Universiteit Leiden
The Netherlands

Bachelor Data Science & Artificial Intelligence

Enhancing PCA-Assisted Bayesian Optimization through Parallel Orthogonal Sampling

Ivan Banny

Under supervision of
Elena Raponi & Ivan Olarte Rodriguez

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

15/07/2025

Abstract

This thesis proposes Orthogonal PCA-assisted Bayesian Optimization (O-PCA-BO), a novel enhancement to the PCA-BO algorithm that addresses fundamental exploration limitations. While PCA-BO effectively reduces dimensionality by operating in a subspace spanned by principal components with high variance, it systematically excludes regions in the orthogonal complement space, potentially missing global optima. Our approach introduces parallel orthogonal sampling, where for each candidate selected in the principal component subspace, multiple points are sampled in the orthogonal complement space spanned by discarded components. This strategy maintains computational efficiency while enabling exploration of previously inaccessible regions. We evaluate O-PCA-BO against PCA-BO on functions F15-F24 from the COCO BBOB benchmark across dimensions 10, 20, and 40 with batch sizes 1, 5, 10, 20, and 42. Results demonstrate that O-PCA-BO achieves statistically significant convergence improvements in 9 out of 15 test configurations while maintaining superior wall-clock performance across all settings, with time reductions increasing with batch size. The method shows particular strength on functions with weak global structure where PCA-BO often fails to locate optimal regions.

Contents

1 Introduction

Bayesian Optimization (BO) has established itself as the method of choice for optimizing expensive black-box functions where function evaluations are costly and gradients are unavailable. The approach constructs a probabilistic surrogate model of the objective function using Gaussian Process regression, then uses an acquisition function to balance exploration and exploitation when selecting the next evaluation point [JSW98]. However, BO faces severe scalability challenges in high-dimensional spaces due to the curse of dimensionality, which manifests in both surrogate model construction and acquisition function optimization.

PCA-assisted Bayesian Optimization (PCA-BO) [RWB+20] addresses these scalability issues through dimensionality reduction. The method applies weighted Principal Component Analysis to identify a lower-dimensional subspace capturing most variance in the objective function landscape, then performs optimization within this reduced space. This approach significantly reduces computational requirements while maintaining optimization performance on problems with adequate global structure.

1.1 Fundamental Limitations of PCA-BO

Despite its computational advantages, PCA-BO exhibits a critical limitation: exploration is restricted exclusively to the subspace spanned by selected principal components. By construction, PCA-BO discards components explaining minimal variance in previously evaluated points. However, these discarded components may contain the global optimum, particularly in functions with weak global structure or deceptive landscapes where optimal regions lie outside the high-variance subspace.

This limitation becomes self-reinforcing. As PCA-BO samples more points within the reduced subspace, variance along those dimensions increases, making them even more likely to be retained as principal components in subsequent iterations. This feedback mechanism can permanently exclude valuable regions in the orthogonal complement space, leading to premature convergence to suboptimal solutions.

Consider a simple example: if the global optimum lies in a direction orthogonal to the primary variation in the data, PCA-BO will systematically avoid this direction regardless of the optimization budget. The algorithm becomes trapped in a subspace that may be entirely disjoint from the optimal region.

1.2 Our Contribution: Orthogonal Sampling

We propose Orthogonal PCA-assisted Bayesian Optimization (O-PCA-BO), which systematically explores the orthogonal complement space while maintaining PCA-BO's computational advantages. The key innovation is parallel orthogonal sampling: for each candidate point selected in the reduced subspace, we generate multiple samples in the $(d - r)$-dimensional orthogonal complement space, where d is the original dimensionality and r is the reduced dimensionality.

Our method works as follows. After PCA-BO selects q candidate points in the r-dimensional reduced space, we map these points back to the original d-dimensional space. For each mapped candidate, we sample m additional points in the orthogonal complement space using Markov Chain Monte Carlo methods, specifically Hit-and-Run sampling [BRS93]. We then evaluate all $q \times m$ points in parallel, incorporating the results into the dataset for subsequent iterations.

This approach addresses PCA-BO's exploration limitation without sacrificing computational efficiency. The orthogonal samples provide information about regions PCA-BO cannot access, potentially rotating the PCA basis in subsequent iterations to better capture the objective function landscape. Parallel evaluation maintains wall-clock efficiency when function evaluations can be performed simultaneously.
Beyond the core orthogonal sampling innovation, we introduce three additional improvements: (1) squared rank-based weighting that reduces the influence of poor-performing points on the PCA transformation, (2) Log Expected Improvement acquisition function wrapped in a penalized formulation for handling bound constraints, and (3) selective GPR training using points ranked by both objective value and mapping distance to reduce model confusion from overlapping projections.

1.3 Experimental Validation

We conduct extensive experiments comparing O-PCA-BO against PCA-BO on functions F15-F24 from the COCO BBOB benchmark suite. These functions divide into two categories: multi-modal functions with adequate global structure (F15-F19) and multi-modal functions with weak global structure (F20-F24). We test across dimensions 10, 20, and 40 with batch sizes 1, 5, 10, 20, and 42. Results demonstrate that O-PCA-BO significantly outperforms PCA-BO, particularly on functions with weak global structure where PCA-BO often plateaus quickly. Wilcoxon signed-rank tests confirm statistical significance in 9 out of 15 test configurations, with O-PCA-BO achieving superior convergence. Wall-clock time analysis reveals consistent efficiency improvements across all configurations, with the advantage growing with batch size. On functions F20-F24, O-PCA-BO maintains consistent improvement throughout the optimization budget while PCA-BO stagnates.

1.4 Thesis Structure

This thesis is organized as follows. Section 2 reviews the mathematical foundations of Bayesian Optimization. Section 3 details the PCA-BO algorithm and analyzes its limitations. Section 4 presents our O-PCA-BO method with complete mathematical formulation and algorithmic description. Section 5 describes our experimental methodology and presents comprehensive results. Section 6 concludes with analysis of contributions and future research directions.

2 Bayesian Optimization

Bayesian Optimization provides a principled framework for global optimization of expensive black-box functions. This section establishes the mathematical foundations underlying both PCA-BO and our proposed O-PCA-BO method.

2.1 Problem Formulation

We consider the optimization problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{1}$$

where $f : \mathcal{X} \to \mathbb{R}$ is an expensive black-box function defined on a bounded domain $\mathcal{X} = [\mathbf{l}, \mathbf{u}] \subset \mathbb{R}^d$. The function f is treated as a black-box because we lack access to its analytical form, derivatives, or internal structure - only function values at queried points are available.
The optimization challenge arises from the expense of function evaluations. In applications such as hyperparameter tuning, engineering design, or experimental optimization, each evaluation may require hours or days of computation, making efficient point selection critical.

2.2 Gaussian Process Surrogate Modeling

BO addresses this challenge by constructing a probabilistic surrogate model of the objective function. A Gaussian Process (GP) serves as the surrogate due to its ability to provide both predictions and uncertainty quantification [Kle09].
A GP is completely specified by a mean function $\mu : \mathcal{X} \to \mathbb{R}$ and a covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We write $f \sim \mathcal{GP}(\mu, k)$ to indicate that f follows a GP with mean μ and covariance k.
Given n observations $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $y_i = f(\mathbf{x}_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, the posterior distribution at a test point $\mathbf{x}$ is Gaussian with:

$$\mu_n(\mathbf{x}) = \mu(\mathbf{x}) + \mathbf{k}_n(\mathbf{x})^T [\mathbf{K}_n + \sigma^2 \mathbf{I}]^{-1} (\mathbf{y}_n - \boldsymbol{\mu}_n) \tag{2}$$

$$\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_n(\mathbf{x})^T [\mathbf{K}_n + \sigma^2 \mathbf{I}]^{-1} \mathbf{k}_n(\mathbf{x}) \tag{3}$$

where $\mathbf{K}_n$ is the $n \times n$ covariance matrix with $[\mathbf{K}_n]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_n(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_n, \mathbf{x})]^T$, $\mathbf{y}_n = [y_1, \ldots, y_n]^T$, and $\boldsymbol{\mu}_n = [\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_n)]^T$.
The GP hyperparameters (kernel parameters and noise variance) are typically learned by maximizing the marginal log-likelihood [WS14]:

$$\log p(\mathbf{y}_n | \mathbf{X}_n, \boldsymbol{\theta}) = -\frac{1}{2} \mathbf{y}_n^T [\mathbf{K}_n + \sigma^2 \mathbf{I}]^{-1} \mathbf{y}_n - \frac{1}{2} \log |\mathbf{K}_n + \sigma^2 \mathbf{I}| - \frac{n}{2} \log(2\pi) \tag{4}$$

2.3 Acquisition Functions

The acquisition function $\alpha : \mathcal{X} \to \mathbb{R}$ guides the selection of the next evaluation point by balancing exploitation (sampling where the surrogate predicts good values) with exploration (sampling where uncertainty is high).

2.3.1 Expected Improvement

Expected Improvement (EI) quantifies the expected improvement over the current best observation $f_{\text{best}} = \min_{i=1,\ldots,n} y_i$:

$$\text{EI}(\mathbf{x}) = \mathbb{E}[\max(f_{\text{best}} - f(\mathbf{x}), 0)] \tag{5}$$

For a GP posterior, this expectation has a closed-form solution:

$$\text{EI}(\mathbf{x}) = (f_{\text{best}} - \mu_n(\mathbf{x}))\Phi(Z) + \sigma_n(\mathbf{x})\phi(Z) \tag{6}$$

where $Z = \frac{f_{\text{best}} - \mu_n(\mathbf{x})}{\sigma_n(\mathbf{x})}$, and Φ and ϕ are the standard normal CDF and PDF respectively.

2.3.2 Log Expected Improvement

Log Expected Improvement (LogEI) [ADE+25] addresses numerical issues with standard EI when improvements become very small:

$$\text{LogEI}(\mathbf{x}) = \log(\text{EI}(\mathbf{x}) + \epsilon) \tag{7}$$

where $\epsilon > 0$ is a small constant preventing logarithm of zero. LogEI provides more stable optimization when acquisition values become very small, which commonly occurs in later optimization stages.

2.4 Parallel Bayesian Optimization

For parallel evaluation of q points, we require batch acquisition functions that select multiple points simultaneously while accounting for their mutual information.

2.4.1 q-Expected Improvement

The q-Expected Improvement (q-EI) generalizes EI to batch selection:

$$\text{q-EI}(\mathbf{X}_q) = \mathbb{E}[\max(f_{\text{best}} - \min_{i=1,\ldots,q} f(\mathbf{x}_i), 0)] \tag{8}$$

where $\mathbf{X}_q = \{\mathbf{x}_1, \ldots, \mathbf{x}_q\}$ is the batch of q points.
Computing q-EI exactly becomes intractable for large q, but Monte Carlo approximations provide practical solutions. The acquisition function naturally accounts for diminishing returns as additional points are added to the batch.

2.5 Bayesian Optimization Algorithm

The standard BO procedure follows an iterative framework:

Algorithm 1 Bayesian Optimization

1: Initialize with n_0 points $\mathcal{D}_{n_0} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_0}$ using Latin Hypercube Sampling [FLS05]
2: **for** $t = n_0 + 1, n_0 + 2, \ldots, N$ **do**
3: Fit GP model to $\mathcal{D}_{t-1}$ using marginal likelihood maximization
4: Select next point(s) $\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}; \mathcal{D}_{t-1})$
5: Evaluate $y_t = f(\mathbf{x}_t)$
6: Augment dataset $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$
7: **end for**
8: Return best found solution $\mathbf{x}^* = \arg\min_{(\mathbf{x},y) \in \mathcal{D}_N} y$

The optimization of the acquisition function in Step 3 typically employs gradient-based methods such as L-BFGS-B [ZBLN97] with multiple random restarts to avoid local optima.

