New Bayesian optimisation framework for robust multi-objective design: decoupling performance and uncertainty

Hao Zhou, Zhongzheng Wang, Didier Lucor, Emilie Sauret

New Bayesian optimisation framework for robust multi–objective design: decoupling performance and uncertainty

Hao Zhou[a], Zhongzheng Wang[a], Didier Lucor[b,*], Emilie Sauret[a,**]

[a] *School of Mechanical, Medical and Process Engineering, Faculty of Engineering, Queensland University of Technology, AU*
[b] *Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91405 Orsay, France*

Abstract

Robust optimisation is increasingly critical in engineering design due to the growing emphasis on robustness under uncertainty. However, existing methods often neglect explicit control of performance stability, particularly regarding the variance of objectives. To address this limitation, this work introduces a Variance–constrained Robust Bayesian Optimisation framework to efficiently perform multi–objective optimisation considering input uncertainty. A robust Gaussian process is employed to quantify input uncertainty by providing both the expected objective value and its associated uncertainty. These two aspects are then integrated using a variance–penalised scalarisation method, which leverages user–defined parameters to flexibly balance mean performance and variability, thereby accommodating different levels of stability tolerance. During optimisation, this scalarisation steers the acquisition function toward regions consistent with the desired stability preference; at reporting time, surrogate estimates are used to discard any designs that violate the uncertainty threshold, and the Pareto set is computed from the remaining stability–feasible points. The novelty lies in enabling a flexible, context–dependent design selection, while preserving the data–efficiency advantages of Bayesian Optimisation. The framework has been thoroughly tested on a series of synthetic benchmark problems, including a higher–dimensional (6D) case, demonstrating its effectiveness in handling varying uncertainty tolerances and substantiating its strong potential for real–world applications.

Keywords: Uncertainty Tolerance, Multi–objective Optimisation, Robust Optimisation, Stability–feasible Pareto fronts

*corresponding: didier.lucor@cnrs.fr
**corresponding: emilie.sauret@qut.edu.au

Nomenclature

Notation	Description
	Multi–objective optimisation
$\mathbf{f}$	Vector of true black–box objective functions
f_i	i–th objective function
N	Number of objectives
x	Scalar design variable
M	Dimension of design variables
$\boldsymbol{x}$	Vector of design variables (decision point)
$\mathcal{X}$	Feasible design space
$\mathcal{X}^*$	Pareto–optimal set
$\mathbb{F}$	Objective value space
$\mathcal{P}^*$	Pareto front in objective space
	Bayesian optimisation
D^n	Dataset of n evaluated design points and their observations
$^{(i)}$	Superscript denoting the i–th data point in the dataset, e.g., $\boldsymbol{x}^{(i)}$, $f^{(i)}$
f^{GP}	Gaussian process surrogate for objective f
$m(\cdot)$	Predictive mean function of the GP
$\sigma(\cdot)$	Predictive standard deviation (uncertainty) from the GP
$\hat{f}$	Predictive mean of the surrogate f^{GP}
$a(\cdot)$	Acquisition function
	Robust Bayesian optimisation
$\boldsymbol{\xi}$	Input perturbation/noise added to $\boldsymbol{x}$
$p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$	Probability distribution over input perturbations
$\mathbf{x}$	Perturbed input: $\boldsymbol{x} + \boldsymbol{\xi}$
J_i	Robust objective: expectation over $f_i(\boldsymbol{x} + \boldsymbol{\xi})$
$\mathbf{J}$	Vector of robust objectives
J_i^{GP}	GP surrogate of robust objective J_i
f^{RGP}	Robust GP surrogate model (same as J^{GP})
V_i	Variance (robust uncertainty) of $f_i(\boldsymbol{x} + \boldsymbol{\xi})$
$\mathbf{V}$	Vector of variance
$\mathbf{p_V}$	Threshold for uncertainty in each objective
$\mathbf{y}_i$	Observation of $f_i(\boldsymbol{x} + \boldsymbol{\xi})$
$\mathcal{X}_{cand}$	Set of candidate points to be evaluated
S	Monte Carlo sample size for uncertainty estimation
z/θ	Reference point in scalarisation (e.g., current best)
$\mathcal{H}$	Augmented robust objective
$\mathbf{H}$	Vector of augmented robust objectives

1. Introduction

In modern engineering practice, design problems are becoming increasingly complex and critical, particularly as performance demands and safety standards rise across industries. These problems often involve multiple conflicting objectives that must be optimised simultaneously. For instance, in battery thermal management, heat–exchanger design must balance peak temperature, heat removal, and pressure drop (hydraulic losses). It must do so within tight packaging and pumping–power limits, while keeping complexity and cost manageable.

Significant research has been conducted on Multi–Objective Optimisation (MOO) as well as optimisation under constraints [1, 2, 3]. To address such challenges, the field of MOO has developed extensively, offering methods to identify sets of trade–off solutions known as Pareto fronts. Conventional MOO algorithms, such as the Non–Dominated Sorting Genetic Algorithm II (NSGA–II) [4] and Strength Pareto Evolutionary Algorithm [5], have been widely applied in engineering design due to their ability to explore complex, high–dimensional design spaces and to approximate the true Pareto–optimal set.

However, conventional MOO methods typically assume deterministic conditions and do not explicitly account for input uncertainty, which limits their robustness when applied to real–world systems affected by noise and variability [6, 7]. For example, small geometrical variations in the surface feature of the interface materials in electronics applications (e.g., 1 micrometer increase in surface roughness) can lead to drastic change in the thermal resistance by a factor of 2 [8]. Therefore, addressing optimisation under input uncertainty has received increasing attention. Nevertheless, although robustness under uncertainty has been extensively explored in single–objective optimisation, its extension to the multi–objective setting has received comparatively less attention. In the broader optimisation literature, input uncertainty is often classified into robust formulations (distribution–free, set–based uncertainty) and stochastic formulations (distribution–based uncertainty) [9, 10]. In line with more recent work [11] that treats these categories under a unified stochastic viewpoint, we use the term robust in this paper in a distribution–based sense: the input uncertainty is described by a probability distribution, and robustness is measured through functionals of the resulting output distribution (in our case, the mean and the variance).

Two well–established paradigms are Robust–Design Optimisation (RDO) [12, 13] and Reliability–based Design Optimisation (RBDO) [14, 15, 16]. RDO expresses objectives (and possibly constraints) in terms of statistics such as expectation and variance, seeking designs with low sensitivity to input variation. RBDO focuses on the probability that constraints are respected; smaller admissible violation probabilities imply higher reliability. When surrogates are used—data–driven emulators of costly simulators that map inputs to outputs at a fraction of the evaluation cost—they

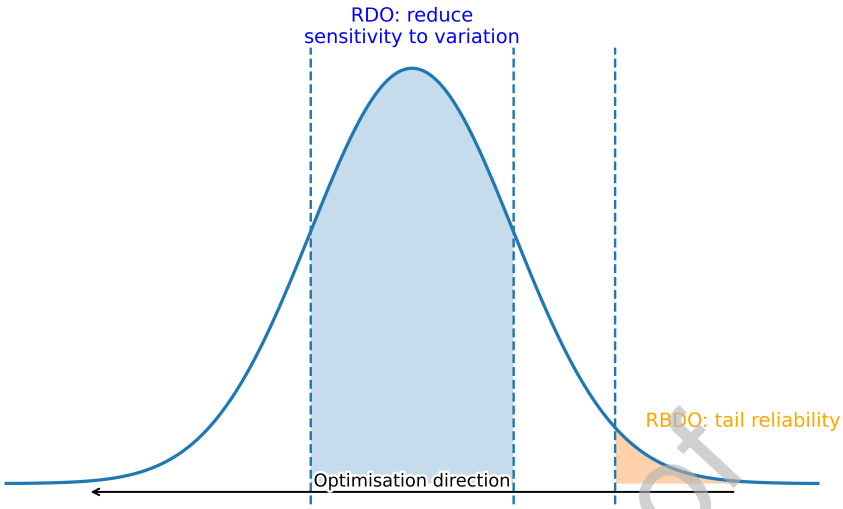

Figure 1: Illustration of Robust design optimisation (RDO) and Reliability–based Design Optimisation (RBDO).

replace repeated high–fidelity runs and, in the case of Gaussian–process models, provide predictive uncertainty to guide exploration. Fig. 1 highlights the practical distinction: we use the term RDO for formulations that optimise performance measures based on response statistics such as the mean and, when appropriate, its spread (e.g., variance) whereas RBDO explicitly limits the tail probability of violating a specified limit. In practice, multi–objective workflows often replace each nominal objective by a robust statistic (e.g., mean or mean–variance) and, when needed, monitor a simple reliability indicator (e.g., a probability of violation), without committing to a single constrained formulation. Reliability–based Robust Design Optimisation (RBRDO) [17, 18] combines these views by using expectations and variances to describe performance while bounding failure probabilities for reliability.

A common way to adapt traditional MOO to uncertainty is to optimise statistics, such as mean and maximum, in place of the nominal objectives. However, estimating these statistics via repeated uncertainty sampling [19] (e.g., Exhausted NSGA–II (ENSGA–II) [20]) is prohibitively costly, especially for expensive high–fidelity black–box evaluations, motivating sample–efficient alternatives.

To address this challenge, Bayesian Optimisation (BO) [21] has emerged as a powerful framework for sample–efficient optimisation of expensive black–box functions. BO constructs a probabilistic surrogate model, typically a Gaussian Process (GP) [22], that not only approximates the objective function but also quantifies predictive uncertainty. Consequently, many studies have combined BO and Robust Optimisation (RO) into Robust Bayesian Optimisation (RBO) [23, 24, 25]. These

methods incorporate input uncertainty directly into the BO loop by applying repeated sampling on GP surrogate rather than the original objective, which significantly reduces computational cost. This leads to a Robust Gaussian Process (RGP) that models the distribution of the perturbed response and provides uncertainty–aware statistics (e.g., expectation or mean–variance). The RGP thus replaces the nominal GP in guiding the acquisition function and downstream optimisation. Kudva et al. [26] proposed flexibility–analysis algorithm that accommodated uncertain parameters and recourse variables across multiple constraints. Li et al. [27] utilised a Bayes' risk in RGP, achieving robust uncertainty quantification to account for the inherent stochasticity of additive manufacturing processes across different geometries and process parameters. Oliveira et al. [28] proposed handling both terms via an upper confidence bound strategy in RBO, offering theoretical guarantees for noisy–input settings and demonstrating competitive performance against other BO approaches dealing with input uncertainty. Inatsu et al. [29] proposed a GP–based MOBO that constructs high–probability bounding boxes for general risk measures by mapping a high–probability credible region of the response, thereby screening out extreme (unsafe) scenarios under input uncertainty.

Despite significant efforts have been made in the development of RBO, existing approaches still share a critical limitation: they focus on either the expectation (Bayes' risk) or a worst–case scenario proxy (e.g., worst–case, Value–at–Risk (VaR)). Expectation–based methods aim to optimise average performance under uncertainty, which might inadvertently tolerate excessively large performance fluctuations around the mean [30]. Conversely, worst–case approaches concentrate on ensuring robust performance under extreme scenarios, potentially resulting in overly conservative solutions that compromise average performance [31]. Crucially, neither approach explicitly controls the variance of the objective induced by input uncertainty [32, 33]. This overlooked variability can pose severe operational risks and ignore the importance of stability, especially in quality–sensitive applications, as significant fluctuations in system response might frequently violate critical design constraints or safety thresholds.

Inspired by the exploration–exploitation trade–off inherent in traditional acquisition functions such as Expected Improvement (EI), we propose a variance–penalised scalarisation method that explicitly balances the expected performance and the uncertainty in objectives caused by input perturbations. Through this approach, the mean and standard deviation of each objective are combined into a single scalar quantity, thereby enabling direct application of conventional acquisition strategies. Recognising that the predictive variance of an RGP mixes two qualitatively different sources—epistemic (model) uncertainty from limited data and aleatory (objective) uncertainty induced by input perturbations—we proposed a bi–loop Variance–constrained Robust Bayesian Optimisation (VRBO) architecture with two parallel

branches updated within each iteration. The mixing occurs because a random input is propagated through a probabilistic model, so the resulting spread reflects both model ignorance and inherent input–driven variability and appears as a single "variance" signal to the optimiser. One branch refines the nominal surrogate to reduce epistemic error; the other targets aleatory variability using the robust surrogate and the variance–penalised scalarisation, steering the search toward stability–feasible regions. Treating these components separately is crucial: epistemic uncertainty should shrink with additional data, whereas aleatory uncertainty is irreducible and ought to shape "stability" preferences; if conflated, the optimiser may mistake undertrained areas for genuinely low–variance regions (or vice versa). This parallel design avoids that confounding and guides the optimisation toward robust solutions with predictable performance under input variability. Stability enters in two complementary forms: a soft influence within the acquisition, controlled by a tunable parameter α that modulates the emphasis placed on uncertainty, and after optimisation, we filter the final Pareto front by a hard stability constraint, discarding any points that violate the user–specified tolerance $\mathbf{p_V}$.

By employing this framework, optimisation results can be controlled based on the specified tolerance to objective uncertainty. In this study, four benchmark cases are utilised to validate the proposed VRBO method. These cases are specifically selected according to categories defined by the relationship between the Pareto fronts obtained by non–robust and robust optimisation. Through the application of varying uncertainty tolerance levels within VRBO, the benchmark cases exhibit distinctly different Pareto front behaviours, demonstrating the method's capability to generate flexible trade–offs between performance stability and optimality.

In summary, this study makes the following main contributions:

1. A variance–penalised scalarisation is introduced that explicitly integrates the input–uncertainty–induced mean and variance into the search, providing a user–tunable trade–off between expected performance and variability. This produces robust Pareto fronts tailored to specified uncertainty tolerances.
2. A bi–loop optimisation strategy VRBO is proposed to effectively balances global surrogate accuracy with local robustness requirements, ensuring stable optimisation performance under noisy inputs.
3. The effectiveness and versatility of the proposed VRBO method are demonstrated through an empirical study on five benchmarks. These results confirm that our approach consistently produces adaptive or competitive Pareto fronts compared to conventional optimisation methods.

The rest of this paper is organised as follows. In Section 2, we introduce the background knowledge about optimisation. Section 3 presents the structure of VRBO. The benchmark cases are discussed in Section 4, 5 and 6. Finally, we conclude and summarise our findings in Section 7.

2. Preliminaries

2.1. Multi–objective Optimisation

Many engineering problems require simultaneous optimisation of multiple conflicting objectives (with or without constraints). Given a vector–valued objective function $\mathbf{f} = (f_1, f_2, \ldots, f_N)$, the goal of MOO is to find solutions that balance trade–offs between objectives [34].

Considering a MOO of minimizing each objective function, a candidate design vector $\boldsymbol{x}^* \in \mathcal{X} \subseteq \mathbb{R}^M$ is preferable to another $\boldsymbol{x}$ one if:

$$f_i(\boldsymbol{x}^*) \leq f_i(\boldsymbol{x}), \forall i \in \{1, \ldots, N\}, \text{ and } \exists i \in \{1, \ldots, N\} : \; f_i(\boldsymbol{x}^*) < f_i(\boldsymbol{x}). \tag{1}$$

In this case, these relations introduce the concept of dominance, denoted here as $\boldsymbol{x} \prec \boldsymbol{x}^*$ (in the case of a minimisation problem) to indicate that $\mathbf{f}(\boldsymbol{x}) \prec \mathbf{f}(\boldsymbol{x}^*)$. In MOO, a candidate $\boldsymbol{x}^*$ is defined as Pareto optimal if: $\nexists \; \boldsymbol{x} \in \mathcal{X}$ such that $\mathbf{f}(\boldsymbol{x}^*) \prec \mathbf{f}(\boldsymbol{x})$.

