each constraint and the objective in the full input space using a trust region heuristic, and (ii) a vanilla BO setup using the logCEI acquisition function, together with a scaled lengthscale prior, as proposed in Hvarfner *et al* (2024). While AERO-BO is specifically designed for high-dimensional and large-constraint settings where SCBO and vanilla BO becomes computationally intractable, this comparison serves to illustrate where each method excels, and how approximation errors introduced by latent modelling affect performance. We assess performance on two benchmark problems: the 7*D* Speed Reducer problem with

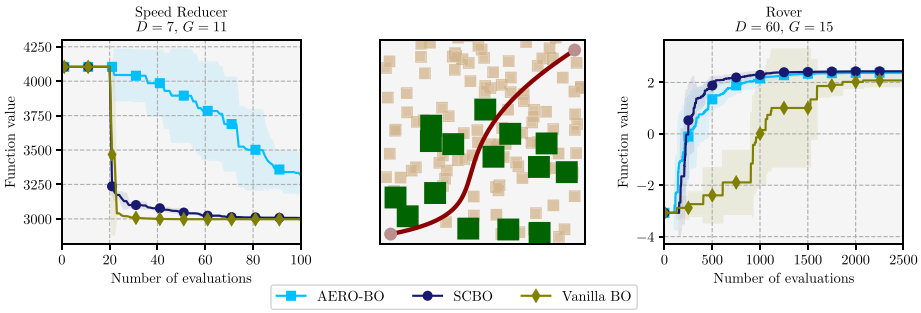

Figure 4. Comparison of AERO-BO with SCBO and Vanilla BO. (left) 7D Speed Reducer benchmark with 11 black-box constraints. Lemonge *et al* (2010) (centre) Example of an optimised trajectory. The green boxes are the impassable objects whereas the tanned objects are passable while adding a penalty to the objective function. (right) 60D Rover trajectory optimisation with 15 black-box constraints. Wang *et al* (2018).

11 black-box constraints (evaluated without up-projecting the input dimension and constraints), and the 60D Rover benchmark from Wang *et al* (2018). In the latter, the task is to optimise a robot trajectory encoded by 30 spline control points, each consisting of an (x, y) coordinate. Following the setup in Eriksson and Poloczek (2021), we introduce impassable rectangular obstacles as constraints (visualised in figure 4, centre).

For the Speed Reducer benchmark, we initialise with 20 samples and use a batch size of $q = 1$. AERO-BO uses arbitrary latent spaces of dimension $\dim(\tilde{\mathcal{X}}) = 4$ and $\dim(\tilde{\mathcal{Y}}) = 7$. For the Rover benchmark, we start with 100 initial samples and set $q = 100$, using latent dimensions $\dim(\tilde{\mathcal{X}}) = 20$ and $\dim(\tilde{\mathcal{Y}}) = 8$. The results, shown in figure 4, reveal that AERO-BO underperforms relative to both SCBO and vanilla BO on the Speed Reducer benchmark. This is expected, as the additional approximation introduced by output-space compression may lead to reduced accuracy in low-data regimes. Vanilla BO with logCEI, performs best in this case, likely due to the modest problem dimensionality and constraint count, which allow full-dimensional GP modelling without significant computational burden. SCBO also performs well, likewise benefiting from constructing the constraint surrogates directly. In contrast, on the high-dimensional Rover benchmark, vanilla BO fails to scale effectively, while AERO-BO performs almost as good as SCBO despite training significantly fewer surrogate models, demonstrating improved scalability. This comparison underscores several important insights. First, SCBO remains a strong choice for problems of moderate dimensionality and constraint count, particularly in early optimisation stages where surrogate models benefit from low uncertainty. However, its scalability is fundamentally limited, as it constructs and updates one full-dimensional GP per constraint. Second, AERO-BO introduces a trade-off: by reducing the dimensionality of the constraint space, it enables optimisation in previously intractable regimes, but at the cost of introducing an approximation. The quality of this approximation depends on the existence of a meaningful low-dimensional manifold in the output space, as well as the availability of sufficient training data. If these conditions are not met, as appears to be the case in the Speed Reducer problem, AERO-BO may fail to model high-frequency or strongly coupled constraint behaviour accurately, resulting in diminished performance.

These results highlight a fundamental limitation of latent-space BO approaches: if the effective dimensionality of the output space is high or the autoencoder is not trained with adequate data, compression may lead to loss of information and poor surrogate performance. This effect is explicitly visible in the reconstruction loss (see equation (14)) and is further compounded when the latent embedding is poorly aligned with the constraint geometry. Conversely, when a low-dimensional structure exists and is well captured, as in the Rover benchmark, AERO-BO offers a scalable alternative to conventional methods, maintaining competitive optimisation performance while drastically reducing the number of GPs.

