A New Acquisition Function for Multi-objective Bayesian Optimization: Correlated Probability of Improvement

Kaifeng Yang[1], Kai Chen[2], Michael Affenzeller[1], Bernhard Werth[1],
[1]Heuristic and Evolutionary Algorithms Laboratory
University of Applied Sciences Upper Austria, Hagenberg, Austria
[2]School of Mathematics and Statistics
Central South University, Changsha, China
Kaifeng.Yang@fh-hagenberg.at

ABSTRACT

Multi-objective Bayesian optimization is a sequential optimization strategy in which an optimizer searches for optimal solutions by maximizing an acquisition function. Most existing acquisition functions assume that objectives are independent, but none of them incorporates the correlations among objectives through an explicit formula for exact computation. This paper proposes a novel acquisition function, namely, correlated probability of improvement (cPoI), for bi-objective optimization problems. The cPoI method builds on the probability of improvement and addresses the correlations between objectives by utilizing 3 distinct approaches to compute the posterior covariance matrix from a multi-task Gaussian process. This paper presents both an explicit formula for exact computation of cPoI and a Monte Carlo method for approximating it. We evaluate the performance of the proposed cPoI against 4 state-of-the-art multi-objective optimization algorithms on 8 artificial benchmarks and 1 real-world problem. Our experimental results demonstrate the effectiveness of cPoI in achieving superior optimization performance.

CCS CONCEPTS

• **Computing methodologies** → **Gaussian processes**; • **Applied computing** → **Multi-criterion optimization and decision-making**; • **Mathematics of computing** → **Multivariate statistics**.

KEYWORDS

Multi-objective Bayesian Optimization, Multi-task Gaussian Process, Posterior Covariance, Probability of Improvement

ACM Reference Format:
Kaifeng Yang[1], Kai Chen[2], Michael Affenzeller[1], Bernhard Werth[1], . 2023. A New Acquisition Function for Multi-objective Bayesian Optimization: Correlated Probability of Improvement. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583133.3596374

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3596374

1 INTRODUCTION

Multi-objective optimization is a branch of optimization that involves simultaneous optimization of multiple objectives that frequently conflict with each other. In many real-world applications, due to the high convergence time of simulation models, the evaluation of the objective functions is computationally expensive. This leads to a lack of efficiency of the evolutionary multi-objective optimization algorithms. The surrogate modeling technique is a widely adopted and straightforward solution for addressing computationally expensive evaluations. In this technique, expensive simulation models are replaced with surrogate models constructed based on previous evaluations of the objectives. These surrogate models are trained using a limited set of function evaluations and used to guide the search for optima within the search space.

Bayesian Optimization (BO), which was initially proposed by Jonas Mockus and Antanas Žilinskas [37, 62] and is more widely known as Efficient Global Optimization (EGO) in [30]. Usually, BO utilizes a Gaussian process (GP) as the surrogate for modelling the objective function. Usually, multi-objective Bayesian optimization (MOBO) is generalized from BO by building up GPs for each objective. Recently, Chugh proposed a mono-surrogate approach where a single surrogate model is built after aggregating the objectives functions [8]. Both BO and MOBO sequentially select a promising solution based on posterior distribution derived from GP(s) in a search space.

To assess the potential of a solution or direct the search towards optimal solutions, an acquisition function (a.k.a, infill criteria) maps a solution to a score representing its potential to improve the overall Pareto front. However, most current acquisition functions assume that objectives are independent, and there is no explicit formula for exact computation that considers the correlations among objectives. Recently, multiple-point expected hypervolume improvement (q-EHVI) is proposed in [11] and is computed by an approximation method that does not require assumptions of independence between objectives. Therefore, q-EHVI can incorporate correlations among objectives by utilizing a multi-task Gaussian Process (see [11, 34] for details). More recently, the expected hypervolume improvement (EHVI) is efficiently approximated using Gauss-Hermite quadrature that can take correlated relationships between objectives into account. However, there is no explicit formula to exactly compute q-EHVI or EHVI that takes objectives correlation into account [34]. Despite the increased computational efficiency, the optimization results using the approximation methods have been demonstrated to be not competitive performance compared to exact computation methods [9, 52]. Multiple-point probability of improvement

(q-PoI) [52] has an explicit formula, but it only incorporates the correlations between/among multiple points for a specific objective instead of correlations among objectives.

In many real-world multi-objective optimization problems, the objectives are highly correlated. The common strategy of simply building up the independent GP model for each objective ignores correlations among objectives and can lead to suboptimal solutions. Recently, the multi-task Gaussian process (MTGP) has attracted more and more interest in both academic and application domains because MTGP models can capture the correlation between tasks/objectives. As a popular framework of MTGP, the linear model of coregionalization (LMC) uses a shared covariance structure to capture the dependencies between tasks/objectives and learns task-specific functions using the training data. In the field of MTGP, a fundamental challenge is to effectively leverage task correlations in order to facilitate the tasks to mutually benefit from each other, thereby resulting in improved prediction accuracy. There have been various advances for MTGPs since the introduction of the "free-form" LMC [3], including GP regression network (GPRN) [51] combining the structural Bayesian neural networks with the nonparametric flexibility of GP, considering quasi-shared or non-shared covariance structures between tasks [6, 36, 45], and scalable MTGPs for big data [1, 60]. Due to the incorporation of task correlation, the MTGP model can be used for tasks such as multi-objective optimization, where the objective functions are treated as different tasks to be jointly optimized. Recently, MOEA/D-ASS proposed in [49] utilizes MTGP as the surrogate model and a so-called *adaptive lower confidence bound* (ALCB) as the acquisition function for each objective to identify the most promising subproblems for further modeling. However, the posterior covariance between different objectives from the posterior covariance matrix from MTGP is not incorporated in ALCB. Other related works either approximate the acquisition function that incorporates the correlation among objectives (e.g., in [34, 42]) or ignore the covariance among objectives by only using the diagonal elements of the covariance matrix (e.g., in [33, 43, 49]).

The main contribution of this paper is to propose a new acquisition function, namely, correlated probability of improvement (cPoI) based on posterior multivariate distribution derived from MTGP, and provides an explicit formula of the proposed cPoI for multi-objective Bayesian optimization, which incorporates the correlation between objectives. For quantifying posterior covariance between objectives, two methods are extended from the existing works.

This paper is structured as follows: Sec. 2 introduces the multiple-task Gaussian process and multi-objective Bayesian optimization; Sec. 3 describes the methods to compute the posterior covariance between different tasks in MTGP; Sec. 4 describes the preliminaries to derive the explicit formula of the cPoI and the formula to exact compute cPoI; Sec. 5 describes the experimental settings and discusses the experiment results.

2 MULTI-OBJECTIVE BAYESIAN OPTIMIZATION

Multi-objective optimization can be mathematically formulated as follows:

$$\arg\min_{\mathbf{x}} \quad \mathbf{f}(\mathbf{x}) := \left(f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})\right)^\top, \text{for } \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$$

Here, f_i denotes the i-th objective function $f_i : \mathcal{X} \to \mathbb{R}$, $i \in \mathbb{M} := \{1, \ldots, m\}$, and m is the number of objective functions. The decision vector $\mathbf{x}$ belongs to a subset $\mathcal{X}$ of a continuous search space $\mathbb{R}^d$, where d denotes the dimensionality of the search space. It is worth noting that $\mathcal{X}$ can also be a subset of a discrete alphabet $\mathcal{X} \subseteq \{0,1\}^d$ or a mixed space $\mathcal{X} \subseteq \mathbb{R}^{d_1} \times \mathbb{N}^{d_2} \times \{0,1\}^{d_3}$, with the appropriate choice of distance metric [58] or encoding strategy [22]. In this paper, d denotes the dimension of the search space $\mathcal{X}$, and the search space is restricted to $\mathbb{R}^d$. Constraints are not considered.

2.1 Multi-task Gaussian Process

A multi-task Gaussian process (MTGP) is a probabilistic machine learning method that enables the joint modeling of multiple related tasks [2]. In MTGP, the relationship between tasks is modeled using a covariance matrix that characterizes the correlation between the tasks. The covariance structure is typically assumed to be a linear combination of fixed and unknown task-specific covariance functions.

