NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World

Learning relevant contextual variables within Bayesian optimization

Julien Martinelli JULIEN.MARTINELLI@AALTO.FI
Department of Computer Science, Aalto University

Ayush Bharti AYUSH.BHARTI@AALTO.FI
Department of Computer Science, Aalto University

Armi Tiihonen ARMI.TIIHONEN@GMAIL.COM
Department of Applied Physics, Aalto University

Louis Filstroff LOUIS.FILSTROFF@CENTRALELILLE.FR
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL

S.T. John ST.JOHN@AALTO.FI
Department of Computer Science, Aalto University

Sabina J. Sloman SABINA.SLOMAN@MANCHESTER.AC.UK
Department of Computer Science, University of Manchester

Patrick Rinke PATRICK.RINKE@AALTO.FI
Department of Applied Physics, Aalto University

Samuel Kaski SAMUEL.KASKI@AALTO.FI
Department of Computer Science, Aalto University
Department of Computer Science, University of Manchester

Abstract

Contextual Bayesian Optimization (CBO) efficiently optimizes black-box, expensive-to-evaluate functions with respect to design variables, while simultaneously integrating *relevant contextual* information regarding the environment, such as experimental conditions. However, the relevance of contextual variables is not necessarily known beforehand. Moreover, contextual variables can sometimes be optimized themselves, an overlooked setting by current CBO algorithms. Optimizing contextual variables may be costly, which raises the question of determining a minimal relevant subset. We address this problem using a novel method, Sensitivity-Analysis-Driven Contextual BO (`SADCBO`). We learn the relevance of context variables by sensitivity analysis of the posterior surrogate model, whilst minimizing the cost of optimization by leveraging recent developments on early stopping for BO. We empirically evaluate our proposed `SADCBO` against alternatives on both synthetic and real-world experiments, and demonstrate a consistent improvement across examples.

Keywords: Contextual Bayesian Optimization, Variable Selection, Gaussian Processes

1. Introduction

Bayesian optimization (BO) is a sample-efficient black-box optimization method, typically used when the expense of computing the objective function makes the problem in-

tractable (Jones et al., 1998; Brochu et al., 2010), e.g./ material and drug discovery(Zhang et al., 2020; Gómez-Bombarelli et al., 2018; Korovina et al., 2020).

A key implicit assumption in BO is that the objective function only depends on the design variables. This assumption is violated in many practical scenarios, wherein various environmental factors and experimental settings, referred to as *contextual variables* (Krause and Ong, 2011; Kirschner et al., 2020; Arsenyan et al., 2023), also affect the objective function. For instance, ambient humidity was found to influence the experiments in robot-assisted material design (Nega et al., 2021), leading to a changing optimal design under different humidity conditions. Moreover, in practice, the domain experts themselves might not know *a priori* which contextual variables are relevant. Identifying the relevant contextual variables is therefore critical not only to guarantee reliable optimization results but also for the practitioners to reliably reproduce experimental results.

Variants of BO have therefore been developed to deal with the uncertainty related to the contextual variables. In particular, Krause and Ong (2011) introduced the Contextual Bayesian optimization (CBO) framework, enabling the inclusion of uncontrollable contextual information in the surrogate model. However, in some applications, contextual variables *can* be controlled. For instance, synthesis conditions of material samples, the used solvents, or certain environment conditions, such as experiment room temperature or ambient humidity (Higgins et al., 2021; Nega et al., 2021), are principally controllable during the course of an experiment, but it may not be straightforward to predict whether they are relevant to include (Abolhasani and Brown, 2023). While gains in BO performance can potentially be obtained by optimizing over all the potential contextual variables, or by determining the relevant ones and optimizing over them, intervening on such variables is usually costly, thus invoking a cost versus efficiency trade-off.