In summary, AERO-BO fills a practical niche in the constrained BO landscape: it enables efficient optimisation in settings characterised by high input dimensionality and a large number of black-box constraints. However, its success is contingent upon the representational capacity of the autoencoders and the quality of the learned latent spaces. In settings where the number of constraints is low, methods such as SCBO or standard BO may still offer superior performance.

5.4. Ablation study

To better understand the behaviour and design trade-offs of AERO-BO, we conduct a series of ablation studies focusing on its key components and hyperparameters. In this subsection, we investigate the influence of the latent input space dimensionality, i.e. the output size of the encoder $\phi_{\mathcal{X}}$, on optimisation

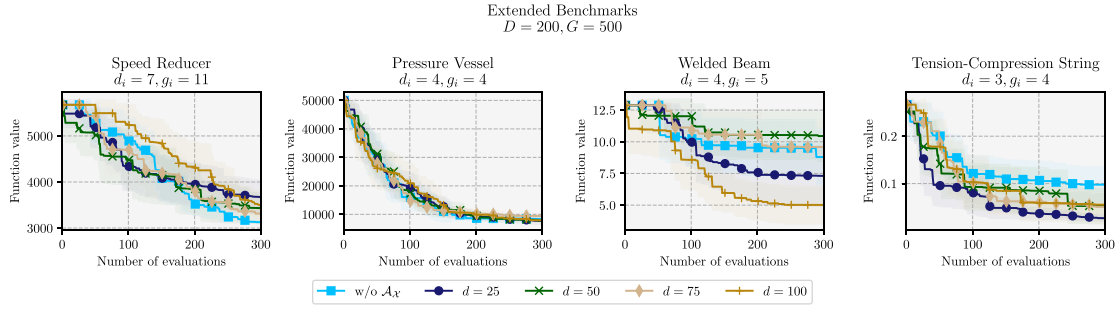

Figure 5. Input Dimension variation, investigating the influence of the input reduction on the performance of the method.

performance. Figure 5 presents results for the previously introduced, up-projected benchmark problems. We also include a baseline that omits input dimensionality reduction altogether, replacing the encoder $\phi_\mathcal{X}$ with the identity function. This baseline corresponds to a variant of SCBO that compresses only the output (constraint) space using an autoencoder rather than PCA.

The results show that input compression via $\mathcal{A}_\mathcal{X}$ is beneficial in most cases. However, the optimal choice of latent dimension is problem-specific. In the first two benchmarks (Speed Reducer and Pressure Vessel), performance differences between latent dimensions are relatively small. This suggests that the problems either admit a wide range of effective latent dimensionalities or that the search space is not strongly sensitive to compression in those cases. Consequently, no single latent size clearly dominates, and several configurations yield near-equivalent performance. In contrast, the third and fourth benchmarks (welded beam and tension-compression string) exhibit more distinct trends, depicting that performance can benefit from input dimensionality reduction. In the third case, a larger latent space ($\dim(\tilde{\mathcal{X}}) = 100$) performs best, followed $\dim(\tilde{\mathcal{X}}) = 25$, while $\dim(\tilde{\mathcal{X}}) = 50$ underperforms. This non-monotonic behaviour likely reflects trade-offs between expressiveness and regularisation: $\dim(\tilde{\mathcal{X}}) = 100$ may better capture complex interactions, whereas dimension $\dim(\tilde{\mathcal{X}}) = 25$ may promote smoother, more regular models. In the fourth benchmark, a smaller latent space ($\dim(\tilde{\mathcal{X}}) = 25$) yields the best performance, followed by dimension $\dim(\tilde{\mathcal{X}}) = 100$, then $\dim(\tilde{\mathcal{X}}) = 50$. This inversion indicates that for some problems, aggressive compression may help by suppressing noise and preventing overfitting.

As is common in latent variable modelling, the optimal latent dimensionality is both problem- and data-dependent. When the input–output mapping lies on a well-defined low-dimensional manifold and enough training data are available, small latent spaces can suffice. However, for intrinsically high-dimensional problems or in low-data regimes, excessive compression may discard informative variation, impairing surrogate fidelity and optimisation performance. These findings underscore the importance of selecting the latent dimension carefully. While AERO-BO is generally robust to this parameter, practitioners may benefit from heuristic guidance such as examining PCA eigenvalue spectra or monitoring surrogate reconstruction loss.

Further ablation studies on learning rate, training epochs, training window size, and the use of the trust region heuristic (Eriksson and Poloczek 2021) are provided in appendix A.3. These experiments confirm that AERO-BO performs robustly across a wide range of configurations, reinforcing its suitability for practical use.

