CORRELATION-AWARE VARIABLE SELECTION FOR HIGH-DIMENSIONAL BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian optimization (BO) is a sample-efficient framework for expensive black-box optimization, but its performance often deteriorates in high-dimensional spaces. Variable selection is a natural remedy because it reduces the search to a smaller active subspace. However, existing methods typically rely on axis-aligned importance scores, which do not model how variables act together, making them vulnerable when the objective is governed by correlated groups. We propose Correlation-Aware Variable Selection (CAVS), which learns variable importance from a lightweight Mahalanobis kernel with a low-rank-plus-diagonal structure. This design captures inter-variable correlations while avoiding the quadratic cost of a full metric. We prove that CAVS can obtain a sublinear cumulative regret bound. Experiments show that CAVS remains robust even when the objective is expressed in non-axis-aligned coordinates, where axis-aligned methods degrade, and achieves the best average rank among a comprehensive set of competitive baselines.

1 INTRODUCTION

Expensive black-box optimization is central to materials design (Frazier & Wang, 2016), engineering (Shields et al., 2021), robotics (Shields et al., 2021), and hyperparameter optimization (Seong-Eun et al., 2026), where evaluations are costly and gradients unavailable. Bayesian optimization (BO) (Shahriari et al., 2015; Frazier, 2018; Garnett, 2023) uses a surrogate model to select candidates sequentially, balancing exploration and exploitation.

Despite its low-dimensional success, BO faces significant challenges in high-dimensional spaces, where distance concentration weakens kernel discriminative power and renders GP posteriors unreliable under limited budgets. Common solutions exploit low-dimensional structures via additive models, random embeddings, or trust regions (Kandasamy et al., 2015; Mutný & Krause, 2018; Wang et al., 2016; Letham et al., 2020; Papenmeier et al., 2022; Eriksson et al., 2019). Variable selection (Li et al., 2017; Eriksson & Jankowiak, 2021; Song et al., 2022; Hoang et al., 2025)—the approach we pursue—directly identifies active dimensions and restricts BO to a reduced subspace. This is particularly attractive when only a few coordinates drive the objective, offering improved interpretability and computational efficiency.

The dominant variable selection view in high-dimensional BO is largely axis-aligned. LassoBO (Hoang et al., 2025), for instance, uses inverse lengthscales in an automatic relevance determination (ARD) kernel as variable importance scores. Such scores work well for nearly independent active variables, but their diagonal kernel metric cannot directly model correlations. This becomes problematic for coordinated groups, where one variable matters mainly through interaction with another or several variables redundantly represent the same latent combination. In such settings, axis-aligned scores can still assign nontrivial marginal sensitivity to individual coordinates, yet they may rank and recover the true active groups less reliably among many inactive dimensions, leading to weaker active-subspace selection under limited evaluation budgets.

This limitation is not merely conceptual. Figure 1 illustrates a rotation stress test: LassoBO (Hoang et al., 2025) and SAASBO (Eriksson & Jankowiak, 2021) are fitted on synthetic functions before and after an orthogonal rotation of the input space. The rotation preserves intrinsic difficulty but destroys its axis-aligned sparsity structure; after rotation, both methods exhibit a marked increase in the average rank of true active variables and a clear drop in group recall. This empirical observation suggests that variable-selection methods should account for correlated directions, not just identify important coordinates in a fixed axis-aligned representation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

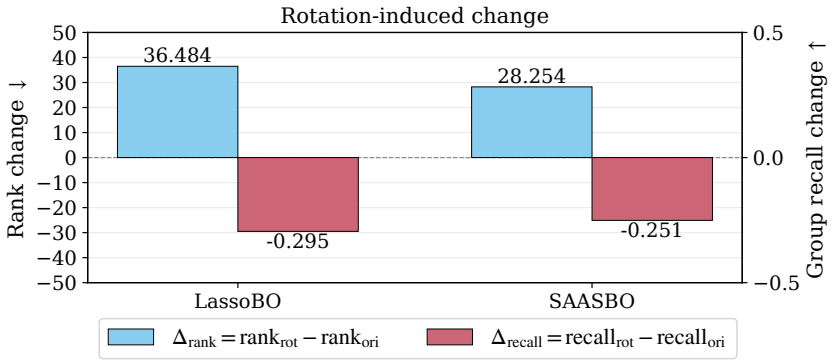

Figure 1: Rotation-induced change in variable identification. Positive rank change (lower is better) and negative recall change (higher is better) indicate degradation under rotation.

In this paper, we propose Correlation-Aware Variable Selection (CAVS) for high-dimensional Bayesian optimization. The central idea is to incorporate variable correlations into the surrogate kernel, and then derive variable importance from the resulting correlation-aware geometry. A full Mahalanobis kernel naturally supports this goal but introduces $O(D^2)$ parameters (where D denotes the number of dimensions), which is costly and prone to overfitting in BO's small-data regime. CAVS therefore uses a lightweight Mahalanobis kernel with a low-rank-plus-diagonal matrix,

$$\mathbf{M} = \text{diag}(\ell^{-1}) + \mathbf{R}\mathbf{R}^\top,$$

where the diagonal term preserves per-variable scales and the low-rank term captures correlations. We summarize our contributions as follows.

- We propose CAVS, a correlation-aware variable selection method for high-dimensional BO that learns variable importance from a lightweight low-rank-plus-diagonal Mahalanobis kernel.
- We provide theoretical analysis of CAVS, proving a sublinear cumulative regret bound.
- We show that CAVS achieves the best average rank across diverse real-world benchmarks, while further experiments validate its correlation-aware scoring mechanism.

2 Preliminaries

We review GP-based BO and the kernel geometry used here; broader treatments of BO and GP surrogates appear in (Shahriari et al., 2015; Frazier, 2018; Garnett, 2023). We maximize an expensive black-box function $f : \mathcal{X} \to \mathbb{R}$ over $\mathcal{X} = [0,1]^D$ within an evaluation budget T; its analytic form and gradients are unavailable. Each query returns $y = f(\mathbf{x}) + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, yielding data $\mathcal{D}_t = \{(\mathbf{x}_j, y_j)\}_{j=1}^t$ after t evaluations. BO uses these observations to select informative queries and find a high-value point.

Gaussian-process surrogate and Bayesian optimization. A Gaussian process (GP) is specified by a mean $\mu(\mathbf{x})$ and positive-definite covariance kernel $k(\mathbf{x}, \mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}\big(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big).$$

Given $\mathcal{D}_t$, GP regression yields posterior mean $\mu_t(\mathbf{x})$ and variance $\sigma_t^2(\mathbf{x})$; marginal likelihood fits kernel hyperparameters by balancing data fit and complexity. BO maximizes an acquisition function $\alpha_t(\mathbf{x})$, evaluates the selected point, and updates the data. Examples include expected improvement (EI) (Mockus, 1994; Jones et al., 1998), probability of improvement (PI) (Kushner, 1964), and upper confidence bound (UCB) (Srinivas et al., 2012), which favors high mean or uncertainty.

Kernel geometry. The kernel determines how the GP transfers information between nearby points. A common choice is the squared-exponential kernel with automatic relevance determination (ARD):

$$k_{\text{ARD}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i - x_i')^2}{\ell_i^2}\right), \tag{1}$$

where $\ell_i > 0$ denotes the lengthscale of the i-th coordinate and $\sigma_f^2 > 0$ is the signal variance. Shorter lengthscales imply faster variation along the corresponding coordinate, so inverse lengthscales are often interpreted as axis-aligned variable importance scores (Eriksson & Jankowiak, 2021; Hoang et al., 2025). Equivalently, Eq. (1) can be written with a diagonal metric $\mathbf{\Lambda} = \mathrm{diag}(\ell_1, \ldots, \ell_D)$ as $k_{\mathrm{ARD}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-(\mathbf{x} - \mathbf{x}')^\top \mathbf{\Lambda}^{-1}(\mathbf{x} - \mathbf{x}')/2\right)$. Because $\mathbf{\Lambda}$ is diagonal, ARD measures smoothness independently along each coordinate and cannot represent directions such as $x_0 + x_1$ or $2x_0 - x_1$ as single correlated directions.

To represent such correlated directions, a Mahalanobis kernel (Letham et al., 2020) replaces the diagonal metric with a positive semidefinite matrix:

$$k_{\mathrm{Mah}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top \mathbf{M}(\mathbf{x} - \mathbf{x}')\right),$$

Here $\mathbf{M} \succeq 0$ replaces ARD's diagonal metric, scaling coordinates along the diagonal and encoding correlations off the diagonal. CAVS structures $\mathbf{M}$ for efficient fitting and correlation-aware scoring.

3 Related Work

High-dimensional BO uses additive decompositions (Kandasamy et al., 2015; Mutný & Krause, 2018; Ziomek & Bou-Ammar, 2023), lower-dimensional embeddings (Wang et al., 2016; Nayebi et al., 2019; Letham et al., 2020; Papenmeier et al., 2022), and adaptive trust regions (Eriksson et al., 2019) to make search tractable. GIT-BO (Yu et al., 2026) instead uses TabPFN v2 (Hollmann et al., 2025) as a surrogate and its predictive gradients to construct an active subspace. These approaches reduce the effective dimension for modeling or acquisition optimization.

Vanilla BO (Hvarfner et al., 2024) and related studies (Xu et al., 2025; Papenmeier et al., 2025) show that robust lengthscale priors and dimension-aware scaling can remain competitive in high dimensions, underscoring the importance of kernel geometry even for sparse objectives.

Variable-selection methods include random coordinate subsampling in Dropout (Li et al., 2017), subset search in MCTS-VS (Song et al., 2022), sparse inverse-lengthscale priors in SAASBO (Eriksson & Jankowiak, 2021), and L_1-regularized inverse-lengthscale estimation with dynamic selection in LassoBO (Hoang et al., 2025). LassoBO provides an iterative subspace workflow, but its ARD RBF scores cannot capture correlated joint effects.