The set of all Pareto optimal solutions is called the Pareto set $\mathcal{X}^*$:

$$\mathcal{X}^* := \{\boldsymbol{x}^* \in \mathcal{X} \mid \nexists \mathbf{f}(\boldsymbol{x}) \in \mathbb{F}_{\mathbf{f}} \text{ s.t. } \mathbf{f}(\boldsymbol{x}^*) \prec \mathbf{f}(\boldsymbol{x})\}, \quad \text{where } \mathbb{F}_{\mathbf{f}} := \{\mathbf{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\}, \tag{2}$$

and its image in the objective space is called the Pareto front $\mathcal{P}^*$ [35], representing optimal trade–offs where improving one objective inevitably worsens another. The aim is thus to accurately approximate the Pareto front with as few expensive queries to $\mathbf{f}$ as possible. In this context, it makes sense to turn to efficient optimisation techniques, for instance relying on objective functions model approximations.

2.2. Bayesian Optimisation framework

BO [21] is a sample–efficient framework for optimising expensive black–box functions. It constructs a probabilistic surrogate – commonly an adaptive GP [36, 37] for which we briefly introduce the key ingredients here – to model the objective function with a continuous representation of $f(\boldsymbol{x})$ from a discrete dataset, and uses an acquisition function to guide the search toward promising regions of the input space in order to refine and improve the surrogate.

Given a dataset $D^n = \{(\boldsymbol{x}^{(i)}, f^{(i)})\}_{i=1}^n$, a GP defines a prior over functions specified by a mean function $m(\boldsymbol{x})$ and a covariance kernel $k(\boldsymbol{x}, \boldsymbol{x}')$, representing the independence of function values at different locations. Conditioned on the data, the GP posterior at any test input $\boldsymbol{x}_*$ is:

$$f^{\mathrm{GP}}(\boldsymbol{x}_*)|D^n \sim \mathcal{N}(m_{f^{\mathrm{GP}}}(\boldsymbol{x}_*), \sigma^2_{f^{\mathrm{GP}}}(\boldsymbol{x}_*)), \tag{3}$$

where the posterior mean and the variance depend on the choice of the kernel function and the observed data. On the diagonal, $k(\boldsymbol{x}, \boldsymbol{x})$ gives the prior marginal variance

at $\boldsymbol{x}$; after conditioning, the posterior predictive variance $\sigma^2_{f^{\mathrm{GP}}}(\boldsymbol{x}_* \mid D^n)$ quantifies local uncertainty (and typically shrinks near observed points). And the surrogate prediction is the GP posterior mean,

$$\hat{f}(\boldsymbol{x}_*) = m_{f^{\mathrm{GP}}}(\boldsymbol{x}_* \mid D^n). \tag{4}$$

The acquisition function $a(\boldsymbol{x})$ quantifies the utility of sampling a new point $\boldsymbol{x}$, balancing exploration (high variance) and exploitation (low predicted mean). The choice of this acquisition function is crucial, as one easily understands that it is better to refine the surrogate only if necessary, and possibly in the close neighbourhood of the optimal design parameters region. Common choices include EI, Probability of Improvement, and Upper Confidence Bound [38, 39]. At each iteration k, the next sample is chosen as:

$$\boldsymbol{x}^{(k+1)} = \arg\max_{\boldsymbol{x}} a(f^{\mathrm{GP}}(\boldsymbol{x})). \tag{5}$$

In Multi–Objective Bayesian Optimisation (MOBO) [40, 41], a simple choice is to rely on separate GP surrogates, one for each objective function f_i, and acquisition functions such as Expected Hypervolume Improvement (EHVI), are used to suggest candidates that efficiently improve the Pareto front.

2.3. Accounting for uncertainties

In many real–world systems, the exact values of actual input parameters are not precisely known. This uncertainty may arise from factors such as manufacturing and assembly tolerances, or inaccuracies in material properties. It is therefore essential to incorporate these parameter uncertainties into the optimisation process to ensure the resulting design is robust.

Such uncertainties introduce non–determinism into the system's response and constraints, making the optimisation process considerably more challenging. To address this, one must first define a deterministic quantity of interest to optimize. This typically involves statistical measures – such as moments (e.g., mean or variance), probabilities, or specific events like worst–case scenarios – related to the system's performance.

Another important consideration is identifying which parameters are affected by uncertainty. In some cases, uncertainties pertain to environmental or operational conditions that are independent of the design variables $\boldsymbol{x}$. However, in the scenario considered in this study, the uncertainties directly impact the design variables themselves. As a result, the actual system input takes the form $\boldsymbol{x} + \boldsymbol{\xi}$, where $\boldsymbol{\xi}$ is a known zero–mean random vector representing manufacturing variabilities. This deviation from the nominal design must be accounted for in order to achieve a robust and reliable solution.

To account for such perturbations, we consider not only the nominal objective value $\mathbf{f}(\boldsymbol{x})$, but its statistical response under uncertainty–specifically, the mean

(Bayes' risk) and variance–leading to the formulation of robust and risk–aware optimisation strategies [42].

3. Variance–constrained Robust Bayesian Optimisation

3.1. Optimisation under uncertainties formulation

We consider a deterministic design $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^M$ and input perturbations $\boldsymbol{\xi} \sim p_{\boldsymbol{\xi}}$. Each objective is evaluated as $f_i(\boldsymbol{x} + \boldsymbol{\xi})$. To summarise uncertainty effects we use:

$$\begin{aligned} J_i(\boldsymbol{x}) &= \mathbb{E}_{\boldsymbol{\xi}\sim p_{\boldsymbol{\xi}}}[f_i(\boldsymbol{x}+\boldsymbol{\xi})]\,, && \text{(robust / Bayes' risk)} \\ V_i(\boldsymbol{x}) &= \mathbb{V}_{\boldsymbol{\xi}\sim p_{\boldsymbol{\xi}}}[f_i(\boldsymbol{x}+\boldsymbol{\xi})]\,, && \text{(stability / uncertainty)} \end{aligned} \tag{6}$$

In our setting, "stability" is operationalised as compliance with a user–specified tolerance on input–induced variability, i.e. $V_i(\boldsymbol{x}) \leq p_{V,i}$. We therefore search in the $\mathbf{J}$–space while enforcing the hard variance bound at decision time.

Robust multi--objective design optimisation. Under the MOO background, we aim to find Pareto--optimal solutions that minimise the Bayes' risk vector $\mathbf{J}(\boldsymbol{x})$ while satisfying user–defined tolerance $\mathbf{p_V}$ on the variance vector $\mathbf{V}(\mathbf{x})$:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & \mathbf{J}(\boldsymbol{x}) = \Big(J_1(\boldsymbol{x}), \ldots, J_N(\boldsymbol{x})\Big), \\ \text{s.t.} \quad & \mathbf{V}(\boldsymbol{x}) = \Big(V_1(\boldsymbol{x}), \ldots, V_N(\boldsymbol{x})\Big) \leq \mathbf{p_V}, \end{aligned} \tag{7}$$

where $\mathcal{X} \subseteq \mathbb{R}^M$ is the feasible domain, and $\mathbf{p_V} \in \mathbb{R}^N_+$ is a user–defined vector of variance thresholds. Note that any maximisation objective can be recast as a minimisation by negating it.

To solve this stochastic optimisation efficiently under limited data budgets, we require surrogate models that can account for the propagation of input uncertainty.

3.2. Robust Gaussian Process surrogate

In order to model each objective f_i under input uncertainty given a dataset, we construct a RGP surrogate. This model approximates the Bayes' risk $J_i(\boldsymbol{x}) = \mathbb{E}_{p_\xi}[f_i(\boldsymbol{x}+\boldsymbol{\xi})]$ while bounding its uncertainty using a modified GP framework.

Kernel expectation formulation. Let $k_{f_i}(\boldsymbol{x}, \boldsymbol{x}')$ denotes the kernel of the nominal GP surrogate f_i^{GP}, and $K_i = K_{f_i} + \sigma_\varepsilon^2 I$ the posterior covariance matrix based on the training outputs $\mathbf{y}_i$. The robust surrogate $J_i^{\mathrm{GP}} \sim \mathcal{GP}(m_{J_i}, \mathrm{Cov}_{J_i})$ is then given by [22]:

$$m_{J_i}(\boldsymbol{x}) = k_{J_i,f_i}(\boldsymbol{x})^\top K_i^{-1} \mathbf{y}_i, \tag{8}$$

$$\mathrm{Cov}_{J_i}(\boldsymbol{x}, \boldsymbol{x}') = k_{J_i}(\boldsymbol{x}, \boldsymbol{x}') - k_{J_i,f_i}(\boldsymbol{x})^\top K_i^{-1} k_{J_i,f_i}(\boldsymbol{x}'), \tag{9}$$

where the kernel expectations are defined as:

$$k_{J_i,f_i}(\boldsymbol{x}) = \int k_{f_i}(\boldsymbol{x}+\boldsymbol{\xi})\, p(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}, \tag{10}$$

$$k_{J_i}(\boldsymbol{x},\boldsymbol{x}') = \iint k_{f_i}(\boldsymbol{x}+\boldsymbol{\xi},\boldsymbol{x}'+\boldsymbol{\xi}')\, p(\boldsymbol{\xi})p(\boldsymbol{\xi}')\, \mathrm{d}\boldsymbol{\xi}\, \mathrm{d}\boldsymbol{\xi}'. \tag{11}$$

For squared–exponential kernels and Gaussian perturbations, the integrals defining the robust objective can be computed analytically. Otherwise, we approximate them using Monte Carlo integration [24]. Specifically, for each candidate input $\boldsymbol{x}$, a set of perturbations $\{\boldsymbol{\xi}^{(s)}\}_{s=1}^{S}$ is drawn from the input noise distribution $p_{\boldsymbol{\xi}}$, and the corresponding perturbed inputs $\boldsymbol{x}+\boldsymbol{\xi}^{(s)}$ are evaluated using the nominal GP surrogate f^{GP}. The empirical mean and variance of these surrogate outputs are then used to approximate the robust mean $m_J(\boldsymbol{x})$ and variance $\mathrm{Cov}_J(\boldsymbol{x})$. These statistics are subsequently used to construct the robust surrogate J^{GP} with heteroscedastic noise (the noise level is various across different input).

Practical construction. The following procedure builds a RGP surrogate by re–using the nominal GP and propagating noise through the model:

Algorithm 1: Constructing a Robust Gaussian Process under input uncertainty (here one–dimensional for mono–objective case)

Input : Dataset $D^n = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$;
Perturbation distribution function $p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$;
Candidate set $\mathcal{X}_{\mathrm{cand}}$;
Sample count for Monte Carlo S

Output: Robust surrogate J^{GP}

Train nominal GP surrogate f^{GP} on D^n;
foreach $\boldsymbol{x} \in \mathcal{X}_{cand}$ **do**
  Draw S samples $\{\boldsymbol{\xi}^{(s)}\}_{s=1}^{S} \sim p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$;
  Predict with surrogate $y^{(s)} \leftarrow f^{\mathrm{GP}}(\boldsymbol{x}+\boldsymbol{\xi}^{(s)})$;
  Estimate:;
    $m_J(\boldsymbol{x}) \leftarrow \frac{1}{S}\sum_{s=1}^{S} y^{(s)}$;
    $\mathrm{Cov}_J(\boldsymbol{x}) \leftarrow \mathbb{V}\left[y^{(1)},\ldots,y^{(S)}\right]$;
Fit J^{GP} on $\{(\boldsymbol{x}, m_J(\boldsymbol{x}))\}$ with heteroscedastic noise $\mathrm{Cov}_J(\boldsymbol{x})$;

In all experiments, the robust statistics are approximated by Monte Carlo with a fixed sample size of $S = 200$ per design point [43], while the candidate set $\mathcal{X}_{\mathrm{cand}}$ is generated by Sobol sequences with a dimension–dependent size. The resulting surrogate captures both the mean Bayes' risk $m_{J_n}(\boldsymbol{x})$ and its epistemic uncertainty $\sigma^2_{J_n}(\boldsymbol{x}) = \mathrm{Cov}_{J_n}(\boldsymbol{x},\boldsymbol{x})$, enabling the construction of variance–penalised acquisition functions discussed in Section 3.4.

3.3. Existing acquisition strategies for (robust) MOBO

Before introducing our variance–penalised scalarisation, we briefly situate it within existing acquisition strategies for multi–objective and RBO. In standard MOBO, a common approach is to build one GP per objective and either (i) scalarise the vector of objectives into a single surrogate (e.g. ParEGO, which combines random weights with EI on a Tchebycheff–type scalarisation), or (ii) use Pareto–based criteria such as EHVI and related hypervolume–based acquisitions, which directly target expansion of the dominated region in the objective space.

When input uncertainty is present, many "robust MOBO" methods follow a similar pattern: they first define robust statistics of each objective—typically expectations, mean–variance combinations, or risk measures–and then apply standard single– or multi–objective acquisitions (e.g. EI/Upper Confidence Boundary (UCB) or EHVI) to these transformed objectives. In other words, robustness enters through the choice of response functional, while the acquisition itself remains conventional.

Our framework follows this general pattern but targets a different robustness notion. In this work, We also adopt EHVI as the underlying MOBO acquisition, yet we construct a dedicated scalar score $H(\mu, \sigma)$ that combines the robust mean and its input–induced variability in a tunable way, and we complement this soft, acquisition–level trade–off with a hard variance filter at selection time. The next subsection details this variance–penalised scalarisation.

3.4. Variance-penalised Scalarisation Algorithm

After constructing J_i^{GP}, we build a soft, risk–aware score that balances the robust mean $m_{J_n}(\boldsymbol{x})$ and the input-induced variability measured by $\sigma^2_{J_n}(\boldsymbol{x})$. The Conditional Value–at–Risk (CVaR) term acts as a tail–severity penalty around a reference level, while the $\ln(1+\sigma_i)$ term softly discourages large variance. This scalarisation guides the search; final stability is enforced by the hard variance bound $V_i(\boldsymbol{x}) \le p_{V,i}$ when reporting the Pareto set.