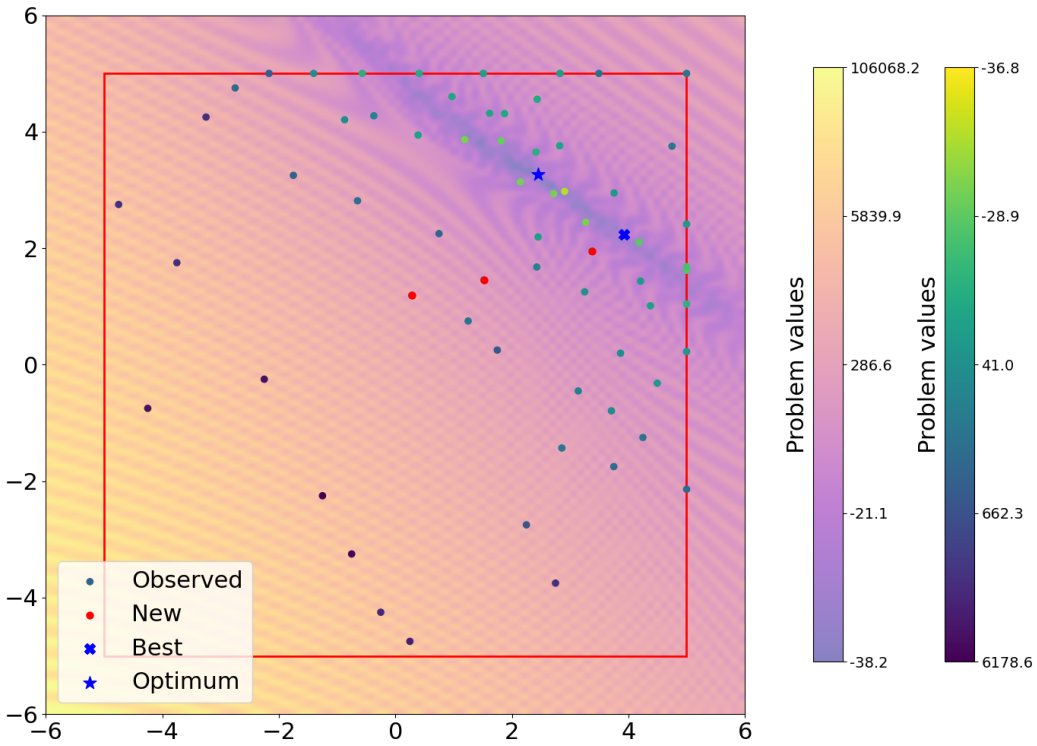

Figure 1: Vanilla Bayesian Optimization on a 2D test function. The landscape shows the true objective function with darker regions indicating better values. Red points indicate candidates selected in the current iteration (batch size 3), while other points are colored by their objective value. The algorithm operates directly in the 2D problem space without dimensionality reduction.

This framework provides the foundation for dimensionality reduction approaches like PCA-BO and our proposed O-PCA-BO method, which modify the spaces in which surrogate modeling and acquisition optimization occur.

3 PCA-Assisted Bayesian Optimization

PCA-assisted Bayesian Optimization addresses the scalability challenges of standard BO through dimensionality reduction. This section analyzes the algorithm in detail and identifies the fundamental limitations that motivate our orthogonal sampling approach.

3.1 Motivation and Challenges

As dimensionality increases, BO faces two critical computational bottlenecks:

1. **Surrogate model complexity**: GP inference scales as $O(n^3)$ in the number of training points, while the effective number of training points needed for reliable modeling grows exponentially with dimension.

2. **Acquisition function optimization**: Finding global optima of the acquisition function becomes increasingly difficult in high-dimensional spaces, requiring exponentially more optimization effort.

Standard BO typically becomes impractical beyond 10-15 dimensions. PCA-BO addresses these challenges by identifying a lower-dimensional subspace that captures most variance in the objective function, performing optimization within this reduced space.

3.2 Weighted Principal Component Analysis

PCA-BO employs a weighted variant of PCA that incorporates objective function values to guide dimensionality reduction toward regions of interest.

3.2.1 Rank-Based Weighting Scheme

Given n evaluated points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$ with corresponding function values $\mathbf{y} = [y_1, \ldots, y_n]^T$, PCA-BO computes rank-based weights:

$$\tilde{w}_i = \ln n - \ln r_i \tag{9}$$

where r_i is the rank of point $\mathbf{x}_i$ according to its function value (rank 1 corresponds to the best observed value). These pre-weights are then normalized:

$$w_i = \frac{\tilde{w}_i}{\sum_{j=1}^{n} \tilde{w}_j} \tag{10}$$

This weighting scheme assigns exponentially decreasing weights to worse-performing points, focusing the PCA transformation on regions with better objective values.

3.2.2 Weighted PCA Procedure

The weighted PCA procedure proceeds as follows:

1. **Center the data**:
$$\overline{\mathbf{X}} = \mathbf{X} - \mathbf{1}_n \boldsymbol{\mu}^T \tag{11}$$
where $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$ and $\mathbf{1}_n$ is an n-dimensional vector of ones.

2. **Apply weights**:
$$\mathbf{X}' = \operatorname{diag}(w_1, \ldots, w_n)\, \overline{\mathbf{X}} \tag{12}$$

3. **Compute weighted mean**:
$$\boldsymbol{\mu}' = \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}'_i \tag{13}$$

4. **Center weighted data**:
$$\mathbf{X}'' = \mathbf{X}' - \mathbf{1}_n \boldsymbol{\mu}'^T \tag{14}$$

5. **Compute covariance matrix**:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}''^T\mathbf{X}'' \tag{15}$$

6. **Eigendecomposition**:

$$\mathbf{C} = \mathbf{P}\mathbf{D}\mathbf{P}^T \tag{16}$$

where columns of $\mathbf{P}$ contain eigenvectors (principal components) and $\mathbf{D} = \text{diag}(\lambda_1, \ldots, \lambda_d)$ contains eigenvalues in descending order.

3.2.3 Component Selection

PCA-BO selects the first r principal components that explain at least α percent of the total variance:

$$r = \inf\left\{k \in \{1, \ldots, d\} : \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{d}\lambda_i} \geq \alpha\right\} \tag{17}$$

In practice, $\alpha = 0.95$ (95%) provides a reasonable balance between dimensionality reduction and information preservation.

3.3 Space Transformations

PCA-BO defines bidirectional transformations between the original d-dimensional space and the reduced r-dimensional space.

3.3.1 Forward Transformation

The transformation from original space to reduced space maps centered data to the principal component subspace:

$$\mathbf{z} = \mathbf{P}_r^T(\mathbf{x} - \boldsymbol{\mu} - \boldsymbol{\mu}') \tag{18}$$

where $\mathbf{P}_r \in \mathbb{R}^{d\times r}$ contains the first r principal components.

3.3.2 Inverse Transformation

The transformation from reduced space back to original space is given by:

$$\mathbf{x} = \mathbf{P}_r\mathbf{z} + \boldsymbol{\mu}' + \boldsymbol{\mu} \tag{19}$$

This transformation projects points from the r-dimensional reduced space back to the original d-dimensional space.

3.4 PCA-BO Algorithm

The complete PCA-BO algorithm integrates weighted PCA with the BO framework:

Algorithm 2 PCA-assisted Bayesian Optimization

Initialize with n_0 points $\mathcal{D}_{n_0} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_0}$ using Latin Hypercube Sampling
Set $n = n_0$
while $n < N$ (budget not exhausted) **do**
  Compute rank-based weights $\{w_i\}$ using Equations 9-10
  Perform weighted PCA using Equations 11-16
  Select r components using Equation 17
  Transform data to reduced space: $\mathbf{Z}_r = \mathbf{P}_r^T(\overline{\mathbf{X}} - \mathbf{1}_n \boldsymbol{\mu}'^T)$
  Fit GP model $\mathcal{GP}(\mu_r, k_r)$ on $(\mathbf{Z}_r, \mathbf{y})$
  Optimize acquisition function: $\mathbf{z}^* = \arg\max_{\mathbf{z} \in \mathcal{Z}} \alpha(\mathbf{z})$
  Transform back to original space: $\mathbf{x}^* = \mathbf{P}_r \mathbf{z}^* + \boldsymbol{\mu}' + \boldsymbol{\mu}$
  Evaluate $y^* = f(\mathbf{x}^*)$ and augment $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{x}^*, y^*)\}$
  $n = n + 1$
end while
Return best found solution

3.5 Computational Advantages

PCA-BO achieves significant computational savings through dimensionality reduction:

1. **GP training**: Model fitting occurs in $\mathbb{R}^r$ rather than $\mathbb{R}^d$, reducing kernel matrix size and computation time.

2. **Acquisition optimization**: The acquisition function optimization operates in the reduced space, significantly decreasing optimization difficulty.

3. **Scalability**: The method remains practical for dimensions where standard BO becomes intractable.

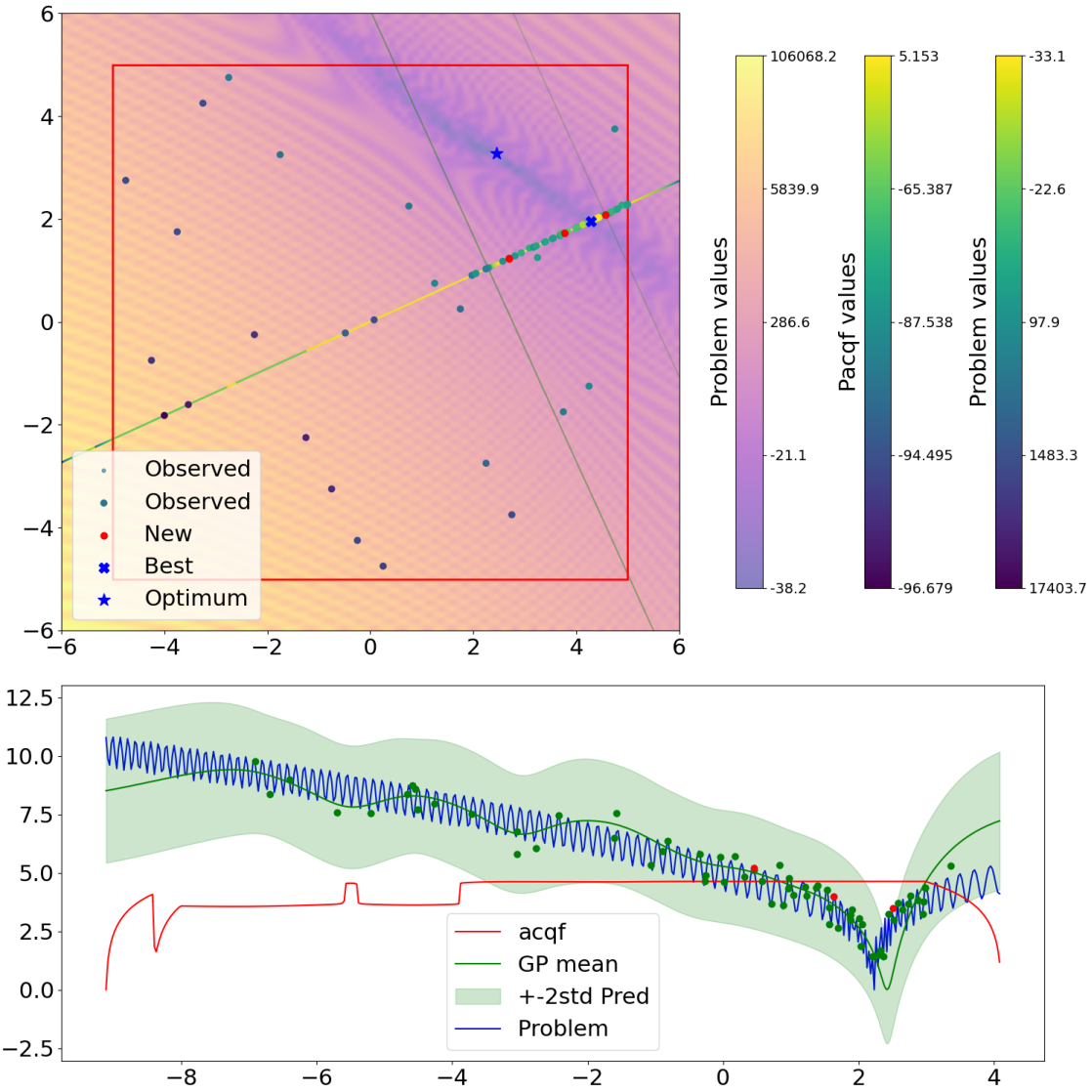

Figure 2: PCA-BO on a 2D test function. Top: The landscape shows the true objective function with the principal components (PCs) overlaid. The bright line represents the selected PC colored by acquisition function values, while the dark line shows the discarded PC. Red points indicate candidates selected via batch acquisition function optimization (batch size 3). Note how all sampled points cluster near the current PC subspace, demonstrating limited exploration. Bottom: Analysis along the main PC showing true function values (blue), penalized acquisition function (red), and GP mean with confidence bounds (green). Points are colored based on their inclusion in GPR training.