Kernel learning provides the complementary perspective. ARD kernels encode per-coordinate smoothness but ignore cross-variable correlations. Mahalanobis kernels replace the diagonal metric with a positive semidefinite matrix and can represent correlated directions. ALEBO (Letham et al., 2020) makes this explicit in an embedding setting: If the true low-dimensional function is generated by an ARD kernel in the intrinsic subspace, then the induced covariance after linear embedding is not axis-aligned and should use a Mahalanobis form. Thus a multiplicative ARD kernel in the true subspace becomes a Mahalanobis kernel after embedding: Euclidean distance should be replaced by Mahalanobis distance when the surrogate geometry is oblique rather than axis-aligned. CAVS adopts the same insight, but applies it in the original high-dimensional space for variable selection rather than inside a fixed embedding. ALEBO depends on a projection matrix and optimizes the acquisition function, whereas CAVS learns a structured Mahalanobis metric directly in the ambient space and reads variable importance from it.

4 Proposed Method

In this section, we introduce CAVS, a correlation-aware variable selection method that replaces axis-aligned importance scoring with a metric learned from a structured Mahalanobis kernel. The core idea is to fit a GP with a Mahalanobis metric $\mathbf{M}$ that encodes both per-variable scales and cross-variable interactions, and then derive importance scores directly from its diagonal entries. An overview of the proposed CAVS framework is illustrated in Figure 2.

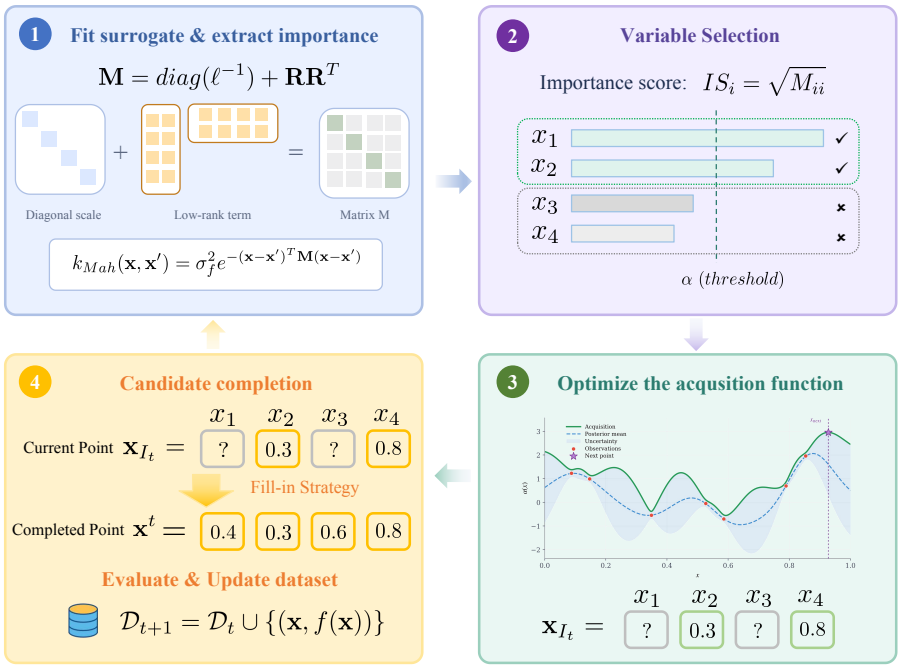

Figure 2: Overview of CAVS. At each iteration, the surrogate is refit, importance scores are computed from the learned metric, variables are selected, the acquisition function is optimized in the active subspace with completion of inactive coordinates, and the new observation updates the dataset.

4.1 A Lightweight Mahalanobis Kernel

A natural way to learn the Mahalanobis distance metric matrix is to parameterize a positive semidefinite matrix directly. Two common parameterizations are:

1. **Cholesky form (Dense Mahalanobis kernel):** $\mathbf{M} = \mathbf{L}\mathbf{L}^\top$, where $\mathbf{L}$ is lower triangular. This ensures $\mathbf{M} \succeq 0$ and is used in dense full-space Mahalanobis kernels.
2. **Low-rank form (Light Mahalanobis kernel without diagonal):** $\mathbf{M} = \mathbf{R}\mathbf{R}^\top$, where $\mathbf{R} \in \mathbb{R}^{D\times d}$ with $d \ll D$. This yields a rank-d metric with fewer parameters.

However, both forms have limitations in high-dimensional BO. The dense Mahalanobis kernel has $O(D^2)$ parameters, increasing memory and time costs and overfitting risk with limited observations. While the low-rank parameterization cuts computation by reducing parameters, it discards independent per-variable scales. Since all directional sensitivity of the objective must be represented through a unified latent subspace, this structure suffers from limited expressive power.

CAVS therefore combines the two ideas in a low-rank-plus-diagonal structure:

$$\mathbf{M} = \mathrm{diag}(\ell^{-1}) + \mathbf{R}\mathbf{R}^\top, \tag{2}$$

where $\ell = (\ell_1, \ldots, \ell_D)$ contains positive lengthscale parameters and $R \in \mathbb{R}^{D\times d}$ is low-rank with $d \ll D$. The diagonal part inherits the ability of ARD kernels to model per-variable scale differences, while the low-rank part captures correlations using only $O(dD)$ parameters. The sum is positive semidefinite, as required for a valid Mahalanobis kernel.

4.2 Hyperparameter Estimation

We estimate the lightweight kernel parameters $\theta = (\sigma_f^2, \ell, \mathbf{R})$ from $\mathcal{D}_t$ by minimizing penalized log marginal likelihood. For observations $\mathbf{y} = [y_1, \ldots, y_t]^\top$ and kernel matrix $\mathbf{K}$ with entries $K_{ij} = k_{\mathrm{Mah}}(\mathbf{x}_i, \mathbf{x}_j)$, the unregularized log marginal likelihood is

$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^\top (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K} + \sigma^2\mathbf{I}| - \frac{t}{2}\log(2\pi).$$

The first term measures data fit through the Mahalanobis norm of the residuals in feature space; the second term penalizes model complexity through the log determinant of the covariance; and the constant term does not affect optimization. Following the sparse-active-dimension assumption common in high-dimensional BO, we introduce an L_1 penalty over entries of the metric matrix $\mathbf{M}$ and minimize the composite objective below

$$U_t(\theta) = -\log p(\mathbf{y} \mid \mathbf{X}, \theta) + \lambda \sum_{i,j \in [D]} |M_{ij}|,$$

Here λ encourages sparse metric entries and hence sparse importance patterns in high dimensions.

4.3 Correlation-Aware Importance Score

After learning the Mahalanobis metric $\mathbf{M}$, we obtain a natural correlation-aware variable importance score. Intuitively, a variable is important if the function varies rapidly along that coordinate. For a GP with kernel k_{Mah}, the magnitude of partial derivatives is controlled by the diagonal entries of the metric $\mathbf{M}$, as formalized in Lemma 4.1. A larger $\sqrt{M_{ii}}$ indicates the GP sample path varies rapidly along the corresponding dimension, implying the variable strongly influences the objective function. Conversely, a smaller $\sqrt{M_{ii}}$ suggests the dimension is relatively flat and less informative.

Lemma 4.1. *For a GP with Mahalanobis kernel k_{Mah}, let $IS_i = \sqrt{M_{ii}}$. For any $L > 0$,*

$$P\left(\left|\frac{\partial f}{\partial x_i}\right| \leq IS_i \cdot L\right) \geq 1 - \exp\left(-\frac{L^2}{2\sigma_f^2}\right).$$

We therefore define the correlation-aware score for variable x_i as:

$$IS_i = \sqrt{M_{ii}}. \tag{3}$$

Unlike conventional axis-aligned importance scores relying solely on independent lengthscale parameters, IS_i integrates per-variable marginal variation and cross-variable correlation information encoded in the low-rank metric structure. Although IS_i is read from a diagonal entry, it is not merely an ARD inverse lengthscale. Under Eq. (2), $M_{ii} = \ell_i^{-1} + (\mathbf{R}\mathbf{R}^\top)_{ii} = \ell_i^{-1} + \|\mathbf{R}_{i,:}\|_2^2$, and thus the score combines an axis-aligned scale with a contribution from the low-rank factor. Because R is learned jointly from all variables, $(\mathbf{R}\mathbf{R}^\top)_{ii}$ reflects how strongly coordinate x_i participates in the correlated directions captured by the kernel. In other words, a variable can receive a high score either because it varies quickly on its own or because it is involved in a jointly active direction. The full derivation and proof of Lemma 4.1 is provided in Appendix A.1

4.4 Variable Selection and Candidate Completion

Variable selection. At each iteration, CAVS sorts variables by IS_i in descending order and selects variables until their cumulative importance reaches a threshold α. Concretely, let $IS_{(1)} \geq \cdots \geq IS_{(D)}$ be the sorted scores and $S = \sum_{i=1}^{D} IS_i$. Starting from the largest score, variables are added to I_t until $\sum_{i \in I_t} IS_i / S \geq \alpha$. To avoid overly large subspaces, the number of selected variables is clipped to the interval $[1, \lfloor \gamma D \rfloor]$. This rule mirrors the spirit of LassoBO (Hoang et al., 2025), which also converts estimated importance into a sparse active set, but replaces axis-aligned inverse lengthscales with the Mahalanobis-based score in Eq. (3).