6. Conclusion

In this work, we introduce AERO-BO, a novel approach for high-dimensional BO that effectively handles problems with hundreds of design variables as well as thousands of constraints. Unlike many traditional high-dimensional BO methods, which rely on random input embeddings and face challenges in constrained settings, AERO-BO employs a unique strategy: two jointly trained autoencoders that map both inputs and outputs to lower-dimensional latent spaces. This approach eliminates the need to construct separate surrogate models for each constraint, making it computationally efficient and scalable. We demonstrate the effectiveness of AERO-BO across a variety of benchmarks, including existing and extended high-dimensional test problems and a complex real-world multidisciplinary design optimisation problem from aerospace engineering. The results showed that AERO-BO consistently outperforms competing methods in finding feasible solutions for highly constrained problems, which are often encountered in engineering optimisation. This capability is particularly crucial when other methods struggle to find feasible points, highlighting the reliability and robustness of AERO-BO.

Data availability statement

No new data were created or analysed in this study.

Acknowledgment

The authors would like to express their sincere gratitude to Embraer S A, especially Pedro Higino Cabral and Alex Pereira do Prado, for their invaluable support and collaboration within the Aeroelastic Tailoring Enabled Design project. Their expertise, resources, and guidance have been instrumental in the successful completion of this study.

Artificial intelligence was used in the writing process to assist with language refinement and clarity. All AI-generated content was reviewed and verified by the authors to ensure accuracy and alignment with the research objectives.

Appendix

A.1. Benchmark problems

This section briefly describes the in this work used benchmark cases.

7D Speed Reducer with 11 black-box constraints. This benchmark poses an optimisation problem to minimise the weight of the so-called speed reducer. The design variables include geometrical measures like the length of two shafts and the width of the face also the number of teeth on a pinion and the module of teeth (Lemonge *et al* 2010).

4D Pressure Vessel with 4 black-box constraints. The goal of this benchmark is to minimise the cost of the design. The design variables include the shell and head thicknesses, as well as the inner radius and length of the cylindrical section including some bounds (Coello Coello and Mezura Montes 2002).

4D Welded Beam with 5 black-box constraints. The Welded Beam benchmark aims to minimise the cost by considering mechanical limits on the structure, such as shear stress, bending stress, buckling load and the deflection of the beam (Hedar and Fukushima 2006).

3D Tension-Compression Spring with 4 black-box constraints This problem also takes into account mechanical constraints such as a limit on the deflection, shear stress and surge frequency. In addition a geometrical parameter is added, describing the outside diameter. While considering these constraints, the aim is to minimise the weight of the tension compression string (Hedar and Fukushima 2006).

10D Ackley with 2 black-box constraints. This benchmark is a synthetic function, considering two black-box constraints and is known to be difficult to optimise taken from Eriksson and Poloczek (2021).

128D MOPTA08 with 68 black-box constraints. Lastly, the MOPTA08 benchmark was proposed by Anjos (2008) and stems form the automotive industry. The design variables herein describe the material and shape of the structure as well as gages while taking into account some performance constraints.

60D Rover with 15 black-box constraints. This problem was originally considered by Wang *et al* (2018), optimising the trajectory of a rover by fitting a B-spline to 30 design points, described by x and a y coordinate. The objective function is $f(x) = c(x) + 5$ where $c(x)$ penalises collisions with objects by -20. Additionally, Eriksson and Poloczek (2021) propose to add 15 impassible objects. More on the constraint formulation can be found in their paper.

A.2. Extending benchmark problems

To evaluate the performance of optimisation algorithms in high-dimensional and highly constrained scenarios, we extend existing benchmark problems in terms of dimensionality and the number of constraints. Considering a d_i-dimensional optimisation problem including g_i black-box constraints as follows

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{x}) \leqslant 0, \tag{A1}$$

with $\mathbf{c} \in \mathbb{R}^{g_i}$. The goal is to extend the existing problem to D dimensions and G constraints, where $d_i \ll D$ and $g_i \ll G$. To achieve this, two projection matrices are defined: $\mathbf{P}_d : \mathbb{R}^D \to \mathbb{R}^{d_i}$, mapping the high-dimensional input space back to the original space, and $\mathbf{P}_{g_i} : \mathbb{R}^G \to \mathbb{R}^{g_i}$, mapping the extended constraint space to the original constraint space. Using these projections, the extended optimisation problem can be written as