3.6 Fundamental Limitations

Despite its computational advantages, PCA-BO exhibits a critical limitation that motivates our orthogonal sampling approach.

3.6.1 Subspace Restriction

PCA-BO restricts exploration to the r-dimensional subspace spanned by the selected principal components. The orthogonal complement space, spanned by the remaining $(d - r)$ discarded components, remains entirely unexplored.
Mathematically, let $\mathbf{P}_r$ contain the first r principal components and $\mathbf{P}_{d-r}$ contain the remaining $(d - r)$ components. Any point $\mathbf{x}$ in the original space can be decomposed as:

$$\mathbf{x} = \mathbf{P}_r \mathbf{z}_r + \mathbf{P}_{d-r} \mathbf{z}_{d-r} + \boldsymbol{\mu}' + \boldsymbol{\mu} \tag{20}$$

PCA-BO only explores points where $\mathbf{z}_{d-r} = \mathbf{0}$, completely ignoring the orthogonal complement space.

3.6.2 Self-Reinforcement Problem

PCA-BO exhibits a self-reinforcing bias. As more points are sampled within the reduced subspace, variance along those dimensions increases, making them even more likely to be retained as principal components in subsequent iterations. This creates a feedback loop that can permanently exclude valuable regions.
Let $V_r(t)$ denote the cumulative variance explained by the first r components at iteration t. If PCA-BO samples exclusively within the current reduced subspace, then:

$$V_r(t + 1) \geq V_r(t) \tag{21}$$

This monotonic increase in explained variance reinforces the current component selection, making it increasingly difficult for the algorithm to explore orthogonal directions even if they contain the global optimum.

3.6.3 Failure Modes

PCA-BO fails when:

1. **Misleading initial sampling**: If initial points exhibit high variance in directions orthogonal to the global optimum, PCA-BO will focus on these misleading directions.
2. **Weak global structure**: On functions where the global optimum lies in a low-variance direction, PCA-BO systematically avoids the optimal region.
3. **Deceptive landscapes**: Functions with local optima in high-variance directions can trap PCA-BO away from the global optimum.

These limitations motivate the development of O-PCA-BO, which systematically explores the orthogonal complement space while maintaining PCA-BO's computational advantages.

4 Orthogonal PCA-assisted Bayesian Optimization

This section presents our proposed Orthogonal PCA-assisted Bayesian Optimization (O-PCA-BO) method, which addresses PCA-BO's exploration limitations through systematic parallel orthogonal sampling while preserving computational efficiency.

4.1 Core Innovation: Orthogonal Sampling

The fundamental insight behind O-PCA-BO is that the orthogonal complement space - the $(d-r)$-dimensional subspace spanned by discarded principal components - may contain valuable regions that PCA-BO cannot access. Rather than ignoring this space entirely, we systematically sample within it to gather information about potentially promising directions.

4.1.1 Mathematical Framework

Given the principal component decomposition $\mathbf{P} = [\mathbf{P}_r, \mathbf{P}_{d-r}]$ where $\mathbf{P}_r \in \mathbb{R}^{d\times r}$ contains the first r components and $\mathbf{P}_{d-r} \in \mathbb{R}^{d\times(d-r)}$ contains the remaining components, any point in the original space can be expressed as:

$$\mathbf{x} = \mathbf{P}_r\mathbf{z}_r + \mathbf{P}_{d-r}\mathbf{z}_{d-r} + \boldsymbol{\mu}' + \boldsymbol{\mu} \tag{22}$$

where $\mathbf{z}_r \in \mathbb{R}^r$ represents coordinates in the principal component subspace and $\mathbf{z}_{d-r} \in \mathbb{R}^{d-r}$ represents coordinates in the orthogonal complement space.
PCA-BO explores only the manifold where $\mathbf{z}_{d-r} = \mathbf{0}$. O-PCA-BO extends exploration by sampling non-zero values of $\mathbf{z}_{d-r}$ around each candidate point selected in the reduced space.

4.1.2 Orthogonal Sampling Strategy

For each candidate point $\mathbf{x}'$ selected by optimizing the acquisition function in the reduced space, O-PCA-BO generates m orthogonal samples:

$$\mathbf{x}'_{\text{ortho},j} = \mathbf{x}' + \mathbf{P}_{d-r}\boldsymbol{\delta}_j, \quad j = 1, \ldots, m \tag{23}$$

where $\boldsymbol{\delta}_j \in \mathbb{R}^{d-r}$ are displacement vectors in the orthogonal complement space.
This formulation ensures that orthogonal samples lie on hyperplanes perpendicular to the principal component subspace, passing through the original candidate points. The samples explore regions entirely inaccessible to PCA-BO while remaining anchored to promising locations identified by the acquisition function.

4.2 Sampling Methods in the Orthogonal Space

We investigate two approaches for generating displacement vectors $\boldsymbol{\delta}_j$ in the orthogonal complement space.

4.2.1 Hit-and-Run Sampling with Distance Control

We employ Hit-and-Run MCMC sampling [BRS93] for all orthogonal sample generation, with an adaptive distance control mechanism that balances exploration breadth with proximity to promising candidates.
For each candidate point $\mathbf{x}'$, we generate orthogonal samples using a two-stage process:

1. **Over-sampling**: Generate $m \cdot s$ samples using Hit-and-Run MCMC, where the sample size multiplier is:

$$s = \max\left(1, \lfloor \text{onorm\_factor} \cdot \max(1, \sqrt{d-r}) \rfloor\right) \tag{24}$$

2. **Distance-based selection**: Select the m samples with smallest Euclidean distance from the origin in the orthogonal space.

The onorm_factor hyperparameter controls the sampling density and resulting proximity:

- When onorm_factor $= 0$: $s = 1$, yielding uniform sampling with exactly m points
- When onorm_factor > 0: $s > 1$, yielding denser sampling followed by selection of the closest points

This mechanism allows adaptive control of exploration intensity in the orthogonal directions. Higher onorm_factor values produce samples closer to the candidate points (more conservative exploration), while lower values produce more uniform coverage of the orthogonal space.
The Hit-and-Run sampling operates within the polytope defined by:

$$\mathbf{l} - \mathbf{x}' \leq \mathbf{P}_{d-r}\boldsymbol{\delta} \leq \mathbf{u} - \mathbf{x}' \tag{25}$$

where the inequality constraints ensure all orthogonal samples remain within the original problem bounds when mapped back to the full-dimensional space.

4.3 Parallel Evaluation Framework

O-PCA-BO evaluates multiple points simultaneously, balancing computational efficiency with exploration breadth. The parallel evaluation strategy operates as follows:

1. Select q candidate points in the reduced space using batch acquisition functions (e.g., q-LogEI)
2. For each candidate, generate m orthogonal samples
3. Evaluate all $q \times m$ points in parallel
4. Incorporate results into the dataset for subsequent iterations

This approach provides $O(qm)$ parallelism while maintaining the computational advantages of reduced-space optimization. When function evaluations can be performed simultaneously (common in simulation-based optimization), the wall-clock time is superior to evaluating points sequentially.

4.4 Algorithm Enhancements

Beyond orthogonal sampling, we introduce three additional algorithmic improvements to address specific challenges in high-dimensional BO.

4.4.1 Squared Rank-Based Weighting

Standard PCA-BO uses linear rank-based weights (Equation 9). We modify this to squared weights:

$$\tilde{w}_i = (\ln n - \ln r_i)^2 \tag{26}$$

Squaring the weights dramatically reduces the influence of poorly-performing points on the PCA transformation. This prevents bad points from corrupting the principal component directions and allows the algorithm to focus more sharply on promising regions.

4.4.2 Log Expected Improvement with Penalization

We replace standard Expected Improvement with Log Expected Improvement (LogEI) to improve numerical stability in later optimization stages. To handle bound constraints when mapping from the reduced space, we wrap LogEI in a penalized acquisition function:

$$\text{PLogEI}(\mathbf{z}) = \text{LogEI}(\mathbf{z}) - \beta \cdot \mathbf{d_z} \tag{27}$$

where $\mathbf{d_z}$ is the distance between the point $\mathbf{P}_r\mathbf{z} + \boldsymbol{\mu}' + \boldsymbol{\mu}$ (reverse PCA transformation) and its projection (via clipping) onto the feasible problem bounds and $\beta > 0$ is a penalty factor. Note that $\mathbf{d_z}$ is 0 if the point is mapped back to the valid range of inputs.

4.4.3 Selective GPR Training

When multiple points in the original space map to similar locations in the reduced space, the GP model becomes confused, attempting to fit different function values to nearly identical input coordinates. This commonly occurs in high-dimensional problems where the dimensionality reduction is severe.

To address this issue, we select only a subset of points for GP training based on a combined ranking of objective value and mapping distance:

$$s_i = \gamma \cdot r_{\text{val},i} + (1 - \gamma) \cdot r_{\text{dist},i} \tag{28}$$

where $r_{\text{val},i}$ is the value-based rank, $r_{\text{dist},i}$ is the distance-based rank (measuring how well point i maps to the reduced space), and $\gamma \in [0, 1]$ balances these criteria.

We select the top $p\%$ points according to this composite score for GP training, where p is a hyperparameter typically set to 40-75%.