Candidate completion. The acquisition function is optimized only over the selected variables in the active set I_t, while the black-box objective requires full D-dimensional inputs. Following LassoBO (Hoang et al., 2025), we fill unselected inactive dimensions with fixed background coordinate vectors, which supply constant values for inactive variables during acquisition maximization. Copying inactive entries from the current best point favors exploitation yet risks local stagnation; uniform random sampling over inactive dimensions boosts exploration but discards historical information. We combine both strategies to build $N_t + 1$ background candidates per iteration: One vector from the inactive subspace of the best observed point (denoted as $\mathbf{x}^{t-1,*}$), plus N_t uniformly sampled inactive vectors over $[0,1]^{D-|I_t|}$. Let $\mathcal{Z}_t$ store these N_t uniform samples, and define the full background pool $\mathcal{B}_t = \mathcal{Z}_t \cup \{\mathbf{x}^{t-1,*}_{[D]\setminus I_t}\}$. For each fixed background vector $\mathbf{x}_{\text{bg}} \in \mathcal{B}_t$, we optimize the

acquisition function over active coordinates:

$$\mathbf{x}^t = \arg \max_{\mathbf{x}_{I_t} \in [0,1]^{|I_t|}} \max_{\mathbf{x}_{\text{bg}} \in \mathcal{B}_t} \alpha_t([\mathbf{x}_{I_t}, \mathbf{x}_{\text{bg}}]).$$

The background vector yielding the highest acquisition value is combined with the optimized active coordinates to form the full D-dimensional candidate. This scheme balances exploitation and exploration, reusing LassoBO's subspace optimization pipeline with our correlation-aware importance scoring as the core modification. Overall, the CAVS procedure is summarized in Algorithm 1.

5 Theoretical Analysis

In this section, we analyze the cumulative regret of CAVS, which measures how much utility the algorithm loses relative to the optimum over a finite horizon. Formally, cumulative regret is defined as $R_T = \sum_{t=1}^{T} (f^* - f(\mathbf{x}_t))$ where f^* is the maximum value of f. Sublinear growth of R_T in T indicates that the average regret vanishes asymptotically.

CAVS inherits the iterative active subspace optimization paradigm of LassoBO (Hoang et al., 2025), allowing us to readily leverage its rigorous regret analysis framework while substituting the axis-aligned inverse-lengthscale score with our score in Eq. (3). To facilitate the regret derivation, we introduce Assumption 5.1, a uniform tail bound on partial derivatives over the domain $\mathcal{X}$, which establishes a connection between our importance score and the coordinate-wise smoothness of the objective. This assumption is commonly used in cumulative regret analysis of BO (Srinivas et al., 2012). Note that it is different from Lemma 4.1, which establishes pointwise probabilistic bounds on partial derivatives of GP sample paths.

Assumption 5.1. *For constants $a, b > 0$ and any $L > 0$, the partial derivatives of f satisfy, for all $i \in [D]$,*

$$P\left(\sup_{\mathbf{x} \in \mathcal{X}} \left| \frac{\partial f}{\partial x_i} \right| < \sqrt{M_{ii}} L\right) \geq 1 - a e^{-(L/b)^2}. \tag{4}$$

Under this assumption, CAVS regret splits into a standard GP-UCB term on the selected subspace and an extra term from fixing inactive coordinates during acquisition optimization.

Theorem 5.1. *Let $m^* = \max_{i \in [D]} M_{ii}$ and $\tilde{n}_t = \max_{1 \leq k \leq t} |I_k|$. Given $\delta \in (0,1)$ and Assumption 5.1, let $\beta_t = 2 \log\left(\frac{\pi^2 t^2}{\delta}\right) + 2\tilde{n}_T \log\left(4\sqrt{m^*} b \tilde{n}_T \sqrt{\log\left(\frac{12Da}{\delta}\right)} t^2\right)$ and $C = \max_{t \in [T]} 2b\sqrt{\log\left(\frac{2Da}{\delta}\right)} \left(\Gamma(D - |I_t| + 1)\right)^{\frac{1}{D-|I_t|}} \log\left(\frac{6}{\delta}\right)$, where $\Gamma(k) = (k-1)!$ denotes the gamma function for integer inputs. Given $\delta \in (0,1)$, under Assumption 5.1, the cumulative regret of CAVS is bounded with probability at least $1 - \delta$:*

$$R_T \leq \sqrt{\beta_T C_1 T \gamma_T} + \frac{\pi^2}{6} + C\sqrt{\left(\sum_{t=1}^{T} \sum_{i \in [D] \setminus I_t} M_{ii}\right) \left(\sum_{t=1}^{T} \left(\frac{1}{N_t}\right)^{\frac{2}{D-|I_t|}}\right)},$$

Here $C_1 = 8/\log(1 + \sigma^{-2})$; γ_T is the maximum information gain from T queries to f.

The first two terms in Theorem 5.1 arise from optimism-based regret analysis on the active subspace. The additional square-root term measures how much importance mass (i.e. $\sum_{t=1}^{T} \sum_{i \in [D] \setminus I_t} M_{ii}$) remains outside the selected set and how aggressively the completion strategy explores inactive coordinates through the background sample size N_t. Larger N_t suppresses this extra regret component, shrinking the overall bound of R_T. The score enters the bound through M_{ii}: better variable selection reduces the penalty from ignored coordinates.

Corollary 5.1. *Under the conditions in Theorem 5.1, if $N_t = \lceil t^{1/n} \rceil$ with $n \in \mathbb{N}$ and $n > 1$, then with probability at least $1 - \delta$,*

$$R_T \leq \sqrt{\beta_T C_1 T \gamma_T} + \frac{\pi^2}{6} + C\sqrt{c_0}\sqrt{\sum_{t=1}^{T} \sum_{i \in [D] \setminus I_t} M_{ii}} \cdot T^{\frac{1}{2} - \frac{1}{n(D-1)}},$$

for a constant $c_0 > n(D-1)/(n(D-1) - 2)$.

Corollary 5.1 shows that with an appropriate background schedule, the completion-induced term grows sublinearly and the overall regret remains sublinear, i.e., $\lim_{T\to\infty} R_T/T = 0$. The proofs of Theorem 5.1 and Corollary 5.1 are provided in Appendix A.2

6 EXPERIMENTS

We evaluate CAVS on synthetic and real-world benchmarks against competitive baselines.

6.1 EXPERIMENTAL SETUP

Benchmarks. We first use synthetic functions to assess kernel efficiency, variable scoring, and rotation robustness. For end-to-end optimization performance, we then use eight real-world tasks used in high-dimensional BO: MOPTA08 (124D) (Eriksson & Jankowiak, 2021) for vehicle lightweight design, Rover (60D) (Wang et al., 2017) for robot trajectory planning, Car Mazda (148D) (Kohira et al., 2018) for automotive design optimization, Lasso-DNA (180D) (Sehic et al., 2022) for DNA sequence analysis, HalfCheetah (102D) (Wang et al., 2019), Ant (888D) (Wang et al., 2020), and Swimmer (16D) (Todorov et al., 2012) for robot locomotion control, and MSLR (136D) (Qin et al., 2010) for web search ranking. Details are provided in Appendix A.3. These tasks cover diverse dimensions and structures, forming a comprehensive testbed.

Baselines. We group the compared methods into three categories:

- **Methods without explicit variable selection.** Random Search (RS), standard BO (Srinivas et al., 2012), TuRBO (Eriksson et al., 2019), ALEBO (Letham et al., 2020), Vanilla BO (Hvarfner et al., 2024), and GIT-BO (Yu et al., 2026).
- **Variable selection methods.** Dropout (Li et al., 2017) samples coordinate subsets, SAASBO (Eriksson & Jankowiak, 2021) encourages sparse variable relevance, and MCTS-VS (Song et al., 2022) searches over candidate variable subsets.
- **Full-space modeling with iterative variable selection.** LassoBO (Hoang et al., 2025) and CAVS fit a GP over all original coordinates to estimate variable importance, then optimize the acquisition function over selected variables. Retaining all coordinates in the model allows currently unselected variables to be reconsidered in subsequent iterations.

Implementation. All methods are implemented with BoTorch (Balandat et al., 2020). CAVS uses UCB as the acquisition function, and others follow the standard settings used by their corresponding baselines. Following LassoBO (Hoang et al., 2025), we set the regularization coefficient $\lambda = 10^{-3}$ and the subspace size clipping ratio $\gamma = 0.6$. In all of our experiments, we fix the low-rank dimension at $d = 10$ and the variable selection threshold at $\alpha = 0.9$. As shown in Appendix A.4 and Appendix A.5, the overall performance is rather robust with respect to these parameter settings. Here d is the rank of $\mathbf{R}$, whereas the number of selected variables is determined by the importance scores and threshold α. We choose $d = 10$ to retain enough capacity for the dominant correlated directions without introducing unnecessary fitting cost. The sensitivity experiment in Appendix A.4 shows favorable optimization performance at this setting. Each task uses a budget of 250 evaluations and five independent seeds, so the comparison emphasizes sample efficiency under a limited evaluation budget. **Empirical regret reference.** For each benchmark, we take the best value observed across all methods, seeds, and evaluations under its maximization convention: $f_{\mathrm{ref}} = \max_{m,s,t} f(\mathbf{x}_{m,s,t})$. The final regret for method m and seed s is $r_{m,s} = f_{\mathrm{ref}} - \max_{1\le t\le T} f(\mathbf{x}_{m,s,t})$. This is an empirical reference, not a known global optimum. Mean minus one standard deviation can fall below zero even though each seed's regret is nonnegative.

6.2 SURROGATE KERNEL EVALUATION

Before evaluating the full BO pipeline, we first examine the behavior of the proposed lightweight Mahalanobis kernel in terms of fitting efficiency and predictive accuracy.