$$\min_{\mathbf{x}' \subseteq \mathcal{X}' \in \mathbb{R}^D} f(\mathbf{P}_{d_i}\mathbf{x}') \quad \text{s.t.} \quad \mathbf{c}(\mathbf{P}_{d_i}\mathbf{x}') \leqslant 0, \tag{A2}$$

where $\mathbf{P}_{d_i}$ retains only the original d_i dimensions, ensuring that the objective function $f(\mathbf{P}_{d_i}\mathbf{x}') = f(\mathbf{x})$ is unchanged. Similarly, the constraints satisfy $\mathbf{c}(\mathbf{P}_{d_i}\mathbf{x}') = \mathbf{c}(\mathbf{x})$. Here, the extended variable $\mathbf{x}'$ is decomposed into $\mathbf{x}' = [\mathbf{x}^\top, \tilde{\mathbf{x}}^\top]^\top$ where $\tilde{\mathbf{x}} \in \mathbb{R}^{(D-d_i)}$ represents some artificial dimensions. To introduce additional constraints, we define the artificial constraints as follows:

$$\tilde{\mathbf{c}}(\tilde{\mathbf{x}}) = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}, \tag{A3}$$

where $\mathbf{A} \in \mathbb{R}^{(G-g_i)\times(D-d_i)}$ and $\mathbf{b} \in \mathbb{R}^{(D-d_i)}$ are a random matrix and vector, respectively, where $\mathbf{b}$ ensures that the artificial constraints are non-violated by construction. The extended constraints are then formulated by combining the original and artificial constraints:

$$\mathbf{c}'(\mathbf{x}') = [\mathbf{c}^\top(\mathbf{P}_{d_i}\mathbf{x}'), \tilde{\mathbf{c}}^\top(\tilde{\mathbf{x}})]^\top \in \mathbb{R}^G. \tag{A4}$$

Thus, the final extended optimisation problem becomes:

$$\min_{\mathbf{x}' \subseteq \mathcal{X}' \in \mathbb{R}^D} f(\mathbf{x}') \quad \text{s.t.} \quad \mathbf{c}'(\mathbf{x}') \leqslant 0. \tag{A5}$$

This approach ensures that the characteristics of the original problem are preserved, while allowing for scalability in terms of dimensionality and constraint complexity.

A.3. Additional ablation studies and sensitivity analyses

In this section we present some additional ablation studies to investigate the performance of AERO-BO with respect to some of its hyperparameters as well as the influence of the trust region heuristic.

A.3.1. Training hyperparameters

We analyse the performance of AERO-BO concerning the learning rate α, used to jointly train the models. Figure A1 shows that the algorithm exhibits robustness across different values of α, with the best performance achieved at $\alpha = 0.001$. The impact of the number of training epochs N_e is investigated while keeping the learning rate fixed. Figure A2 demonstrates robust performance across different values of N_e. Notably, when the models are not updated ($N_e = 0$), performance significantly deteriorates in some cases, underscoring the importance of periodic updates. Lastly, the influence of the training window size or also called lookback factor N_b is investigated, inspired by Maus *et al* (2024). The lookback factor determines the number of recent samples within the current dataset $\mathcal{D}$ used during training. Figure A3 illustrates that utilising all samples ($N_b = N$) generally yields the best results, except for the Welded Beam benchmark. However, considering all samples may increase training costs.

A.3.2. Influence of the trust region heuristic

The role of the trust region heuristic is evaluated, as introduced by Eriksson *et al* (2019) and extended for constrained problems in Eriksson and Poloczek (2021). Figure A4 compares two scenarios: Enforcing that N_c candidate points lie within a trust region centred around the best solution so far and allowing candidate sampling across the entire design space. The use of perturbation probability, as noted by Rashidi *et al* (2024), remains integral to the SCBO framework and is used in both cases. The results in figure A4 suggest that while the trust region heuristic has limited impact on the speed of finding a feasible point, it plays a crucial role in identifying higher-quality solutions later in the optimisation process.