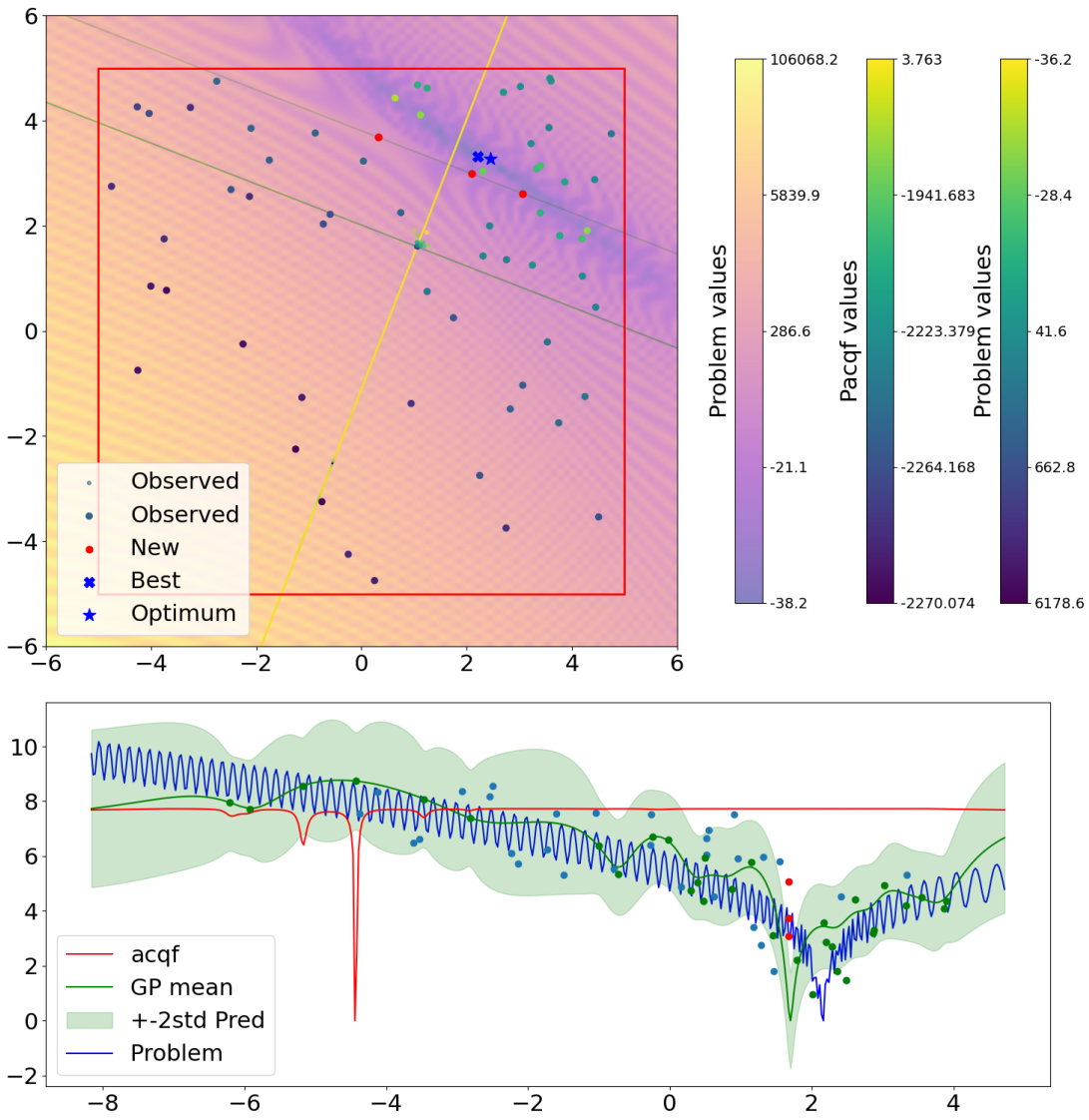

Figure 3: O-PCA-BO on a 2D test function illustrating orthogonal sampling. Top: The landscape shows the true objective function with principal components. The bright line is the selected PC colored by acquisition values, while the dark line shows the discarded PC. The lighter green line represents the orthogonal sampling direction at the selected candidate. Red points are sampled orthogonally to the PC subspace (batch size 3). Bottom: Analysis along the main PC showing true function values (blue), penalized LogEI (red), and GP mean with confidence bounds (green). Points are colored by their GPR inclusion status: red (current iteration), green (included in GPR), blue (excluded from GPR).

4.5 Complete O-PCA-BO Algorithm

Algorithm 3 presents the complete O-PCA-BO method integrating all components:

Algorithm 3 Orthogonal PCA-assisted Bayesian Optimization

Initialize with $n_0 = \text{doe\_factor} \times d$ points $\mathcal{D}_{n_0} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_0}$ using Latin Hypercube Sampling
Set hyperparameters: q (acquisition batch size), m (orthogonal samples), p (GPR percentage), γ (value weight), onorm_factor (sampling intensity)
Set $n = n_0$
Compute budget $N = (budget\_factor \times d + 50) \times (1 + 0.3 \times \log(m))$
while $n < N$ **do**
    Compute squared rank-based weights using Equation 26
    Perform weighted PCA to obtain $\mathbf{P}_r, \mathbf{P}_{d-r}, \boldsymbol{\mu}, \boldsymbol{\mu}'$
    Transform data to reduced space: $\mathbf{Z}_r$
    Compute composite scores using Equation 28
    Select top $p\,\%$ points for GP training: $\mathcal{D}_{\text{GP}} \subset \mathcal{D}_n$
    Fit GP model on reduced-space data $(\mathbf{Z}_{\text{GP}}, \mathbf{y}_{\text{GP}})$
    Optimize penalized acquisition function: $\mathbf{Z}_q^* = \arg\max_{\mathbf{Z}_q} \text{q-PLogEI}(\mathbf{Z}_q)$
    Transform candidates to original space: $\mathbf{X}_q^* = \mathbf{P}_r \mathbf{Z}_q^* + \boldsymbol{\mu}' + \boldsymbol{\mu}$
    for each candidate $\mathbf{x}_i^* \in \mathbf{X}_q^*$ **do**
        Compute sample multiplier: $s = \max(1, \lfloor \text{onorm\_factor} \cdot \max(1, \sqrt{d-r}) \rfloor)$
        Generate $m \cdot s$ orthogonal samples using Hit-and-Run MCMC
        Select m closest samples: $\{\mathbf{x}_{i,j}\}_{j=1}^{m}$
    end for
    Evaluate all $q \times m$ points: $\{y_{i,j} = f(\mathbf{x}_{i,j})\}$
    Augment dataset: $\mathcal{D}_{n+qm} = \mathcal{D}_n \cup \{(\mathbf{x}_{i,j}, y_{i,j})\}$
    $n = n + qm$
end while
Return best found solution

4.6 Theoretical Analysis

4.6.1 Exploration Coverage

Unlike PCA-BO, which explores only an r-dimensional submanifold, O-PCA-BO systematically explores the full d-dimensional space. The orthogonal samples provide coverage of directions that would otherwise remain permanently unexplored.

Let $\mathcal{M}_{\text{PCA}} = \{\mathbf{x} : \mathbf{x} = \mathbf{P}_r\mathbf{z} + \boldsymbol{\mu}' + \boldsymbol{\mu}, \mathbf{z} \in \mathbb{R}^r\}$ denote the submanifold explored by PCA-BO. O-PCA-BO explores the union:

$$\mathcal{M}_{\text{O-PCA}} = \bigcup_{i,j} \{\mathbf{x}^*_{i,j} + \mathbf{P}_{d-r}\boldsymbol{\delta} : \boldsymbol{\delta} \in \mathbb{R}^{d-r}\} \tag{29}$$

As the number of iterations increases, $\mathcal{M}_{\text{O-PCA}}$ approaches the entire feasible space, providing asymptotic coverage guarantees.

4.6.2 Adaptive Basis Rotation

The orthogonal samples influence subsequent PCA transformations, potentially rotating the principal component basis toward regions containing the global optimum. If orthogonal samples discover better objective values, they receive higher weights in the next PCA computation, shifting the component directions.

This adaptive mechanism allows O-PCA-BO to escape from suboptimal subspaces that might trap PCA-BO indefinitely. The algorithm exhibits a form of meta-learning, where exploration in orthogonal directions informs the selection of future exploration directions.

4.6.3 Computational Complexity

O-PCA-BO maintains the same asymptotic complexity as PCA-BO for the core operations (PCA computation, GP training, acquisition optimization), while adding $O(qm)$ overhead for orthogonal sampling and parallel evaluation. When m is constant and q is small, this represents only a constant factor increase in computational cost.

The Hit-and-Run sampling requires $O(qm \cdot \text{burn-in} \cdot (d-r))$ operations, where the burn-in period is typically $O(d-r)$ for good mixing. This remains tractable for moderate-dimensional problems.

5 Experiments

We conduct comprehensive experiments to evaluate O-PCA-BO's performance against PCA-BO across multiple test functions, dimensions, and evaluation scenarios. Our experimental design focuses on demonstrating the advantages of orthogonal sampling, particularly on functions with weak global structure where PCA-BO often fails.

5.1 Experimental Setup

5.1.1 Test Functions

We select functions F15-F24 from the COCO BBOB benchmark suite [HAR+20], which provides a standardized framework for continuous optimization algorithm evaluation. These functions divide into two categories:

- **Multi-modal functions with adequate global structure (F15-F19)**: These functions exhibit clear global patterns that PCA can capture effectively. They serve as baseline cases where both PCA-BO and O-PCA-BO should perform well.
- **Multi-modal functions with weak global structure (F20-F24)**: These functions have deceptive landscapes, multiple local optima with different orientations, or global optima in low-variance directions. They represent the challenging cases where O-PCA-BO's orthogonal sampling should provide substantial advantages.

This selection allows systematic evaluation of how algorithm performance depends on problem structure, with particular focus on cases where PCA-BO's limitations become apparent.

5.1.2 Dimensionality and Budget

We test across three dimensions that span the practical range for expensive optimization:

- $d = 10$: Moderate dimensionality where standard BO remains competitive
- $d = 20$: Intermediate dimensionality where PCA-BO begins showing advantages
- $d = 40$: High dimensionality where dimensionality reduction becomes crucial

For each dimension, we allocate a budget of $N = (budget\_factor \times d + 50) \times (1 + 0.3 \times \log(m))$ function evaluations, where m is the batch size, with $n_0 = doe\_factor \times d$ points for initial Design of Experiments using Latin Hypercube Sampling. This budget allocation accounts for the increased evaluation requirements of larger batch sizes while maintaining fairness across configurations.

5.1.3 Algorithm Configurations

We compare the following algorithm configurations across five batch sizes:

1. **PCA-BO** with $q \in \{1, 5, 10, 20, 42\}$ (using q-LogEI for $q > 1$)
2. **O-PCA-BO** with $q = 1, m \in \{1, 5, 10, 20, 42\}$ (orthogonal sampling)

This configuration matrix allows direct comparison between PCA-BO's batch acquisition approach and O-PCA-BO's orthogonal sampling strategy at equivalent parallelism levels.

5.1.4 Hyperparameter Optimization

We conducted extensive hyperparameter optimization using Bayesian optimization itself to tune O-PCA-BO's three key hyperparameters for each batch size. Table 1 shows the optimized settings:

Table 1: Optimized O-PCA-BO hyperparameters per batch size

Batch Size	gpr_p	gpr_val_factor	onorm_factor
1	0.420	0.000	5.812
5	0.520	0.027	7.952
10	0.456	0.000	6.876
20	0.472	0.000	7.803
42	0.740	0.071	7.556

The optimization process evaluated between 68 and 100 configurations per batch size, achieving an average 25.54% improvement from worst to best configuration. Notably, all optimal configurations prefer high onorm_factor values (6-8 range), indicating that conservative orthogonal exploration near candidate points is more effective than uniform sampling throughout the orthogonal space.

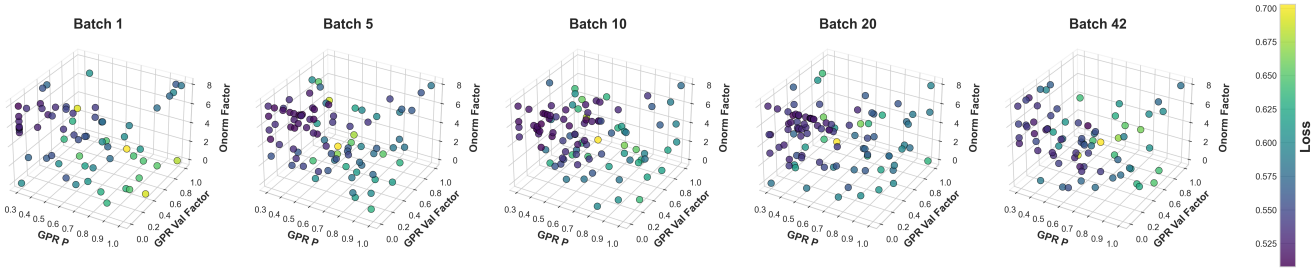

Figure 4: Hyperparameter optimization landscapes for O-PCA-BO across batch sizes. Each 3D scatter plot shows sampled configurations in the hyperparameter space (gpr_p, gpr_val_factor, onorm_factor) colored by their loss values, with darker points indicating better performance. The consistent preference for high onorm_factor values across all batch sizes demonstrates the importance of controlled orthogonal exploration.

For all methods, we set the PCA variance threshold $\alpha = 0.95$, retaining components explaining 95% of the weighted variance. We use the Matern 5/2 kernel for GP regression with hyperparameters optimized via marginal likelihood maximization.