Threshold–CVaR. Let the best observed value in the dataset D^t for a minimisation problem be $\theta_i = \min_{y_i \in \mathcal{D}^t} y_i$. For any candidate $\boldsymbol{x}$ with predictive mean $\mu_i(\boldsymbol{x}) = m_{J_i}(\boldsymbol{x})$ and standard deviation $\sigma_i(\boldsymbol{x}) = \sqrt{\mathbb{V}[f_i(\boldsymbol{x}+\boldsymbol{\xi})]}$, we consider the conditional expectation of $f_i(\boldsymbol{x}+\boldsymbol{\xi})$ above the target level θ_i:

$$\mathrm{CVaR}_{\theta_i}(\boldsymbol{x}) \;=\; \mathbb{E}\Big[f_i(\boldsymbol{x}+\boldsymbol{\xi}) \;\Big|\; f_i(\boldsymbol{x}+\boldsymbol{\xi}) \ge \theta_i\Big].$$

Under the usual Gaussian–process assumption $f_i(\boldsymbol{x}+\boldsymbol{\xi}) \sim \mathcal{N}(\mu_i(\boldsymbol{x}), \sigma_i^2(\boldsymbol{x}))$, this quantity has the closed form

$$\mathrm{CVaR}_{\theta_i}(\boldsymbol{x}) \;=\; \mu_i(\boldsymbol{x}) + \sigma_i(\boldsymbol{x})\,\frac{\phi\big(z_i(\boldsymbol{x})\big)}{1-\Phi\big(z_i(\boldsymbol{x})\big)}, \qquad z_i(\boldsymbol{x}) = \frac{\theta_i - \mu_i(\boldsymbol{x})}{\sigma_i(\boldsymbol{x})}, \tag{12}$$

where ϕ and Φ denote the standard normal density and cumulative distribution function, respectively. In other words, $z_i(\boldsymbol{x})$ is the standardised distance between the target θ_i and the predictive mean.

This quantity is closely related to the classical CVaR [32, 44]. In the standard definition, CVaR_p at level $p \in (0,1)$ is the expected value of the loss conditional on the loss exceeding its p-quantile. If, instead of fixing θ_i to the current best observed value, we chose θ_i so that $\mathbb{P}\big(f_i(\boldsymbol{x}+\boldsymbol{\xi}) \leq \theta_i\big) = p$, then $\mathrm{CVaR}_{\theta_i}(\boldsymbol{x})$ would coincide with the usual CVaR_p. In this work, however, we fix θ_i as the current best observed value and use $\mathrm{CVaR}_{\theta_i}(\boldsymbol{x})$ as a threshold–based tail risk measure: candidates for which the upper tail above θ_i has smaller conditional mean are preferred, as they are less likely to produce severely poor outcomes relative to the current best performance.

Variance–penalised scalarisation. We then propose a modification and construct:

$$\mathcal{H}_i(\boldsymbol{x}) \;=\; \mathrm{CVaR}_{\theta_i}(\boldsymbol{x}) \;+\; \alpha \,\ln\Big[1 + \sigma_i(\boldsymbol{x})\Big], \tag{13}$$

where $\alpha \geq 0$ is a trade-off parameter that modulates the trade–off between tail–risk aversion and exploration. Data samples for the next evaluation are chosen via $\boldsymbol{x}^{(t+1)} = \arg\min_{\boldsymbol{x}\in\mathcal{X}_{cand}} \mathcal{H}_i(\boldsymbol{x})$.

Interpretation. In order to summarise, we can say that the first term in Eq. (13) captures a target–based exploitation strategy. Rather than simply being risk–averse, it evaluates the distribution relative to a specific threshold; minimising this term penalises any region whose performance is expected to be worse than the current best. This ensures the acquisition function actively avoids areas where the objective is likely to underperform compared to the best solution found so far. The second log–variance term serves as a complementary exploration incentive that grows with uncertainty. It acts as an exploration bonus: points with higher uncertainty get a higher value, but not excessively so, due to the logarithm. Finally, the $+1$ term is included to avoid singularities when the standard deviation tends to zero ($\sigma \to 0 \Rightarrow \ln(1+\sigma) \approx \sigma$). This yields a search strategy that actively avoids underperformance relative to the current best while remaining explorative, making it suitable for robust optimisation under input uncertainty as we will see in the applications section.

3.5. Effect of α

To visualise how the variance–penalised objective $\mathcal{H}$ trades off mean performance and predictive uncertainty, we evaluate $\mathcal{H}$ on a grid $\mu \in [0,1]$, $\sigma \in [0, 0.5]$ with the CVaR reference fixed at $\theta = 0.2$. Fig. 2 shows maps for four settings of the trade–off parameter α. When $\alpha = 0$ (Fig. 2 (a)), $\mathcal{H}$ reduces to the CVaR term: isolines are mostly vertical over much of the domain (mean dominates), but they can flatten for very small μ and become diagonal where σ is large. With a small penalty ($\alpha = 0.1$, Fig. 2 (b)) the contours begin to bend, reflecting a mild preference for smaller σ. At

$\alpha = 1$ (Fig. 2 (c)) the contours tilt towards $\approx 45°$, indicating a balanced emphasis on reducing both tail loss and variance. For a strongly risk–averse choice ($\alpha = 10$, Fig. 2 (d)), the variance penalty dominates: contours become almost horizontal and designs with larger σ are penalised irrespective of μ. Overall, as α increases, the acquisition places more weight on low variance, so the search shifts from mean–seeking (small α) to variance–averse sampling that concentrates evaluations in low–σ regions.

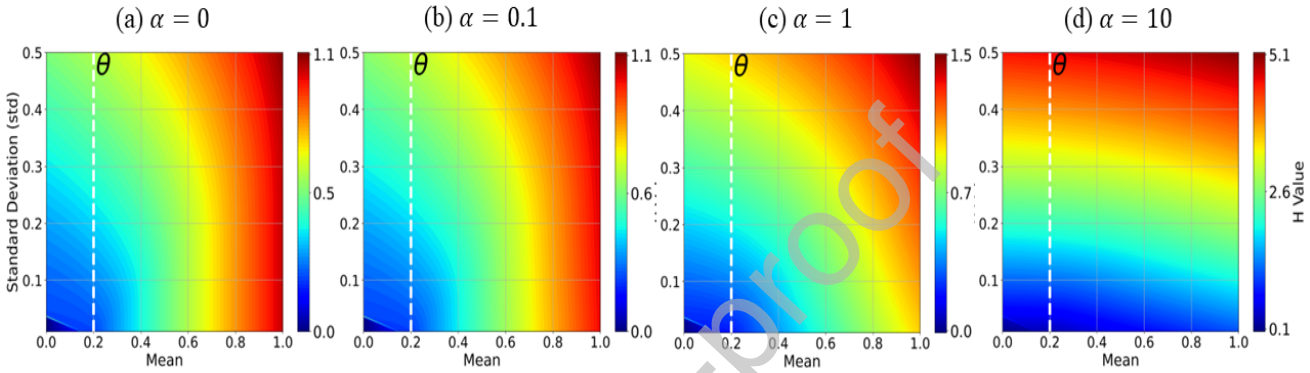

Figure 2: Contour maps of the scalar objective $\mathcal{H}(\mu, \sigma)$ for (a) $\alpha = 0.$, (b) $\alpha = 0.1$, (c) $\alpha = 1.0$, (d) $\alpha = 10$. The horizontal axis is the predictive mean μ, the vertical axis the standard deviation σ. Warmer colours indicate a higher (worse) objective value.

3.6. Variance–constrained Robust Bayesian Optimisation Framework

In a RGP, the predictive variance $\sigma^2_{J_i}(\mathbf{x})$ will be affected by two distinct components: the model uncertainty that reflects the surrogate's lack of data and the objective uncertainty that originates from the random input perturbations $\boldsymbol{\xi}$. An illustrative example is shown in Fig. 3. Panel (a) displays the nominal GP for $f(x)$: the blue curve is the posterior mean and the light–blue ribbon is the 95% epistemic band. Panel (b) shows the robust surrogate $J(x) = \mathbb{E}[f(x + \boldsymbol{\xi})]$ obtained by propagating input noise through the GP: the orange curve is the robust mean and the hatched ribbon is the 95% total band (epistemic $\oplus$ aleatory). This highlights that J carries both model and input-induced uncertainty.

The VRBO framework therefore employs two nested optimisation loops, each targeting one of these sources, as shown in Fig. 4.

MO loop – Surrogate-model refinement. The MO loop maintains nominal GP surrogates GP_{f_i} for every objective and selects new points that minimise model uncertainty. Typical criteria include maximising the GP posterior variance or using information–based acquisition functions; the goal is to densify data where the surrogate is least confident, thereby shrinking epistemic error.

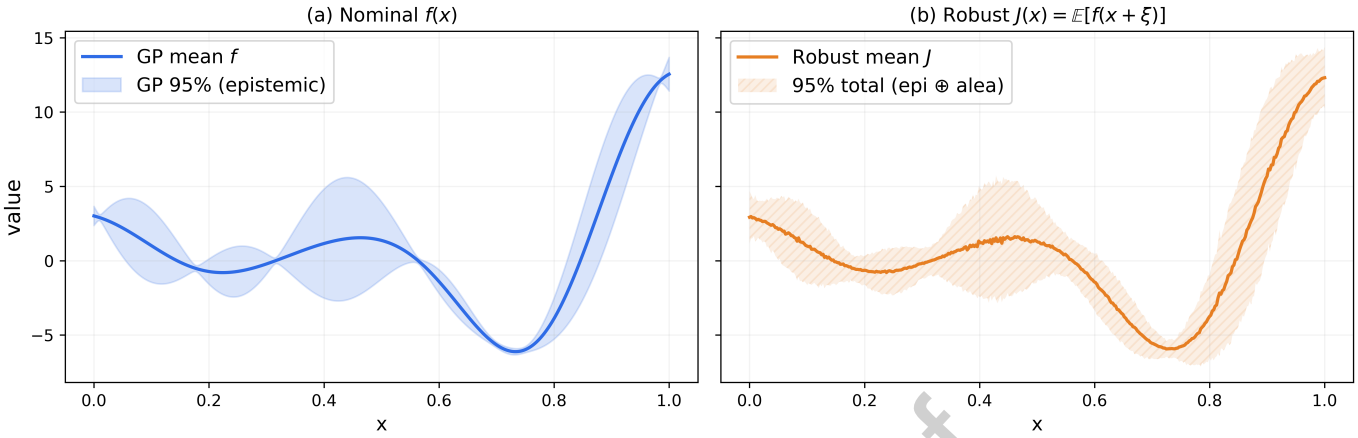

Figure 3: Uncertainty propagation from the nominal GP to the robust surrogate. (a) Nominal $f(x)$: GP mean (blue) with 95% epistemic band. (b) Robust $J(x) = \mathbb{E}[f(x+\boldsymbol{\xi})]$: mean (orange) with 95% total band (epistemic $\oplus$ aleatory) obtained by propagating input perturbations through the GP.

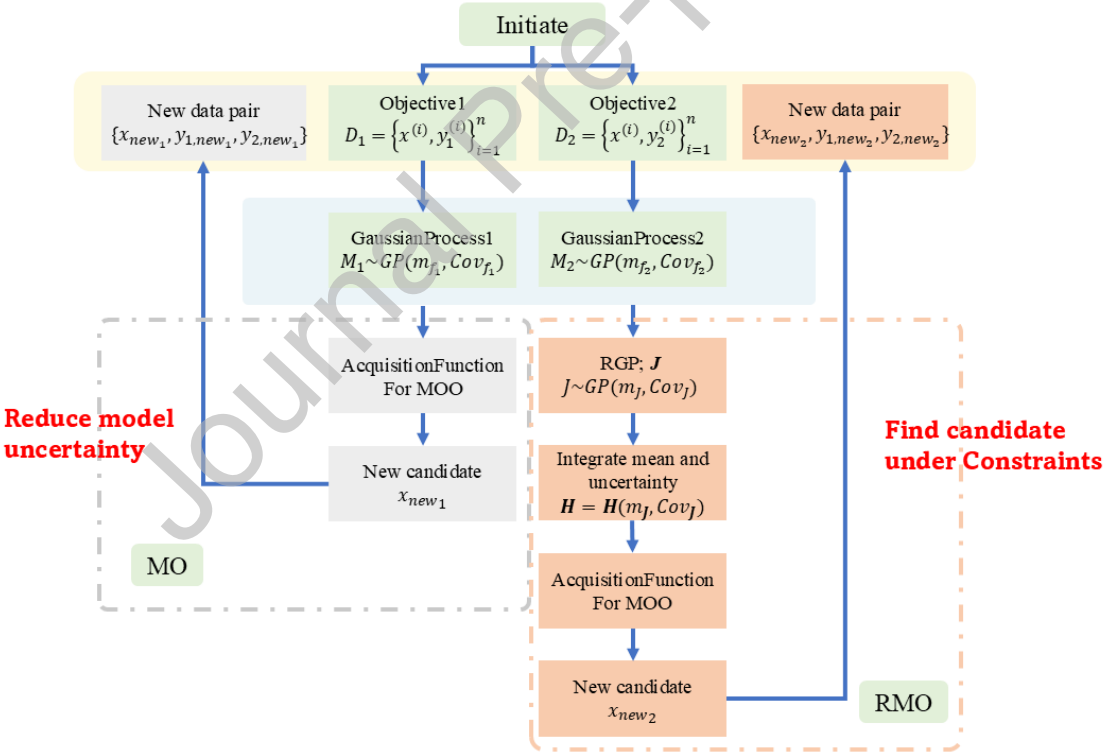

Figure 4: Bi–loop structure of the proposed VRBO framework. The MO loop (left) updates nominal GP models to reduce model uncertainty, while the RMO loop (right) uses a RGP surrogate and a variance–penalised scalarisation function H to find candidates under input uncertainty.

RMO loop – Robust candidate selection. The RMO loop builds RGP surrogates J_i^{GP} (Section 3.2) and employs the variance-penalised robust acquisition $\mathcal{H}_i(\boldsymbol{x})$ in Eq. (13). Optimisation of $\mathcal{H}_i$ focuses on controlling objective uncertainty introduced by input noise, producing designs whose Bayes–risk means and variances approximately satisfy the user–set tolerance $\mathbf{p_V}$.

For the multi–objective case, the robust scores are aggregated—-e.g. with EHVI, ParEGO, or an ε–constraint strategy—and the next candidate is

$$\boldsymbol{x}^{(t+1)} = \arg\min_{\boldsymbol{x}\in\mathcal{X}_{\mathrm{cand}}} a\left(\mathbf{H}(\boldsymbol{x})\right),$$

where $a(\cdot)$ denotes the chosen acquisition function for MOO, $\mathbf{H} = (\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_N)$. In this work, EHVI is adopted as the acquisition function for MOO, since EHVI directly maximises expected hypervolume improvement and tends to provide a more balanced coverage of the Pareto front.

Iterative update. After evaluating the black–box model at $\boldsymbol{x}^{(t+1)}$ and recording observation $\mathbf{y}^{(t+1)}$, the data set D^t is augmented and both loops repeat until a stopping criterion (maximum calls or Pareto convergence) is reached.

By decoupling model and objective uncertainties, this nested framework simultaneously (i) refines global surrogate fidelity (MO) and (ii) steers sampling toward designs that remain performant under input noise (RMO), achieving sample–efficient and uncertainty–aware optimisation. Overall, the algorithm for the entire optimisation process is presented below in Algorithm 2.