Fitting efficiency. Dense Mahalanobis kernel implementations parameterize $\mathbf{M} = \mathbf{L}\mathbf{L}^\top$ and scale poorly with dimension. Appendix Figure 3 compares GP fitting time on Hartmann6 embedded at increasing ambient dimension D, with 300 training points. Dense Mahalanobis fitting time rises

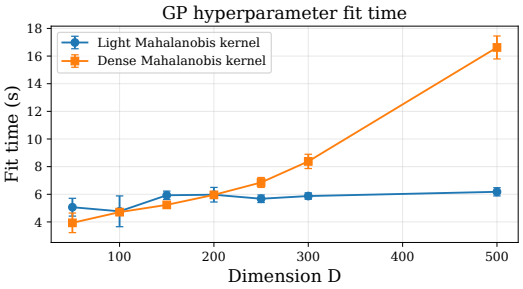

Figure 3: GP fitting time on Hartmann6 with 300 training points as ambient dimension D increases. Dense Mahalanobis kernel fitting grows rapidly, while the light one remains stable.

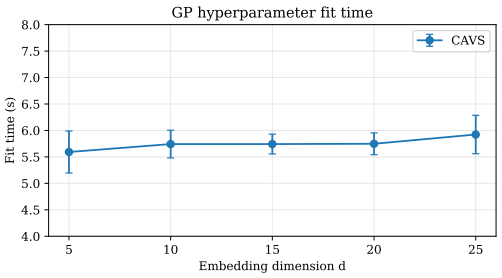

Figure 4: CAVS GP fitting time as intrinsic effective dimension varies from 5 to 25, with ambient dimension $D = 300$, 300 training points, and kernel rank fixed at 10.

Table 1: Final regret and average rank on eight real-world benchmarks. CAVS is evaluated against diverse state-of-the-art highdimensional BO methods covering variable selection, embedding, trust-region and foundation-model frameworks, alongside Vanilla BO, standard GP-BO and random search. All results are mean ± standard deviation from five independent runs.

Method	MOPTA08	Rover	HalfCheetah	Mazda	Lasso-DNA	Ant	Swimmer	MSLR	Avg. Rank
RS	78.9 ± 4.3	8.9 ± 0.9	1202 ± 329.9	23.2 ± 2.4	0.040 ± 0.004	19.8 ± 3.2	142 ± 41.7	0.051 ± 0.022	9.6
BO	57.4 ± 7.4	4.7 ± 0.6	1316 ± 133.8	17.3 ± 0.8	0.045 ± 0.005	15.0 ± 2.9	99 ± 55.4	0.026 ± 0.003	7.2
TuRBO	32.5 ± 4.4	2.5 ± 0.8	925 ± 318.6	11.5 ± 0.76	**0.005 ± 0.004**	21.0 ± 4.1	9 ± 8.3	0.008 ± 0.003	4.5
ALEBO	61.8 ± 4.6	5.4 ± 1.0	654 ± 378.9	21.9 ± 3.40	0.018 ± 0.006	14.7 ± 4.2	35 ± 37.9	0.028 ± 0.008	6.6
Vanilla BO	**7.2 ± 3.0**	1.8 ± 1.0	619 ± 426.5	**2.5 ± 2.7**	0.009 ± 0.005	15.1 ± 2.4	104 ± 46.8	0.050 ± 0.035	4.0
GIT-BO	81.8 ± 4.0	11.3 ± 2.2	905 ± 352.6	25.8 ± 2.7	0.011 ± 0.003	22.7 ± 3.1	140 ± 28.4	0.259 ± 0.340	9.4
Dropout	43.1 ± 4.1	4.5 ± 1.0	1097 ± 269.2	12.8 ± 0.6	0.008 ± 0.004	16.6 ± 2.1	105 ± 39.2	0.010 ± 0.004	5.8
SAASBO	59.9 ± 3.8	5.2 ± 0.3	1514 ± 85.3	17.5 ± 1.2	0.065 ± 0.006	22.9 ± 5.8	48 ± 8.9	0.018 ± 0.002	8.6
MCTS-VS	32.1 ± 5.8	2.4 ± 0.6	799 ± 413.5	9.8 ± 1.0	0.006 ± 0.002	19.3 ± 4.0	34 ± 44.8	0.011 ± 0.006	4.1
LassoBO	19.4 ± 4.3	1.2 ± 1.3	908 ± 474.9	8.6 ± 1.2	0.013 ± 0.002	19.5 ± 5.7	**7 ± 5.3**	**0.008 ± 0.005**	3.6
CAVS	11.0 ± 4.2	**0.7 ± 0.8**	**500 ± 404.4**	9.6 ± 0.6	0.010 ± 0.002	**12.6 ± 11.4**	18 ± 21.1	0.011 ± 0.002	**2.5**

rapidly, whereas the lightweight kernel remains comparatively stable. Appendix Figure 4 instead varies the intrinsic effective dimension d_{eff} (labeled d in the plot) from 5 to 25 at fixed $D = 300$, with 300 training points and kernel rank 10. Across this range, CAVS fitting time stays around 5.6 to 5.9 seconds and does not increase substantially as the effective dimension grows. This supports using Eq. (2) as the surrogate in CAVS rather than a dense metric.

Predictive accuracy. Runtime efficiency alone is insufficient to justify a surrogate model; it must also preserve strong modeling capacity for the black-box objective. We therefore compare model predictions against ground-truth values both visually, to assess predictive fidelity, and quantitatively, using negative log-likelihood (NLL) and root mean squared error (RMSE). All tests are on Hartmann6 with ambient dimensions $50, 100, 200, 300$, and 500, using 100 training points and 50 test points for consistent evaluation. Appendix Figure 7 shows that, with $d = 10$ fixed, predictions continue to track ground-truth values as D increases from 50 to 500. RMSE remains below that of ARD RBF and competitive with dense Mahalanobis throughout this range. This suggests that the structured metric can still identify important dimensions as its parameter count grows.

Beyond individual metrics, we further analyze the Pareto trade-off between GP fitting efficiency and predictive performance across 22 diagnostic tasks. As illustrated in Figure 6, our proposed lightweight Mahalanobis kernel lies on the observed Pareto frontier and achieves the lowest average normalized RMSE. The dense Mahalanobis kernel has a higher average RMSE in this comparison. Its $O(D^2)$ parameterization requires fitting many more hyperparameters from limited observations, which can impair the reliability of the fitted GP and its test predictions. ARD RBF fits fastest but suffers from degraded predictive accuracy, while the pure low-rank kernel offers marginal computational savings at the cost of substantially worse prediction quality. Overall, the lightweight kernel balances fitting cost and prediction quality.

6.3 Main Results

Having established the surrogate's fitting efficiency and predictive accuracy, we turn to end-to-end optimization. Table 1 reports final regret $r_T = f_{\text{ref}} - \max_{1 \le t \le T} f(\mathbf{x}_t)$ and average rank on eight real-world tasks. CAVS achieves the best average rank of 2.5 across these tasks. Although specialized methods may lead on individual benchmarks, CAVS offers balanced performance across diverse problems. Figure 5 shows that CAVS maintains a strong rank throughout the evaluation budget, indicating that the gains are not tied to a single final budget. It also runs faster than LassoBO, ALEBO, and SAASBO (Appendix A.6).

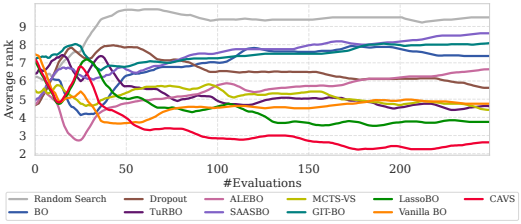

Figure 5: Average rank trajectory across all benchmarks over 250 evaluations and five random seeds.

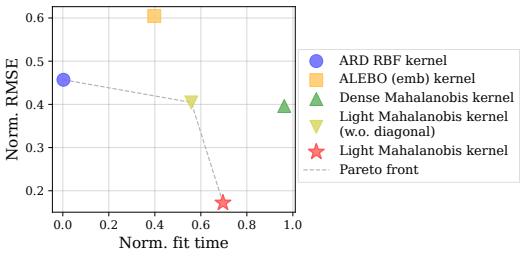

Figure 6: Average normalized Pareto trade-off between GP fitting time and test RMSE across 22 synthetic functions.

6.4 Ablation Study

We assess kernel components, variable identification under rotation, and hyperparameter sensitivity.

Kernel components. We compare the full low-rank-plus-diagonal kernel with ARD RBF, which omits the low-rank correlation term, and a pure low-rank Mahalanobis kernel, which omits independent coordinate scales. Figure 6 reports their GP fitting time and average normalized RMSE across 22 diagnostic tasks. The full kernel achieves the lowest average normalized RMSE among these variants, supporting the combination of both components.

Variable identification under rotation. Because our method relies on the sparsity assumption of active variables, we rotate only the intrinsic effective dimensions, preserving the relevant coordinate set while changing the relationships within it (Appendix A.7). We compare LassoBO (Hoang et al., 2025), SAASBO (Eriksson & Jankowiak, 2021), and CAVS before and after rotation using the average rank of true active coordinates and group recall among the top-k selected variables, where k is the number of true active dimensions. Appendix Figure 8 shows that CAVS exhibits a smaller increase in the rank of active variables and a smaller drop in group recall than LassoBO. SAASBO achieves similar variable identification performance to CAVS, but requires more computation time and performs worse on the real-world tasks (Appendix A.6; Table 1).

Hyperparameter sensitivity. We vary the low-rank dimension d and selection threshold α across eight benchmarks. Optimization trajectories vary little, and $d = 10$ and $\alpha = 0.9$ perform well overall. Full settings and curves appear in Appendices A.4 and A.5.

7 Conclusion

We presented CAVS, which replaces axis-aligned variable scoring in high-dimensional BO with importance from a lightweight Mahalanobis kernel. Its low-rank-plus-diagonal structure captures correlations at linear parameter complexity, and provides a principled selection criterion. We proved a sublinear regret bound and empirically showed that CAVS achieves the best average rank across eight real-world benchmarks and remains robust under coordinate rotation. Overall, CAVS advocates selecting variables that jointly shape the objective, offering a robust alternative to axis-aligned scoring in coupled settings that widely exist in real-world applications.