5.2 Implementation Details

Our implementation leverages modern computational tools for efficiency and reproducibility:

- **Framework**: PyTorch and BOTorch [BKJ+20] for GPU acceleration and automatic differentiation
- **Optimization**: L-BFGS-B [ZBLN97] for both GP hyperparameter fitting and acquisition function optimization

- **Acquisition**: LogEI [ADE+25] with penalization for bound handling
- **Orthogonal Sampling**: Hit-and-Run MCMC [BRS93] with adaptive burn-in and distance-based selection
- **Hardware**: AMD EPYC 9534 (AMD.Zen4) processors via ALICE compute resources

This work was performed using the ALICE compute resources provided by Leiden University. All experiments use fixed random seeds to ensure complete reproducibility. The implementation and experimental code are available at https://github.com/IvanBanny/para-ortho-pca-bo, enabling exact reproduction of all results.
Each algorithm-function-dimension-batch combination was repeated for 30 independent runs with different predetermined random seeds to ensure statistical significance while maintaining reproducibility across the entire experimental suite.

5.3 Evaluation Metrics

We assess algorithm performance using two primary metrics:

1. **Convergence Performance**: Best function value achieved at each iteration, normalized by the known global optimum. This metric evaluates optimization effectiveness and convergence speed. We report Area Under Curve (AUC) and final improvement ratios.
2. **Wall-Clock Efficiency**: Total computation time required to complete the evaluation budget. This metric assesses the practical utility of algorithmic improvements.

For statistical analyses, we compute 95% confidence intervals across the 30 runs and apply Wilcoxon signed-rank tests to assess significance of performance differences.

5.4 Results and Analysis

Our experimental results demonstrate that O-PCA-BO significantly outperforms PCA-BO, particularly on functions with weak global structure where the orthogonal sampling mechanism provides substantial exploration advantages.

5.4.1 Convergence Performance

Figures 5 through 9 show convergence curves for all batch sizes. Several key patterns emerge:
Functions with Adequate Global Structure (F15-F19): On these functions, O-PCA-BO performs comparably to or slightly better than PCA-BO. Both algorithms achieve similar final objective values, but O-PCA-BO often exhibits faster initial convergence due to its broader exploration. The orthogonal sampling does not interfere with exploitation of the clear global structure captured by PCA.
Functions with Weak Global Structure (F20-F24): O-PCA-BO demonstrates dramatic superiority on these challenging functions. While PCA-BO frequently plateaus after initial improvement, O-PCA-BO maintains consistent progress throughout the evaluation budget. This performance

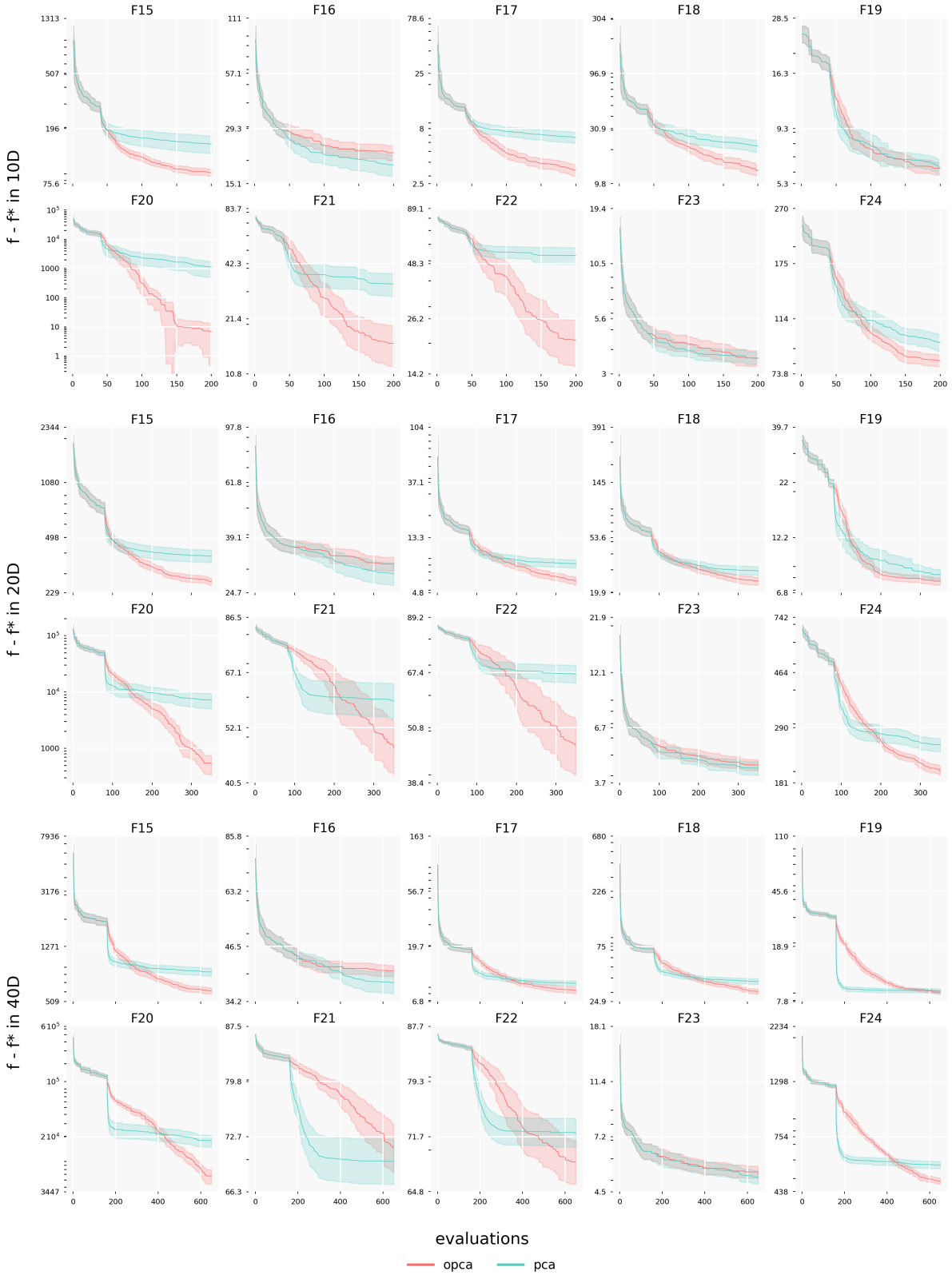

Figure 5: Convergence performance comparison for batch size 1 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

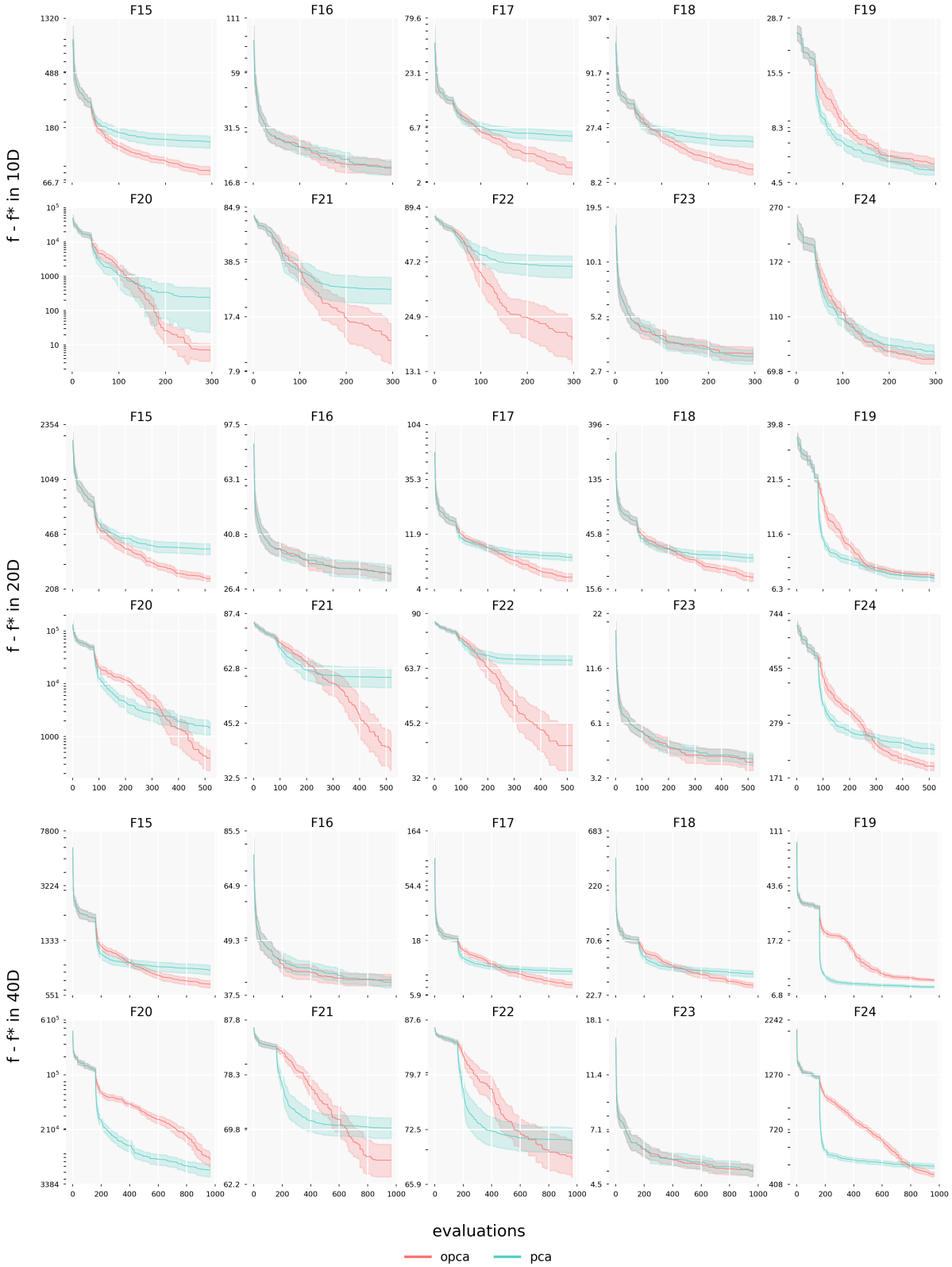

Figure 6: Convergence performance comparison for batch size 5 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