Algorithm 2: Variance–constrained Robust Bayesian Optimisation for MOO

Input : Initial dataset $D^t = \{(\boldsymbol{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{t}$;
Perturbation distribution $p(\boldsymbol{\xi})$;
Feasible domain $\mathcal{U}$, variance–tolerance $\mathbf{p_V}$;
Trade–off coefficient α, evaluation budget T;
MOO acquisition $a_{\text{MOO}}(\cdot)$, candidate pool $\mathcal{X}_{\text{cand}}$
Output: Final surrogates $\hat{\mathbf{f}}(\boldsymbol{x})$

Initialise nominal $\text{GP}_{\mathbf{f}} = (\text{GP}_{f_1}, \text{GP}_{f_2}, \cdots, \text{GP}_{f_n})$ with D^t;
while $t < T$ **do** `// single loop, dual branches`
  Retrain $\text{GP}_{\mathbf{f}}$ on D^t;
  ; `// MO loop`
  $\boldsymbol{x}_{\text{MO}} = \arg\max_{\boldsymbol{x} \in \mathcal{X}_{\text{cand}}} \left[\text{GP}_{\mathbf{f}}(\boldsymbol{x})\right]$;
  ; `// RMO loop`
  Construct robust surrogates $\text{GP}_{\mathbf{J}} \sim \mathcal{GP}(m_{\mathbf{J}}, \text{Cov}_{\mathbf{J}})$ via Alg. 1 using $\text{GP}_{\mathbf{f}}$ and $p(\boldsymbol{\xi})$;
  foreach $n = 1, \ldots, N$ **do**
    $\theta_n \leftarrow \min_{\boldsymbol{x} \in D^t} y_n(\boldsymbol{x})$;
  foreach $\boldsymbol{x} \in \mathcal{X}_{cand}$ **do**
    $\mathbf{H}(\boldsymbol{x}) = \text{CVaR}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \alpha \ln(1 + \sigma_{\mathbf{J}}(\boldsymbol{x}))$;
  $\boldsymbol{x}_{\text{RMO}} = \arg\max_{\boldsymbol{x} \in \mathcal{X}_{\text{cand}}} a_{\text{MOO}}\big(\mathbf{H}(\boldsymbol{x})\big)$;
  Evaluate black box at $\{\boldsymbol{x}_{\text{MO}}, \boldsymbol{x}_{\text{RMO}}\}$ and append results to D^t to obtain D^{t+2};
  $t \leftarrow t + 2$
Refit surrogate model $\hat{\mathbf{f}}(\boldsymbol{x})$ based on final dataset D^t $\mathcal{P}_{\mathbf{H}}$;
return $\hat{\mathbf{f}}(\boldsymbol{x})$;

3.7. Stability–filtered robust Pareto front

Given the final surrogates $\hat{\mathbf{f}}(\cdot)$ by VRBO, we estimate for each candidate $\boldsymbol{x} \in \mathcal{X}$ the robust mean $\boldsymbol{\mu}(\boldsymbol{x}) \approx \mathbf{J}(\boldsymbol{x})$ and uncertainty $\boldsymbol{\sigma}(\boldsymbol{x}) \approx \sqrt{\mathbf{V}(\boldsymbol{x})}$ by propagating input noise through the surrogate (RGP in closed form, or MC otherwise). We then apply a hard stability filter $\boldsymbol{\sigma}(\boldsymbol{x}) \leq \mathbf{p_V}$ and perform non–dominated sorting, using the implementation provided in BoTorch [45], on $\boldsymbol{\mu}(\boldsymbol{x})$. The resulting set is our reported robust Pareto front (RMO). The process is presented in Algorithm 3. In our pipeline, the parameter α primarily affects the sampling policy during optimisation. After the data have been collected, however, we refit the surrogate on all observations and perform a separate global scan over the design space, applying the variance filter

$\boldsymbol{\sigma}(\boldsymbol{x}) \preceq \mathbf{p_V}$ and non–dominated sorting on $\boldsymbol{\mu}(\boldsymbol{x})$. Thus, α mainly controls where we sample during the optimisation loop, while the final robust Pareto set is determined in a post–hoc manner by the converged surrogate and the chosen tolerance $\mathbf{p_V}$.

Algorithm 3: Robust evaluation & variance–filtered Pareto selection

Input: $\hat{\mathbf{f}}(\cdot)$: trained surrogate(s);
$\mathcal{X} \subset \mathbb{R}^d$: candidate designs;
$p_{\boldsymbol{\xi}}$: input–perturbation law;
S: Monte–Carlo samples per design;
$\mathbf{p_V}$: per–objective uncertainty thresholds
Output: $\mathcal{P}_{\text{RMO}}$: stability–feasible robust Pareto set

foreach $\boldsymbol{x} \in \mathcal{X}$ **do**
  Draw $\{\boldsymbol{\xi}^{(s)}\}_{s=1}^{S} \sim p_{\boldsymbol{\xi}}$;
  $\mathbf{y}^{(s)} \leftarrow \hat{\mathbf{f}}\big(\boldsymbol{x} + \boldsymbol{\xi}^{(s)}\big)$, $s = 1, \ldots, S$;
  $\boldsymbol{\mu}(\boldsymbol{x}) \leftarrow \frac{1}{S}\sum_{s=1}^{S} \mathbf{y}^{(s)}$;
  $\boldsymbol{\sigma}(\boldsymbol{x}) \leftarrow \sqrt{\frac{1}{S-1}\sum_{s=1}^{S}\big(\mathbf{y}^{(s)} - \boldsymbol{\mu}(\boldsymbol{x})\big)^2}$;
$\mathcal{X}_{\text{feas}} \leftarrow \{\, \boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{\sigma}(\boldsymbol{x}) \preceq \mathbf{p_V} \,\}$;
$\mathcal{P}_{\text{RMO}} \leftarrow \textsc{NonDominatedSort}\big(\mathcal{X}_{\text{feas}},\ \text{objectives} = \boldsymbol{\mu}\big)$;
return $\mathcal{P}_{\text{RMO}}$;

4. Noise–free Observations

In practical engineering optimisation, data typically come from two sources: high–fidelity simulations and physical experiments. In the simulation regime, the commanded design $\boldsymbol{x}$ is realised exactly and the outputs are deterministic, so the training data are noise–free. Input uncertainty is therefore not part of the data generation; instead, it is accounted for at evaluation time by propagating perturbations through the RGP (Algorithm 2). This setup allows us to isolate the intrinsic performance of the proposed framework under ideal observations, which is validated by three benchmark cases in this section. The complementary experimental regime—inaccurate in observations–is addressed later in Section 5, where we demonstrate our method under noisy observations.

4.1. Detailed Analysis: MDTP3

This subsection presents a comprehensive investigation of the MDTP3 problem, which serves as the primary benchmark for assessing the proposed framework.

4.1.1. Ground Truth Analysis

The MDTP3 problem is defined over a two–dimensional domain $(x_1, x_2) \in [0, 1]^2$ and employs trigonometric or polynomial components to generate complex non–dominated fronts. By adjusting problem parameters or the input domain, the degree of conflict between objectives can be controlled. In this work, we adopt the following definition:

$$\begin{aligned} f_1(x_1, x_2) &= x_1, \\ f_2(x_1, x_2) &= 1 - 0.9\,\exp\Big[-\Big(\frac{x_2 - 0.8}{0.1}\Big)^2\Big] - 1.3\,\exp\Big[-\Big(\frac{x_2 - 0.3}{0.03}\Big)^2\Big] - x_1. \end{aligned} \tag{14}$$

The optimisation task aims to minimise both objectives:

$$\min_{(x_1,x_2)\in[0,1]^2} \Big[f_1(x_1, x_2),\ f_2(x_1, x_2)\Big].$$

To evaluate the robustness of the proposed framework, Gaussian input noise with standard deviation $\sigma_{x_1} = \sigma_{x_2} = 0.02$ is introduced to both design variables. This noise level is chosen because it produces a ground truth mean and standard deviation landscape with conflicting characteristics, as illustrated in Fig. 5, generated by high–dimension grid search combined with Monte Carlo evaluation. At $x_2 = 0.3$ the mean performance is better than at $x_2 = 0.8$, whereas the standard deviation at $x_2 = 0.3$ is significantly higher than at $x_2 = 0.8$. Such conflict induces a challenging trade–off between performance and robustness.

As a consequence, the location of the ground truth Pareto front depends on the uncertainty threshold $\mathbf{p_v}$: for a high threshold ($\mathbf{p_v} = 0.5$), the Pareto–optimal solutions are concentrated around $x_2 = 0.3$, prioritising mean performance despite higher variability; for a low threshold ($\mathbf{p_v} = 0.1$), the front shifts towards $x_2 = 0.8$, favouring lower variability over mean performance.

4.1.2. Searching Strategy

As presented in Section 3 and Eq. (7), the optimisation problem is formulated with a hard constraint. However, due to resource limitations, direct evaluation of the constraint is infeasible. We anticipate that incorporating the constraint information can still guide the optimisation process towards promising regions of the design space. To achieve this, the variance–penalised scalarisation algorithm is proposed that transforms the hard constraint into a soft form, enabling control over the influence of uncertainty through the parameter α and thereby steering the search process accordingly.

Fig. 6 illustrates the distribution of observed points in MDTP3 for different α values. When $\alpha = 0$, shown in Fig. 6 (a), the search favours regions $x_2 = 0.3$ with low values regardless of variability. As α increases, the emphasis of the searching strategy gradually shifts from $x_2 = 0.3$ to $x_2 = 0.8$ where the mean is higher while

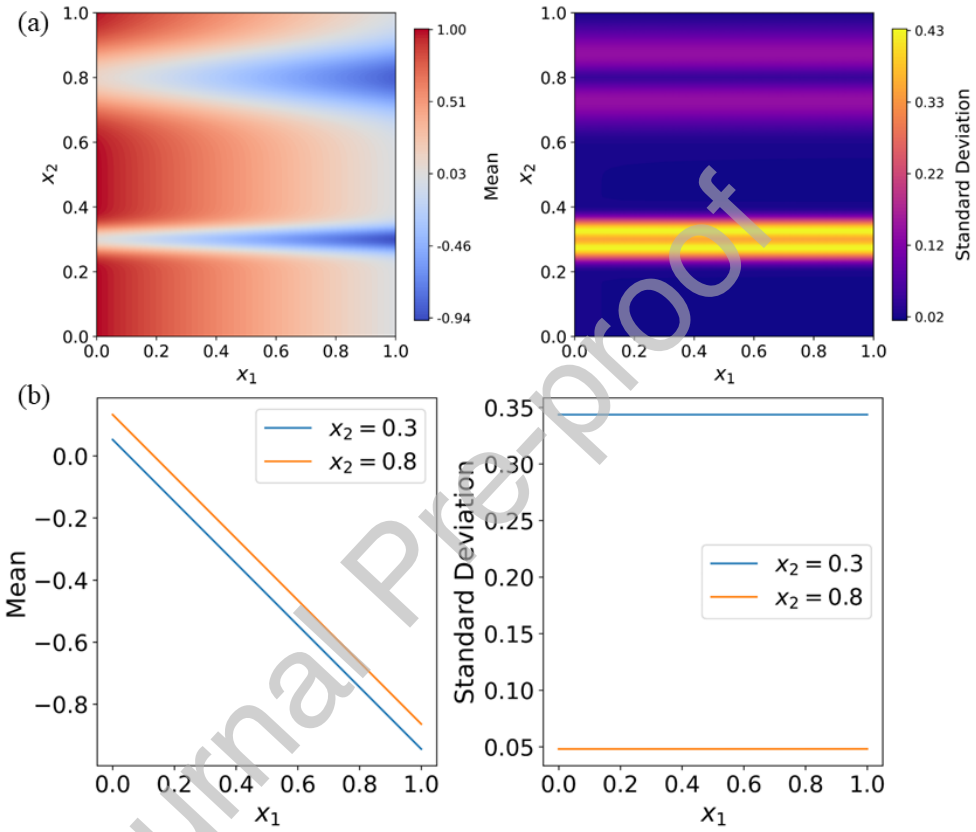

Figure 5: Ground truth performance landscape of MDTP3 objective 2 (f_2) under zero–mean Gaussian input noise ($\sigma_{x_1} = \sigma_{x_2} = 0.02$). (a) ($f_2$)–mean (left) and –standard deviation (right) over the 2D domain. (b) Cross–sectional statistics at $x_2 = 0.3$ and $x_2 = 0.8$. While $x_2 = 0.3$ yields a lower mean than $x_2 = 0.8$, it also exhibits substantially higher variability, creating a clear trade–off between performance and robustness. This conflict results in distinct Pareto front locations for different uncertainty thresholds: for $\mathbf{p_V} = 0.5$, the front is near $x_2 = 0.3$; for $\mathbf{p_V} = 0.1$, it shifts towards $x_2 = 0.8$.

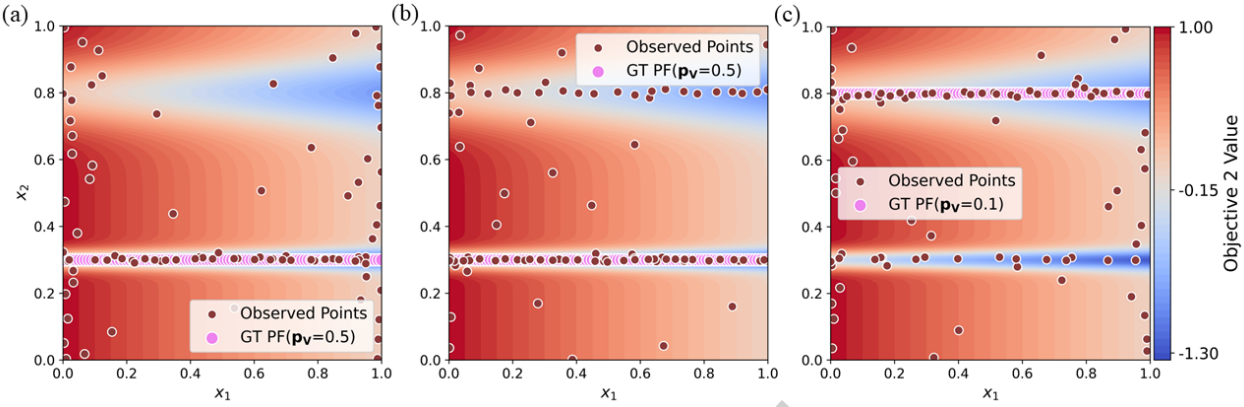

Figure 6: Distribution of observed points (shown in red circles) in the MDTP3 search process under different values of α in the soft–constrained acquisition function. The pink circles indicate the ground truth Pareto front (GT PF) for the corresponding uncertainty threshold $\mathbf{p_V}$. (a) $\alpha = 0$. (b) $\alpha = 0.4$. (c) $\alpha = 1$.

uncertainty is lower. By adjusting α according to the specified constraint threshold, the search can be directed to regions of the design space that probably satisfy the desired trade–off between performance and robustness. This flexibility allows the proposed framework to adapt to different stability requirements without redesigning the entire optimisation process, thereby improving efficiency and applicability in practical scenarios.

4.1.3. Final Performance

The VRBO procedure is used to collect data and fit the final surrogates. The reported Pareto front is the robust front (RMO), which directly solves the hard–constrained problem of Section 3.7 by enforcing the uncertainty threshold $\mathbf{p_V}$ and performing non–dominated sorting on the robust means. For reference, we also post–process the final surrogate model to obtain a soft, scalarisation–based front (RMO–H): starting from the same GP posteriors used by RMO, we compute the H–criterion given by Eq. 13 for a dense candidate set in the design space and extract the non–dominated set in $\mathbf{H}$–space. This RMO–H front is shown purely as a comparison front to illustrate the effect of scalarising mean and variance; it does not define the optimisation target or drive the sampling in our experiments.