Let $f_i(\mathbf{x})$ be the function representing the i-th task, where $\mathbf{x}$ is the input vector. The MTGP model assumes that each task function is drawn from a Gaussian process with mean 0 and covariance $k_i(\mathbf{x}, \mathbf{x}'; \theta_i)$. Here, θ_i is the set of hyperparameters that control the shape and amplitude of the covariance function for task i. The joint MTGP model over all tasks can be expressed as follows:

$$\mathbf{y} = \begin{bmatrix} f_1(\mathbf{x}_1) \\ \vdots \\ f_m(\mathbf{x}_m) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{bmatrix}, \tag{1}$$

where $\mathbf{y} \sim \mathcal{GP}(0, K_M)$ is the stacked vector of observations for all tasks, $\boldsymbol{\epsilon} = [\epsilon_1, ..., \epsilon_m]^\top$ is the vector of i.i.d Gaussian noise with variance σ_m^2 in task m, and K_M is the kernel of MTGP to capture the relationship between the input vectors and the tasks. When $m = 1$, we obtain the single-task GP (STGP). The coefficient vectors in K_M can be learned from data using the standard maximum likelihood or Bayesian inference [2, 40].

At the heart of MTGP lies the kernel function k_M [2]. Formally, let i and j denote the corresponding task index vectors, where $i, j \in \mathbb{M}$. Then, the multi-task kernel function can be written as:

$$k_M(\mathbf{x}_i, \mathbf{x}_j, i, j) = k_x(\mathbf{x}_i, \mathbf{x}_j) k_T(i, j), \tag{2}$$

where $k_x(\mathbf{x}_i, \mathbf{x}_j)$ and $k_T(i, j)$ are the input and task-specific kernel functions, respectively. Furthermore, the kernel matrix K_M for the MTGP model can be formulated as:

$$K_M = K_x(X_i, X_j) \otimes K_T(i, j) = \begin{bmatrix} K_{1,1} & \cdots & K_{1,m} \\ \vdots & \ddots & \vdots \\ K_{m,1} & \cdots & K_{m,m} \end{bmatrix}, \tag{3}$$

where $K_{i,j}$ is the cross kernel matrix between tasks i and j.

There are different choices for $k_T(i, j)$ and $k_x(\mathbf{x}_i, \mathbf{x}_j)$ in MTGP, depending on the specific modeling assumptions. A popular choice for $k_T(i, j)$ is the linear model of coregionalization (LMC) function [3], which assumes that tasks share a common underlying function with some task-specific differences:

$$k_T(i, j) = \mathbf{w}_i^\top \mathbf{w}_j + b\delta_{i,j}, \tag{4}$$

where $\mathbf{w}_i$ is a vector of weights for task i, b is a bias term, and $\delta_{i,j}$ is the Kronecker delta function [2]. This function can capture varying degrees of correlation between tasks by adjusting the values of the weights and bias term. On the other hand, for $k_x(\mathbf{x}, \mathbf{x}')$, a common choice is a squared exponential (SE) kernel, which assumes that outputs are correlated when inputs are close to each other:

$$k_x(\mathbf{x}, \mathbf{x}') = k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\left(\frac{\mathbf{x} - \mathbf{x}'}{\boldsymbol{\ell}}\right)^2\right), \tag{5}$$

where σ_f^2 is the signal variance, and $\boldsymbol{\ell}$ is the length scale across input dimensions. Other kernel functions such as Matérn, periodic, and linear can also be used for $k_x(\mathbf{x}, \mathbf{x}')$, depending on problem nature and data characteristics. Furthermore, the kernel function can be customized based on specific requirements for better model performance. In this paper, we only employ the LMC framework for MTGP because the LMC has a succinct form, good interpretability, computational efficiency, and extensive applications.

Given a kernel function k_M, the joint distribution of the MTGP can be written as:

$$\Pr(\mathbf{f}(\mathbf{x})) = \mathcal{N}(\mathbf{f}(\mathbf{x}) \mid \mathbf{0}, K_M), \tag{6}$$

where $\mathcal{N}$ denotes the joint multivariate normal distribution of multiple tasks. Inference in MTGP involves finding the posterior distribution of the function values $\hat{\mathbf{f}}(X^*)$ at a test vector $X^* = (\mathbf{x}^{*(1)}, \cdots, \mathbf{x}^{*(m)})$. The posterior distribution $p(\hat{\mathbf{f}}(X^*))$ can be expressed as follows:

$$\Pr(\hat{\mathbf{f}}(X^*)) \mid \mathbf{y}, \mathbf{X}, \Theta) = \mathcal{N}(\hat{\mathbf{f}}(X^*) \mid \boldsymbol{\mu}^*, \Sigma^*), \tag{7}$$

where $\boldsymbol{\mu}^*$, Σ^*, and Θ are the mean vector of $\hat{\mathbf{f}}(X^*)$, covariance matrix of $\hat{\mathbf{f}}(X^*)$, and the hyperparameters.

MTGP is widely applied in machine learning, including prediction in structured data [15], transfer learning [3, 21], and modeling complex interactions between multiple outputs [6, 51].

2.2 Framework of MOBO

Multi-objective Bayesian optimization (MOBO) is an optimization algorithm that seeks to find a set of Pareto-optimal solutions by simultaneously optimizing multiple objective functions. MOBO starts by sampling an initial design of experiment (DoE) with a size of η (line 2 in Algorithm 1) from the search space $\mathcal{X}$. This DoE is typically generated using simple random sampling or Latin Hypercube Sampling [41]. Usually, the surrogate models are constructed for each objective under the assumption of no correlation between different objectives, such as in [52, 57]. In this paper, a MTGP, which is used to estimate the multivariate distribution $\Pr(\hat{f}(\mathbf{x})|X, \mathbf{y})$, is trained by the initial DoE set (X) and its corresponding responses $(\mathbf{y} = \mathbf{f}(\mathbf{x}^{(1)}), \cdots, \mathbf{f}(\mathbf{x}^{(\eta)}) \subseteq \mathbb{R}^{m \times \eta})$.

After constructing a multi-task GP model ($\mathcal{M}$), MOBO iterates through the main loop until a stopping criterion is met. This loop, outlined in Alg. 1 from line 6 to line 12, begins with a search for a decision vector $\mathbf{x}'$ in a search space $\mathcal{X}$ by maximizing the acquisition function $\mathscr{A}$ with parameters γ and MTGP model $\mathcal{M}$ (line 7 in Alg. 1). To search for the optimal decision vector $\mathbf{x}^*$, a single-objective optimization algorithm, such as GA, PSO, ACO, CMA-ES, or gradient-ascent algorithms [55], can be utilized. Following evaluation by the 'true' objective functions $\mathbf{f}(\mathbf{x}^*)$, $\mathbf{x}^*$ and its corresponding objective values $\mathbf{f}(\mathbf{x}^*)$ is used to update the training dataset, and

Algorithm 1: Multi-objective Bayesian Optimization

1 **MOBO**($\mathbf{f}, \mathscr{A}, \mathcal{X}, \mathcal{M}, \gamma, \eta, T_c$)
2 **Output:** $\mathcal{PF}$
/* $\mathbf{f}$: objective functions, $\mathscr{A}$: acquisition function, $\mathcal{X}$: search space, γ: parameters of $\mathscr{A}$, $\mathcal{M}$: a surrogate model to train, T_c: maximum number of function evaluations, $\mathcal{PF}$: Pareto-front approximation set */
3 Generate the initial DoE: $\mathrm{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\eta)}\} \subset \mathcal{X}$;
4 Evaluate $\mathbf{y} \leftarrow \{\mathbf{f}(\mathbf{x}^{(1)}), \ldots, \mathbf{f}(\mathbf{x}^{(\eta)})\}$;
5 Train multi-task GP model $\mathcal{M}$ on $(\mathrm{X}, \mathbf{y})$;
6 $g \leftarrow \eta$;
7 **while** $g < T_c$ **do**
8 $\quad \mathbf{x}^* \leftarrow \arg\max_{\mathbf{x}} \mathscr{A}(\mathbf{x}'; \mathcal{M}, \gamma)$, where $\mathbf{x}' \in \mathcal{X}$;
9 $\quad \mathbf{y}^* \leftarrow \mathbf{f}(\mathbf{x}^*)$;
10 $\quad \mathrm{X} \leftarrow \mathrm{X} \cup \{\mathbf{x}^*\}$;
11 $\quad \mathbf{y} \leftarrow \mathbf{y} \cup \{\mathbf{y}^*\}$;
12 $\quad$ Re-train the MTGP model $\mathcal{M}$;
13 $\quad g \leftarrow g + 1$;
14 $\mathcal{PF} \leftarrow$ Non-dominated($\mathbf{y}$)

then the MTGP rebuilds the surrogate model $\mathcal{M}$. *Non-dominated* ($\mathbf{y}$) at line 15 represents the function of generating the Pareto front for a set $\mathbf{y}$.