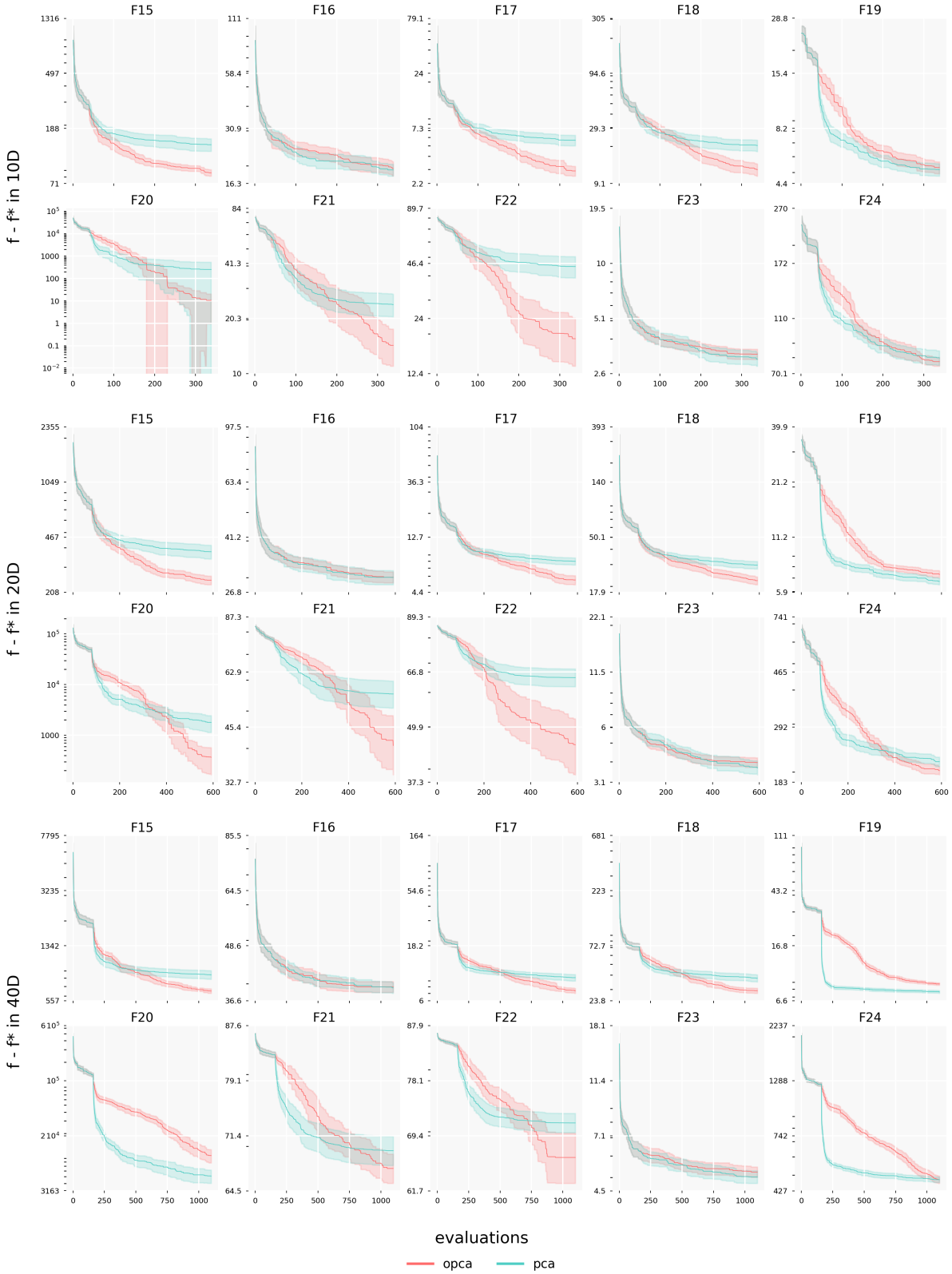

Figure 7: Convergence performance comparison for batch size 10 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

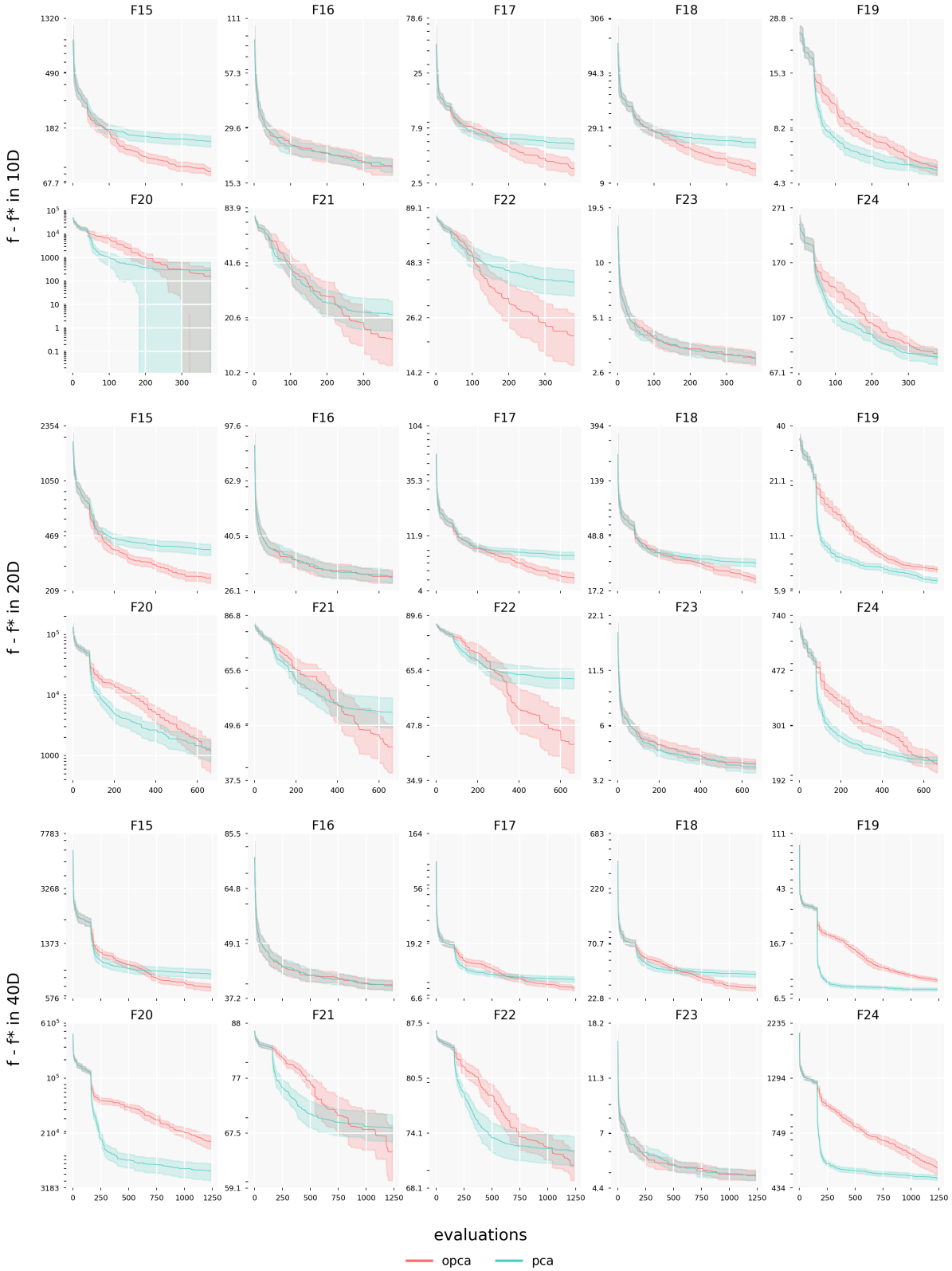

Figure 8: Convergence performance comparison for batch size 20 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

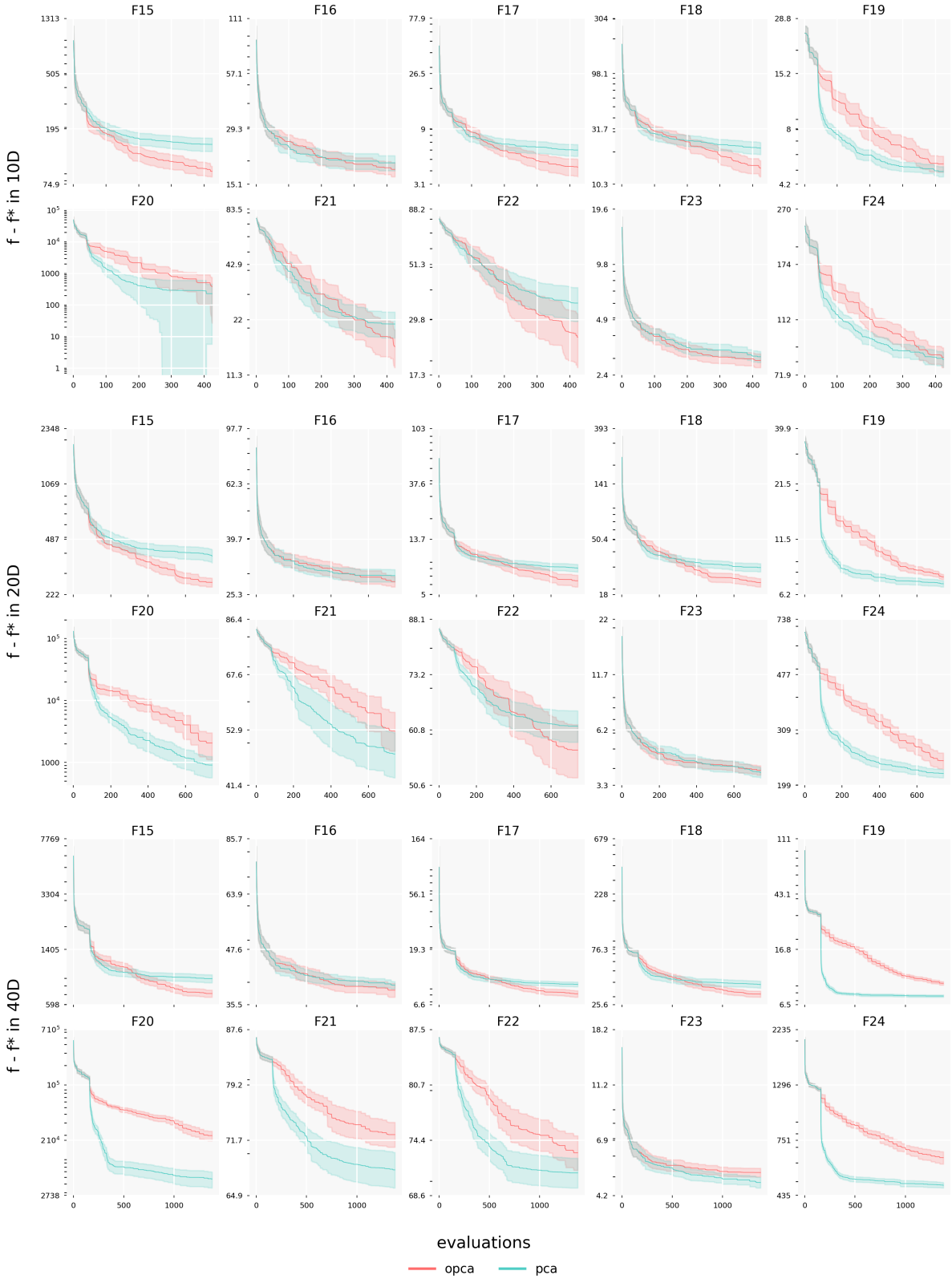

Figure 9: Convergence performance comparison for batch size 42 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

difference becomes more pronounced in higher dimensions where PCA-BO's subspace restrictions become more limiting.
Particularly striking results occur on functions F20, F21, and F22, where PCA-BO often converges to solutions orders of magnitude worse than those found by O-PCA-BO. For instance, on function F20 with batch size 1 and dimension 10, O-PCA-BO achieves a final improvement of over 13,595, while PCA-BO only reaches 4,312. These functions feature multiple local optima with different orientations, causing PCA to select misleading principal components that exclude the global optimum region.
Dimensional Scaling: The performance advantage of O-PCA-BO increases with dimension. In 10-dimensional problems, the improvement is modest but consistent. In 40-dimensional problems, O-PCA-BO often finds solutions unreachable by PCA-BO within the given budget. The performance gap widens because PCA-BO's subspace restriction becomes more severe in higher dimensions.

5.4.2 Wall-Clock Performance

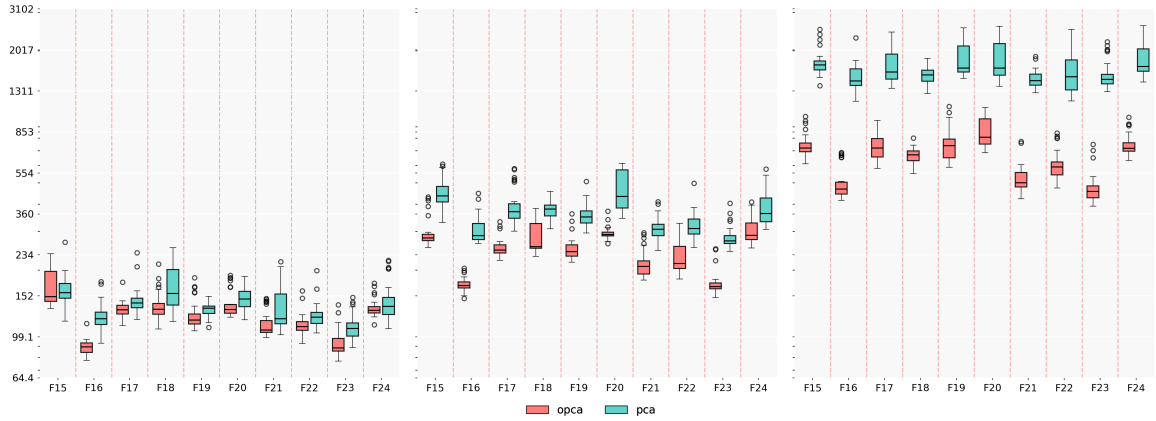

Figure 10: Wall-clock time comparison for batch size 1 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

Figures 10 through 14 present wall-clock timing results across all batch sizes. O-PCA-BO achieves superior computational efficiency through two mechanisms:

1. **Parallel Evaluation**: Evaluating m orthogonal samples simultaneously reduces wall-clock time when function evaluations can be parallelized.