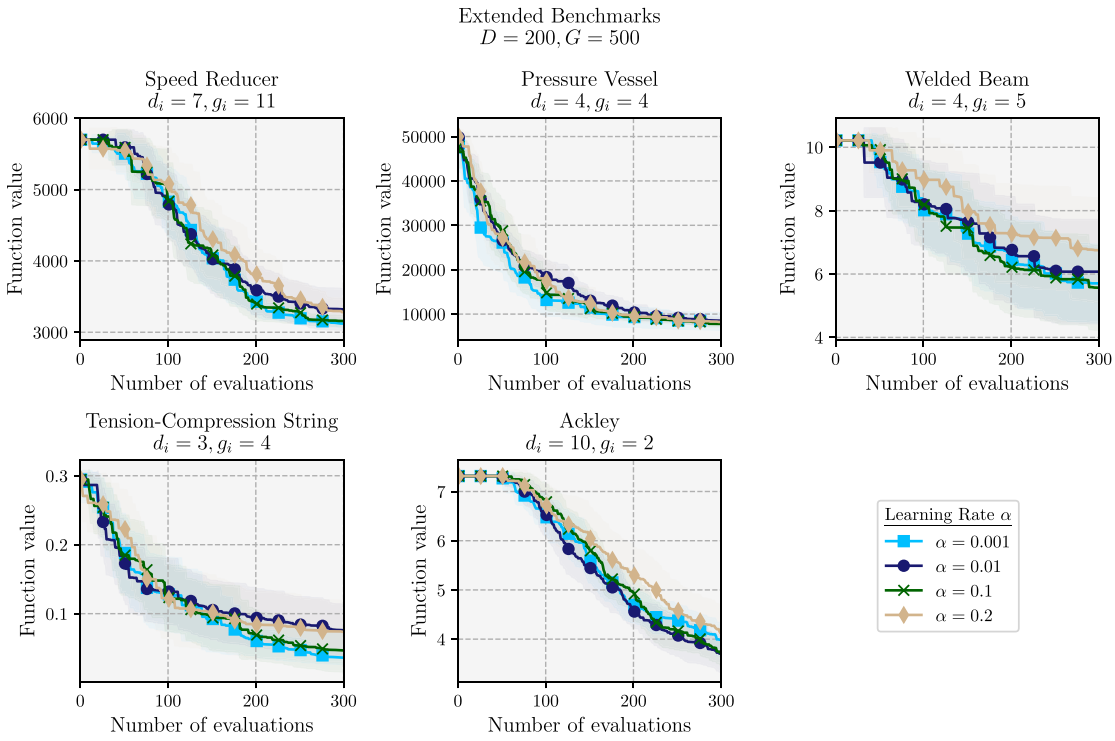

Figure A1. Sensitivity analysis of the learning rate α.

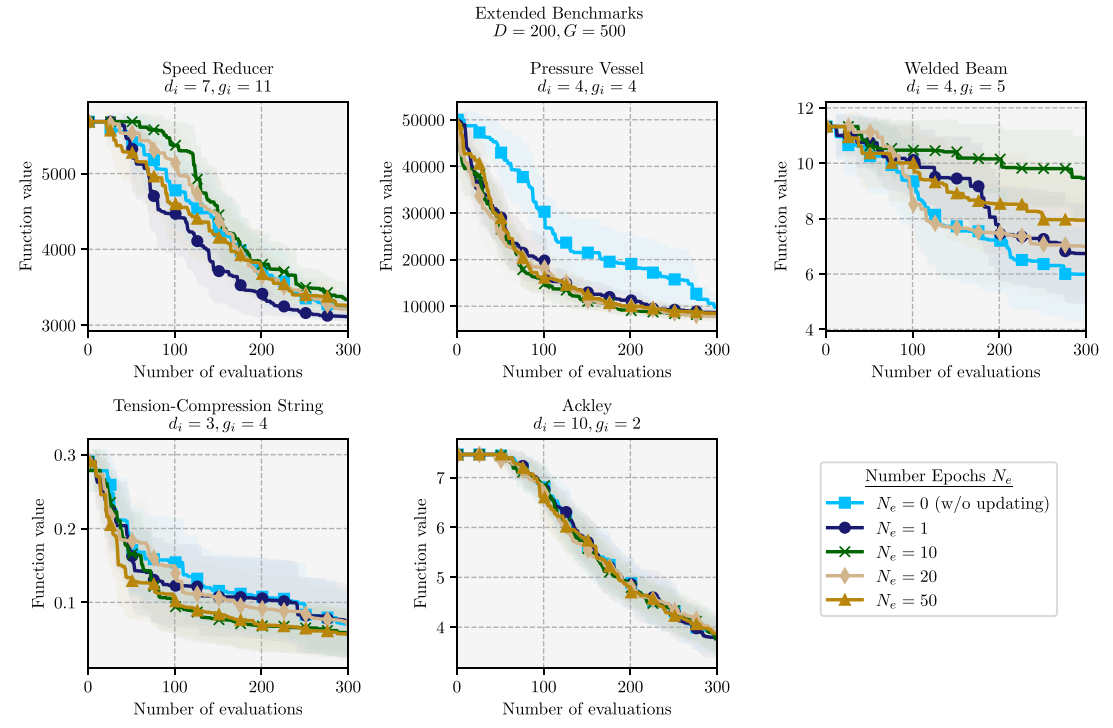

Figure A2. Sensitivity analysis of the number of training epochs N_e.