3 POSTERIOR COVARIANCE BETWEEN OBJECTIVES IN MTGPS

In this section, we introduce a posterior covariance between objectives to measure the strength of their posterior correlation. Inspired by the posterior covariance between kernels [7], a function $\boldsymbol{f}$ obtained from a MTGP using the kernel $K_{\text{M}} = B \otimes k_{\text{SE}}$ follows the form $\boldsymbol{f} \sim \mathcal{GP}(0, K_{\text{M}})$. To simplify notation, we refer to the function values at training location $\mathbf{x}$ as f and the function values at query location $\mathbf{x}^*$ as $\hat{f}$. By employing the formula of Gaussian conditionals, we can obtain the conditional distribution of a GP-distributed function $\hat{f}$ conditioned on $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$:

$$\hat{f}|\boldsymbol{f}_{i+j} \sim \mathcal{N}\left(\mathbf{k}_{\text{M}}^{*\top} K_{\text{M}}^{-1} \boldsymbol{f}_{i+j}, k_{\text{M}}^{**} - \mathbf{k}_{\text{M}}^{*\top} K_{\text{M}}^{-1} \mathbf{k}_{\text{M}}^{*}\right), \tag{8}$$

where the subscript labels tasks, $\mathbf{k}_{\text{M}}^{*}$ denotes vector covariance between $\mathbf{x}^*$ and X, and $\boldsymbol{f}_{i+j} = [\boldsymbol{f}_i, \boldsymbol{f}_j]^\top$. The Gaussian conditionals reflect the posterior uncertainty of the model regarding the distinct tasks of the signal, while simultaneously integrating the potential configurations of the remaining tasks. This approach enables a comprehensive understanding of the underlying relationships and dependencies between the different tasks within the model. In addition, the posterior variances conditioned on a single task [5] can be formulated as follows:

$$\hat{f}_i|\boldsymbol{f}_i \sim \mathcal{N}\left(\mathbf{k}_{\text{SE}}^{*\top} K_{\text{SE}}^{-1} \boldsymbol{f}_i, k_{\text{SE}}^{**} - \mathbf{k}_{\text{SE}}^{*\top} K_{\text{SE}}^{-1} \mathbf{k}_{\text{SE}}^{*}\right). \tag{9}$$

Leveraging the above equations, we can derive the posterior covariance between the random variables of any two tasks, conditioned

on their sum:

$$\mathrm{Cov}\left(\hat{f}_i, \hat{f}_j | \boldsymbol{f}_{i+j}\right) = -\mathbf{k}_{\mathrm{M},i}^{*\top} K_{\mathrm{M}}^{-1} \mathbf{k}_{\mathrm{M},j}^{*}. \tag{10}$$

We use $\mathrm{Cov}(\hat{f}_i, \hat{f}_j)$ instead of $\mathrm{Cov}\left(\hat{f}_i, \hat{f}_j | \boldsymbol{f}_{i+j}\right)$ to simplify the notation. For MTGPs, the posterior covariance indicates the possible presence of statistical dependencies between random variables of different tasks and provides a principled justification for modeling such dependencies within a kernel.

3.1 Diversified Covariance Between Objectives

In this paper, we extend the two methods [5, 7], which quantify the covariance between different kernels in MTGP, into quantifying the posterior covariance between different tasks/objectives (see V1 and V2 below). Besides, a third intuitive method is to use the LCM function, described in Eq. (4). Specifically, these three methods are:

- V1: Employing Eq. (10) that explicitly encodes the correlation between tasks in k_M.
- V2: Employing $\mathrm{Cov}\left(\hat{f}_i, \hat{f}_j | \boldsymbol{f}_{i+j}\right) = -\mathbf{k}_{\mathrm{SE},i}^{*\top} (K_{\mathrm{SE},i} + K_{\mathrm{SE},j})^{-1} \mathbf{k}_{\mathrm{SE},j}^{*}$ that implicitly describes the latent correlation between random variables of any two tasks i and j.
- V3: Employing the non-diagonal element of task correlation matrix k_T in Eq. (4) instead of Eq. (10), considering the fact that the final values (posterior) of k_T are inferred by the optimization on training data.

It is important to note that these three versions are general for cases where $m \geq 2$. This is because they measure the correlation between any two objectives, irrespective of the number of objectives considered. In this paper, we utilize them to measure the underlying correlation between objectives.

3.2 Discussion of Diversified Covariance

To guide the choice of covariance between tasks, we discuss characteristics of the aforementioned computation methods of covariance. The LMC assumes that the correlation between tasks can be explained by a set of latent functions, represented as linear combinations of the observed variables. The third covariance computation method (V3) described in LMC has several shortcomings and challenges. Some of these are:

- It assumes that the correlation between tasks remains constant over space. However, in many cases, the correlation may vary across regions or over time, leading to dynamic correlations between tasks [2].
- The kernel and random variables associated with each task are not relevant in this context, as $\mathbf{w}_i$ in Eq. (4) remains independent of $\mathbf{x}_i$.
- It lacks robustness and is sensitive to outliers, missing data, and task errors, leading to unreliable results.

In contrast, the proposed posterior covariance (V1) between tasks holds multiple advantages:

- Observed from Eq. (10), the posterior covariance can vary across locations of random variables, leading to dynamic correlations between tasks.
- The posterior covariance is determined by each task's kernel of random variables.
- The posterior covariance is much more robust.

4 CORRELATED PROBABILITY OF IMPROVEMENT

This section mainly describes the preliminaries and the explicit formula of the proposed acquisition function, namely, correlated probability of improvement (cPoI).

4.1 Preliminaries

In multi-objective optimization, a fundamental concept is *Pareto dominance*, or briefly *dominance*, representing an ordering relationship among a set of potential solutions. Specifically, it is a mechanism for comparing solutions, wherein one solution is said to dominate another if it performs better in at least one objective function and is not worse in any other objective function. *Dominance* is defined as follows:

Definition 4.1 (Dominance $-\prec$ [10]). Given two decision vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathbb{R}^d$ and their corresponding objective values $\mathbf{y}^{(1)} = \mathbf{f}(\mathbf{x}^{(1)})$, $\mathbf{y}^{(2)} = \mathbf{f}(\mathbf{x}^{(2)})$ in a minimization problem, it is said that $\mathbf{y}^{(1)}$ dominates $\mathbf{y}^{(2)}$, being represented by $\mathbf{y}^{(1)} \prec \mathbf{y}^{(2)}$, iff $\forall i \in \mathbb{M} : f_i(\mathbf{x}^{(1)}) \leq f_i(\mathbf{x}^{(2)})$ and $\exists j \in \mathbb{M} : f_j(\mathbf{x}^{(1)}) < f_j(\mathbf{x}^{(2)})$.

A Pareto-front approximation set $\mathcal{PF}$ is a collection of non-dominated solutions that approximate the Pareto front. A non-dominate space represents the objective space where any arbitrary solution $\mathbf{f}(\mathbf{x}') \prec \mathcal{PF}$, and it is expressed as:

Definition 4.2 (Non-Dominated Space of a Set [54]). Let $\mathcal{PF}$ be a subset of $\mathbb{R}^m$ and let a reference point $\mathbf{r} \in \mathbb{R}^m$ be such that $\forall \mathbf{p} \in \mathcal{PF} : \mathbf{p} \prec \mathbf{r}$. The non-dominated space of $\mathcal{PF}$ with respect to $\mathbf{r}$, denoted as $\mathrm{ndom}(\mathcal{PF})$, is then defined as:

$$\mathrm{ndom}(\mathcal{PF}, \mathbf{r}) := \{\mathbf{y} \in \mathbb{R}^m \mid \mathbf{y} \prec \mathbf{r} \text{ and } \nexists \mathbf{p} \in \mathcal{PF} \ \ s.t. \ \ \mathbf{p} \prec \mathbf{y}\} \tag{11}$$

The *Hypervolume Indicator*, introduced in [63], is one of the most important unary indicators for evaluating the quality of a Pareto-front approximation. Its theoretical properties are discussed in [64]. An important advantage of the HV is that it does not require a true Pareto front to be known a priori, and the maximization of HV promotes the discovery of diverse and high-quality approximation sets of the Pareto front. The *Hypervolume Indicator* is defined as follows:

Definition 4.3 (Hypervolume Indicator). Given a finite approximation to a Pareto front, say $\mathcal{PF} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\} \subset \mathbb{R}^d$, the *Hypervolume Indicator* (HV) of $\mathcal{PF}$ is defined as a d-dimensional Lebesgue measure of the subspace, dominated by $\mathcal{PF}$ and bounded below by a reference point $\mathbf{r}$:

$$\mathrm{HV}(\mathcal{PF}) = \lambda_d(\cup_{\mathbf{y} \in \mathcal{PF}}[\mathbf{r}, \mathbf{y}]) \tag{12}$$

with λ_d being the Lebesgue measure on $\mathbb{R}^d$.