Additionally, for comparison, an ENSGA–II baseline is also included. ENSGA–II adopts a standard NSGA–II procedure operating on a normalised $[0, 1]^d$ design space (population–based evolution, nondominated sorting, crowding–distance selection, simulated–binary crossover and polynomial mutation). The only modification lies in how each individual is evaluated under input noise. For a given design $\boldsymbol{x}$, we draw S_{ENSGA} perturbed inputs from the prescribed input–noise distribution centred

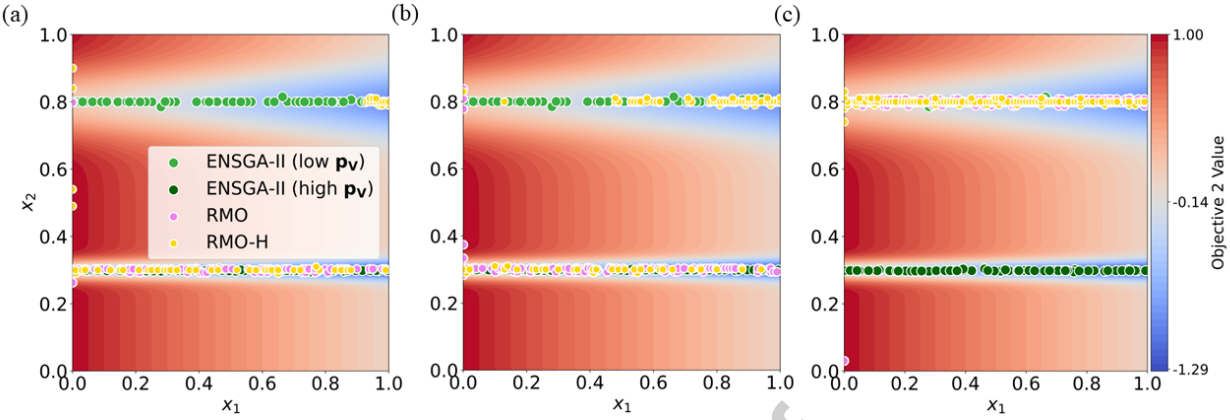

Figure 7: Final optimisation results on MDTP3 for different $\mathbf{p_V}$ values. ENSGA–II (low $\mathbf{p_V}$) and ENSGA-II (high $\mathbf{p_V}$) denote the exhaustive NSGA–II baseline for the respective uncertainty thresholds. RMO: robust Pareto front obtained by applying the final GP to solve the hard–constrained problem. RMO–H: robust Pareto front obtained using the H criterion with a soft uncertainty constraint. (a) $\alpha = 0, \mathbf{p_V} = 0.5$, (b) $\alpha = 0.4, \mathbf{p_V} = 0.5$, and (c) $\alpha = 1.0, \mathbf{p_V} = 0.1$.

at $\boldsymbol{x}$, evaluate the underlying ground–truth benchmark function at each perturbed input, and aggregate the resulting objective vectors. The sample means of these S_{ENSGA} realisations are used as the objective values passed to NSGA–II, while the corresponding sample standard deviations are used to check whether the design satisfies the prescribed variability threshold for the given $\mathbf{p_V}$ level. With a sufficiently large exhaustive budget, ENSGA–II provides a high–quality approximation to the robust Pareto front. In this paper, $S_{\text{ENSGA}} = 2000$ to approximate the real mean and standard deviation.

The results are illustrated in Fig. 7. It can be observed that for $\alpha = 0, \mathbf{p_V} = 0.5$ (Fig. 7a), both of RMO and ENSGA–II correctly locate the front near $x_2 \approx 0.3$. RMO–H produces a slight shift due to the soft constraint. While for $\alpha = 1.0, \mathbf{p_V} = 0.1$ (Fig. 7c), all methods correctly locate the front near $x_2 \approx 0.8$. In the transition setting with $\alpha = 0.4$ and $\mathbf{p_V} = 0.5$ (Fig. 7b), RMO–H produces a smooth shift from the high–$\mathbf{p_V}$ front towards the low–$\mathbf{p_V}$ front, with the transition initiating from the trailing end of the branch, i.e., at larger x_1 values. This is because RMO–H applies a soft constraint integrating the mean and standard deviation. Although the entire branch exhibits similar uncertainty, the soft formulation of H makes the importance of uncertainty vary with the mean value. In the large–x_1 region, where the mean is closer to the global minimum, uncertainty will play a more important role, causing the shift to begin from this end of the branch.

4.1.4. Evaluation of Performance and Computational Cost

To quantitatively assess the quality of the obtained robust Pareto fronts, three widely used performance indicators are considered. The Generational Distance (GD) [46] measures the average Euclidean distance from each point on the obtained Pareto front to its nearest point on the ground–truth front; smaller GD values indicate better convergence towards the true front. The Inverted Generational Distance (IGD) [46] measures the average distance from each point on the ground–truth front to its nearest point on the obtained front; this metric reflects both convergence and diversity, with smaller values indicating better coverage of the true front. The hypervolume difference (ΔHV) computes the loss in hypervolume between the obtained front and the ground–truth front with respect to a common reference point; a smaller ΔHV means the obtained front captures a larger portion of the dominated region defined by the true front.

For each benchmark problem and each variance threshold $\mathbf{p_V}$, the ground–truth robust Pareto front is first obtained via a high–resolution grid search over the normalised design space, using the same input–uncertainty level and variance constraint $\mathbf{p_V}$ as in our experiments. The fronts returned by VRBO and ENSGA–II are then compared against this ground–truth front using GD, IGD and ΔHV.

To make the deviations interpretable and comparable across benchmarks, we report GD and IGD as percentages relative to the objective space. Concretely, we first compute GD/IGD in the original objective space and then normalise the resulting Euclidean distances by the size of that space (the Euclidean length of the bounding box defined by the lower and upper bounds of each objective). The normalised values are expressed as percentages, so that a GD or IGD of only a few percent corresponds to average deviations that are small relative to the full objective range. For ΔHV, we additionally report the absolute hypervolume of the ground–truth front, $\mathrm{HV}_{\mathrm{GT}}$, which allows the relative impact of ΔHV to be judged by comparing it directly to the total attainable hypervolume in each benchmark.

Table 1 summarises, for one representative benchmark, the quantitative results of VRBO and ENSGA–II under two variance thresholds ($\mathbf{p_V} = 0.5$ and $\mathbf{p_V} = 0.1$), together with the associated computational cost measured as the total number of expensive evaluations of the underlying function. In our sequential surrogate–based setting, these black–box evaluations dominate the overall computational effort, whereas updating and querying the surrogate is comparatively cheap. The number of function evaluations is therefore the most natural and comparable proxy for cost across all methods. The same evaluation protocol (high–resolution grid for ground truth, normalised GD/IGD, and ΔHV with $\mathrm{HV}_{\mathrm{GT}}$) is applied consistently across all benchmark problems considered in this work.

From Table 1, we observe that RMO, evaluated on the VRBO–trained surrogate, attains GD and IGD values on the order of only a few percent of the objective

Table 1: Quantitative comparison between RMO and ENSGA–II in terms of ΔHV, GD, IGD, and computational cost for MDTP3. Lower values of ΔHV, GD, and IGD indicate better performance.

Method	GD(%)	IGD(%)	ΔHV	HV_{GT}	Cost (evaluations)
RMO $\alpha = 0$ [$\mathbf{p_V} = 0.5$]	5.7	1.0	0.0151	1.14	90
RMO $\alpha = 0.4$ [$\mathbf{p_V} = 0.5$]	3.4	0.6	0.0151	1.14	85
ENSGA–II [$\mathbf{p_V} = 0.5$]	0.4	0.8	0.0094	1.14	144×2000
RMO $\alpha = 1.0$ [$\mathbf{p_V} = 0.1$]	1.4	0.01	0.0105	1.05	73
ENSGA–II [$\mathbf{p_V} = 0.1$]	0.3	0.9	0.0087	1.05	300×2000

space, together with small ΔHV values relative to the corresponding ground–truth hypervolumes HV_{GT} for both variance thresholds $\mathbf{p_V} = 0.5$ and $\mathbf{p_V} = 0.1$. While ENSGA–II is included as a reference optimisation baseline, the results indicate that RMO is capable of producing robust Pareto fronts with comparable solution quality under the considered uncertainty settings.

This observation is further supported by the visual comparisons in Fig. 7, where the Pareto fronts obtained by RMO are almost indistinguishable from the ground–truth fronts obtained via exhaustive search. This efficiency stems from VRBO's bi–loop design and variance-penalised scalarisation (via α), which decouple surrogate refinement from uncertainty control and steer sampling to stability–feasible regions.

4.1.5. Bi–loop Effect

All results reported in the previous subsection (e.g., Fig. 7 and Table 1) were obtained with the full bi–loop VRBO framework. To further examine the contribution of the proposed bi–loop strategy, we compare the optimisation results obtained under the same α setting but without the bi–loop. Instead of integrating the traditional acquisition function and $\mathcal{H}$, the baseline here directly applies the $\mathcal{H}$ criterion as the sole acquisition function throughout the optimisation.

The resulting mean and standard–deviation landscapes are shown in Fig. 8. Compared with the bi–loop results, the single–loop variant exhibits limited exploration and a Pareto set that stalls in local minima. This behaviour arises because in the RGP model, both the predictive model uncertainty and the propagated input uncertainty are intertwined; when they are not decoupled through the bi-loop, the optimiser may over–exploit regions with underestimated uncertainty, leading to sub-optimal coverage of the true Pareto front.

4.2. Additional Benchmarks: VLMOP2 and MDTP2

To further examine the generality of the proposed framework, we evaluate it on two additional problems: VLMOP2 and MDTP2. These benchmarks represent distinct optimisation landscapes, enabling an assessment of performance across both

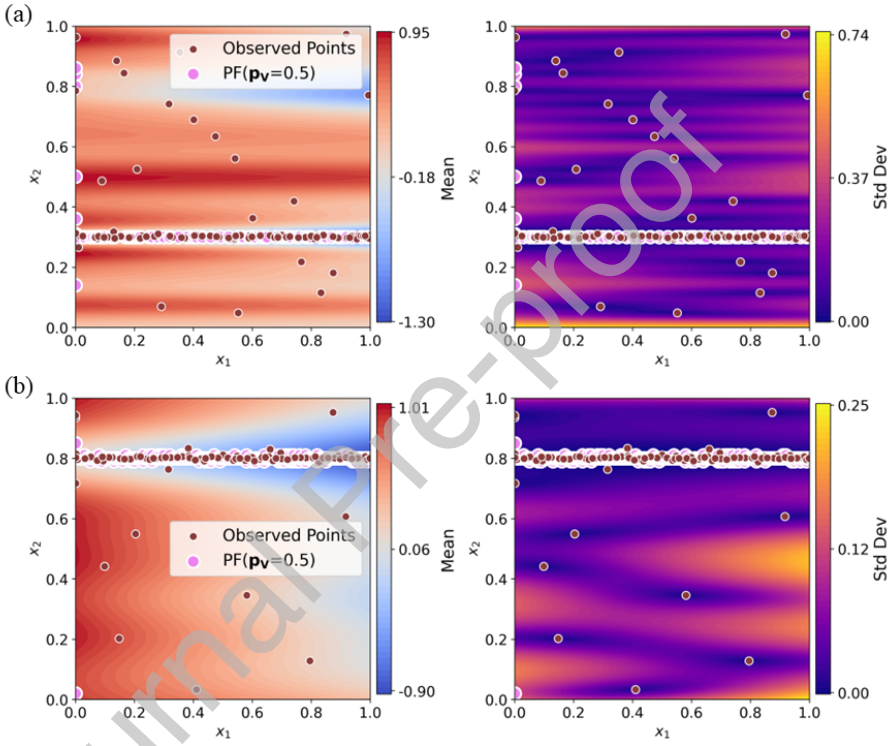

Figure 8: Optimisation results for MDTP3 without the proposed bi–loop strategy, using only the H criterion as the acquisition function. The lack of alternating updates between model training and acquisition search leads to premature convergence and entrapment in local minima, due to the coupling of model and input uncertainties in RGP. Both examples correspond to $\alpha = 0$, illustrating the consistency of this failure mode.

smooth and highly nonlinear multi–objective problems. Complete problem formulations and experimental settings are provided in Appendix A.

4.2.1. VLMOP2

The VLMOP2 problem is a smooth bi–objective benchmark with a convex and continuous Pareto front, commonly used to assess convergence accuracy in noise–free multi–objective optimisation. Fig. 9 compares the RMO obtained by the proposed VRBO framework and the exhaustive ENSGA–II baseline. The contour maps depict the objective function landscapes, with the left panel showing Objective 1 and the right panel showing Objective 2. Both methods successfully locate the true Pareto front (GT PF), with the VRBO solutions (RMO) closely aligned along the optimal trade–off curve. Despite using far fewer function evaluations, the VRBO method achieves accuracy comparable to the exhaustive baseline, confirming its efficiency even on smooth, well-behaved problems.

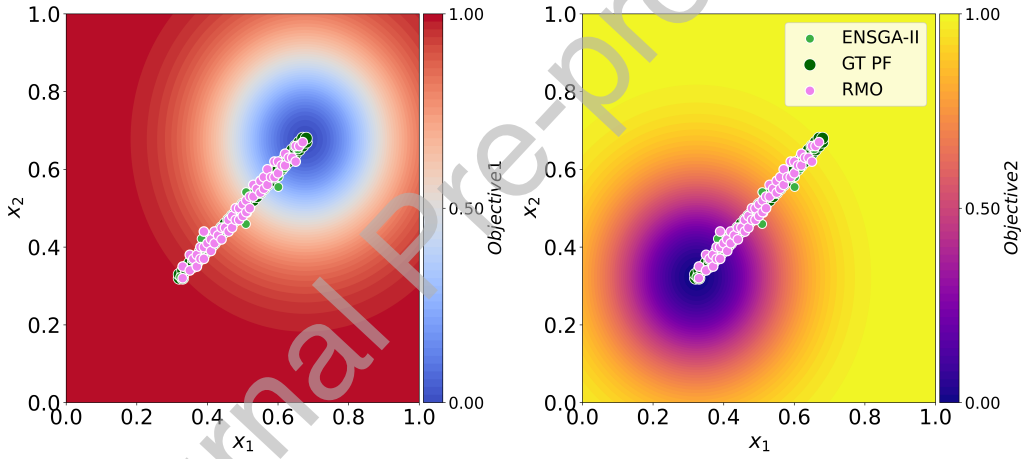

Figure 9: Optimisation results for the VLMOP2 problem: comparison between RMO, ENSGA–II, and the ground–truth Pareto front (GT PF). Contour plots show Objective 1 (left) and Objective 2 (right).

4.2.2. MDTP2

The MDTP2 problem is a two–objective extension of the Branin–Hoo function, featuring strong nonlinearity and multiple local minima in the second objective. This structure makes it a more challenging test case, particularly for algorithms that must balance global exploration with convergence to the true front. Fig. 10 shows that both VRBO (result in RMO) and ENSGA–II identify the ground–truth robust Pareto front, with RMO achieving similar accuracy while requiring only a small fraction of the evaluations. These results highlight the framework's ability to avoid local optima and capture complex front structures under uncertainty.