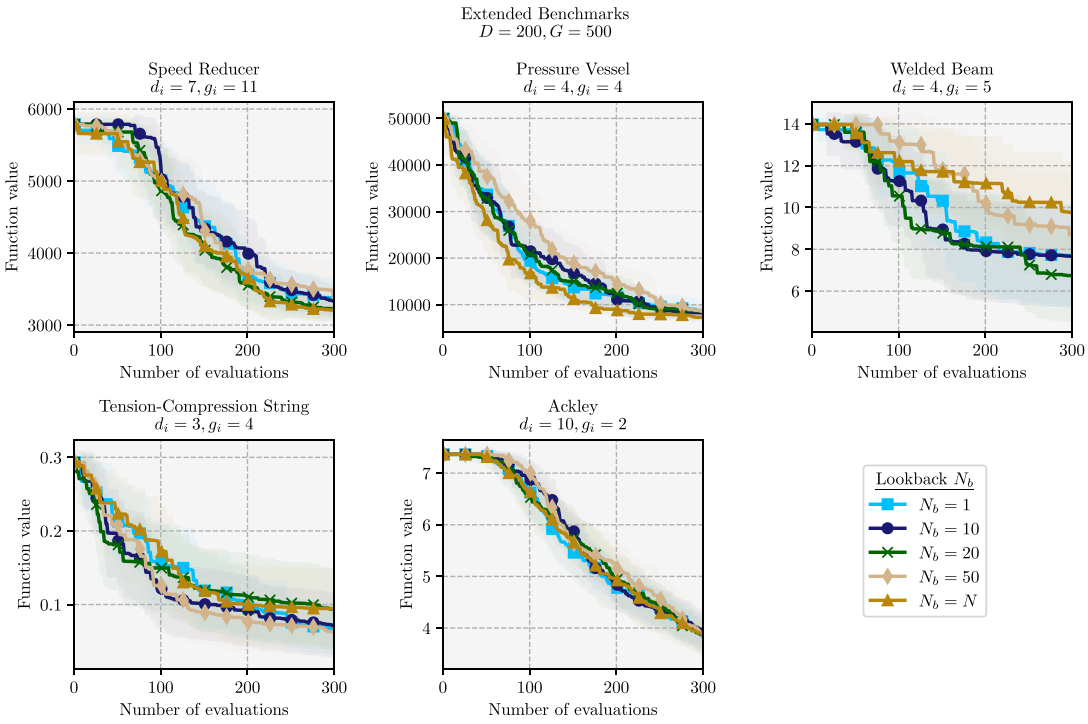

Figure A3. Ablation study of the lookback factor N_b with $N_b = N$ denoting the case where all samples within the current $\mathcal{D}$ are taken into account for training.