The hypervolume (HV) measures the size of the dominated subspace bounded below by a reference point $\mathbf{r}$. Selecting a reference point $\mathbf{r}$ dominated by all feasible solutions is important. The element in the reference point ($\mathbf{r}$) must be a finite value to avoid infinite improvements along a specific axis, which, otherwise, would lead to biased results.

Probability of improvement (PoI) [12, 17, 29, 61] assesses the probability of a posterior distribution dominating a Pareto-front approximation set ($\mathcal{PF}$). Compared with greedy EHVI, PoI is more explorative and assesses a joint distribution over the entire non-dominated space.

Definition 4.4 (Probability of Improvement [18, 61]*).* Given the predictions $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_m)^\top$, where $\hat{y}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)|_{i\in\mathbb{M}}$, with the parameters of the multivariate predictive distribution $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_m)^\top$, $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_m)^\top$ and the Pareto-front approximation set $\mathcal{PF}$, the *Probability of Improvement* is defined as:

$$\text{PoI}\ (\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{PF}) := \int_{\mathbb{R}^m} \text{I}(\hat{\mathbf{y}} \prec \mathcal{PF}) \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\hat{\mathbf{y}}) d\hat{\mathbf{y}} \text{ and } \text{I}(v) = \begin{cases} 1, & v = \text{true} \\ 0, & v = \text{false} \end{cases} \tag{13}$$

where $\xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is the multivariate *independent* normal distribution with the parameters of $\boldsymbol{\mu} \in \mathbb{R}^m$ and $\boldsymbol{\sigma} \in \mathbb{R}_+^m$.

4.2 Definition of cPoI

The correlated probability of improvement (cPoI) is a generalization of PoI, whereas the correlation of the joint distribution for each objective is considered. The cPoI uses a correlation matrix, as described in Sec. 3, to model the relationships among the objectives. The cPoI is then defined as:

Definition 4.5 (Correlated Probability of Improvement). Given the predictions $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_m)^\top$, where $\hat{\mathbf{y}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, with the parameters of the multivariate predictive distribution $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_m)^\top$, Σ is the covariance matrix of $\hat{\mathbf{y}}$, where $\Sigma_{i,j} = \text{Cov}(\hat{y}_i, \hat{y}_j)$ and the Pareto-front approximation set $\mathcal{PF}$, the *correlated Probability of Improvement* (cPoI) is defined as:

$$\text{cPoI}\ (\boldsymbol{\mu}, \Sigma, \mathcal{PF}) := \int_{\mathbb{R}^m} \text{I}(\hat{\mathbf{y}} \prec \mathcal{PF}) \xi_{\boldsymbol{\mu},\Sigma}(\hat{\mathbf{y}}) d\hat{\mathbf{y}}, \tag{14}$$

where $\xi_{\boldsymbol{\mu},\Sigma}$ is the multivariate *dependent* normal distribution with the parameters of $\boldsymbol{\mu} \in \mathbb{R}^m$ and Σ.

4.3 Approximation of cPoI

A Monte Carlo method to approximate cPoI using an acceptance-rejection method [28] is illustrated in Algorithm 2.

Algorithm 2: Monte Carlo approximation for cPoI

Input: Mean vector $\boldsymbol{\mu}$, covariance matrix Σ, number of samples n_{sample}, Pareto-front approximation set $\mathcal{PF}$, reference point $\mathbf{r}$.
Output: cEHVI

```
count = 0 ;
for i = 1 to n_sample do                    /* Main loop */
    Random generate a vector ỹ = (ỹ_1, ỹ_2) ~ N(μ, Σ) ;
    count = count + I(ỹ ≺ PF) ;
cPoI = count/n_sample ;
```

4.4 Formula Derivation

Based on the definition of cPoI in Eq (14), cPoI computes the integral of a multivariate normal distribution in the non-dominated objective space (a.k.a, $\text{ndom}(\mathcal{PF}, \infty^m)$).

To ensure clarity, we provide an overview of the partitioning technique proposed for bi-objective optimization problems in [19, 54], of which computation complexity holds $\theta(n \log n)$ for $m = 2$, where n represents the number of solutions in $\mathcal{PF}$. Given a Pareto-front approximation set $\mathcal{PF} = \left(\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(n)}\right)$ sorted in descending order by f_2, the set is augmented with two sentinels: $\mathbf{y}^{(0)} = (r_1, -\infty)$ and $\mathbf{y}^{(n+1)} = (-\infty, r_2)$[1]. The non-dominated space of $\mathcal{PF}$ is then represented by the stripes $S^{(i)}$, defined as:

$$\begin{aligned} S^{(i)} :&= \left(\begin{pmatrix} l_1^{(i)} \\ l_2^{(i)} \end{pmatrix}, \begin{pmatrix} u_2^{(i)} \\ u_2^{(i)} \end{pmatrix} \right) \\ &= \left(\begin{pmatrix} y_1^{(i)} \\ -\infty \end{pmatrix}, \begin{pmatrix} y_1^{(i-1)} \\ y_2^{(i)} \end{pmatrix} \right), i = 1, \ldots, n+1 \end{aligned} \tag{15}$$

A multivariate cumulative density function (MCDF) of normal distributions is a prerequisite to computing the cPoI, and it is defined as:

Definition 4.6 (MCDF function). Given a multivariate normal distribution $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with the parameters of a mean vector $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_m)^\top$, a covariance matrix Σ with a size of $m \times m$, and a vector $\mathbf{x} = (x_1, \cdots, x_m)^\top$, the probability of multivariate cumulative density function is defined as:

$$\begin{aligned} &\text{MCDF}(\mathbf{x}, \boldsymbol{\mu}, \Sigma) := \text{P}(\mathbf{X} \le \mathbf{x}) = \text{P}(X_1 \le x_1, \cdots, X_m \le x_m) \\ &= \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_m} \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})} \text{d}x_1 \cdots \text{d}x_m \\ &= \int_{(-\infty,\cdots,-\infty)}^{(x_1,\cdots,x_m)} \xi_{\boldsymbol{\mu},\Sigma}(\mathbf{x}) d\mathbf{x}, \end{aligned} \tag{16}$$

where $|\Sigma|$ denotes the determinant of the covariance matrix.

Remark 1: The present definition of MCDF is intended for general scenarios. This paper's followings apply solely to bivariate normal distributions if there is no special statement.

Remark 2: This is no closed-form formula to compute MCDF defined in Eq. (16). However, various numerical estimation techniques can be utilized to evaluate MCDF. These methods include adaptive quadrature on a transformed t-density for bivariate and trivariate distributions [13, 14, 23], quasi-Monte Carlo integration algorithms for higher dimensional distributions ($m \ge 4$) [24, 25], and other approaches as outlined in [4, 26].