We extend the CBO framework to settings in which the relevant contextual variables are not known beforehand and can be intervened on at some cost. We introduce Sensitivity-Analysis-Driven CBO (`SADCBO`), a method which leverages recent advances in sensitivity-analysis-driven variable selection (Sebenius et al., 2022) and early stopping criteria for BO (Ishibashi et al., 2023). `SADCBO` combines the *contextual observational* setting, where the context information is only observed, and the *contextual interventional* setting, where contextual variables are intervened on (similar to design variables), into a sequential algorithm. We provide a thorough evaluation of the performance of `SADCBO`, comparing it against methods from the CBO and high-dimensional BO literature, on both synthetic and real-world examples, demonstrating that `SADCBO` favorably compares to existing methods.

2. Contextual Bayesian Optimization (CBO)

The CBO framework (Krause and Ong, 2011) deals with a black-box function $f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ defined on the space of both the *design variables* $\mathcal{X} \subset \mathbb{R}^d$ and *contextual variables* $\mathcal{Z} \subset \mathbb{R}^c$. We observe noisy evaluations of the function, $y = f(\mathbf{x}, \mathbf{z}) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$.

A Gaussian process (GP) prior (Rasmussen and Williams, 2006) is placed on f; with the notation $\mathbf{v} = [\mathbf{x}, \mathbf{z}]$, we write $f(\mathbf{v}) \sim \mathcal{GP}(0, k(\mathbf{v}, \mathbf{v}'))$. This means that, for any finite-dimensional collection of inputs $[\mathbf{v}_1, \ldots, \mathbf{v}_t]$, the function values $\mathbf{f} = [f(\mathbf{v}_1), \ldots, f(\mathbf{v}_t)]^\top \in \mathbb{R}^t$ follow a multivariate normal distribution $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K} \in \mathbb{R}^{t \times t} = (k(\mathbf{v}_i, \mathbf{v}_j))_{1 \leq i,j \leq t}$ is the kernel matrix computed from the kernel k. Given a dataset $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{z}_i, y_i)\}_{i=1}^t =$

$\{(\mathbf{v}_i, y_i)\}_{i=1}^t$, the posterior distribution of $f(\mathbf{v})$ given $\mathcal{D}_t$ is Gaussian for all $\mathbf{v}$ with closed-form expressions for the mean $\mu_t(\mathbf{v}|\mathcal{D}_t)$ and variance $\sigma_t^2(\mathbf{v}|\mathcal{D}_t)$.

In the CBO setting, we sequentially observe the context variables and choose the design variables in response to this observation. More precisely, at iteration $t+1$, a context vector $\mathbf{z}_{t+1}$ is observed, assumed to have been drawn from a distribution $p(\mathbf{z})$, and the optimal design $\mathbf{x}_{t+1}^\star$ is such that $\mathbf{x}_{t+1}^\star = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}, \mathbf{z}_{t+1})$. Given $\mathbf{z}_{t+1}$ and the previous t observations $\mathcal{D}_t$, the next candidate design point $\mathbf{x}_{t+1}$ is selected using the Upper Confidence Bound acquisition function (Srinivas et al., 2012):

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathcal{X}}{\arg\max}\ \mu_t(\mathbf{x}, \mathbf{z}_{t+1}|\mathcal{D}_t) + \beta_t^{1/2}\sigma_t(\mathbf{x}, \mathbf{z}_{t+1}|\mathcal{D}_t). \tag{1}$$

We extend the problem setting of CBO in two ways. Firstly, we assume that only a subset of the contextual variables truly affect f. Let $\mathbf{z} = [z^{(1)}, \ldots, z^{(c)}]$ be the vector of contextual variables. For a set J belonging to the power set of $\{1, \ldots, c\}$, denote by $\mathbf{z}^{(J)} \in \mathbb{R}^{|J|}$ the vector of reduced dimension whose variables are indexed by J. We assume there exists a set $J^\star$, $c \gg |J^\star|$, such that $f(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z}^{(J^\star)})\ \forall(\mathbf{x}, \mathbf{z})$. Secondly, we enable setting the value of any of the contextual variables at a cost, in addition to the design query cost. This means that for all $j \in \{1, \ldots, c\}$, the context variable $z^{(j)}$ can be intervened on for a cost λ_j. We then aim to maximize f in a cost-efficient manner, by identifying $J^\star$.