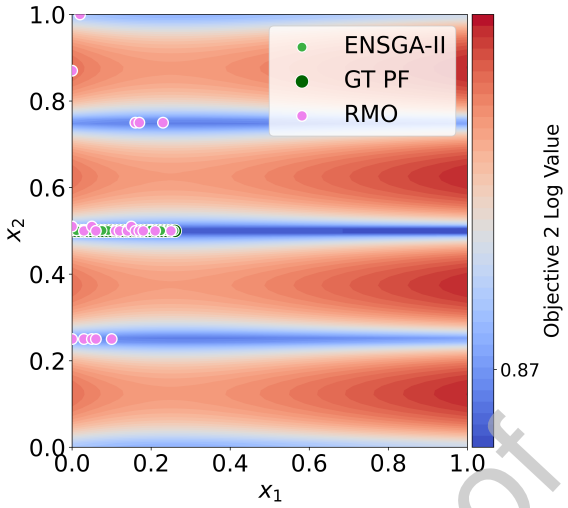

Figure 10: Optimisation results for the MDTP2 problem: comparison between RMO (given by VRBO), ENSGA–II, and the ground–truth Pareto front (GT PF). Contour plots show Objective 2.

The final quantitative performance for both problems, including ΔHV, GD, and computational cost, is summarised in Table 2. Lower values of ΔHV, GD, and IGD indicate better performance.

Table 2: Quantitative comparison between RMO and ENSGA–II in terms of ΔHV, GD, IGD, and computational cost for VLMOP2 and MDTP2. Lower values of ΔHV, GD, and IGD indicate better performance. Note that a negative ΔHV indicates the algorithm outperformed the grid search benchmark, finding denser solutions in certain regions than the discretised baseline.

Benchmark	Method	GD(%)	IGD(%)	ΔHV	HV_GT	Cost (evaluations)
VLMOP2	RMO [$\mathbf{p_V}$ = 0.1]	0.16	0.02	0.004	0.506	126 × 2000
	ENSGA–II [$\mathbf{p_V}$ = 0.1]	0.16	0.02	0.004	0.506	126 × 2000
MDTP2	RMO [$\mathbf{p_V}$ = 10]	2.35	1.29	0.133	2.01	72 × 2000
	ENSGA–II [$\mathbf{p_V}$ = 10]	0.35	0.12	-0.0090	1	72 × 2000

5. Noisy Observations

When optimisation relies on experimental data, different from simulations, measurement errors, fabrication tolerances, or other perturbations may cause deviations between the intended and actual design variables when generating the dataset and observation. Therefore, the dataset inherently contains noise. Since experiments play an equally important role in engineering design, it is essential to evaluate the proposed framework under these realistic, noisy conditions. Therefore, our method

is tested in noisy (experimental) scenarios to comprehensively assess its robustness and applicability, as illustrated in Fig. 11.

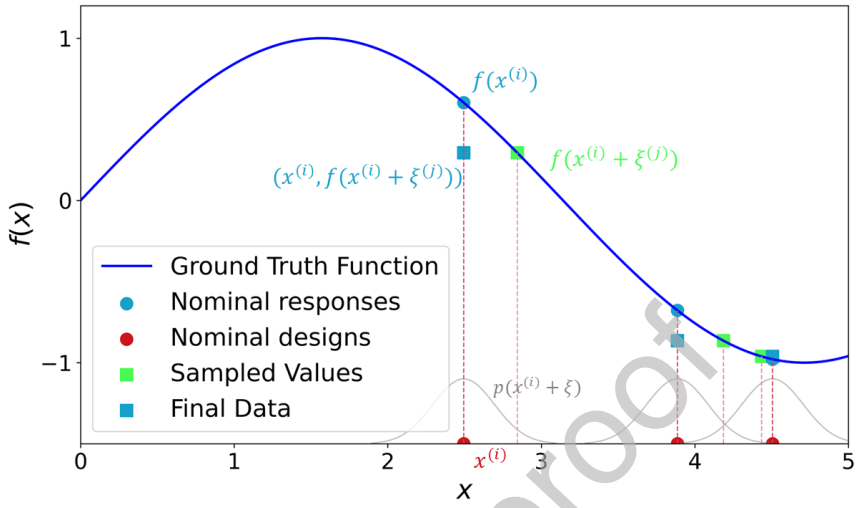

Figure 11: Illustration of input design uncertainty in system response optimisation. Each deterministic proposed design $x^{(i)}$ (red dot) is in fact perturbed $x^{(i)}+\xi$ and associated with a distribution, drawn from $p_\xi(\xi)$ (e.g. shown as grey Gaussian curves). The function evaluation at a particular perturbed location $f(x^{(i)}+\xi^{(j)})$ (green squares) is therefore wrongly attributed at $x^{(i)}$ as $f(x^{(i)})$ (blue squares) which modifies the perceived response and then affects the subsequent optimisation.

5.1. SinLinForrester

SinLinForrester benchmark is utilised to validate the effectiveness under this scenario.

SinLinForrester combines two one–dimensional test objectives, the Sine-Linear and Forrester functions, yielding a bi–objective problem in a single–input domain $x \in [0,1]$. Despite its low dimensionality, it demonstrates conflicting objectives, since one has oscillatory behaviour while the other is smoother.

The aim for this case is to minimise both $f_1(x)$ and $f_2(x)$ defined as

$$\begin{aligned} f_1(x) &= \sin\left(5\pi x^2\right) + 0.5\,x, \\ f_2(x) &= \left(6x-2\right)^2 \sin\left(12\,x-4\right). \end{aligned} \tag{15}$$

Hence, the nominal optimisation problem is $\min_{0\le x\le 1}[f_1(x), f_2(x)]$. And Gaussian input noise with standard deviation $\sigma = 0.04$ is introduced to the design parameter.

The final results are shown in Fig. 12 and Table 3. In Fig. 12, the solid curves depict the ground–truth objectives, the dashed curves show the GP posterior means, and the shaded ribbons indicate the input–induced (ground–truth) uncertainty. Objective 2 exhibits stronger variability: around $x \approx 0.80$ it attains lower values but with

higher uncertainty, whereas around $x \approx 0.55$ the trend is reversed. Observed data from the optimisation run are overlaid (blue dots for Objective 1, red dots for Objective 2), along with vertical ticks illustrating the distribution of sampled inputs. Color hue distinguishes the source (blue for MO loop, orange for RMO loop), while color saturation encodes the temporal progression, with deeper intensities indicating later iterations. Comparing the two scenarios highlights the distinct roles of the two loops and α: the MO loop (indicated by blue ticks) maintains global exploration across the entire region, whereas the RMO loop (orange ticks) concentrates on the specific area of interest. The location of RMO samples is governed by α: with a smaller α (panel (a), $\alpha = 0$, $\mathbf{p_V} = 5$), the search concentrates near the high–uncertainty region around $x \approx 0.80$ gradually; increasing α (panel (b), $\alpha = 0.1$, $\mathbf{p_V} = 2$) shifts sampling toward safer, lower–uncertainty regions (roughly $x \approx 0.55$–0.65). This behaviour confirms that VRBO flexibly steers exploration–exploitation according to the uncertainty preference encoded by α.

Quantitatively, Table 3 compares the final RMO solutions with the exhaustive ENSGA–II baseline using ΔHV, GD, and IGD against the ground–truth Pareto front. For α=0 and $\mathbf{p_V}$=5, RMO achieves a small GD because the search concentrates on the low–mean yet high–uncertainty region around $x \approx 0.80$; the GP is well trained there, so the recovered front lies close to that portion of the ground–truth front. However, this focus leaves the branch near $x \approx 0.55$ sparsely explored, leading to under–coverage and hence larger IGD and ΔHV. For α=0.1 and $\mathbf{p_V}$=2, sampling shifts toward safer, lower–uncertainty regions and coverage improves markedly relative to α=0 (smaller IGD/ΔHV). Overall, these results highlight how the choice of robustness parameter α and variance threshold $\mathbf{p_V}$ influences the exploration–exploitation balance and the resulting approximation quality of the robust Pareto front.

Table 3: Quantitative comparison between RMO and ENSGA-II in terms of ΔHV, GD, IGD, and computational cost for SinLinForrester. Lower values of ΔHV, GD, and IGD indicate better performance.

Method	GD(%)	IGD(%)	ΔHV	$\mathrm{HV_{GT}}$	Cost (evaluations)
RMO $\alpha = 0$ [$\mathbf{p_V} = 5$]	0.34	2.41	0.2366	5.39	45
ENSGA–II [$\mathbf{p_V} = 5$]	1.79	1.11	-0.0367	5.39	78×2000
RMO $\alpha = 0.1$ [$\mathbf{p_V} = 2$]	3.08	1.18	0.2749	4.39	45
ENSGA–II [$\mathbf{p_V} = 2$]	1.2	0.61	0.0087	4.39	108×2000

The fact that the $\alpha = 0$ run in Fig. 12 does not recover the solution near $x \approx 0.55$ reflects the effect of input–uncertainty–corrupted observations on a surrogate-based optimiser, rather than a fundamental failure of global exploration; further runs and discussion are reported in the Supplementary Material.

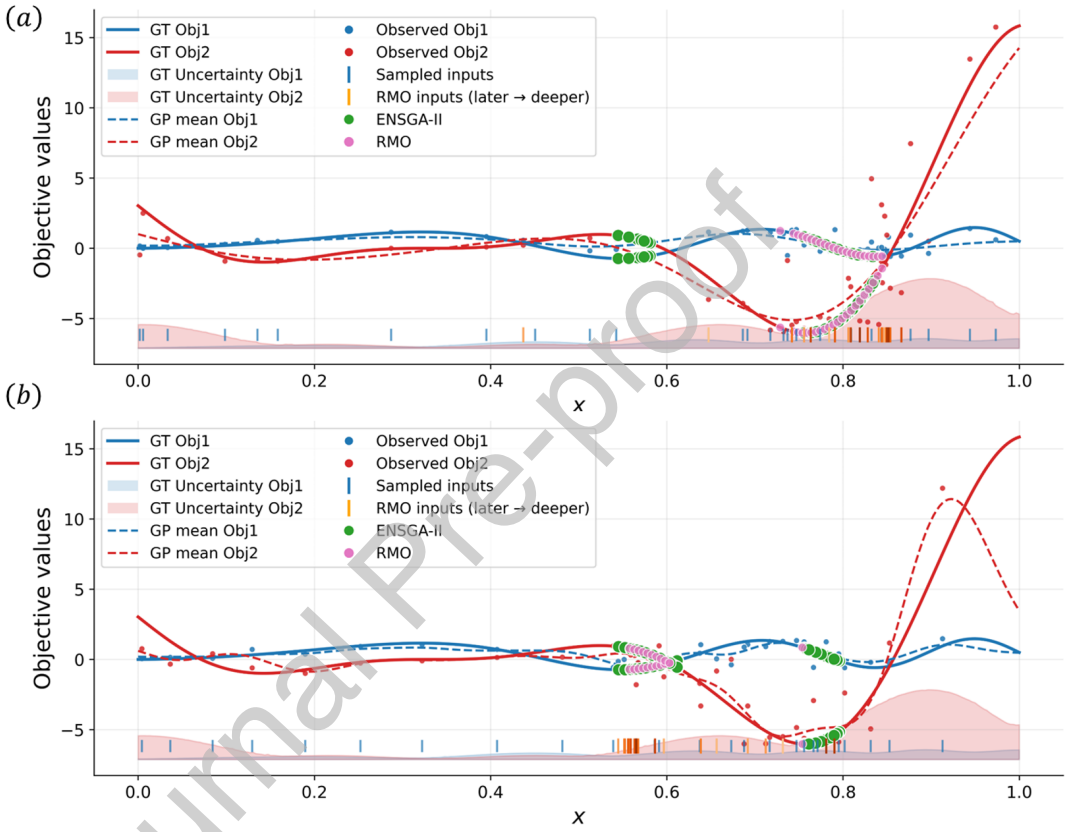

Figure 12: One–dimensional, two–objective example under input noise. Solid curves are the ground–truth objectives; dashed curves are GP posterior means; shaded bands show the ground–truth uncertainty (standard deviation). Dots are observed data collected at nominal x with noisy outputs (so they do not lie exactly on the ground–truth curves). Short vertical ticks on the x–axis mark sampled input locations: blue ticks are proposed by the MO loop and orange ticks by the RMO loop (deeper color means later iterations). Green and pink markers denote ENSGA–II and RMO solutions, respectively. (a) $\alpha = 0, \mathbf{p_V} = 5$, (b) $\alpha = 0.1, \mathbf{p_V} = 1$.

6. High–Dimension Benchmark case – Fonseca–Fleming

The last test problem is a moderately high–dimensional benchmark based on the Fonseca–Fleming (FON) formulation. It is a smooth bi–objective problem that is frequently used in multi–objective optimisation to assess convergence and diversity in spaces of dimension larger than two. In contrast to the previous two–dimensional benchmarks, this case allows us to investigate how the proposed VRBO framework scales when both the decision space and the input–uncertainty space become higher dimensional.

For a design vector $\boldsymbol{x} = (x_1, \ldots, x_M)$, the FON objectives are defined as

$$f_1(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i=1}^{M}\left(x_i - 1/\sqrt{M}\right)^2\right), \qquad f_2(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i=1}^{M}\left(x_i + 1/\sqrt{M}\right)^2\right), \tag{16}$$

and the optimisation task is to minimise both objectives

$$\min_{\boldsymbol{x}\in\mathcal{X}}\left[f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x})\right]. \tag{17}$$

In the original definition, the physical design domain is the hypercube $\boldsymbol{x} \in [-2, 2]^M$.

To test robustness under input uncertainty, independent Gaussian perturbations $\mathcal{N}\left(\mathbf{0},\, \sigma^2\right)$ are applied to all decision variables in the normalised coordinates, where $\sigma = 0.02$. In this study, we consider $M = 6$ decision variables, a setting that is representative of typical engineering design problems [47, 48].

6.1. Ground Truth Analysis

Under this setting, the ground–truth robust Pareto fronts under different uncertainty constraints $\mathbf{p_V}$ are shown in Fig. 13. To construct this reference, we evaluate the FON on a dense 21^6 grid in the normalised design space and, for each design, perform 100 Monte Carlo samples of the input noise. For the chosen noise level, the resulting standard deviation of each objective lies roughly in $[0, 0.08]$. Consequently, the looser threshold $\mathbf{p_V} = 0.10$ (blue dashed curve) is effectively unconstrained and reproduces the nominal Pareto front, whereas the stricter threshold $\mathbf{p_V} = 0.05$ (orange solid curve) removes high–variance designs and yields a shorter, more conservative front with noticeably higher f_2 for the same f_1 in the middle region. This illustrates how tighter variability constraints shrink the feasible robust Pareto set and push the trade–off towards safer but less optimal designs.