To compute the cumulative probability density for the Cartesian product domain of the form $(a, b] \times (c, d]$, we introduce the notation $\Gamma(a, b, c, d, \boldsymbol{\mu}, \Sigma)$ [52] as defined below.

$$\begin{aligned} &\Gamma(a, b, c, d, \boldsymbol{\mu}, \Sigma) := \int_{(a,c)}^{(b,d)} \xi_{\boldsymbol{\mu},\Sigma}(\mathbf{y}) d\mathbf{y} \\ &= \left(\int_{(-\infty,-\infty)}^{(b,d)} + \int_{(-\infty,-\infty)}^{(a,c)} - \int_{(-\infty,-\infty)}^{(a,d)} - \int_{(-\infty,-\infty)}^{(b,c)} \right) \xi_{\boldsymbol{\mu},\Sigma}(\mathbf{y}) d\mathbf{y} \\ &= \text{MCDF}((b, a), \boldsymbol{\mu}, \Sigma) + \text{MCDF}((a, c), \boldsymbol{\mu}, \Sigma) - \\ &\quad \text{MCDF}((a, d), \boldsymbol{\mu}, \Sigma) - \text{MCDF}((b, c), \boldsymbol{\mu}, \Sigma) \end{aligned} \tag{17}$$

Due to the dependence of the multivariate Gaussian distribution, the product of the normal distribution can not be computed using Fubini's theorem, which is widely applied for acquisition functions

[1] For the computation of PoI and cPoI, $\mathbf{r} = (\infty, \infty)$.

that don't consider the correlations among the objectives. The correlated PoI can be computed by a sum of the double integral of the bi-variate probability density function over all the non-dominated space. The formula of cPoI can then be written as:

$$\begin{aligned}\text{cPoI}\ (\boldsymbol{\mu}, \Sigma, \mathcal{PF}) : &= \int_{\mathbb{R}^m} \mathrm{I}(\hat{\mathbf{y}} \prec \mathcal{PF}) \xi_{\boldsymbol{\mu},\Sigma}(\hat{\mathbf{y}}) d\hat{\mathbf{y}} \\ &= \int_{\hat{\mathbf{y}}=(-\infty,-\infty)}^{\hat{\mathbf{y}}=(\infty,\infty)} \mathrm{I}(\hat{\mathbf{y}} \prec \mathcal{PF}) \xi_{\boldsymbol{\mu},\Sigma}(\hat{\mathbf{y}}) d\hat{\mathbf{y}} \\ &= \sum_{j=1}^{n+1} \int_{(l_1^{(j)}, l_2^{(j)})}^{(u_1^{(j)}, u_2^{(j)})} \xi_{\boldsymbol{\mu},\Sigma}(\hat{\mathbf{y}}) d\hat{\mathbf{y}} \\ &= \sum_{j=1}^{n+1} \Gamma(l_1^{(j)}, u_1^{(j)}, l_2^{(j)}, u_2^{(j)}, \boldsymbol{\mu}, \Sigma) \end{aligned} \tag{18}$$

For a more general case of $m > 2$, Eq. (18) can be derived from the same spirit presented above, while the Cartesian product domain of function $\Gamma(.)$ should be expended to $(lb_1, ub_1] \cdots \times (lb_i, ub_i] \times (lb_m, ub_m]$, where lb_i and ub_i represent the lower and upper bounds of the partitioned hyper box for the i-th objective, respectively.

4.5 Computational Complexity

Assuming the computation of a bi-dimensional cumulative probability density function takes $O(2)$ time units, the computational complexity of cPoI is $O(2 \times 4n \log n) = O(n \log n)$ by using the method of decomposing the non-dominated space in [19]. A practical trick for reducing computation time [52] is to only compute the MCDF(.) in Eq. (18) for a non-dominated space that is covered by $\mu - 3\sigma$ and $\mu + 3\sigma$.

4.6 Accuracy Comparison

Figure 1 shows the accuracy comparison of the Monte Carlo approximation and the exact computational method described in Sec. 4.4 with different correlation coefficient (ρ) between two objectives on a simple Pareto front approximation set. The results show that the MC method yields similar results to that of the exact method with at least $1e4$ samples. However, the MC method requires more samples for a sufficiently accurate value. Additionally, a negative and a positive relationship lead the distribution to be elongated along a line with a negative and positive slope, respectively. Therefore, compared with PoI, when $\boldsymbol{\mu}$ is located in the non-dominated space, a negative ρ increases the cPoI value and a positive ρ decreases the cPoI value.

5 EXPERIMENTS

5.1 Experimental Setup

This section presents a series of experiments on 8 artificial bi-objective test problems and 1 real-world optimization problem. The artificial benchmark is WOSGZ1-8 [48], and real-world optimization problem is a hatch cover design problem (RE) [44]. The search space dimension (d) is 15, and 2 for WOSGZ and RE problems, respectively. The performance metric is hypervolume due to its Pareto-compliant property. The reference point[2] for HV is

[2]The choice of a reference point has no effect on the optimization behavior of PoI and cPoI but affects the optimization behavior of EHVI.

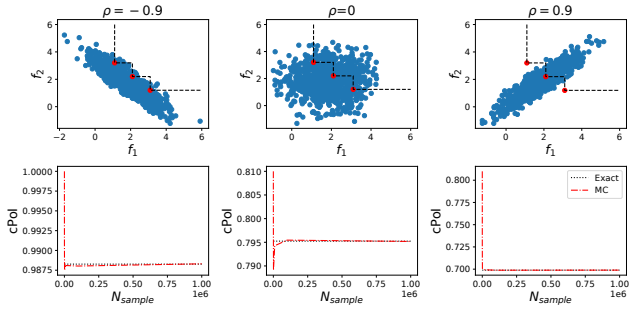

Figure 1: Accuracy comparison between the MC and the exact methods. The first row shows the $\mathcal{PF}$ (red points) and the random samples (with a size of 1,000) based on three different multivariate normal distributions, where $\mathcal{PF} = ((3.1, 1.2)(2.1, 2.2), (1.1, 3.2))$, $\boldsymbol{\mu} = (1.81, 1.82)^\top$, $\boldsymbol{\sigma} = (1, 1)^\top$, $\rho = -0.9, 0, 1$ for the left, middle, and right column, respectively. The second row shows the cPoI values obtained by averaging 10 independent runs of the MC method (red line) using varying sample sizes $N_{sample} \in \{1e1, 1e2, 1e3, 1e4, 1e5, 1e6\}$, and the exact method (black line).

$(1.2, 1.2)$ [48, 52] and $(5885.4870, 5.5063)$ [32] for WOSGZ and RE problems, respectively. The DoE size (η) for all problems is set as $\min(6d, 60)$, and the maximum function evaluation (T_c) is $\eta + 170$. The optimizer is BI-population CMA-ES to maximize the acquisition functions due to the favorable performance on BBOB function testbed [27], with 2,000 iterations and 3 restarts, which is different from the parameter setting of ParEGO in [39]. cPoI-V$i|_{i\in\{1,2,3\}}$ represent different methods to compute the posterior covariance between objectives, as indicated in Sec. 3.1.

The proposed acquisition function cPoI is compared with baseline algorithms that do not consider correlations among objectives. These compared state-of-the-art acquisition functions (algorithms) : PoI [61], EHVI [16], ParEGO [31], MOEA/D-EGO [59]. In all experiments, the Matérn5/2 kernel with default parameter settings [35] is used for STGP and MTGP, where STGP is used for the predictions of mean and variance, and MTGP is only used to compute the posterior covariance for cPoI.

5.2 Empirical Results

Fig. 2 shows the average HV convergence curves of 15 independent runs of all the test algorithms on the WOSGZ test problems. Among all the test algorithms, EHVI converges slower than PoI-based acquisition functions due to the reduction or shrinkage of the entire non-dominated space, $\text{ndom}(\mathcal{PF}, \mathbf{r} = \infty^m)$, which is caused by $\mathbf{r} \neq \infty^m$ being a must set for the EHVI. This evidence is also found in the work [54], where EHVI still underperforms PoI on bi-objective DTLZ2, DTLZ4, and MaF13 problems when $d = 18$, even a large reference point $\mathbf{r}$ is set. A remedy for improving the performance of EHVI is to use a dynamic reference point strategy, that is, to set a relatively large reference at the early stage of iteration and then decrease the reference point when the extreme solutions or nadir points are found. The poor performances of all algorithms

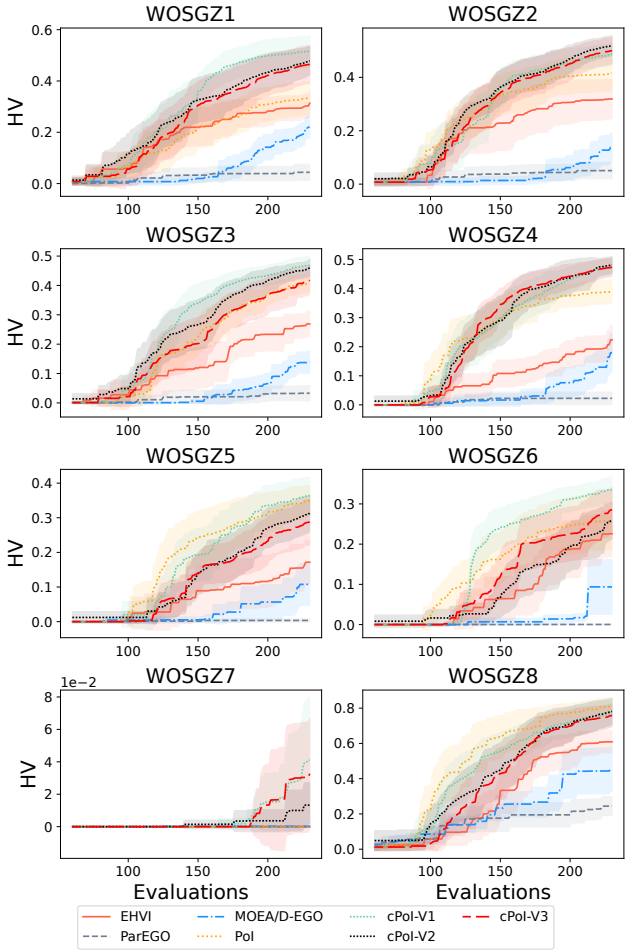

Figure 2: The mean hypervolume of the best-so-far Pareto approximation set $\mathcal{PF}$ plotted against the number of function evaluations. The shaded area represents the 95% confidence interval of the mean, in which 15 independent runs are conducted for each acquisition function.