3. Methodology

This section introduces our method to solve the aforementioned extended CBO problem. It relies on a variable selection technique from the GP literature (Sebenius et al., 2022), which we adapt to the optimization framework.

3.1 Variable selection for CBO via sensitivity analysis

One approach for handling the presence of contextual variables that can be intervened on is to include them in the design space. However, such a strategy can become infeasible when their relevance is not known *a priori*. In such cases, identifying the relevance of the contextual variables is key, not only for efficient optimization of the function, but also as additional information to the experts about the experiment.

To that end, we adapt the Feature Collapsing (FC) method (Sebenius et al., 2022) to identify the relevant contextual variables. The FC method applies a perturbation to a training point (namely, setting one feature to zero), and measures the induced shift in the posterior predictive distribution in terms of KL divergence. Given a dataset $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{z}_i, y_i)\}_{i=1}^t$, the relevance $r(i,j)$ of the i^{th} sample of the j^{th} contextual variable $z_i^{(j)}$ is computed as $r(i,j) = \mathrm{KL}\left(p(y_\star|\mathbf{x}_i, \mathbf{z}_i, \mathcal{D}_t)\ ||\ p(y_\star|\mathbf{x}_i, \mathbf{z}_i \odot \boldsymbol{\xi}[j], \mathcal{D}_t)\right)$, where $\boldsymbol{\xi}[j] = [\xi^{(1)}, \ldots, \xi^{(c)}]$ is a vector s.t. $\xi^{(j)} = 0$, and $\xi^{(j')} = 1$, for $j' \neq j$, and $\odot$ is the element-wise multiplication. The relevance score of the j^{th} contextual variable is then computed as

$$\mathrm{FC}(j) = \frac{1}{|\mathcal{D}_t|}\sum_{i=1}^{|\mathcal{D}_t|}\left(\frac{r(i,j)}{\sum_{j=1}^c r(i,j)}\right). \tag{2}$$

The FC scores computed in this manner reveal the variables that are relevant for output prediction across $\mathcal{D}_t$. However, as our goal is to maximize f, we are interested in identifying

contextual variables that are relevant for high function values. Hence, we modify the dataset over which the FC scores are averaged in Equation (2) Denote that dataset by $\mathcal{D}_t^{\text{FC}} = \mathcal{D}_t^{\gamma_t} \cup \mathcal{D}_t^{Q}$. Here, $\mathcal{D}_t^{\gamma_t}$ is a subset of $\mathcal{D}_t$, comprised of only high output values, defined as

$$\mathcal{D}_t^{\gamma_t} = \{(\mathbf{x}_i, \mathbf{z}_i, y_i) \in \mathcal{D}_t \mid y_i / y_{\text{best}} \geq \gamma_t\}, \tag{3}$$

where $y_{\text{best}} = \max_{1 \leq i \leq t} \ y_i$ is the current observed maximum. For instance, using $\gamma_t = 0.8 \ \forall t$ would yield a $\mathcal{D}_t^{\gamma_t}$ that consists of the highest 20% observations obtained so far. As for $\mathcal{D}_t^{Q} := \{(\mathbf{x}_q^\star, \mathbf{z}_{t+1})\}_{q=1}^{Q}$, it contains promising points given by a batch acquisition function.

Once the FC scores are computed and sorted in descending order, we select the indices of those contextual variables whose cumulative FC score is greater than $\eta \in [0, 1]$, meaning that the selected variables explain the fraction η of the output sensitivity amongst all contextual variables. Let J_η denote the set of indices of the selected contextual variables. We train a GP based on $\{(\mathbf{x}_i, \mathbf{z}_i^{(J_\eta)}, \mathbf{y}_i\}_{i=1}^t$ and select the designs through maximization of the UCB acquisition function: $\mathbf{x}_{t+1} = \underset{\mathbf{x} \in \mathcal{X}}{\arg\max} \ \mu_t(\mathbf{x}, \mathbf{z}_{t+1}^{(J_\eta)} | \mathcal{D}_t) + \beta_t^{1/2} \sigma_t(\mathbf{x}, \mathbf{z}_{t+1}^{(J_\eta)} | \mathcal{D}_t)$.