AI USE STATEMENT

In this work, we used generative AI tools for translation. We have not used generative AI tools for generating synthetic datasets, developing theoretical models, formulating mathematical claims, writing proofs, designing methods or experiments. We have reviewed all AI-assisted work. We reviewed all AI-assisted translations against the original text to ensure that they preserved its meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The appendices document benchmark specifications, implementation settings, evaluation budgets, and numbers of independent runs, together with the protocols for the rotation and sensitivity analyses. The code is provided in the supplementary material.

REFERENCES

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. In *Advances in Neural Information Processing Systems 33*, pp. 21524–21538, Virtual, 2020.

D. Eriksson and M. Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, pp. 493–503, Virtual, 2021.

D. Eriksson, M. Pearce, J. R. Gardner, R. D. Turner, and M. Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems 32*, pp. 5497–5508, Vancouver, Canada, 2019.

P. I. Frazier. A tutorial on Bayesian optimization. *CoRR abs/1807.02811*, 2018.

Peter I. Frazier and Jialei Wang. Bayesian optimization for materials design. In *Information Science for Materials Discovery and Design*, pp. 45–75. Springer International Publishing, 2016.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

Vu Viet Hoang, Hung The Tran, Sunil Gupta, and Vu Nguyen. High dimensional Bayesian optimization using lasso variable selection. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, pp. 3106–3114, Mai Khao, Thailand, 2025.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nat.*, 637(8044):319–326, 2025.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20793–20817, Vienna, Austria, 2024.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

K. Kandasamy, J. G. Schneider, and B. Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 295–304, Lille, France, 2015.

Takehisa Kohira, Hiromasa Kemmotsu, Akira Oyama, and Tomoaki Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the 20th ACM Conference on Genetic and Evolutionary Computation*, pp. 183–184, Kyoto, Japan, 2018.

H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

B. Letham, R. Calandra, A. Rai, and E. Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, pp. 1546–1558, Vancouver, Canada, 2020.

C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton. High dimensional Bayesian optimization using dropout. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2096–2102, Melbourne, Australia, 2017.

J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

M. Mutný and A. Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems 31*, pp. 9005–9016, Montreal, Canada, 2018.

A. Nayebi, A. Munteanu, and M. Poloczek. A framework for Bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4752–4761, Long Beach, CA, 2019.

Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. In *Advances in Neural Information Processing Systems 35*, pp. 11586–11601, New Orleans, LA, 2022.

Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Understanding high-dimensional Bayesian optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, Vancouver, Canada, 2025.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval Journal*, 13(4):346–374, 2010.

Kenan Sehic, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. Lassobench: A high-dimensional hyperparameter optimization benchmark suite for lasso. In *Proceedings of the 1st International Conference on Automated Machine Learning*, pp. 2/1–24, Baltimore, MD, 2022.

Baek Seong-Eun, Lee Jung-Mok, Kim Sung-Bin, and Tae-Hyun Oh. A language-guided Bayesian optimization for efficient lora hyperparameter search. *arXiv:2602.11171*, 2026.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Benjamin J. Shields, Jason M. Stevens, Jun Li, Marvin Parasram, Farhan N. Damani, Jesus I. Martinez Alvarado, Jacob M. Janey, Ryan P. Adams, and Abigail G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. Monte carlo tree search based variable selection for high dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 35*, pp. 28488–28501, New Orleans, LA, 2022.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Proceedings of the 25th IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, Algarve, Portugal, 2012.

Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. In *Advances in Neural Information Processing Systems 33*, pp. 19511–19522, Virtual, 2020.

Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv:1907.02057*, 2019.

Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Freitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55(1):361–387, 2016.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, pp. 745–754, Canary Islands, Spain, 2017.

Zhitong Xu, Haitao Wang, Jeff M. Phillips, and Shandian Zhe. Standard gaussian process is all you need for high-dimensional Bayesian optimization. In *Proceedings of the 13th International Conference on Learning Representations*, Singapore, 2025.

Rosen Ting-Ying Yu, Cyril Picard, and Faez Ahmed. Git-bo: High-dimensional Bayesian optimization with tabular foundation models. *arXiv:2505.20685*, 2026.

Juliusz Krysztof Ziomek and Haitham Bou-Ammar. Are random decompositions all we need in high dimensional Bayesian optimisation? In *Proceedings of the 40th International Conference on Machine Learning*, pp. 43347–43368, Honolulu, Hawaii, 2023.

A APPENDIX

A.1 PROOF FOR LEMMA 4.1

We first state a standard Gaussian tail bound.

Lemma A.1. *If $r \sim \mathcal{N}(0,1)$, then for any $c > 0$,*

$$\mathbb{P}(|r| > c) \leq e^{-c^2/2}.$$

Proof. For any $c > 0$,

$$\begin{aligned}\mathbb{P}(|r| > c) &= 2\mathbb{P}(r > c) \\ &= 2e^{-\frac{c^2}{2}}(2\pi)^{-1/2}\int_c^\infty e^{-\frac{(r-c)^2}{2}-c(r-c)}dr \\ &\leq 2e^{-\frac{c^2}{2}}\mathbb{P}(r > 0) = e^{-\frac{c^2}{2}}.\end{aligned}$$

□

Now we prove the Lemma 4.1.

Proof. For notational convenience, denote k_{Mah} by k. Consider any unit direction vector $\mathbf{v} \in \mathbb{R}^D$. Let $\mathbf{x} = (x_1, \ldots, x_D)$ and $\mathbf{x}' = \mathbf{x} + \epsilon\mathbf{v}$ with $\epsilon > 0$. By the definition of a Gaussian process, the finite-dimensional marginals $(f(\mathbf{x}), f(\mathbf{x}'))$ follow a bivariate normal distribution:

$$\begin{pmatrix} f(\mathbf{x}) \\ f(\mathbf{x}') \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{bmatrix} k(\mathbf{x},\mathbf{x}) & k(\mathbf{x},\mathbf{x}') \\ k(\mathbf{x},\mathbf{x}') & k(\mathbf{x}',\mathbf{x}') \end{bmatrix}\right).$$

Thus the difference $f(\mathbf{x}') - f(\mathbf{x})$ is also normally distributed with variance

$$\begin{aligned}V_M &= k(\mathbf{x},\mathbf{x}) + k(\mathbf{x}',\mathbf{x}') - 2k(\mathbf{x},\mathbf{x}') \\ &= 2\sigma_f^2\left(1 - \exp\left(-\frac{\mathbf{v}^\top\mathbf{M}\mathbf{v}}{2}\epsilon^2\right)\right).\end{aligned}$$

Therefore,

$$\frac{f(\mathbf{x}') - f(\mathbf{x})}{\epsilon} \sim \mathcal{N}\left(0,\ \frac{2\sigma_f^2}{\epsilon^2}\left(1 - e^{-\frac{\mathbf{v}^\top\mathbf{M}\mathbf{v}}{2}\epsilon^2}\right)\right).$$

Taking the limit $\epsilon \to 0$, we obtain

$$\frac{\partial f}{\partial \mathbf{v}}(\mathbf{x}) \sim \mathcal{N}\big(0,\ \sigma_f^2\mathbf{v}^\top\mathbf{M}\mathbf{v}\big).$$

In particular, for the coordinate direction $\mathbf{e}_i$, we have

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) \sim \mathcal{N}\big(0,\ \sigma_f^2 M_{ii}\big).$$

Applying Lemma A.1 with $c = \frac{L}{\sigma_f}$ gives

$$\mathbb{P}\left(\left|\frac{\partial f}{\partial x_i}\right| > L\sqrt{M_{ii}}\right) \leq e^{-\frac{L^2}{2\sigma_f^2}}.$$

Equivalently,

$$\mathbb{P}\left(\left|\frac{\partial f}{\partial x_i}\right| \leq IS_i \cdot L\right) \geq 1 - e^{-\frac{L^2}{2\sigma_f^2}},$$