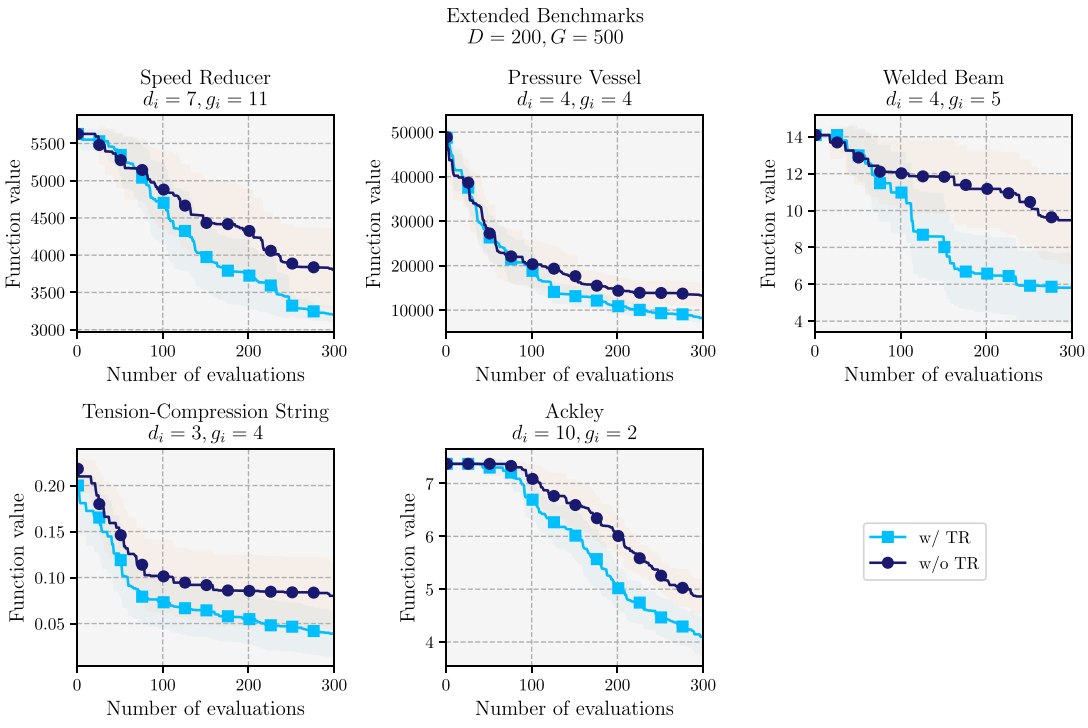

Figure A4. Ablation study investigating the influence of the trust region heuristic on the performance of AERO-BO

ORCID iDs

Hauke Maathuis 0000-0001-7002-4823
Roeland De Breuker 0000-0002-7882-2173
Saullo G P Castro 0000-0001-9711-0991

References

Ament S, Daulton S, Eriksson D, Balandat M and Bakshy E 2023 Unexpected improvements to expected improvement for Bayesian optimization *37th Conf. on Neural Information Processing Systems* (*New Orleans*)

Anjos M 2008 The MOPTA 2008 Benchmark (available at: www.miguelanjos.com/jones-benchmark)

Balandat M, Karrer B, Jiang D R, Daulton S, Letham B, Wilson A G and Bakshy E 2020 BoTorch: a framework for efficient Monte-Carlo Bayesian optimization *Advances in Neural Information Processing Systems* vol 33

Binois M and Wycoff N 2022 A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization *ACM Trans. Evol. Learn. Optim.* **2** 1–26

Bonilla E V, Chai K and Williams C 2007 Multi-task Gaussian process prediction *Advances in Neural Information Processing Systems* vol 20, ed J Platt, D Koller, Y Singer and S Roweis (Curran Associates, Inc.)

Coello C A and Mezura Montes E 2002 Constraint-handling in genetic algorithms through the use of dominance-based tournament selection *Adv. Eng. Inf.* **16** 193–203

Eriksson D and Jankowiak M 2021 High-dimensional Bayesian optimization with sparse axis-aligned subspaces *37th Conf. on Uncertainty in Artificial Intelligence*

Eriksson D, Pearce M, Gardner J, Turner R D and Poloczek M 2019 Scalable global optimization via local Bayesian optimization *33rd Conf. on Neural Inf. Processing Systems* (*Vancouver*)

Eriksson D and Poloczek M 2021 Scalable constrained Bayesian optimization *24th Int. Conf. on Artificial Intelligence and Statistics* (*San Diego*)

Frazier P I 2018 A tutorial on Bayesian optimization (arXiv:1807.02811 [cs, math, stat])

Gardner J R, Kusner M J and Jake G 2014 Bayesian optimization with inequality constraints *31st Int. Conf. on Machine Learning*

Gardner J R, Pleiss G, Bindel D, Weinberger K Q and Wilson A G 2018 Gpytorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration *Advances in Neural Information Processing Systems*

Gelbart M A, Snoek J and Adams R P 2014 Bayesian optimization with unknown constraints *30th Conf. on Uncertainty in Artificial Intelligence*

Gómez-Bombarelli R *et al* 2018 Automatic chemical design using a data-driven continuous representation of molecules *ACS Cent. Sci.* **4** 268–76

Grosnit A *et al* 2021 High-dimensional Bayesian optimisation with variational autoencoders and deep metric learning (arXiv:2106.03609)

Hansen N 2006 The CMA evolution strategy: a comparing review *Towards a New Evolutionary Computation*

Hedar A-R and Fukushima M 2006 Derivative-free filter simulated annealing method for constrained continuous global optimization *J. Glob. Optim.* **35** 521–49

Heifetz A ed 2024 *High Performance Computing for Drug Discovery and Biomedicine* (*Methods in Molecular Biology*) vol 2716 (Springer)

Hensman J, Matthews A and Ghahramani Z 2014 Scalable variational Gaussian process classification *18th Int. Conf. on Artificial Intelligence and Statistics* (*San Diego*)

Hernández-Lobato J M, Gelbart M A, Adams R P, Hoffman M W and Ghahramani Z 2016 A general framework for constrained Bayesian optimization using information-based search *J. Mach. Learn. Res.* **17** 1–53

Higdon D, Gattiker J, Williams B and Rightley M 2008 Computer model calibration using high-dimensional output *J. Am. Stat. Assoc.* **103** 570–83