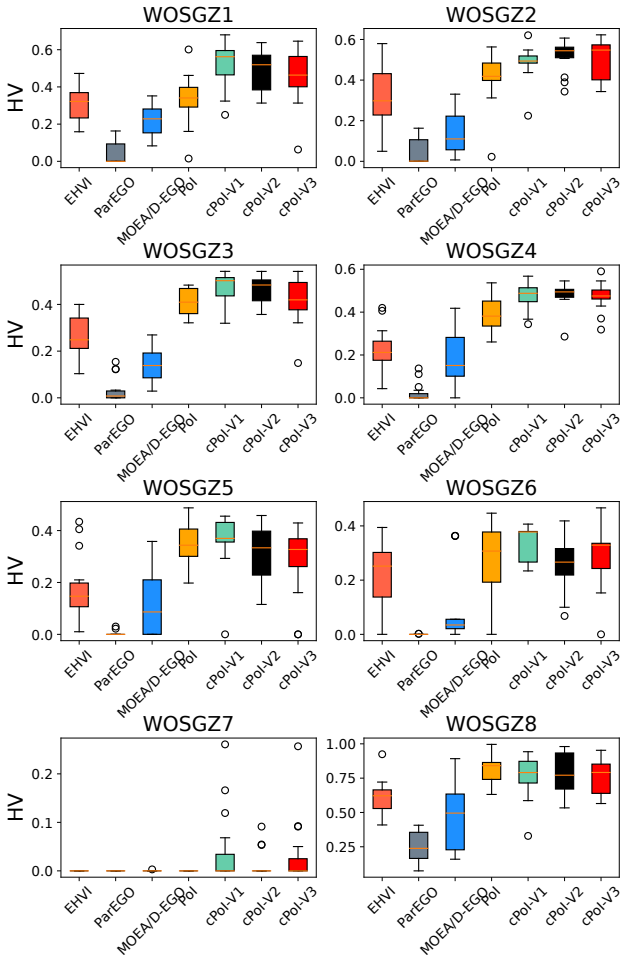

Figure 3: Boxplot comparison on the test problems.

on WOSGZ7 have two reasons: 1). the landscape of WOSGZ7 between the middle and the boundary regions of the Pareto front is more imbalanced [48], leading to more difficulties for optimization algorithms in fulfilling breadth diversity. 2). the kernel (Matérn5/2) used in STGP and MTGP is highly sensitive to the hyperparameter choices and is not robust [40].

Figure 3 shows the boxplot comparison of the WOSGZ problems over 15 independent runs of the last iteration in MOBO, w.r.t. HV values. The results show that cPoI-V1 outperforms PoI w.r.t. median, mean, and standard deviation (std.) of HV values on WOSGZ1-7 problems. On the WOSGZ8 problem, PoI is more robust and yields higher mean and median HV values than cPoI-V1.

Here, we only represent the empirical attainment curves of PoI and cPoI on the WOSGZ8 problem in Figure 4, by using the empirical first-order attainment function in [20]. Compared with cPoI-V1, PoI and cPoI-V2 are more efficient for searching for the nadir points. On the other hand, cPoI-V1 generates more non-dominated solutions around the knee point compared with PoI and cPoI-V2,3. A possible reason is that the distribution of cPoI-V1's covariance coefficient ρ over the entire computation is positive skewness, indicating that ρ is more likely to be positive during the optimization. The rotation of the multivariate normal distribution will lead to fewer samples around the nadir points but more samples around the knee point, as shown in Figure 1. Following the same idea in [38, 56, 57], a remedy is to use PoI to search for the nadir points at the early optimization stage and then use cPoI to increase the resolution between the nadir points. Please see the supplementary associated with this paper for the empirical attainment curves of all the experiments.

Table 1 presents the results of pairwise comparisons using the Wilcoxon Rank-Sum test (+/≈ /−) across all tested algorithms on the WOSGZ1-8 problems. The total sum of the Wilcoxon Rank-Sum test statistics (+/≈ /−) indicates that the cPoI-V1 algorithm outperforms all other algorithms, producing at least similar results to the other algorithm-problem instance pairs. Because the cPoI

Table 1: The pairwise Wilcoxon's Rank-Sum test (+/≈ /−) matrix at a 0.05 significance level is performed among all test algorithms, where algorithms in the first row are compared with the algorithms in the first column.

	EHVI	ParEGO	MOEA/D-EGO	PoI	cPoI-V1	cPoI-V2	cPoI-V3
EHVI	0/0/0	0/1/7	0/4/4	5/3/0	8/0/0	6/2/0	7/1/0
ParEGO	7/1/0	0/0/0	7/1/0	7/1/0	8/0/0	7/1/0	8/0/0
MOEA/D-EGO	4/4/0	0/1/7	0/0/0	7/1/0	8/0/0	7/1/0	8/0/0
PoI	0/3/5	0/1/7	0/1/7	0/0/0	**5/3/0**	**4/4/0**	**4/4/0**
cPoI-V1	0/0/8	0/0/8	0/0/8	0/3/5	0/0/0	0/7/1	0/6/2
cPoI-V2	0/2/6	0/1/7	0/1/7	0/4/4	1/7/0	0/0/0	0/8/0
cPoI-V3	0/1/7	0/0/8	0/0/8	0/4/4	2/6/0	0/8/0	0/0/0
Sum of +/≈ /−	11/11/26	0/4/44	7/7/34	19/16/13	**32/16/0**	24/23/1	27/19/2

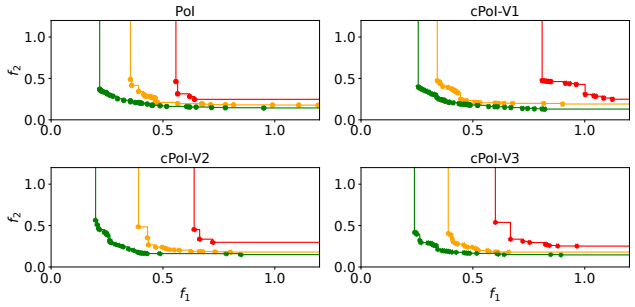

Figure 4: The best, median, and the worst empirical attainment curves on WOSGZ8, where ●, ●, and ● represent the best, the worst, and the median Pareto-front approximation sets over 15 independent runs, respectively.

algorithms benefit from considering the correlation between objectives in guiding the search, enabling them to perform better than PoI and EHVI. This experiment indicates that the posterior covariance using k_M is better than single GPs because MTGP has a more powerful representation capacity. On the other hand, the task correlation encoded in MTGP enables the tasks to transfer knowledge, leading to enhanced prediction accuracy. The performance difference between V1 and V3 is caused by the kernel structures of their posterior covariances. The posterior covariance of V1 has the capacity to connect different objectives by using k_M. However, The posterior covariance of V3 is implicit and does not connect observations from different objectives in a task-specific kernel. This result corroborates our discussion of their difference in Section 3.2.

Real-world Problem. For the real-world problem of hatch cover design optimization, ParEGO and MOEA/D-EGO are not compared due to their poor performance on the artificial benchmarks. Table 2 shows the experiment results w.r.t. HV indicator. The cPoI-V1,2,3 outperform EHVI and PoI w.r.t. mean and std. of HV over 15 independent runs. For the statistical test on the RE problem, please refer to the supplementary associated with this paper.