3.2 Sensitivity-Analysis Driven CBO (SADCBO)

We now present SADCBO, a sequential method for performing BO in the presence of irrelevant contextual variables (Algorithm S1). SADCBO utilizes the variable selection method of Section 3.1 and proceeds in two phases. In the first phase, we choose to only *observe* the values of the contextual variables without optimizing over them, thus preventing costly contextual variable queries, when their relevance is computed based on a limited amount of data. Therefore, we only leverage the available contextual information for design selection. This information, however, will eventually saturate. This is when the second phase starts.

In the second phase, we begin to *intervene* on the contextual variables selected at each iteration based on their FC relevance. As there is a cost λ_j associated with intervening on the context variable $z^{(j)}$, we substitute FC(j) for FC(j)/λ_j in Equation (2). Our variable selection criterion can then be interpreted as the degree of sensitivity *per unit cost*. As previously, once the contextual variables $\mathbf{z}^{(J_\eta)}$ have been selected, we train a GP surrogate based on $\{(\mathbf{x}_i, \mathbf{z}_i^{(J_\eta)}, \mathbf{y}_i\}_{i=1}^t$ and select the next design-context pair to query as

$$(\mathbf{x}_{t+1}, \mathbf{z}_{t+1}^{(J_\eta)}) = \underset{(\mathbf{x}, \mathbf{z}^{(J_\eta)}) \in \mathcal{X} \times \prod_{j \in J_\eta} \mathcal{Z}_j}{\arg\max} \mu_t(\mathbf{x}, \mathbf{z}^{(J_\eta)} | \mathcal{D}_t) + \beta_t^{1/2} \sigma_t(\mathbf{x}, \mathbf{z}^{(J_\eta)} | \mathcal{D}_t). \tag{4}$$

The switch from the observational to the interventional phase in SADCBO relies on a stopping criterion proposed by Ishibashi et al. (2023). We detect the point at which the gain in the optimization from purely observing the contextual variables diminishes, following which the interventional phase begins. Further details are provided in Appendix A.

4. Related work

Bogunovic et al. (2018) and Kirschner et al. (2020) perform worst-case optimization under fluctuations of the contextual variables. In particular, Distributionally-Robust BO (DRBO) tries to maximize the expected black-box function value under the worst-case distribution of the contextual variables. However, as in Krause and Ong (2011), these works assume

Table 1: Methods used in experiments.

	Name	Description	Reference
Baselines with no variable selection	CUBO	Context-Unaware BO over designs $\mathbf{x}$ only	-
	VBO	Vanilla BO over $[\mathbf{x}, \mathbf{z}]$	-
	CBO	Contextual BO using all contexts $\mathbf{z}$	(Krause and Ong, 2011)
Baselines performing variable selection	Dropout	Randomly drop half of the context variables	(Li et al., 2018)
	MMDBO	Maximum mean discrepancy-driven BO	(Spagnol et al., 2019)
	MMDCBO	Maximum mean discrepancy-driven CBO	-
This work	SADBO	**Sensitivity analysis-driven BO**	-
	SADCBO	**Sensitivity analysis-driven CBO**	-
Oracle	OBO	Oracle vanilla BO optimising only $[\mathbf{x}, \mathbf{z}^{(J)}]$	-

that the relevant contextual variables are known *a priori*, and can only be observed and not controlled. On another note, due to the curse of dimensionality, the performance of standard BO is severely degraded when applied in high-dimensional input spaces. To tackle this problem, most proposed approaches either aim at carrying out BO in a lower-dimensional space instead of the original (Li et al., 2018; Ziomek and Bou-Ammar, 2023) or work with a structured GP surrogate, equipped with an additive kernel or a sparsifying prior (Eriksson and Jankowiak, 2021; Liu et al., 2023). Data-driven methods based on various measures of feature relevance have also been proposed (Spagnol et al., 2019; Shen and Kingsford, 2021).