where $IS_i = \sqrt{M_{ii}}$. This completes the proof. □

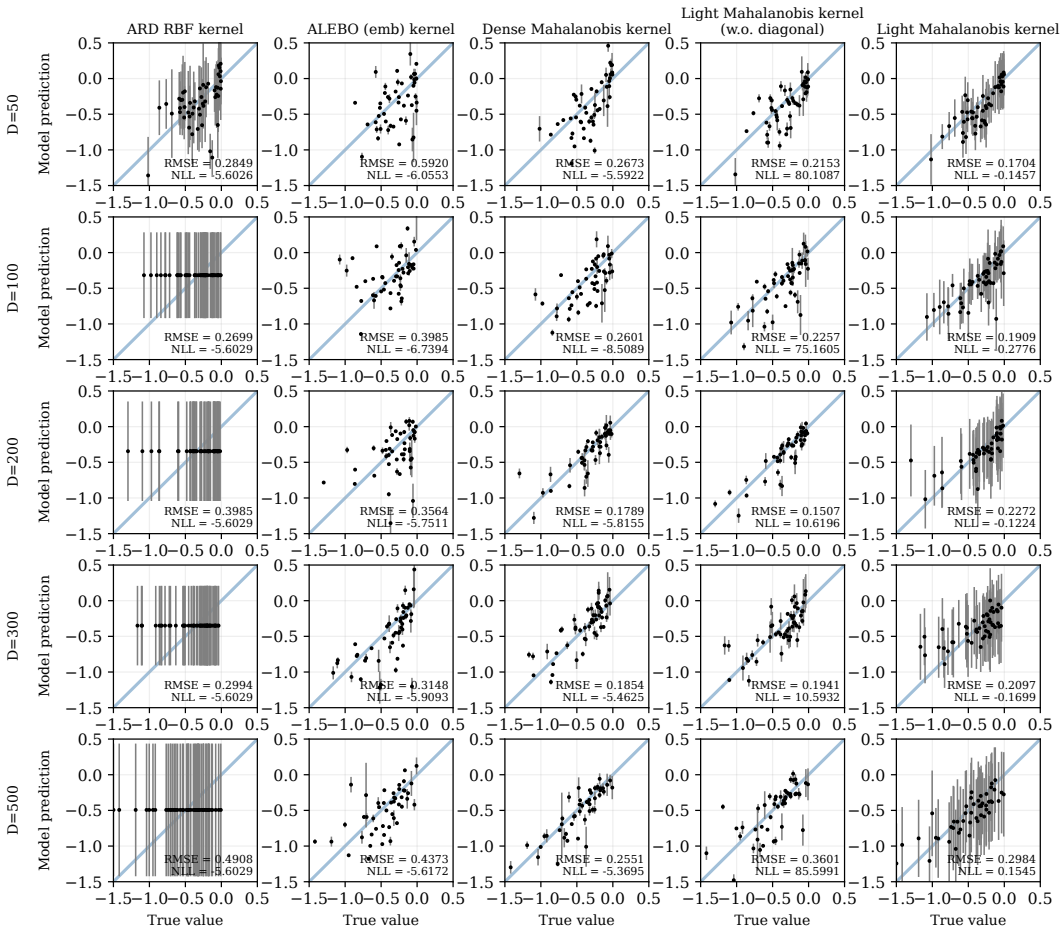

Figure 7: Fitting (NLL) and prediction (RMSE) quality on Hartmann6 embedded in ambient dimensions from 50 to 500.

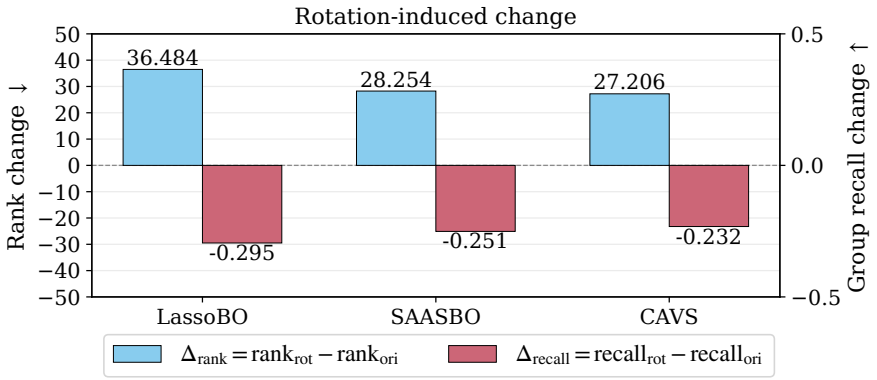

Figure 8: Rotation-induced change in variable identification. Positive rank change (lower is better) and negative recall change (higher is better) indicate degradation under rotation.

Algorithm 1 Correlation-Aware Variable Selection (CAVS)

Input: Initial dataset $\mathcal{D}_0$; budget T; threshold α; low-rank dimension d; clipping ratio γ
Output: Best solution $\mathbf{x}^{T,*}$ and dataset $\mathcal{D}_T$
$\mathbf{x}^{0,*} \leftarrow \mathbf{x}_{i^*}$, where $i^* \in \arg\max_{i:\,(\mathbf{x}_i, y_i) \in \mathcal{D}_0} y_i$
for $t = 1, 2, \ldots, T$ **do**
  Fit GP with $\mathbf{M} = \mathrm{diag}(\ell^{-1}) + \mathbf{R}\mathbf{R}^\top$ in Eq. (2)
  Compute $IS_i = \sqrt{M_{ii}}$ and select active set I_t by cumulative importance $\geq \alpha$, clipping its size to $[1, \lfloor \gamma D \rfloor]$
  Sample N_t uniform background points to form $\mathcal{Z}_t$; let $\mathcal{B}_t = \mathcal{Z}_t \cup \{\mathbf{x}^{t-1,*}_{[D]\setminus I_t}\}$.
  $\mathbf{x}^t \leftarrow \arg\max_{\mathbf{x}_{I_t}} \max_{\mathbf{x}_{\mathrm{bg}} \in \mathcal{B}_t} \alpha_t([\mathbf{x}_{I_t}, \mathbf{x}_{\mathrm{bg}}])$
  Evaluate $y^t = f(\mathbf{x}^t)$, update $\mathcal{D}_t$ and $\mathbf{x}^{t,*}$
end for
return $\mathbf{x}^{T,*}$, $\mathcal{D}_T$

A.2 Proof of Theorem 5.1 and Corollary 5.1

Our regret analysis builds upon the theoretical framework originally developed for LassoBO (Hoang et al., 2025), but we adapt all core concentration and bounding arguments to our correlation-aware variable importance metric $IS_i = \sqrt{M_{ii}}$, where $\mathbf{M} = \mathrm{diag}(\ell^{-1}) + \mathbf{R}\mathbf{R}^\top$ denotes our lightweight low-rank-plus-diagonal Mahalanobis metric matrix. The whole proof is grounded on Assumption 5.1. We recap the critical proof pipeline and restate key intermediate lemmas and bounds under our modified importance scoring.

Per-Iteration Regret Decomposition At iteration t, denote I_t as the selected active variable set, $[D] \setminus I_t$ the inactive coordinate subspace, and $\mathcal{B}_t$ the pool of background imputation vectors for inactive dimensions. We define the constrained candidate subspace

$$\mathcal{S}_t^0 = \big\{[\mathbf{v}, \mathbf{z}] \,\big|\, \mathbf{v} \in [0,1]^{|I_t|},\ \mathbf{z} \in \mathcal{B}_t\big\},$$

and let $f^{\max}_{\mathcal{S}_t^0} = \max_{\mathbf{x} \in \mathcal{S}_t^0} f(\mathbf{x})$ be the maximum objective value attainable within this subspace.

Let $r_t = f^* - f(\mathbf{x}_t)$ denote the instantaneous regret at iteration t. Following the decomposition adopted in LassoBO, we split per-step regret as

$$\begin{aligned} r_t &= f^* - f(\mathbf{x}_t) \\ &= \underbrace{f^* - f^{\max}_{\mathcal{S}_t^0}}_{\text{Term 1}} + \underbrace{f^{\max}_{\mathcal{S}_t^0}}_{\text{Term 2}} - \underbrace{f(\mathbf{x}_t)}_{\text{Term 3}}. \end{aligned}$$

All bounding logic for Term 2 and Term 3 remains identical to LassoBO, with only hyperparameter β_t adjusted to fit our metric definition:

$$\begin{aligned}\beta_t =& 2\log\left(\frac{\pi^2 t^2}{\delta}\right) \\ &+ 2\tilde{n}_t \log\left(4\sqrt{m^*}b\tilde{n}_t\sqrt{\log\left(\frac{12Da}{\delta}\right)}t^2\right),\end{aligned}$$

where $\tilde{n}_t = \max_{1\le k\le t}|I_k|$, $m^* = \max_{i\in[D]} M_{ii}$.

Bound for Term 1 Term 1 captures the approximation error caused by restricting optimization to the finite background sample pool $\mathcal{B}_t$ for inactive dimensions. Under Assumption 5.1, we obtain the bound

$$f^* - f^{\max}_{\mathcal{S}_t^0} \le C_t\sqrt{\sum_{i\in[D]\setminus I_t} M_{ii}}\left(\frac{1}{N_t}\log\left(\frac{6}{\delta}\right)\right)^{\frac{1}{D-|I_t|}},$$

with $C_t = 2b\sqrt{\log\left(\frac{2Da}{\delta}\right)}\left(\Gamma(D-|I_t|+1)\right)^{\frac{1}{D-|I_t|}}$. In LassoBO, the summation inside the square root uses unselected inverse lengthscales $\sum_{i\in[D]\setminus I_t}\rho_i$, while we substitute each ρ_i with our correlation-aware diagonal entry $M_{ii} = \ell_i^{-1} + \|R_{i,:}\|_2^2$, which jointly encodes individual coordinate smoothness and cross-variable correlation contributions. The high-dimensional volume and uniform sampling concentration arguments for this term are unchanged from (Hoang et al., 2025).

Bounds for Term 2 and Term 3 The upper bounds for Term 2 and Term 3 directly inherit the GP-UCB concentration results from LassoBO, with no structural modifications to their algebraic forms. The only distinction is that all Lipschitz constants and discretization radii in their derivations rely on $m^* = \max_i M_{ii}$ instead of the axis-aligned ρ^*. Combining the bounds of Term 1, Term 2 and Term 3 yields a pointwise upper bound for instantaneous regret r_t.

Cumulative Regret Bound (Theorem 5.1) Summing r_t over all iterations $t = 1,\dots,T$ and applying the Cauchy–Schwarz inequality to cross terms produces the final cumulative regret bound:

$$\begin{aligned}R_T \le& \sqrt{\beta_T C_1 T\gamma_T} + \frac{\pi^2}{6} \\ &+ C\sqrt{\left(\sum_{t=1}^{T}\sum_{i\in[D]\setminus I_t} M_{ii}\right)\left(\sum_{t=1}^{T}\left(\frac{1}{N_t}\right)^{\frac{2}{D-|I_t|}}\right)},\end{aligned}$$

where $C = \max_t C_t$, $C_1 = 8/\log(1+\sigma^{-2})$, and γ_T denotes the maximum information gain of f over T observations. The first two standard GP-UCB terms only update β_T with our m^*, whereas the third cross term is fully reparameterized around our Mahalanobis metric entries, marking the core theoretical difference from LassoBO's axis-aligned regret bound.