6 CONCLUSIONS AND FUTURE WORKS

This paper proposes a novel acquisition function (cPoI) for MOBO, which accounts for the correlations among objectives using a covariance matrix computed by the posterior covariance derived from

Table 2: HV Comparison on the RE problem.

	EHVI	PoI	cPoI-V1	cPoI-V2	cPoI-V3
mean	31041.38	31077.46	**31269.62**	31224.66	31229.91
std.	187.8018	813.3861	130.6397	168.9738	**120.3714**
min	30618.97	28145.11	31002.58	30777.44	**31080.75**
median	31113.24	31302.27	**31332.76**	31296.34	31230.75
max	31299.55	31373.54	**31397.39**	31377.67	31380.37

MTGP. Two methods of computing the covariance between objectives are extended from the existing works. The explicit formula of cPoI is provided for exact computation. The proposed cPoI is compared with PoI and EHVI on 8 artificial benchmarks and 1 real-world optimization problem. The experiment results show that cPoI outperforms EHVI and PoI and confirms the effectiveness of incorporating the correlation between objectives in MOBO.

For future works, the following directions are worthwhile to investigate:

- Constructing more powerful MTGPs by using expressive and interpretable kernels. The current Matérn 5/2 is mainly used in interpolation scenarios other than extrapolation [50] that is involved, which limits the learning and representation capacity of an MTGP model.
- Integration of objectives correlation into existing state-of-the-art acquisition functions for exact computation, such as EHVI and truncated EHVI [53]. Although the MC, sample-average approximation and Gauss–Hermite quadrature can approximate correlated EHVI and truncated EHVI, their optimization performance via approximation methods may suffer due to the inherent approximation error, as highlighted by [9]. A potential solution is to generalize the probability distribution of hypervolume improvement [47] by utilizing the multivariate probability density function and using Taylor expansion to facilitate the computation of the product of two Gaussian random variables.
- Incorporating the $\epsilon-$improvement technique in cPoI. The utilization of the $\epsilon-$improvement technique within the context of cPoI has been demonstrated to yield significant benefits, as evidenced in the works [18, 47].
- Applying the proposed acquisition functions on more time-consuming and multi-fidelity real-world problems [46].

ACKNOWLEDGEMENT

This work is supported by the Austrian Science Fund (FWF – Der Wissenschaftsfonds) under the project (I 5315, 'ML Methods for Feature Identification Global Optimization). This work is partly supported by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence, The Chinese University of Hong Kong, Shenzhen, under Grant No. 2022B1212010001, by the Natural Science Foundation of China (NSFC) with grant No. 62106212, and by the High Performance Computing Center of Central South University (CSU), China.

REFERENCES

[1] Mauricio A. Álvarez and Neil D. Lawrence. 2008. Sparse Convolved Gaussian Processes for Multi-output Regression. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou (Eds.). Curran Associates, Inc., 57–64.

[2] Mauricio A. Alvarez, Lorenzo Rosasco, Neil D. Lawrence, et al. 2012. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning* 4, 3 (2012), 195–266.

[3] Edwin V. Bonilla, Kian M. Chai, and Christopher Williams. 2008. Multi-task Gaussian process prediction. In *Advances in neural information processing systems*. 153–160.

[4] Zdravko I Botev. 2017. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79, 1 (2017), 125–148.

[5] Kai Chen, Twan van Laarhoven, Jinsong Chen, and Elena Marchiori. 2019. Incorporating dependencies in spectral kernels for Gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 565–581.

[6] Kai Chen, Twan van Laarhoven, Perry Groot, Jinsong Chen, and Elena Marchiori. 2019. Multioutput convolution spectral mixture for Gaussian processes. *IEEE Transactions on Neural Networks and Learning Systems* 31 (2019), 2255–2266.

[7] Kai Chen, Twan van Laarhoven, Perry Groot, Jinsong Chen, and Elena Marchiori. 2020. Generalized Convolution Spectral Mixture for Multitask Gaussian Processes. *IEEE transactions on neural networks and learning systems* 31, 12 (2020), 5613–5623.

[8] Tinkle Chugh. 2022. Mono-Surrogate vs Multi-Surrogate in Multi-Objective Bayesian Optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 2143–2151. https://doi.org/10.1145/3520304.3533972

[9] Carlos A. Coello Coello, Silvia González Brambila, Josué Figueroa Gamboa, Ma Guadalupe Castillo Tapia, and Raquel Hernández Gómez. 2020. Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex & Intelligent Systems* 6, 2 (2020), 221–236.

[10] Carlos A. Coello Coello. 2011. Evolutionary Multi-Objective Optimization: Basic Concepts and Some Applications in Pattern Recognition. In *Proceedings of the Third Mexican conference on Pattern recognition*, José Francisco Martínez-Trinidad, Jesús Ariel Carrasco-Ochoa, Cherif Ben-Youssef Brants, and Edwin Robert Hancock (Eds.). Springer, Berlin, Heidelberg, 22–33. https://doi.org/10.1007/978-3-642-21587-2_3

[11] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. 2020. Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 826, 14 pages.

[12] George De Ath, Richard M. Everson, Alma A.M. Rahat, and Jonathan E. Fieldsend. 2021. Greed is good: Exploration and exploitation trade-offs in Bayesian optimisation. *ACM Transactions on Evolutionary Learning and Optimization* 1, 1 (2021), 1–22.

[13] Zvi Drezner. 1994. Computation of the trivariate normal integral. *Math. Comp.* 62, 205 (1994), 289–294.

[14] Zvi Drezner and G. O. Wesolowsky. 1990. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation* 35, 1-2 (1990), 101–107. https://doi.org/10.1080/00949659008811236 arXiv:https://doi.org/10.1080/00949659008811236

[15] Robert Dürichen, Marco AF Pimentel, Lei Clifton, Achim Schweikard, and David A Clifton. 2015. Multitask Gaussian processes for multivariate physiological time-series analysis. *IEEE Transactions on Biomedical Engineering* 62, 1 (2015), 314–322.

[16] Michael Emmerich, André Deutz, and Jan-Willem Klinkenberg. 2008. The computation of the expected improvement in dominated hypervolume of Pareto front approximations. *Technical Report, Leiden University* 34 (2008).

[17] Michael Emmerich, Kyriakos C. Giannakoglou, and Boris Naujoks. 2006. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *Evolutionary Computation, IEEE Transactions on* 10, 4 (2006), 421–439.

[18] Michael Emmerich, Kaifeng Yang, and André Deutz. 2020. *Infill Criteria for Multiobjective Bayesian Optimization*. Springer International Publishing, Cham, 3–16. https://doi.org/10.1007/978-3-030-18764-4_1

[19] Michael Emmerich, Kaifeng Yang, Andre H. Deutz, Hao Wang, and Carlos M. Fonseca. 2016. A Multicriteria Generalization of Bayesian Global Optimization. In *Advances in Stochastic and Deterministic Global Optimization*. 229–242. https://doi.org/10.1007/978-3-319-29975-4_12

[20] Carlos M. Fonseca, Andreia P. Guerreiro, Manuel López-Ibáñez, and Luís Paquete. 2011. On the Computation of the Empirical Attainment Function. In *Evolutionary Multi-Criterion Optimization*, Ricardo H. C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 106–120.

[21] Joseph Futoma, Sanjay Hariharan, and Katherine Heller. 2017. Learning to detect sepsis with a multitask Gaussian process RNN classifier. In *International Conference on Machine Learning*. PMLR, 1174–1182.

[22] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. 2020. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing* 380 (2020), 20 – 35. https://doi.org/10.1016/j.neucom.2019.11.004

[23] Alan Genz. 2004. Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing* 14, 3 (2004), 251–260.

[24] Alan Genz and Frank Bretz. 1999. Numerical computation of multivariate t-probabilities with application to power calculation of multiple contrasts. *Journal of Statistical Computation and Simulation* 63, 4 (1999), 103–117. https://doi.org/10.1080/00949659908811962 arXiv:https://doi.org/10.1080/00949659908811962

[25] Alan Genz and Frank Bretz. 2002. Comparison of Methods for the Computation of Multivariate t Probabilities. *Journal of Computational and Graphical Statistics* 11, 4 (2002), 950–971. https://doi.org/10.1198/106186002394 arXiv:https://doi.org/10.1198/106186002394

[26] Alan Genz and Frank Bretz. 2009. *Computation of multivariate normal and t probabilities*. Vol. 195. Springer Science & Business Media.