5. Experimental results

We now evaluate our approach on several real-world examples and synthetic functions described in Table S1 and appendix C.2 and compare it with a number of baselines (Table 1). A number of additional experiments have also been carried out and can be found in Appendices D and E.

In real-world experiments (Figure 1**a**), SADCBO (in red with white markers) achieves promising results, although certain advanced baselines perform on par. SADCBO almost consistently overperforms the first baselines VBO and CUBO. There is next to no difference between the performances of SADCBO and CBO (in blue). This observation combined with the fact that optimizing only design variables (CUBO, in yellow) produces poor results for the Portfolio and Yacht problems suggests that contextual variables play a significant part in maximizing these objectives, but their interventional cost is high. Among baselines with variable selection, it is worth noticing that the Dropout baseline, which randomly drops half of the contextual variables, consistently underperforms, while SADCBO performs similarly to MMDCBO, except for the Alanine experiment where SADCBO performs slightly better.

Next, Figure 1**b** displays the best value found by each baseline for synthetic experiments. On the Hartmann4D problem, SADCBO follows the oracle OBO (green). For Hartmann6D it turns out that SADBO outperforms SADCBO . This is due to the fact that SADBO optimizes contextual variables from the start. For a sufficiently low cost, this proves a better strategy than beginning with only paying a cost for optimizing the designs. Our approach slightly outperforms the MMD-based measure, but consistently across both test functions. CUBO and CBO perform poorly as they do not optimize the context. VBO generally does a poor job, as it considers every variable, thus spending a large fraction of the budget every iteration.

For Hartmann6D and Hartmann4D, Figure 1**c** reports the time at which the stopping criterion kicks in for SADCBO, demonstrating that both phases are leveraged in our approach.

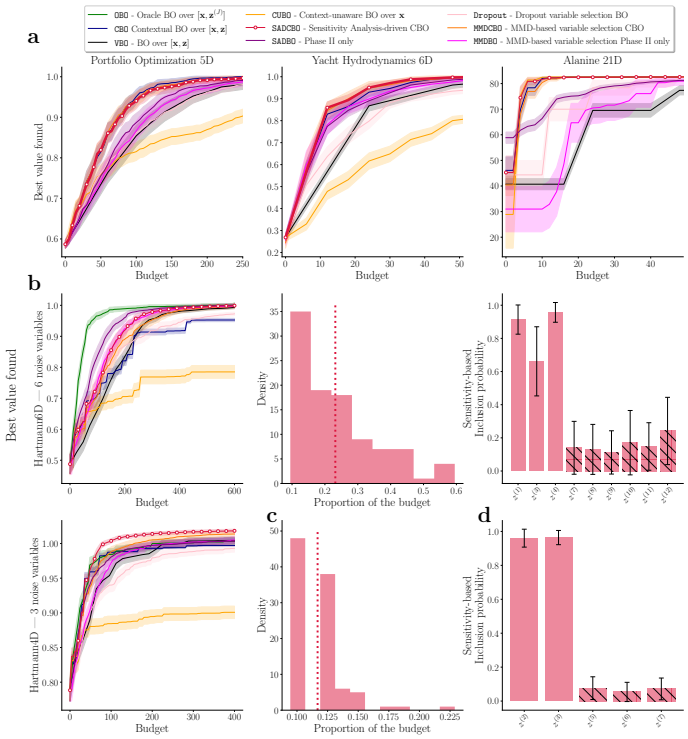

Figure 1: Benchmark of the different methods. (**a**) Real-world examples. (**b**) Synthetic functions. **(c)** Histograms of early stopping criterion hitting time for `SADCBO`, in proportion of the budget. Vertical lines refer to the mean of each distribution. **(d)** Inclusion probability of each contextual variable for `SADCBO`. Irrelevant contextual variables are hatched. Mean ± 2 standard error computed across $N = 100$ trials.