6.2. Performance

As presented in Fig. 14 and summarised in Table 4, the behaviour of VRBO under different robustness settings can be clearly observed. When $\alpha = 0$ and $\mathbf{p_V} = 1$, the

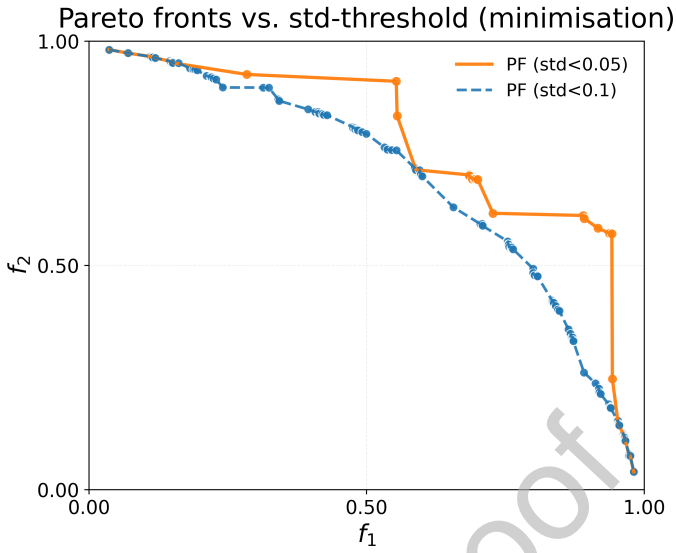

Figure 13: Ground–truth robust Pareto fronts of the FON benchmark under two uncertainty thresholds $\mathbf{p_V} \in \{0.05, 0.10\}$. When $\mathbf{p_V} = 0.1$, the constraint is not activated. The fronts are obtained by exhaustive evaluation on a 21^6 grid with 100 Monte Carlo samples per design.

RMO front aligns closely with the GT Pareto front as shown in Fig. 14 (a), with GD and IGD both around 0.03, and a small ΔHV indicating negligible hypervolume loss. However, under the stricter uncertainty requirement $\mathbf{p_V} = 0.05$, the approximation quality deteriorates noticeably when $\alpha = 0$ in Fig. 14 (b). Increasing the α to $\alpha = 5$ significantly improves the fit under the same $\mathbf{p_V} = 0.05$ in Fig. 14 (c), as reflected by lower GD, IGD, and ΔHV, showing that a larger α helps recover a robust Pareto front closer to the GT reference when tighter variability control is desired.

Table 4: Quantitative comparison between RMO fronts and GT Pareto fronts on the FON problem.

Case	GD(%)	IGD(%)	ΔHV	$\mathrm{HV_{GT}}$
$\alpha = 0,\ \mathbf{p_V} = 1$	3.4	3.4	0.043	0.492
$\alpha = 0,\ \mathbf{p_V} = 0.05$	7.5	4.6	0.066	0.413
$\alpha = 5,\ \mathbf{p_V} = 0.05$	4.3	2.5	0.011	0.413

7. Conclusion

This work introduced VRBO, a sample—efficient framework for multi–objective optimisation under input uncertainty that unifies robustness and stability within a single workflow. The core of VRBO is a RGP surrogate that propagates input perturbations to produce Bayes' risk (means and variances) for each objective. On top of

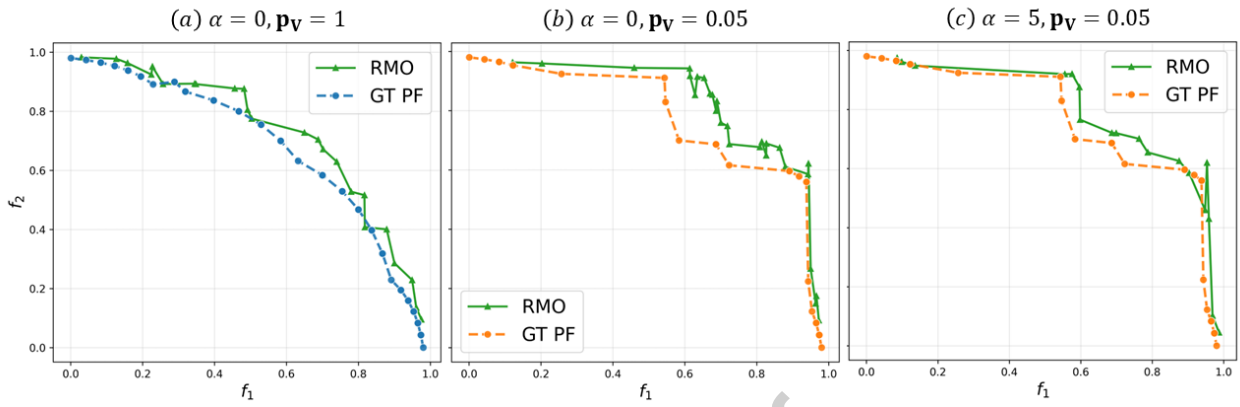

Figure 14: Comparison between the robust Pareto fronts obtained by VRBO (RMO) and the Monte Carlo ground–truth fronts on the 6D Fonseca–Fleming problem under different robustness parameters. (a) $\alpha = 0$, $\mathbf{p_V} = 1$; (b) $\alpha = 0$, $\mathbf{p_V} = 0.05$; (c) $\alpha = 5$, $\mathbf{p_V} = 0.05$.

this surrogate, we proposed a variance-penalised scalarisation that balances expected performance with uncertainty through a tunable parameter α, and a bi–loop strategy that decouples model (epistemic) from objective (aleatory) uncertainty. Stability enters the pipeline twice: softly during search via α, and as a hard uncertainty filter using the user tolerance $\mathbf{p_V}$ when reporting the robust Pareto set.

Comprehensive tests of the proposed framework in both noise–free (simulation) and noisy (experimental) regimes demonstrate that VRBO reliably locates appropriate robust Pareto fronts as $\mathbf{p_V}$ varies. On MDTP3, VRBO reproduces the expected shift of the front between high–variance and low–variance bands, and the RMO–H variant explains the gradual transition induced by the soft weighting. Across VLMOP2, MDTP2, SinLinForrester, and a six–dimensional Fonseca–Fleming benchmark, the recovered Pareto sets match ground truth to within small GD/IGD/ΔHV gaps. These results confirm the ability of the proposed framework to accurately characterise robust Pareto structures across a range of problem settings, uncertainty levels, and dimensionalities.

The scalarisation parameter α, although theoretically related to the uncertainty tolerance, still lacks direct interpretability and intuitive selection guidelines. Within the current VRBO framework, certain designs (e.g., the variance filter based on $\mathbf{p_V}$) help to reduce the practical sensitivity to α, but a clearer connection between $\mathbf{p_V}$ and α remains desirable. Future work will therefore focus on establishing a more explicit link between robustness tolerance and scalarisation, potentially through a normalisation scheme that treats mean performance and uncertainty on a comparable scale, and on increasing the adaptivity of the acquisition so that α can evolve automatically with the search process as a function of $\mathbf{p_V}$. In parallel, we also provide in the Appendix B a supplementary split-–acquisition variant that does not

rely on α and instead incorporates $\mathbf{p_V}$ directly into the optimisation objectives, offering an alternative way to encode robustness within the algorithm. Beyond these methodological developments, applying the framework to an industrial battery heat–exchanger design problem is the next step toward demonstrating the practical utility of VRBO in real engineering settings.

Data and code availability

The code and example structure of models used in this paper is available at https://github.com/HaoZhou-713/rbro-vrbo.git.

Declaration of interests

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Emilie Sauret reports financial support was provided by Australian Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Australian Research Council Discovery Project DP230102229 awarded to Emilie Sauret. Hao Zhou gratefully acknowledges the QUT for support through a Ph.D. scholarship.

Abbreviations

BO	Bayesian Optimisation
CVaR	Conditional Value–at–Risk
EHVI	Expected Hypervolume Improvement
EI	Expected Improvement
ENSGA–II	Exhausted NSGA–II
FON	Fonseca–Flemin
GD	Generational Distance
GP	Gaussian Process
IGD	Inverted Generational Distance

MOBO	Multi–Objective Bayesian Optimisation
MOO	Multi–Objective Optimisation
NSGA–II	Non–Dominated Sorting Genetic Algorithm II
RBDO	Reliability–based Design Optimisation
RBO	Robust Bayesian Optimisation
RBRDO	Reliability–based Robust Design Optimisation
RDO	Robust–Design Optimisation
RGP	Robust Gaussian Process
RO	Robust Optimisation
UCB	Upper Confidence Boundary
VaR	Value–at–Risk
VRBO	Variance–constrained Robust Bayesian Optimisation

Appendix A. Benchmark Definitions: VLMOP2 and MDTP2

This appendix specifies the exact formulations, domains, and input–noise settings used for the two auxiliary benchmarks. In both cases we conduct optimisation and robust evaluation in the normalised coordinate system $\mathbf{x} = (x_1, x_2) \in [0,1]^2$, and apply independent Gaussian input perturbations to each component:

$$\boldsymbol{\xi} \sim \mathcal{N}\big(\mathbf{0}, (0.02)^2 \mathbf{I}_2\big), \qquad \text{i.e., } x_j \leftarrow x_j + \xi_j.$$

Unless stated otherwise in the main text, robust means and variances (or standard deviations) are estimated by Monte Carlo under this noise model; feasibility is then enforced via the uncertainty tolerance used in each experiment before extracting the non–dominated set on robust means.

Appendix A.1. VLMOP2

We use the standard VLMOP2 bi–objective problem defined on the physical coordinates $\mathbf{u} = (u_1, u_2) \in [-2,2]^2$:

$$f_1(\mathbf{u}) = 1 - \exp\Big(-\big(u_1 - \tfrac{1}{\sqrt{2}}\big)^2 - \big(u_2 - \tfrac{1}{\sqrt{2}}\big)^2\Big),$$
$$f_2(\mathbf{u}) = 1 - \exp\Big(-\big(u_1 + \tfrac{1}{\sqrt{2}}\big)^2 - \big(u_2 + \tfrac{1}{\sqrt{2}}\big)^2\Big).$$

Optimisation is carried out in the normalised domain $\mathbf{x} \in [0,1]^2$. For function evaluation, we map $\mathbf{x}$ back to the physical domain $\mathbf{u}$, then compute f_1, f_2. Input noise is applied in the normalised coordinates (STD $= 0.02$ per dimension) prior to evaluation. The uncertainty tolerance used for VLMOP2 is as stated in the main text (see Table 2).

Appendix A.2. MDTP2

The MDTP2 bi-objective variant used in this work is

$$\begin{aligned}
&\mathbf{u} = (u_1, u_2), \qquad u_1 \in [0,1], \quad u_2 \in [-1,1],\\
&f_1(\mathbf{u}) = u_1,\\
&f_2(\mathbf{u}) = \left(1 - u_1^2\right) + \left(10 + u_2^2 - 10\cos(4\pi u_2)\right)\left(\tfrac{1}{0.2+u_1} + 10u_1^2\right).
\end{aligned}$$

As with VLMOP2, optimisation and robust evaluation are performed in the normalised space $\mathbf{x}$; independent Gaussian input noise with STD 0.02 is applied to each normalised coordinate. The uncertainty tolerance and evaluation details match those reported in the main text for this case (e.g., Table 2).

Appendix B. Alternative Split Formulation for Parameter–Free Optimisation

To address the challenge of selecting the scalarisation parameter α in Eq. 13 and to provide a parameter–free alternative, this appendix presents a split formulation where the mean performance and the uncertainty are treated as two explicit, separate objectives. Recall that the variance–penalised scalarisation for the i–th objective is originally defined as $\mathcal{H}_i(x) = \mathrm{CVaR}_{\theta_i}(x) + \alpha \ln(1 + \sigma_i(x))$. In the split formulation, this scalarised acquisition is decoupled into two distinct objectives to be minimised simultaneously: the mean performance, denoted as $f_i^{(1)}(x) = \mathrm{CVaR}_{\theta_i}(x)$, and the normalised uncertainty, denoted as $f_i^{(2)}(x)$.

To ensure the uncertainty objective is commensurate with the user–specified variance tolerance $p_{V,i}$, a normalisation factor is introduced. The uncertainty objective is thus defined as:

$$f_i^{(2)}(x) = \frac{\ln(1 + \sigma_i(x))}{\ln(1 + p_{V,i})}. \tag{B.1}$$

This mapping scales the uncertainty term relative to the allowable variability budget, thereby removing the need for the manual trade–off weight α. Consequently, for an original problem with N objectives, the optimisation transforms into a $2N$–objective problem:

$$\min_{x \in \mathcal{X}} \left(\mathrm{CVaR}_{\theta_1}(x), \ldots, \mathrm{CVaR}_{\theta_N}(x),\ f_1^{(2)}(x), \ldots, f_N^{(2)}(x)\right). \tag{B.2}$$

The split acquisition formulation was evaluated on two benchmark problems: MDTP3 and the noisy SinLinForrester case. The framework demonstrates the anticipated behavior when the variability tolerance is stringent. Specifically, with $\mathbf{p_V} = 0.1$ on MDTP3 (Fig. B.15(a)) and $\mathbf{p_V} = 1$ on SinLinForrester (Fig. B.16(a)), the sampler concentrates on regions satisfying the prescribed variability constraints while simultaneously exploiting high–quality areas of the design space.

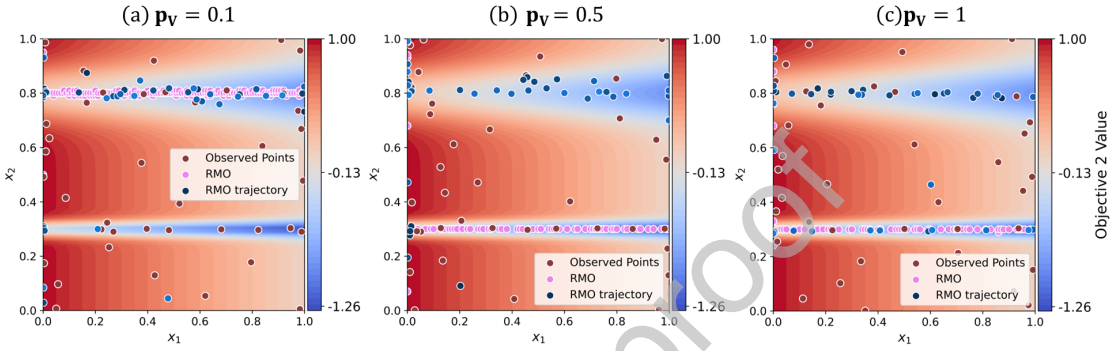

Figure B.15: Distribution of sampled points in the MDTP3 search under different values of $\mathbf{p_V}$. Red circles denote points proposed by the MO loop, blue circles (darker shades indicate later iterations) denote points proposed by the RMO loop, and pink circles indicate the robust Pareto front corresponding to the specific uncertainty threshold $\mathbf{p_V}$.

However, moderate increases in $\mathbf{p_V}$ result in less discernible changes in sampling behavior. As observed with $\mathbf{p_V} = 0.5$ on MDTP3 (Fig. B.15(b)) and $\mathbf{p_V} = 5$ on SinLinForrester (Fig. B.16(b)), the sampling locations remain similar to those obtained with tighter tolerances. Significant shifts in sampling behaviour–specifically, the allocation of points into higher–uncertainty bands—are observed only when $\mathbf{p_V}$ is substantially increased (e.g., $\mathbf{p_V} = 1$ on MDTP3 as shown in Fig. B.15(c), and $\mathbf{p_V} = 10$ on SinLinForrester as shown in Fig. B.16(c)).