Corollary 5.1 We follow LassoBO's background sample schedule $N_t = \lceil t^{1/n}\rceil$ with integer $n > 1$. Substituting this schedule into the cross loss term and applying standard p-series summation bounds gives

$$\begin{aligned}R_T \le& \sqrt{\beta_T C_1 T\gamma_T} + \frac{\pi^2}{6} \\ &+ C\sqrt{c_0}\sqrt{\sum_{t=1}^{T}\sum_{i\in[D]\setminus I_t} M_{ii}}\cdot T^{\frac{1}{2}-\frac{1}{n(D-1)}},\end{aligned}$$

for constant $c_0 > n(D-1)/(n(D-1)-2)$. The term $T^{\frac{1}{2}-\frac{1}{n(D-1)}}$ grows sublinearly in T, so $\lim_{T\to\infty} R_T/T = 0$, verifying sublinear cumulative regret for CAVS. The series convergence arguments stay identical to (Hoang et al., 2025).

Derivational Basis and Key Modifications All low-level supporting lemmas are similar to those presented in LassoBO (Hoang et al., 2025), with every occurrence of $\sqrt{\rho_i}$ replaced by our $\sqrt{M_{ii}}$. We do not replicate lengthy intermediate algebraic calculations, while all modified key bounds and substitution positions are explicitly restated above to clearly separate our correlation-aware regret characterization from LassoBO's axis-aligned analysis.

A.3 Details of Benchmark Problems

MOPTA08. This vehicle design task requires optimizing 124 design variables to reduce vehicle weight. The original formulation aims to minimize vehicle mass subject to 68 constraints. For Bayesian optimization experiments, these constraints are relaxed, transforming the task into a single-objective minimization problem. Specifically, its design variables characterize materials, gauges, and the geometric shape of the vehicle.

Rover. This problem is a trajectory optimization task designed to emulate autonomous rover navigation (Wang et al., 2017). The goal is to minimize the trajectory cost while ensuring that the trajectory starts and ends at specified locations. Specifically, the optimization variable $x \in [0, 1]^{60}$ represents a set of control points used to fit a B-spline trajectory. The objective function is non-smooth, discontinuous, and concave with respect to the dimensions controlling the start and goal positions.

HalfCheetah. This benchmark originates from reinforcement learning locomotion tasks within the MuJoCo (Todorov et al., 2012) simulation suite. Its objective maximizes the cumulative reward obtained under a linear policy parameterized by a high-dimensional matrix. Specifically, HalfCheetah is a 102-dimensional RL problem derived from the MuJoCo Half-Cheetah-v4 environment.

Car Mazda. This constrained multi-objective benchmark originates from automotive structural design optimization. The original task aims to minimize total vehicle weight while maximizing the number of shared standard plate thickness components across different vehicle models. Specifically, we adopt the 148-dimensional configuration for simultaneous design optimization of two vehicle types, which includes 148 discrete design parameters and 36 constraints. Following the practice in GIT-BO (Yu et al., 2026), we apply penalty transforms to incorporate constraints and use average weighting to convert the multi-objective problem into an unconstrained single-objective minimization task suitable for Bayesian optimization.

Lasso-DNA. This is a hyperparameter optimization problem from LassoBench (Sehic et al., 2022). The task targets adjusting hyperparameters to strike a balance between least-squares fitting and the sparsity-inducing penalty term. Specifically, LassoDNA forms a 180-dimensional hyperparameter optimization benchmark built upon a microbiology DNA dataset.

Ant. This benchmark corresponds to the quadruped robot locomotion task implemented in MuJoCo. We aim to maximize the cumulative reward via optimizing a linear policy. Specifically, Ant forms an 888-dimensional optimization problem, where the decision variables consist of all entries of the linear policy matrix that transforms state observations into actuator commands.

Swimmer. This benchmark is also a locomotion reinforcement learning task from the MuJoCo simulation suite. The goal is to maximize the cumulative reward by searching for a linear locomotion policy. Specifically, Swimmer is a 16-dimensional optimization problem. The dimension originates from all weights of the linear policy mapping environment observations to control actions.

MSLR. This problem originates from the Microsoft Learning to Rank dataset (Qin et al., 2010). The goal is to tune model parameters to achieve favorable ranking performance. The dataset covers over 10,000 distinct queries, with every web page encoded using a set of 136 features.

A.4 Sensitivity Analysis of Parameter d

We conduct sensitivity experiments to investigate the influence of the hyperparameter d in the Mahalanobis metric matrix $\mathbf{R}$, which controls the dimensionality of the local embedding space. The

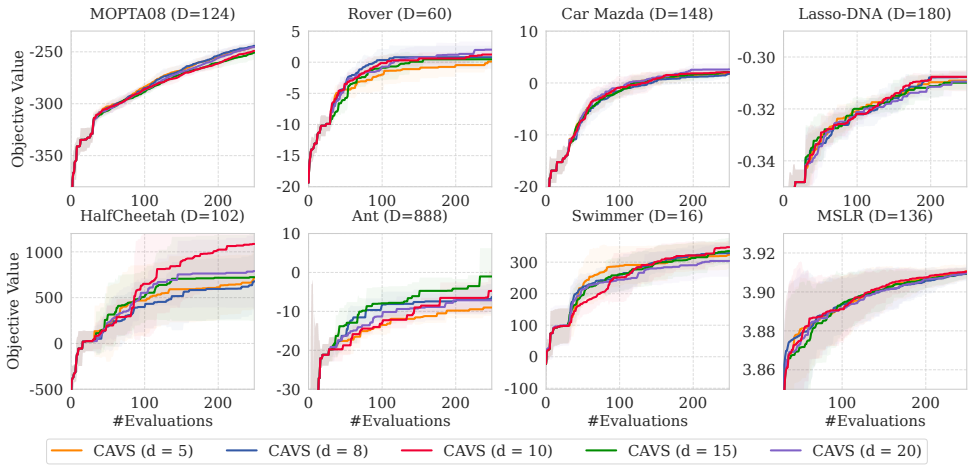

Figure 9: Sensitivity analysis of hyperparameter d. Optimization curves of CAVS with $d \in \{5, 8, 10, 15, 20\}$ evaluated on 8 benchmark functions, each repeated over 5 random seeds. The performance discrepancy across different choices of d is mild, and $d = 10$ yields favorable overall results.

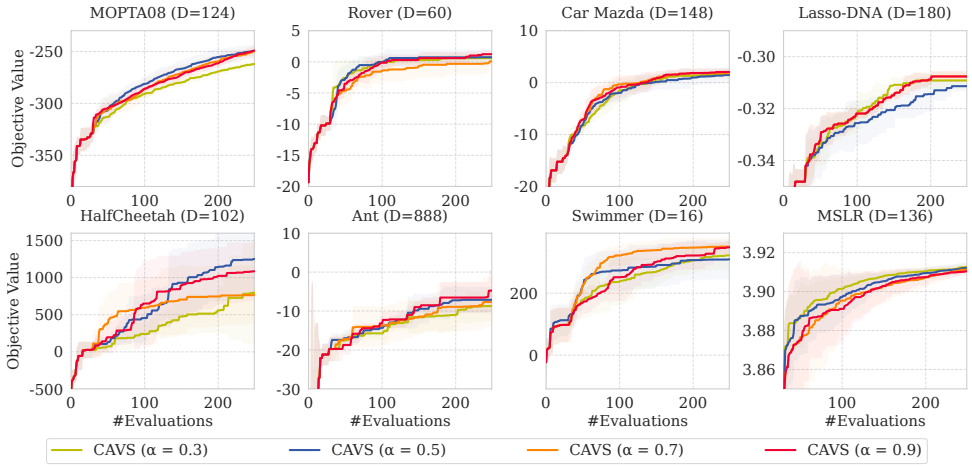

Figure 10: Sensitivity analysis of variable selection threshold α. Optimization curves of CAVS with $\alpha \in \{0.3, 0.5, 0.7, 0.9\}$ on 8 benchmark functions, with 5 random seeds for each configuration. Performance across different α is comparable, and $\alpha = 0.9$ is chosen as the default setting.

experiments are performed on 8 benchmark functions, where we set $d \in \{5, 8, 10, 15, 20\}$. For each configuration, we run 5 random seeds to ensure stable and reproducible results.

We visualize the optimization curves of CAVS under different d values across all benchmark tasks in Figure 9. The experimental results show that varying d does not lead to significant performance differences in the optimization trajectories. Among all tested configurations, $d = 10$ achieves consistently favorable and slightly better average performance across tasks. Accordingly, we set $d = 10$ as the default hyperparameter for all main experiments throughout our framework.

A.5 Sensitivity Analysis of Threshold α

We perform sensitivity experiments to study the effect of threshold α used for variable selection in CAVS. Experiments are carried out on 8 benchmark functions with $\alpha \in \{0.3, 0.5, 0.7, 0.9\}$. We run 5 random seeds for each setting to obtain reliable outcomes.

Table 2: Wall-clock runtime for 250 function evaluations on the Lasso-DNA benchmark. All results report mean ± standard deviation across 5 random seeds.

Method	Total Runtime (min)
Random Search	1.89 ± 0.04
BO	2.04 ± 0.10
TuRBO	3.04 ± 0.17
ALEBO	115.87 ± 11.39
Vanilla BO	4.97 ± 0.48
GIT-BO	11.06 ± 0.21
Dropout	2.15 ± 0.13
SAASBO	366.49 ± 14.02
MCTS-VS	4.95 ± 0.20
LassoBO	40.47 ± 2.41
CAVS	27.54 ± 1.02

We visualize the optimization curves of CAVS under different α values across all benchmark tasks in Figure 10. The results indicate that the optimization trajectories exhibit only minor differences across the tested values of α. Overall, $\alpha = 0.9$ achieves better empirical performance on average. Therefore, we adopt $\alpha = 0.9$ as the default threshold in all main experiments.