Lastly, Figure 1**d** reports the sensitivity indices computed at each iteration for each contextual variable, averaged across whole trajectories of multiple trials for Hartmann6D and Hartmann4D.

6. Conclusion

We introduced `SADCBO`, an algorithm designed to sort out relevant context variables affecting the experimental outcomes by efficiently leveraging information present when observing or optimizing contextual variables. `SADCBO` reduces the surrogate model to only relevant variables and ensures the reproducibility of experiments by controlling for such relevant variables. In that respect, `SADCBO` should be used for practical applications where contextual variables can have an influence while being controllable. This would include, for example, high-throughput materials and molecule exploration loops that are being increasingly utilized in both academic and industrial laboratories for drug design and new material development (Zhang et al., 2020; Gómez-Bombarelli et al., 2018).

Supplementary Materials

Outline of the Appendix. In Appendix A, further details about SADCBO are provided. In Appendix B, we detail the experimental settings with respect to benchmarked baselines and implementation details. In particular, Appendix B.2 introduces one of the baselines used in the main text, based on maximum mean discrepancy. Appendix C contains a description of the real-world experiments performed throughout the paper, along with the analytical expressions of the synthetic examples used. Appendix D contains further experimental results regarding:

- The distribution of early stopping time for SADCBO (Figure S1)
- Varying the number of irrelevant contextual variables (Section D.1)
- Varying contextual variables query cost (Section D.2)
- Varying the surrogate model kernel structure (Section D.3)
- Varying SADCBO hyperparameters (Section D.4)

Finally, Appendix E presents a forward selection approach to perform variable selection once the sensitivity indices have been computed and evaluates this approach on synthetic examples.

Appendix A. Stopping criterion and algorithmic description of SADCBO

We here briefly describe the details of the stopping criterion initially proposed by (Ishibashi et al., 2023). This criterion was adapted to suit our method and effectively determines when we switch from the observational phase to the interventional phase.

Let $\mathbf{v}_t^\star = \arg\max_{\mathbf{v}\in\mathcal{D}_t} f(\mathbf{v})$ be the current best candidate point in the dataset up to time t, where $\mathbf{v} = [\mathbf{x}, \mathbf{z}]$. and denote $f^\star := \max_{\mathbf{v}\in\mathcal{V}} f(\mathbf{v})$. Let $R_t = f^\star - \mathbb{E}_{\hat{f}\sim p(f|\mathcal{D}_t)}[\max_{\mathbf{v}\in\mathcal{V}}\ \hat{f}(\mathbf{v})]$ be the expected minimum simple regret. Then, $\Delta R_t = |R_t - R_{t-1}|$ can be upper bounded:

$$\begin{aligned}\Delta R_t &\leq v(\phi(g) + g\Phi(g)) + |\Delta\mu_t^\star| \\ &\quad + \kappa_{\delta,t-1}\sqrt{\frac{1}{2}\mathrm{KL}(p(f|\mathcal{D}_t)||p(f|\mathcal{D}_{t-1}))} \\ &:= \Delta\tilde{R}_t,\end{aligned}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the p.d.f. and c.d.f. of a standard Gaussian distribution, respectively, $\Delta\mu_t^\star := \mu_{t-1}(\mathbf{v}_{t-1}^\star) - \mu_t(\mathbf{v}_t^\star), v := \sqrt{\sigma_t^2(\mathbf{v}_t^\star) - 2\sigma_t^2(\mathbf{v}_t^\star, \mathbf{v}_{t-1}^\star)) + \sigma_t^2(\mathbf{v}_{t-1}^\star)}, g := -\Delta\mu_t^\star/v$, and $\kappa_{\delta,t-1}$ is a sequence indexed by t and depending on δ. Then, we switch from the observational to the interventional phase in SADCBO when $\Delta\tilde{R}_t \leq s_t$, where

$$s_t := \frac{(\sigma_{t-1}^2(\mathbf{v}_t^\star) + \kappa_{\delta,t-1}/2)\sigma_{t-1}^2(\mathbf{v}_t)\sqrt{-2\log\delta}}{\sqrt{\sigma_{\text{noise}}}(\sigma_{t-1}^2(\mathbf{v}_t) + \sigma_{\text{noise}}^{-1})}.$$