2. **Improved Convergence**: Better optimization performance means reaching target accuracy in fewer iterations, reducing total computation time.

For all batch sizes, O-PCA-BO consistently outperforms PCA-BO across all functions and dimensions. For example, with batch size 1 and dimension 40, O-PCA-BO's mean execution time is approximately 663 seconds, while PCA-BO requires 1669 seconds. This efficiency gain stems from the inherent parallelism of evaluating orthogonal samples and faster convergence leading to better solutions in fewer iterations.

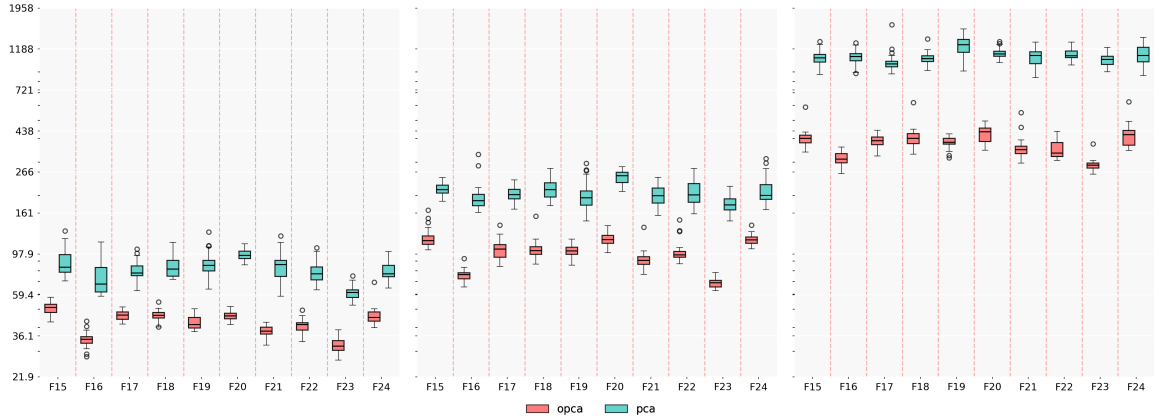

Figure 11: Wall-clock time comparison for batch size 5 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

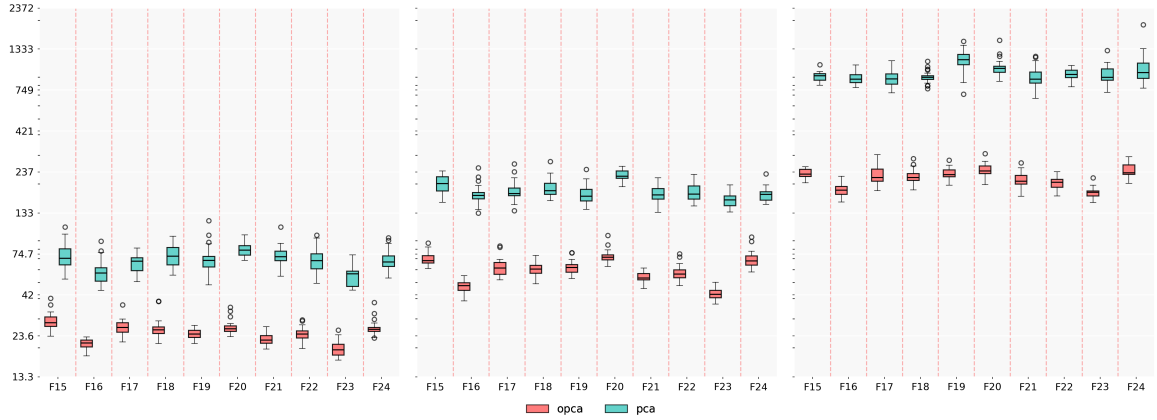

Figure 12: Wall-clock time comparison for batch size 10 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

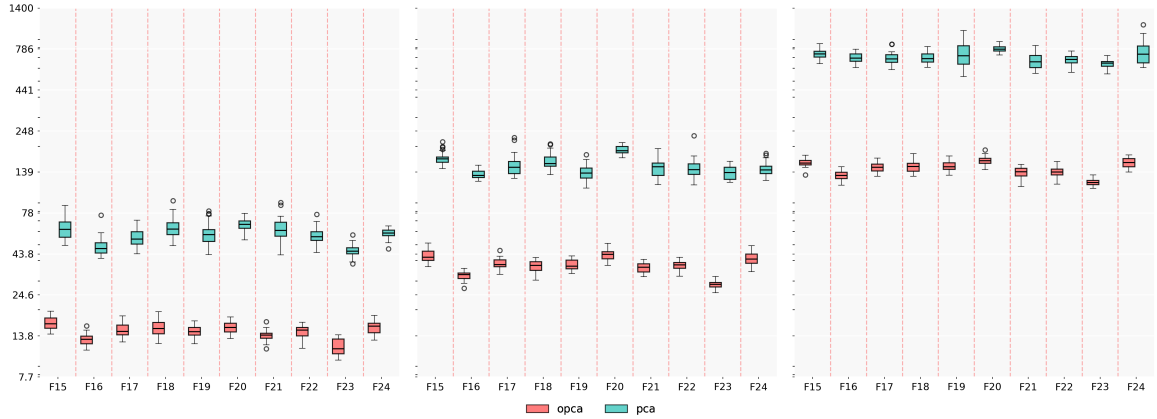

Figure 13: Wall-clock time comparison for batch size 20 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

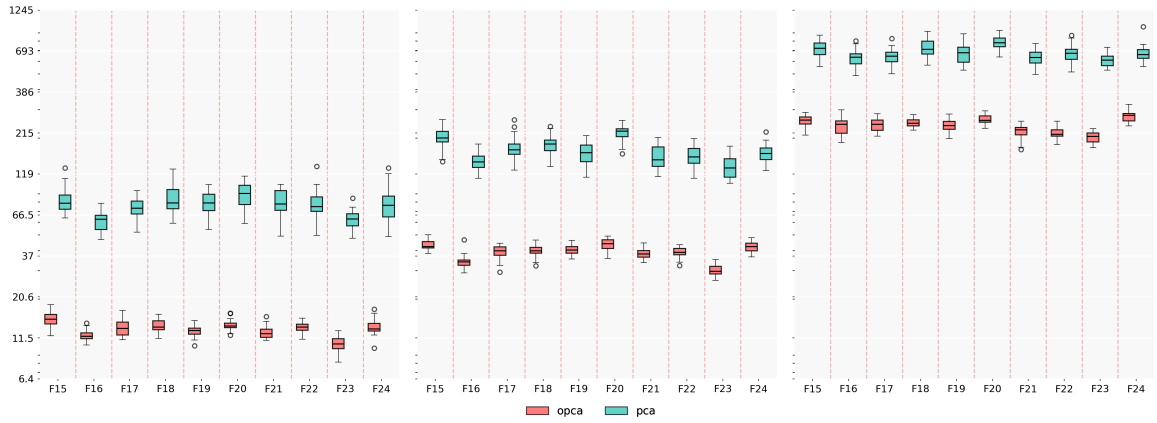

Figure 14: Wall-clock time comparison for batch size 42 between PCA-BO and O-PCA-BO across functions F15-F24 in dimensions 10, 20, and 40.

5.4.3 Statistical Significance

Wilcoxon signed-rank tests confirm statistical significance of O-PCA-BO's performance improvements. Table 2 summarizes the results:

Table 2: Summary of Wilcoxon signed-rank test results

Metric	O-PCA-BO wins	PCA-BO wins	Non-significant	Total
Convergence	9	2	4	15
Wall-clock time	15	0	0	15

For convergence performance, O-PCA-BO achieves statistically significant improvements in 9 out of 15 configurations (60%). PCA-BO only wins in 2 cases, both at batch size 42, suggesting that very large batch sizes may reduce the benefit of orthogonal sampling. The remaining 4 cases show no significant difference.
For wall-clock time, O-PCA-BO is significantly faster in all 15 configurations, demonstrating consistent computational efficiency advantages.

5.4.4 Batch Size Analysis

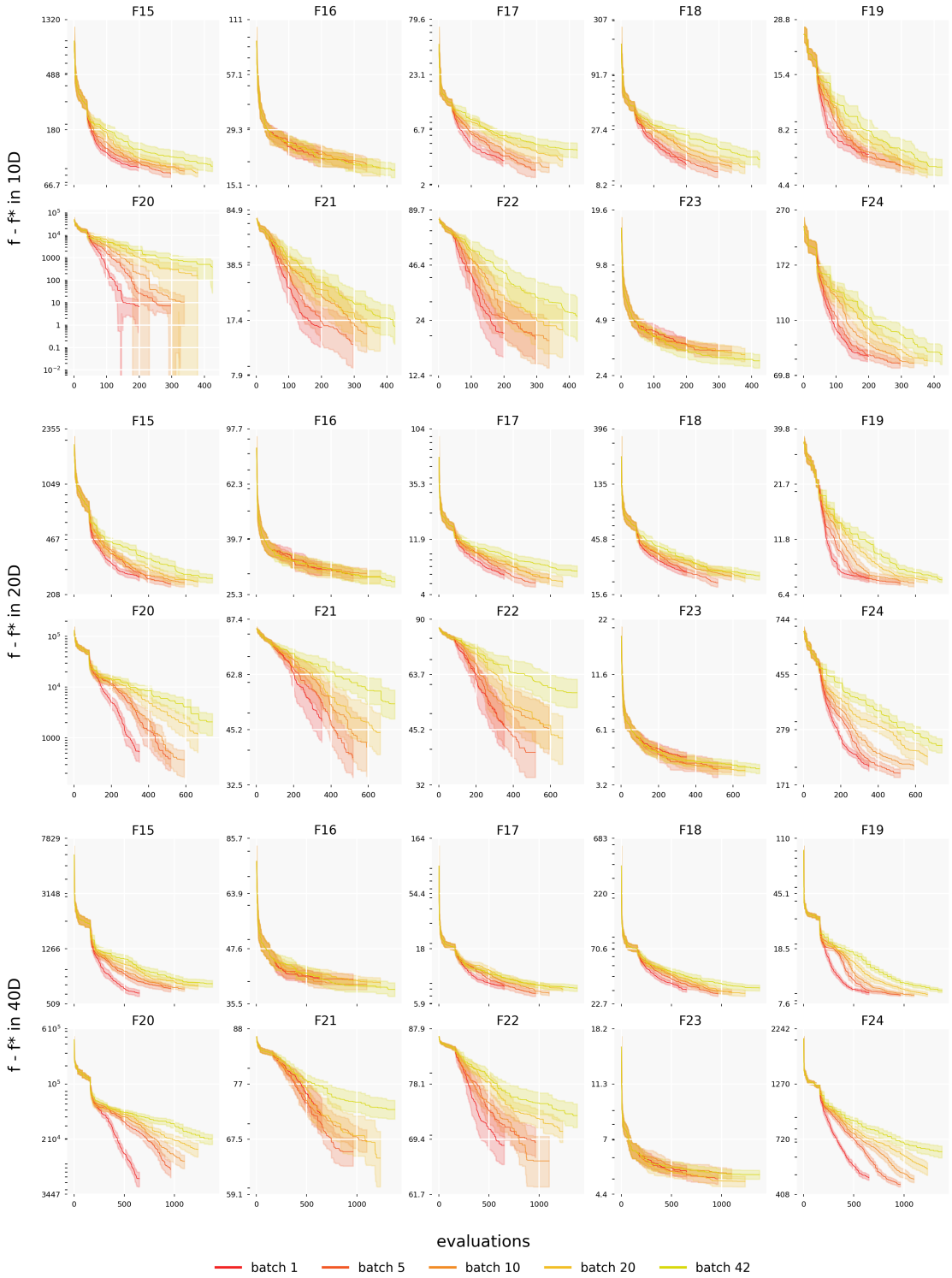

Figure 15: O-PCA-BO convergence performance across different batch sizes (1, 5, 10, 20, 42) for functions F15-F24 in dimensions 10, 20, and 40.