Hvarfner C, Hellsten E O and Nardi L 2024 Vanilla Bayesian optimization performs great in high dimensions *Proc. 41st Int. Conf. on Machine Learning* (*Proc. Machine Learning Research*) vol 235, ed R Salakhutdinov, Z Kolter, K Heller, A Weller, N Oliver, J Scarlett and F Berkenkamp (PMLR) pp 20793–817

Kingma D P and Ba J 2017 Adam: A Method for Stochastic Optimization *Int. Conf. on Learning Representations*

Kushner H J 1962 A versatile stochastic model of a function of unknown and time varying form *J. Math. Anal. Appl.* **5** 150–67

Kushner H J 1964 A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise *J. Basic Eng.* **86** 97–106

Lemonge A C C, Barbosa H J C, Hasenclever Borges C C and dos Santos Silva F B 2010 Constrained optimization problems in mechanical engineering design using a real-coded steady-state genetic algorithm (available at: https://api.semanticscholar.org/CorpusID:54994542)

Letham B, Calandra R, Rai A and Bakshy E 2020 Re-examining linear embeddings for high-dimensional Bayesian optimization (arXiv:2001.11659 [cs, stat])

Maathuis H F, De Breuker R and Castro S G P 2025 Scaling Bayesian optimization for high-dimensional and large-scale constrained spaces *AIAA J.* 1–11

Maathuis H, De Breuker R and Castro S G P 2024 High-dimensional Bayesian optimisation with large-scale constraints – an application to aeroelastic tailoring *AIAA SCITECH 2024 Forum*

Maddox W J, Balandat M, Wilson A G and Bakshy E 2021 Bayesian optimization with high-dimensional outputs *35th Conf. on Neural Information Processing Systems*

Maus N, Jones H T, Moore J S, Kusner M J, Bradshaw J and Gardner J R 2023 Local latent space Bayesian optimization over structured inputs *36th Conf. on Neural Information Processing Systems*

Maus N, Lin Z J, Balandat M and Bakshy E 2024 Joint composite latent space Bayesian optimization *41 st Int. Conf. on Machine Learning*

Nayebi A, Munteanu A and Poloczek M 2019 A framework for Bayesian optimization in embedded subspaces *Proc. 36th Int. Conf. on Machine Learning* (*Proc. Machine Learning Research*) ed K Chaudhuri and R Salakhutdinov pp 4752–61

Papenmeier L, Nardi L and Poloczek M 2023 Increasing the scope as you learn: adaptive Bayesian optimization in nested subspaces *36th Conf. on Neural Information Processing Systems*

Powell M J D 1994 A direct search optimization method that models the objective and constraint functions by linear interpolation (available at: https://api.semanticscholar.org/CorpusID:118045691)

Rashidi B, Johnstonbaugh K and Gao C 2024 Cylindrical Thompson sampling for high-dimensional Bayesian optimization *Proc. 27th Int. Conf. on Artificial Intelligence and Statistics* (*Proc. Machine Learning Research* vol 238) ed S Dasgupta, S Mandt and Y Li (PMLR) pp 3502–10 (available at: https://proceedings.mlr.press/v238/rashidi24a.html)

Rasmussen C E and Williams C K I 2006 Gaussian processes for machine learning *Adaptive Computation and Machine Learning* (MIT Press)

Rumelhart D E, Hinton G E and Williams R J 1986 *Learning Internal Representations by Error Propagation* (MIT Press) pp 318–62

Thompson W R 1933 On the likelihood that one unknown probability exceeds another in view of the evidence of two samples *Biometrika* **25** 285

Tripp A, Daxberger E and Miguel Hernández-Lobato J 2020 Sample-efficient optimization in the latent space of deep generative models via weighted retraining *34th Conf. on Neural Information Processing Systems*

Wang Z, Gehring C, Kohli P and Jegelka S 2018 Batched large-scale bayesian optimization in high-dimensional spaces *21st Int. Conf. on Artificial Intelligence and Statistics*

Wang Z, Hutter F, Zoghi M, Matheson D and de Freitas N 2016 Bayesian optimization in a billion dimensions via random embeddings *J. Artif. Intell. Res.* **55** 361–87

Xing W W, Triantafyllidis V, Shah A A, Nair P B and Zabaras N 2016 Manifold learning for the emulation of spatial fields from computational models *J. Comput. Phys.* **326** 666–90

Xing W, Shah A A and Nair P B 2015 Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on Isomap *Proc. R. Soc. A* **471** 20140697

Zhe S, Xing W and Kirby R M 2019 Scalable high-order Gaussian process regression *Proc. 22nd Int. Conf. on Artificial Intelligence and Statistics* (*Proc. Machine Learning Research*) vol 89, ed K Chaudhuri and M Sugiyama (PMLR) pp 2611–20