This phenomenon is attributed to the problem–dependent uncertainty landscape. For instance, in the SinLinForrester case, the maximum ground–truth standard deviation is approximately 5 (see Fig. B.17). Consequently, setting $\mathbf{p_V} = 5$ results in the normalisation of the term

$$\mathrm{unc}(x) = \frac{\ln(1 + \sigma(x))}{\ln(1 + \mathbf{p_V})}$$

to the interval $[0, 1]$. This functions similarly to a standard min–max normalisation, causing the acquisition function to retain a preference for lower–uncertainty regions. Conversely, increasing $\mathbf{p_V}$ to 10 results in a denominator that exceeds the maximum realisable data value. This effectively compresses the typical uncertainty scores toward zero, thereby diminishing the relative influence of the uncertainty objective.

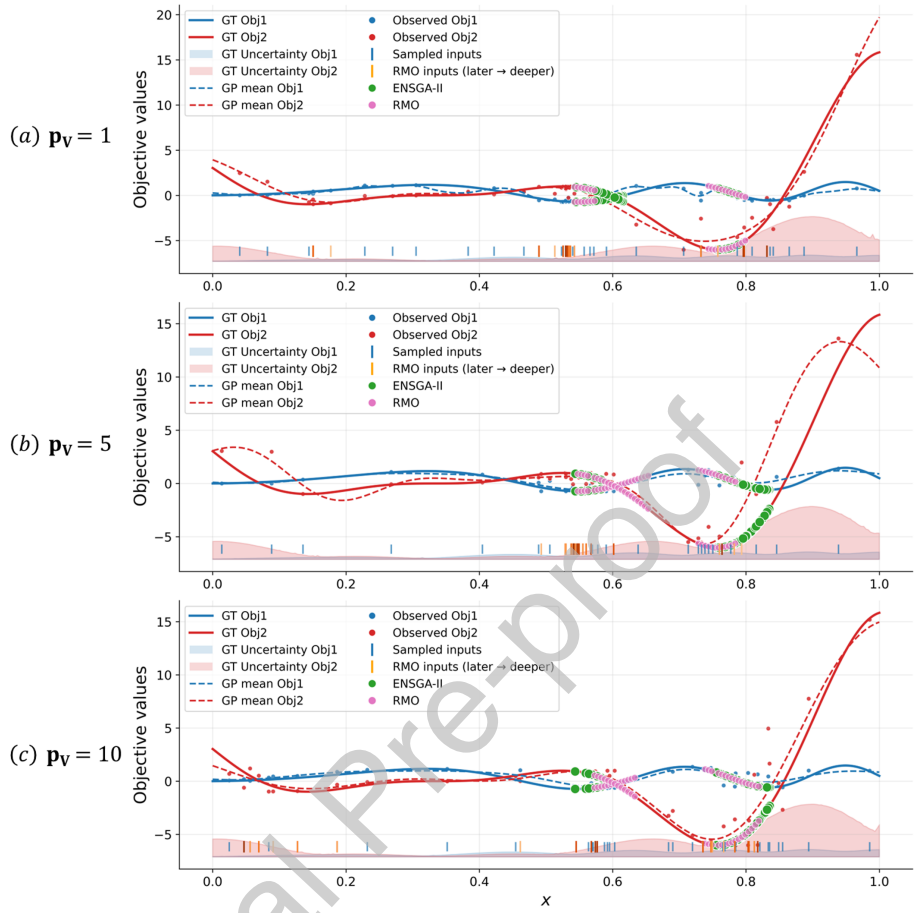

Figure B.16: One–dimensional, two–objective example under input noise. Solid curves represent the ground–truth objectives; dashed curves indicate GP posterior means; shaded bands show the ground–truth uncertainty (standard deviation). Dots represent observed data collected at nominal x with noisy outputs. Short vertical ticks on the x–axis mark sampled input locations: blue ticks correspond to the MO loop and orange ticks to the RMO loop (deeper color intensity indicates later iterations). Green and pink markers denote ENSGA–II and RMO solutions, respectively. (a) $\mathbf{p_V} = 1$, (b) $\mathbf{p_V} = 5$, (c) $\mathbf{p_V} = 10$.

As a result, the subsequent EHVI calculation penalises moderate uncertainty less severely, facilitating selection in higher–uncertainty regions.

In summary, while the split formulation is feasible and eliminates the need for manual tuning of α, effective practical implementation suggests setting $\mathbf{p_V}$ slightly above the expected system variability (approximately 1–2×) to ensure the uncertainty objective remains sufficiently active.

A critical consideration for this approach is the associated computational cost. Decomposing each objective into separate CVaR and unc components doubles the

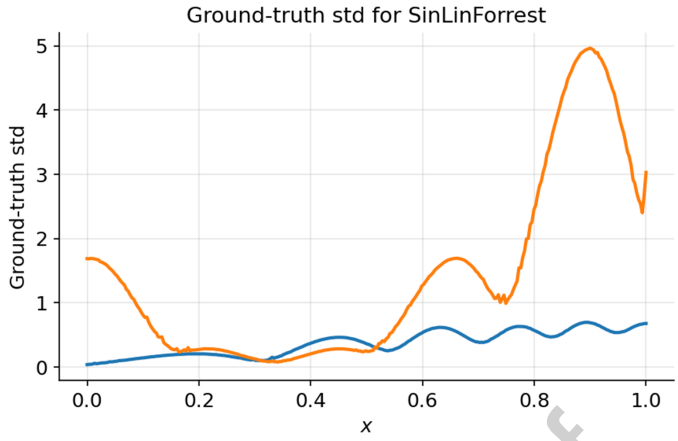

Figure B.17: Ground–truth input–noise profiles for the SinLinForrester benchmark. The orange curve shows the standard deviation of the disturbed objective, which varies with x, while the blue curve corresponds to the lower–variance objective.

dimensionality of the objective space passed to the HV computation (from d to $2d$). Exact HV calculation algorithms are highly sensitive to dimension: complexity typically scales from $O(n \log n)$ or $O(n^2)$ for $d = 2, 3$ to polynomial time $O(n^{f(d)})$ for $d \geq 4$, where the exponent grows with d. In the conducted experiments, this dimensionality shift increased the runtime from approximately 5 minutes to 23 minutes. This overhead is an inherent trade–off for making the uncertainty trade–off explicit; however, it may be mitigated in practice through candidate pre–filtering or the use of approximate HV algorithms.

Bibliography

[1] J. L. J. Pereira, G. A. Oliver, M. B. Francisco, S. S. Cunha Jr, G. F. Gomes, A review of multi-objective optimization: methods and algorithms in mechanical engineering problems, Archives of Computational Methods in Engineering 29 (2022) 2285–2308.

[2] I. Rahimi, A. H. Gandomi, F. Chen, E. Mezura-Montes, A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization, Archives of Computational Methods in Engineering 30 (2023) 2181–2209.

[3] B. Y. Qu, P. N. Suganthan, Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods, Engineering Optimization 43 (2011) 403–416.

[4] K. Deb, Multi-objective optimisation using evolutionary algorithms: an introduction, in: Multi-objective evolutionary optimisation for product design and manufacturing, Springer, 2011, pp. 3–34.

[5] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the strength pareto evolutionary algorithm, TIK report 103 (2001).

[6] J. Luo, F. Liu, Performance impact of manufacturing tolerances for a turbine blade using second order sensitivities, in: Turbo Expo: Power for Land, Sea, and Air, volume 51029, American Society of Mechanical Engineers, 2018, p. V02DT46A010.

[7] M. A. Elfarra, M. N. Önel, G. Yalin, S. Malik, Impact of production tolerances on turbine performance: A cfd based monte-carlo simulations study, Alexandria Engineering Journal 117 (2025) 289–300.

[8] S. Sudhindra, F. Kargar, A. A. Balandin, Noncured graphene thermal interface materials for high-power electronics: Minimizing the thermal contact resistance, Nanomaterials 11 (2021) 1699.

[9] J. Ide, A. Schöbel, Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts, OR spectrum 38 (2016) 235–271.

[10] J. R. Birge, F. Louveaux, Introduction to stochastic programming, Springer, 1997.

[11] S. Mandal, A. Shiuly, D. Sau, A. K. Mondal, K. Sarkar, Study on the use of different machine learning techniques for prediction of concrete properties from their mixture proportions with their deterministic and robust optimisation, AI in Civil Engineering 3 (2024) 7.

[12] S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, E. Bakshy, Robust multi-objective bayesian optimization under input noise, in: International Conference on Machine Learning, PMLR, 2022, pp. 4831–4866.

[13] H. Cheng, Z. Zhang, et al., Aerodynamic robustness optimization of aeroengine fan performance based on an interpretable dynamic machine learning method, Reliability Engineering & System Safety 254 (2025) 110654.

[14] W. Hu, S. Cheng, J. Yan, J. Cheng, X. Peng, H. Cho, I. Lee, Reliability-based design optimization: a state-of-the-art review of its methodologies, applications, and challenges, Structural and Multidisciplinary Optimization 67 (2024) 168.

[15] D. Meng, H. Yang, S. Yang, Y. Zhang, A. M. De Jesus, J. Correia, T. Fazeres-Ferradosa, W. Macek, R. Branco, S.-P. Zhu, Kriging-assisted hybrid reliability design and optimization of offshore wind turbine support structure based on a portfolio allocation strategy, Ocean Engineering 295 (2024) 116842.

[16] J. Song, Y. Cui, P. Wei, M. A. Valdebenito, W. Zhang, Constrained bayesian optimization algorithms for estimating design points in structural reliability analysis, Reliability Engineering & System Safety 241 (2024) 109613.

[17] R. d. S. Motta, S. M. Afonso, An efficient procedure for structural reliability-based robust design optimization, Structural and Multidisciplinary Optimization 54 (2016) 511–530.

[18] M. Yang, H. Zhang, D. Zhang, X. Han, Q. Li, Time-variant reliability-based robust optimization for structures with material degradation, Computer Methods in Applied Mechanics and Engineering 432 (2024) 117337.

[19] S. Sarangi, K. K. Bodla, S. V. Garimella, J. Y. Murthy, Manifold microchannel heat sink design using optimization under uncertainty, International Journal of Heat and Mass Transfer 69 (2014) 92–105.

[20] D. Fernández-Sánchez, E. C. Garrido-Merchán, D. Hernández-Lobato, Improved max-value entropy search for multi-objective bayesian optimization with constraints, arXiv preprint arXiv:2011.01150 (2020).

[21] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (1998) 455–492.

[22] C. E. Rasmussen, Gaussian processes in machine learning, in: Summer school on machine learning, Springer, 2003, pp. 63–71.

[23] Q. Zhou, P. Jiang, X. Huang, F. Zhang, T. Zhou, A multi-objective robust optimization approach based on gaussian process model, Structural and Multidisciplinary Optimization 57 (2018) 213–233.

[24] J. Qing, I. Couckuyt, T. Dhaene, A robust multi-objective bayesian optimization framework considering input uncertainty, Journal of Global Optimization 86 (2023) 693–711.

[25] Z. Abulawi, R. Hu, P. Balaprakash, Y. Liu, Bayesian optimized deep ensemble for uncertainty quantification of deep neural networks: a system safety case study on sodium fast reactor thermal stratification modeling, Reliability Engineering & System Safety (2025) 111353.

[26] A. Kudva, W.-T. Tang, J. A. Paulson, Robust bayesian optimization for flexibility analysis of expensive simulation-based models with rigorous uncertainty bounds, Computers & Chemical Engineering 181 (2024) 108515.

[27] Z. Li, K. Fu, B. Bevans, P. Rao, R. Wang, A. Carrington, A. Riensche, C. Barrett, S. Halliday, M. Kim, Scalable and robust bayesian transfer learning for in-situ qualification in laser powder bed fusion additive manufacturing, Authorea Preprints (2025).

[28] R. Oliveira, L. Ott, F. Ramos, Bayesian optimisation under uncertain inputs, in: The 22nd international conference on artificial intelligence and statistics, PMLR, 2019, pp. 1177–1184.

[29] Y. Inatsu, S. Takeno, H. Hanada, K. Iwata, I. Takeuchi, Bounding box-based multi-objective bayesian optimization of risk measures under input uncertainty, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2024, pp. 4564–4572.

[30] J. Goh, M. Sim, Distributionally robust optimization and its tractable approximations, Operations research 58 (2010) 902–917.

[31] D. Bertsimas, D. B. Brown, C. Caramanis, Theory and applications of robust optimization, SIAM review 53 (2011) 464–501.

[32] R. T. Rockafellar, S. Uryasev, et al., Optimization of conditional value-at-risk, Journal of risk 2 (2000) 21–42.

[33] X. Chen, M. Sim, P. Sun, A robust optimization perspective on stochastic programming, Operations research 55 (2007) 1058–1071.

[34] S. Kang, K. Li, R. Wang, A survey on pareto front learning for multi-objective optimization, Journal of Membrane Computing (2024) 1–7.

[35] T. M. Deist, M. Grewal, F. J. Dankers, T. Alderliesten, P. A. Bosman, Multi-objective learning using hv maximization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2023, pp. 103–117.

[36] G. Su, L. Peng, L. Hu, A gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis, Structural Safety 68 (2017) 97–109.

[37] K. Zhang, A. Guliani, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, P. Beckman, Machine learning-based temperature prediction for runtime thermal management across system components, IEEE Transactions on parallel and distributed systems 29 (2017) 405–419.

[38] J. Wilson, F. Hutter, M. Deisenroth, Maximizing acquisition functions for bayesian optimization, Advances in neural information processing systems 31 (2018).

[39] W. Gan, Z. Ji, Y. Liang, Acquisition functions in bayesian optimization, in: 2021 2nd international conference on big data & artificial intelligence & software engineering (ICBASE), IEEE, 2021, pp. 129–135.

[40] S. Daulton, M. Balandat, E. Bakshy, Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization, Advances in Neural Information Processing Systems 33 (2020) 9851–9864.

[41] K. Yang, M. Emmerich, A. Deutz, T. Bäck, Efficient computation of expected hypervolume improvement using box decomposition algorithms, Journal of Global Optimization 75 (2019) 3–34.

[42] J. Qing, T. Dhaene, I. Couckuyt, Spectral representation of robustness measures for optimization under input uncertainty, in: 38th International Conference on Machine Learning (ICML), 2022.

[43] L. Yang, J. Lyu, W. Lyu, Z. Chen, Efficient robust bayesian optimization for arbitrary uncertain inputs, Advances in Neural Information Processing Systems 36 (2023) 43156–43176.

[44] Y. Chow, A. Tamar, S. Mannor, M. Pavone, Risk-sensitive and robust decision-making: a cvar optimization approach, Advances in neural information processing systems 28 (2015).

[45] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, E. Bakshy, Botorch: A framework for efficient monte-carlo bayesian optimization, Advances in neural information processing systems 33 (2020) 21524–21538.

[46] H. Ishibuchi, H. Masuda, Y. Tanigaki, Y. Nojima, Modified distance calculation in generational distance and inverted generational distance, in: International conference on evolutionary multi-criterion optimization, Springer, 2015, pp. 110–125.

[47] A. Verma, T. Saikia, P. Saikia, D. Rakshit, C. E. Ugalde-Loo, Thermal performance analysis and experimental verification of lithium-ion batteries for electric vehicle applications through optimized inclined mini-channels, Applied Energy 335 (2023) 120743.

[48] W. Li, M. Xiao, X. Peng, A. Garg, L. Gao, A surrogate thermal modeling and parametric optimization of battery pack with air cooling for evs, Applied Thermal Engineering 147 (2019) 90–100.