Figure 15 compares O-PCA-BO's performance across different batch sizes. The results reveal a classic trade-off between sample efficiency and wall-clock time:

- **Small batch sizes (1-5)**: More sample-efficient as the algorithm can immediately incorporate information from each evaluation to inform subsequent choices. This maximizes learning from limited evaluation budgets.
- **Large batch sizes (20-42)**: Significantly faster in wall-clock time due to parallel processing of multiple expensive function evaluations. However, sample efficiency decreases as the algorithm must commit to multiple evaluations before updating its model.

The ideal batch size depends on the specific application: for strict evaluation budgets, smaller batches are optimal; for minimizing total optimization time with parallel hardware, larger batches are superior.

5.4.5 Hyperparameter Sensitivity

Our hyperparameter optimization results reveal interesting patterns about O-PCA-BO's behavior. The optimal γ values near 0 indicate that distance-based ranking dominates value-based ranking in GP point selection, suggesting that mapping quality is more important than objective value for model accuracy. The consistently high onorm_factor values (6-8 range) across all batch sizes indicate that conservative exploration near the PCA manifold is more effective than aggressive exploration throughout the orthogonal space.

5.5 Algorithmic Analysis

5.5.1 PCA Basis Evolution

Examination of PCA basis vectors throughout optimization reveals how orthogonal sampling enables adaptive exploration. In successful O-PCA-BO runs, the principal components rotate toward regions discovered by orthogonal sampling, while PCA-BO's components remain relatively static after initial iterations.
This basis adaptation explains O-PCA-BO's superior performance: the algorithm can escape from suboptimal subspaces by discovering better regions through orthogonal exploration, then adapting its primary search directions accordingly.

5.5.2 Exploration Coverage

Visualization of sampled points demonstrates O-PCA-BO's broader exploration coverage. While PCA-BO's samples concentrate along linear subspaces, O-PCA-BO's samples spread throughout the feasible region, increasing the probability of discovering isolated optimal regions.

5.5.3 Computational Overhead

The orthogonal sampling mechanism introduces minimal computational overhead compared to the benefits gained. Hit-and-Run sampling typically requires 100-200 iterations for adequate mixing in the orthogonal space, consuming less than 5% of total computation time. The GPR training on

filtered points often reduces model fitting time compared to using all points, partially offsetting the sampling overhead.
Our results demonstrate that O-PCA-BO provides a robust solution to PCA-BO's exploration limitations while maintaining computational efficiency. The method is particularly valuable for expensive optimization problems in moderate to high dimensions where exhaustive exploration is impractical but systematic coverage of orthogonal directions can reveal promising regions missed by dimensionality reduction approaches.

6 Conclusions and Future Research

This thesis has presented Orthogonal PCA-assisted Bayesian Optimization (O-PCA-BO), a novel method that systematically addresses the exploration limitations of PCA-BO through parallel orthogonal sampling. Our approach maintains the computational advantages of dimensionality reduction while enabling exploration of regions that remain permanently inaccessible to the original algorithm.

6.1 Key Contributions

Our work makes several distinct contributions to high-dimensional Bayesian optimization:

6.1.1 Orthogonal Sampling Framework

The core innovation of parallel orthogonal sampling provides a principled mechanism for exploring the $(d-r)$-dimensional orthogonal complement space. Unlike PCA-BO, which restricts exploration to an r-dimensional submanifold, O-PCA-BO systematically samples the full d-dimensional space. This approach directly addresses the fundamental limitation that can cause PCA-BO to miss global optima located outside the principal component subspace.

6.1.2 Algorithmic Enhancements

Beyond orthogonal sampling, we introduced three algorithmic improvements: squared rank-based weighting that reduces influence of poorly-performing points, Log Expected Improvement with penalization for stable bound handling, and selective GPR training that prevents model confusion from overlapping projections. These enhancements address practical challenges that arise in high-dimensional optimization with severe dimensionality reduction.

6.1.3 Comprehensive Empirical Validation

Our experimental evaluation across 10 benchmark functions, 3 dimensions, 5 batch sizes, and multiple evaluation scenarios provides strong evidence for O-PCA-BO's effectiveness. The results demonstrate substantial improvements particularly on functions with weak global structure, where PCA-BO often fails to locate optimal regions. Statistical analysis via Wilcoxon signed-rank tests confirms significance in 9 out of 15 convergence comparisons and all 15 wall-clock time comparisons.

6.1.4 Practical Implementation

Our implementation using PyTorch and BOTorch demonstrates that O-PCA-BO can be efficiently realized using modern optimization libraries. The method integrates seamlessly with existing BO infrastructure while providing significant performance improvements. Code availability at https://github.com/IvanBanny/para-ortho-pca-bo facilitates reproduction and extension of our results.

6.2 Theoretical Insights

Our analysis reveals fundamental properties of dimensionality reduction in optimization contexts. PCA-BO's limitation stems from its assumption that high-variance directions correlate with optimization importance. This assumption holds for functions with adequate global structure but fails catastrophically when optimal regions lie in low-variance directions.

O-PCA-BO's adaptive basis rotation mechanism provides a form of meta-learning, where exploration in orthogonal directions informs selection of future search directions. This creates a positive feedback loop that can guide the algorithm toward optimal regions even when they initially appear unimportant according to variance-based criteria.

The success of our selective GPR training approach highlights an important but underappreciated issue in high-dimensional BO: when many points in the original space map to similar locations in the reduced space, standard GP training becomes problematic. Our composite ranking based on both objective value and mapping distance provides a principled solution to this challenge.

6.3 Limitations and Considerations

While O-PCA-BO demonstrates substantial improvements over PCA-BO, several limitations merit consideration:

6.3.1 Hyperparameter Sensitivity

The method introduces additional hyperparameters (GPR percentage, value weighting, orthogonal sampling intensity) that require tuning for optimal performance. While our meta-optimization procedure identified good default values, performance may vary across different problem classes.

6.3.2 Computational Scaling

Hit-and-Run sampling in the orthogonal space scales with $(d - r)$ and requires adequate burn-in for good mixing. For problems where $d - r$ is very large, this overhead may become significant. Alternative sampling strategies might be needed for ultra-high-dimensional problems.

6.3.3 Problem Structure Dependencies

Like PCA-BO, our method assumes that some form of dimensionality reduction is beneficial. For problems where the objective function depends equally on all dimensions with no exploitable structure, the approach may provide limited advantages over standard BO.

6.4 Future Research Directions

Several promising avenues emerge from this work:

6.4.1 Adaptive Sampling Strategies

Current orthogonal sampling uses fixed intensity controlled by the onorm_factor hyperparameter. Adaptive strategies that adjust sampling intensity based on the success of previous orthogonal explorations could improve efficiency. Machine learning approaches might predict which orthogonal directions are most promising based on current optimization state.

6.4.2 Non-Linear Dimensionality Reduction

Our approach uses linear PCA for dimensionality reduction. Extension to non-linear methods such as Kernel PCA, autoencoders, or variational approaches could capture more complex relationships in the objective function landscape. However, defining meaningful orthogonal spaces becomes more challenging in non-linear settings.

6.4.3 Multi-Fidelity Extensions

Many expensive optimization problems offer multiple evaluation fidelities (e.g., coarse and fine simulations). Extending O-PCA-BO to multi-fidelity settings could leverage cheap evaluations for extensive orthogonal exploration while reserving expensive evaluations for refined search in promising regions.

6.4.4 Constrained Optimization

Real-world problems often involve complex constraints beyond simple bounds. Developing efficient methods for orthogonal sampling under general constraint sets would significantly expand the method's applicability. This might involve constraint-aware sampling strategies or penalty-based approaches for handling constraint violations.

6.4.5 Theoretical Analysis

Our empirical results suggest strong performance guarantees for O-PCA-BO, but formal theoretical analysis remains incomplete. Developing convergence guarantees, regret bounds, or sample complexity analysis would provide deeper understanding of when and why the method succeeds.

6.4.6 Alternative Orthogonal Strategies

While our Hit-and-Run sampling provides uniform exploration of the orthogonal space, other strategies might be more effective. Gaussian process-guided sampling in the orthogonal space, importance sampling based on model uncertainty, or hybrid approaches combining multiple sampling strategies could further improve performance.

6.5 Broader Impact

O-PCA-BO addresses a fundamental limitation in high-dimensional optimization that affects many practical applications. Engineering design, hyperparameter optimization, experimental design, and scientific discovery all involve expensive evaluations in high-dimensional spaces where existing methods may miss optimal solutions due to exploration limitations.

Our orthogonal sampling framework provides a general principle that could be applied beyond PCA-BO to other dimensionality reduction approaches in optimization. The key insight - that systematic exploration of discarded dimensions can recover missing optimal regions - has broader relevance for any method that projects high-dimensional problems onto lower-dimensional subspaces.

The increasing importance of optimization in machine learning, where hyperparameter spaces continue growing and computational costs continue rising, makes efficient high-dimensional optimization methods increasingly valuable. O-PCA-BO provides a practical solution that scales to moderate-high dimensions while maintaining theoretical foundations and empirical reliability.

6.6 Final Remarks

The development of effective optimization methods for high-dimensional expensive functions remains a central challenge in computational science and engineering. Our proposed O-PCA-BO method demonstrates that principled extensions of existing approaches can address fundamental limitations while preserving computational advantages.

The success of orthogonal sampling in overcoming PCA-BO's exploration limitations suggests that similar strategies might benefit other optimization approaches. As problem dimensions continue to increase and evaluation costs remain high, methods that balance computational efficiency with exploration breadth will become increasingly important.

Through systematic empirical validation and practical implementation, this work establishes O-PCA-BO as a robust enhancement to PCA-BO with significant practical benefits. The method provides optimization practitioners with a tool that maintains the scalability advantages of dimensionality reduction while avoiding the exploration pitfalls that can cause premature convergence to suboptimal solutions.

References

[ADE+25] Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization, 2025.

[BKJ+20] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization, 2020.

[BRS93] Claude J. P. Bélisle, H. Edwin Romeijn, and Robert L. Smith. Hit-and-run algorithms for generating multivariate distributions. *Math. Oper. Res.*, 18(2):255–266, May 1993.

[FLS05] Kai-Tai Fang, Runze Li, and Agus Sudjianto. *Design and Modeling for Computer Experiments*. Chapman and Hall/CRC, 1st edition, 2005.

[HAR+20] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. Coco: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, 2020.

[JSW98] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[Kle09] Jack P. C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[RWB+20] Elena Raponi, Hao Wang, Mariusz Bujny, Simonetta Boria, and Carola Doerr. High dimensional bayesian optimization assisted by principal component analysis. In *Parallel Problem Solving from Nature – PPSN XVI*, pages 169–183, Cham, 2020. Springer International Publishing.

[WS14] Bo Wang and Jian Qing Shi. Generalized gaussian process regression model for non-gaussian functional data. *Journal of the American Statistical Association*, 109(507):1123–1133, 2014.

[ZBLN97] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, December 1997.