[27] Nikolaus Hansen. 2009. Benchmarking a BI-Population CMA-ES on the BBOB-2009 Function Testbed. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers* (Montreal, Québec, Canada) *(GECCO '09)*. Association for Computing Machinery, New York, NY, USA, 2389–2396. https://doi.org/10.1145/1570256.1570333

[28] Wilfred K. Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109.

[29] Donald R. Jones. 2001. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *J. Glob. Optim.* 21, 4 (2001), 345–383. https://doi.org/10.1023/A:1012771025575

[30] Donald R. Jones, Matthias Schonlau, and William J. Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.

[31] Joshua Knowles. 2006. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 10, 1 (2006), 50–66.

[32] Mina K. Lukovic, Yunsheng Tian, and Wojciech Matusik. 2020. Diversity-guided multi-objective bayesian optimization with batch evaluations. *Advances in Neural Information Processing Systems* 33 (2020), 17708–17720.

[33] Jianping Luo, Jiqiang Feng, and RuoFan Jin. 2019. A New Approach to Building the Gaussian Process Model for Expensive Multi-objective Optimization. In *2019 9th International Conference on Information Science and Technology (ICIST)*. 374–379. https://doi.org/10.1109/ICIST.2019.8836854

[34] Wesley J. Maddox, Maximilian Balandat, Andrew G. Wilson, and Eytan Bakshy. 2021. Bayesian Optimization with High-Dimensional Outputs. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 19274–19287. https://proceedings.neurips.cc/paper_files/paper/2021/file/a0d3973ad100ad83a64c304bb58677dd-Paper.pdf

[35] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. 2017. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research* 18, 40 (apr 2017), 1–6. http://jmlr.org/papers/v18/16-537.html

[36] Arman Melkumyan and Fabio Ramos. 2011. Multi-kernel Gaussian processes. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 1408.

[37] Jonas Močkus. 1975. *On bayesian methods for seeking the extremum*. Springer Berlin Heidelberg, Berlin, Heidelberg, 400–404. https://doi.org/10.1007/3-540-07165-2_55

[38] Pramudita S. Palar, Kaifeng Yang, Koji Shimoyama, Michael Emmerich, and Thomas Bäck. 2018. Multi-objective Aerodynamic Design with User Preference Using Truncated Expected Hypervolume Improvement. In *Proceedings of the*

Genetic and Evolutionary Computation Conference (Kyoto, Japan) *(GECCO '18)*. ACM, New York, NY, USA, 1333–1340. https://doi.org/10.1145/3205455.3205497

[39] Alma A. M. Rahat, Richard M. Everson, and Jonathan E. Fieldsend. 2017. Alternative Infill Strategies for Expensive Multi-Objective Optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Berlin, Germany) *(GECCO '17)*. Association for Computing Machinery, New York, NY, USA, 873–880. https://doi.org/10.1145/3071178.3071276

[40] Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian processes for machine learning*. MIT Press. https://www.worldcat.org/oclc/61285753

[41] Thomas J. Santner, Brian J. Williams, and William I. Notz. 2003. *The Design and Analysis of Computer Experiments*. Springer. https://doi.org/10.1007/978-1-4757-3799-8

[42] Amar Shah and Zoubin Ghahramani. 2016. Pareto Frontier Learning with Expensive Correlated Objectives. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1919–1927. https://proceedings.mlr.press/v48/shahc16.html

[43] Qi Sun, Tinghuan Chen, Siting Liu, Jianli Chen, Hao Yu, and Bei Yu. 2022. Correlated Multi-Objective Multi-Fidelity Optimization for HLS Directives Design. *ACM Trans. Des. Autom. Electron. Syst.* 27, 4, Article 31 (mar 2022), 27 pages. https://doi.org/10.1145/3503540

[44] Ryoji Tanabe and Hisao Ishibuchi. 2020. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing* 89 (2020), 106078. https://doi.org/10.1016/j.asoc.2020.106078

[45] Kyle R. Ulrich, David E. Carlson, Kafui Dzirasa, and Lawrence Carin. 2015. GP kernels for cross-spectrum analysis. In *Advances in neural information processing systems*. 1999–2007.

[46] Handing Wang, Yaochu Jin, and John Doherty. 2018. A Generic Test Suite for Evolutionary Multifidelity Optimization. *IEEE Transactions on Evolutionary Computation* 22, 6 (2018), 836–850. https://doi.org/10.1109/TEVC.2017.2758360

[47] Hao Wang, Kaifeng Yang, Michael Affenzeller, and Michael Emmerich. 2022. Probability Distribution of Hypervolume Improvement in Bi-objective Bayesian Optimization. arXiv:2205.05505 [cs.LG]

[48] Zhenkun Wang, Yew-Soon Ong, Jianyong Sun, Abhishek Gupta, and Qingfu Zhang. 2019. A Generator for Multiobjective Test Problems With Difficult-to-Approximate Pareto Front Boundaries. *IEEE Transactions on Evolutionary Computation* 23, 4 (2019), 556–571. https://doi.org/10.1109/TEVC.2018.2872453

[49] Zhenkun Wang, Qingfu Zhang, Yew-Soon Ong, Shunyu Yao, Haitao Liu, and Jianping Luo. 2021. Choose Appropriate Subproblems for Collaborative Modeling in Expensive Multiobjective Optimization. *IEEE Transactions on Cybernetics* (2021), 1–14. https://doi.org/10.1109/TCYB.2021.3126341

[50] Andrew Wilson and Ryan Adams. 2013. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 1067–1075.

[51] Andrew G. Wilson, David A. Knowles, and Zoubin Ghahramani. 2012. Gaussian process regression networks. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*. 1139–1146.

[52] Kaifeng Yang, Michael Affenzeller, and Guozhi Dong. 2022. A parallel technique for multi-objective Bayesian global optimization: Using a batch selection of probability of improvement. *Swarm and Evolutionary Computation* 75 (2022), 101183. https://doi.org/10.1016/j.swevo.2022.101183

[53] Kaifeng Yang, Andre Deutz, Zhiwei Yang, Thomas Bäck, and Michael Emmerich. 2016. Truncated expected hypervolume improvement: Exact computation and application. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 4350–4357. https://doi.org/10.1109/CEC.2016.7744343

[54] Kaifeng Yang, Michael Emmerich, Andre Deutz, and Thomas Bäck. 2019. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization* 75, 1 (01 Sep 2019), 3–34. https://doi.org/10.1007/s10898-019-00798-7

[55] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. 2019. Multi-Objective Bayesian Global Optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation* 44 (2019), 945 – 956. https://doi.org/10.1016/j.swevo.2018.10.007

[56] Kaifeng Yang, Longmei Li, André Deutz, Thomas Bäck, and Michael Emmerich. 2016. Preference-based Multiobjective Optimization using Truncated Expected Hypervolume Improvement. In *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 276–281. https://doi.org/10.1109/FSKD.2016.7603186

[57] Kaifeng Yang, Pramudita S. Palar, Michael Emmerich, Koji Shimoyama, and Thomas Bäck. 2019. A Multi-Point Mechanism of Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Global Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 656–663. https://doi.org/10.1145/3321707.3321784

[58] Kaifeng Yang, Koen van der Blom, Thomas Bäck, and Michael Emmerich. 2019. Towards Single- and Multiobjective Bayesian Global Optimization for Mixed Integer Problems. *AIP Conference Proceedings* 2070, 1 (2019), 020044. https://doi.org/10.1063/1.5090011 arXiv:https://aip.scitation.org/doi/pdf/10.1063/1.5090011

[59] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. 2009. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation* 14, 3 (2009), 456–474.

[60] Jing Zhao and Shiliang Sun. 2016. Variational dependent multi-output Gaussian process dynamical systems. *The Journal of Machine Learning Research* 17, 1 (2016), 4134–4169.

[61] Antanas Zilinskas. 1992. A review of statistical models for global optimization. *J. Glob. Optim.* 2, 2 (1992), 145–153. https://doi.org/10.1007/BF00122051

[62] Antanas Žilinskas and Jonas Mockus. 1972. On one Bayesian method of search of the minimum. *Avtomatica i Vychislitel'naya Teknika* 4 (1972), 42–44.

[63] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271.

[64] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane G. Da Fonseca. 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* 7, 2 (2003), 117–132.