A.6 Runtime Comparison on DNA Benchmark

To evaluate the total practical wall-clock overhead under a fixed budget, we measure the end-to-end runtime of all methods over 250 function evaluations and 5 random seeds. Experiments are conducted on a server with Intel Xeon Gold 6530 processors and 256 GB RAM, running on Ubuntu 22.04.1. All methods are run on CPU, except for GIT-BO, which relies on the TabPFN foundation model and is therefore run on a single NVIDIA RTX 4090 GPU.

Table 2 summarizes the timing results. Lightweight baselines finish within 3.1 minutes, and mid-cost approaches take 5–11 minutes. Our CAVS runs in 27.54 minutes, primarily due to adaptive subspace identification and Mahalanobis-kernel GP fitting. More importantly, CAVS is $1.5\times$, $4.2\times$, and $13.3\times$ faster than LassoBO, ALEBO, and SAASBO, respectively, demonstrating a strong balance between expressiveness and computational cost.

A.7 Rotation Robustness Experiment Setup

We conduct rotation robustness experiments on a diverse set of classical, synthetic, and local interaction benchmark functions. In our experiments, we further augment the input dimension of each of the 11 original test functions to 300 and 500 dimensions by appending redundant useless dimensions that do not participate in the objective function computation, resulting in a total of 22 test functions, each with 5 random seeds, averaged over all tasks. These test functions cover fully separable classical optimization tasks and non-separable synthetic tasks with complex variable interactions, which are suitable for evaluating the rotation robustness of variable selection methods. The detailed specifications of all test functions used in our rotation stress test are summarized in Table 3, and the complete mathematical formulations are presented below.

A.7.1 Classical Functions

Hartmann6 The 6-dimensional Hartmann function is defined on $[0,1]^6$ as:

$$f(x) = \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right),$$

where $\alpha = [1.0, 1.2, 3.0, 3.2]$, and A, P are predefined constant matrices. All 6 dimensions are active variables.

Table 3: Summary of test functions for rotation robustness experiments.

Function	Dimension	Input domain	Active variables
Classical Functions			
Hartmann6	6	$[0,1]^6$	$\{0,1,2,3,4,5\}$
Levy15	15	$[-3,3]^{15}$	$\{0,1,\ldots,14\}$
Rastrigin10	10	$[-2.56,2]^{10}$	$\{0,1,\ldots,9\}$
SumSq16	16	$[-5,5]^{16}$	$\{0,1,\ldots,15\}$
Synthetic Functions			
Twinh6	56	$[0,1]^{56}$	$\{0,\ldots,5\}\cup\{50,\ldots,55\}$
Linred	10	$[-1,1]^{10}$	$\{0,1\}\cup\{2,\ldots,9\}$
Coupledquad	10	$[-1,1]^{10}$	$\{0,1\}\cup\{2,\ldots,9\}$
Phasemod	10	$[-1,1]^{10}$	$\{0,1\}\cup\{2,\ldots,9\}$
Local Interaction Functions			
Tripleprod	10	$[-1,1]^{10}$	$\{0,1,2\}\cup\{3,\ldots,9\}$
Quadcross	10	$[-1,1]^{10}$	$\{0,1,2\}\cup\{3,\ldots,9\}$
Sinprod4	10	$[-1,1]^{10}$	$\{0,1\}\cup\{2,3\}\cup\{4,\ldots,9\}$

Levy15 The 15-dimensional Levy function is defined on $[-3,3]^{15}$ as:

$$\begin{aligned} w_i &= 1+\frac{x_i-1}{4},\\ f(x) &= -\Big[\sin^2(\pi w_1)\\ &\quad +\sum_{i=2}^{14}(w_i-1)^2\left(1+10\sin^2(\pi w_i+1)\right)\\ &\quad +(w_{15}-1)^2\left(1+\sin^2(2\pi w_{15})\right)\Big]. \end{aligned}$$

All 15 dimensions are active variables.

Rastrigin10 The 10-dimensional Rastrigin function is defined on $[-2.56,2]^{10}$ as:

$$f(x) = -\left[10\cdot 10+\sum_{i=1}^{10}\left(x_i^2-10\cos(2\pi x_i)\right)\right].$$

All 10 dimensions are active variables.

SumSq16 The 16-dimensional weighted sum of squares function is defined on $[-5,5]^{16}$ as:

$$f(x) = -\sum_{i=1}^{16}\frac{x_i^2}{i^2}.$$

All 16 dimensions are active variables.

A.7.2 Synthetic Functions

Twinh6 This function consists of two disjoint 6-dimensional Hartmann blocks in a 56-dimensional space:

$$f(x) = -\left[\text{Hartmann6}(x_{0:6})+0.65\cdot\text{Hartmann6}(x_{50:56})\right].$$

The active variables are two independent groups $\{0,\ldots,5\}$ and $\{50,\ldots,55\}$.

Linred This function combines linearly redundant variable pairs and an 8-dimensional Ackley background:

$$\begin{aligned} f_{\text{ridge}}(x_0,x_1) &= \sin(\pi(x_0+x_1)),\\ f_{\text{ack}}(x) &= 0.35\cdot\text{Ackley}(2x_{2:10}-1),\\ f(x) &= -\big(f_{\text{ridge}}+f_{\text{ack}}\big). \end{aligned}$$

Active variables include the redundant pair $\{0,1\}$ and the 8-dimensional Ackley group $\{2,\ldots,9\}$.

Coupledquad This function adopts coupled quadratic variable interaction with an 8-dimensional Ackley background:

$$\begin{aligned} f_{\text{pair}}(x_0, x_1) &= 0.45\left[(2x_0 - x_1)^2 + 0.08(x_0^2 + x_1^2)\right], \\ f_{\text{ack}}(x) &= 0.35 \cdot \text{Ackley}(2x_{2:10} - 1), \\ f(x) &= -\big(f_{\text{pair}} + f_{\text{ack}}\big). \end{aligned}$$

Active variables include the quadratic interaction group $\{0, 1\}$ and the 8-dimensional Ackley group $\{2, \ldots, 9\}$.

Phasemod This function introduces phase-modulated trigonometric coupling with an 8-dimensional Ackley background:

$$\begin{aligned} f_{\text{pair}}(x_0, x_1) &= \sin\left(2\pi\left(x_0 + 0.35\sin(2\pi \cdot 2 \cdot x_1)\right)\right), \\ f_{\text{ack}}(x) &= 0.35 \cdot \text{Ackley}(2x_{2:10} - 1), \\ f(x) &= -\big(f_{\text{pair}} + f_{\text{ack}}\big). \end{aligned}$$

Variable x_1 modulates the phase of x_0, and active variables consist of the coupling group $\{0, 1\}$ and the background group $\{2, \ldots, 9\}$.

A.7.3 Local Interaction Functions

Tripleprod This function uses triple-product sine interaction with a 7-dimensional Ackley background:

$$\begin{aligned} f_{\text{core}}(x_0, x_1, x_2) &= \sin(2\pi x_0 x_1 x_2), \\ f_{\text{ack}}(x) &= 0.35 \cdot \text{Ackley}(2x_{3:10} - 1), \\ f(x) &= -\big(f_{\text{core}} + f_{\text{ack}}\big). \end{aligned}$$

Active variables include the triple-interaction group $\{0, 1, 2\}$ and the 7-dimensional Ackley group $\{3, \ldots, 9\}$.

Quadcross This function contains non-separable quadratic cross interaction with a 7-dimensional Ackley background:

$$\begin{aligned} f_{\text{core}}(x_0, x_1, x_2) &= -(x_0^2 + 2x_0 x_1 + x_2^2), \\ f_{\text{ack}}(x) &= 0.35 \cdot \text{Ackley}(2x_{3:10} - 1), \\ f(x) &= -\big(f_{\text{core}} + f_{\text{ack}}\big). \end{aligned}$$

Active variables include the quadratic cross-interaction group $\{0, 1, 2\}$ and the background group $\{3, \ldots, 9\}$.

Sinprod4 This function integrates two independent sine product interaction pairs with a 6-dimensional Ackley background:

$$\begin{aligned} f_{\text{core}}(x_0, x_1, x_2, x_3) &= \sin(\pi x_0 x_1) \cdot \cos(\pi x_2 x_3), \\ f_{\text{ack}}(x) &= 0.35 \cdot \text{Ackley}(2x_{4:10} - 1), \\ f(x) &= -\big(f_{\text{core}} + f_{\text{ack}}\big). \end{aligned}$$

Active variables include two product interaction groups $\{0, 1\}, \{2, 3\}$ and the 6-dimensional Ackley background group $\{4, \ldots, 9\}$.

A.8 Orthogonal Rotation Setup for Robustness Test

We build rotated benchmark functions to test variable selection robustness under non-axis-aligned active subspaces. All orthogonal transforms only operate on each task's intrinsic effective dimensions, leaving redundant inactive coordinates unchanged. This keeps the target set of important coordinates fixed while changing the relationships among them, allowing a direct comparison of variable identification before and after rotation.

For intrinsic dimension d, orthonormal rotation matrices are generated via QR decomposition of seeded standard Gaussian matrices. We adjust diagonal signs of the triangular factor after factorization to remove sign ambiguity, guaranteeing identical rotation matrices across repeated experiments.

All coordinate transforms are conducted on the symmetric domain $[-1, 1]^d$ for stable numerical computation. Given raw input $\mathbf{x}$ within original bounds $[\mathbf{lb}, \mathbf{ub}]$, we first rescale inputs to the symmetric hypercube, multiply by the orthogonal matrix, map rotated coordinates back to the original domain, and clip values within the input range to resolve boundary overflow before evaluating the base objective.

Orthogonal rotations preserve distances and the core optimization structure of the original function, yet break axis-aligned sparsity of active variables. We compare variable ranking and group recall metrics between unrotated and rotated variants to measure how each method degrades under oblique coordinate representations.