Algorithm S1 Sensitivity-Analysis-Driven Contextual BO (`SADCBO`)

Input: initial dataset $\mathcal{D}_0$, hyperparameters η and γ, batch size Q, budget Λ, costs $\lambda_{\mathbf{x}}, \lambda_1, \dots, \lambda_c$
Initialize GP using all variables $[\mathbf{x}, \mathbf{z}]$. `phase` = observational
while $\Lambda \geq \lambda_{\mathbf{x}} + \min_j \lambda_j$ **do**
  Receive context $\mathbf{z}_t \sim p(\mathbf{z})$
  Assemble dataset $\mathcal{D}_t^{\text{FC}}$ (Equation (3) and $\mathcal{D}_t^Q$)
  Compute sensitivity measure FC(j) based on $\mathcal{D}_t^{\text{FC}}$ Equation (2)
  In descending order, add indices to J_η until $\sum_{j \in J_\eta} \text{FC}(j) > \eta$
  Train reduced GP on $[\mathbf{x}, \mathbf{z}^{(J_\eta)}]$
  Get $\mathbf{x}_t$ (and $\mathbf{z}_t$ if `phase` = interventional) (Equation (4))
  $y_t \leftarrow f(\mathbf{x}_t, \mathbf{z}_t) + \varepsilon_t$
  $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, \mathbf{z}_t, y_t)\}$
  Retrain full GP
  if `phase` = observational and $\Delta \tilde{R}_t \leq s_t$ [based on $p(f|\mathcal{D}_t)$] (Appendix A) **then**
    `phase` = interventional `// Never check again once criterion satisfied`
  end if
  $\Lambda \leftarrow \Lambda - \lambda_{\mathbf{x}} + \sum_{j \in J_\eta} \lambda_j,\ t \leftarrow t + 1$
end while

Appendix B. Details on experimental settings

B.1 Baselines and implementation details

We benchmark our approach, coined `SADCBO`, against baselines referenced in Table 1. `SADCBO` is expected to compete with or even outperform Oracle Vanilla BO (`OBO`) upon identification of the relevant contextual variables, and outperform the other baselines. We also report `SADBO`, an analog to `SADCBO` but without the first observational phase, which amounts to performing BO with variable selection at each step. The baselines `MMDBO` and `MMDCBO` follow our approach but use an MMD-based measure of sensitivity (Spagnol et al., 2019) instead of the FC scores. Further details are provided in Appendix B.2.

We fix the hyperparameter of `SADCBO` and `SADBO` to $\eta = 0.8, Q = 10, \gamma_t = 0.8\ \forall t$. For the GP surrogate, an RBF kernel with independent lengthscales for each variable is employed. We use the UCB acquisition strategy, as well as Q-UCB for computing $\mathcal{D}_t^Q$ (Wilson et al., 2017). In all experiments, we assume that all variables, design or contextual ones, have cost $\lambda_j = 1\ \forall j \in \{1, \dots, d + c\}$.

B.2 Maximum Mean Discrepancy-based variable selection

Spagnol et al. (2019) introduced a BO algorithm with a variable selection procedure based on the Hilbert Schmidt Independence Criterion (HSIC). This measure can be used in our setting as well. We now briefly describe how it is defined.

As introduced in the main text, let $\mathcal{Z} \subset \mathbb{R}^c$ be the space of contextual variables, and $\mathcal{H}$ be a Hilbert space of $\mathbb{R}$-valued functions on $\mathcal{Z}$. Assume that $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is the unique positive definite kernel associated with the Reproducing Kernel Hilbert Space $\mathcal{H}$. Let $\mu_{\mathbb{P}_Z}$ be the

kernel mean embedding of the distribution $\mathbb{P}_Z$, $\mu_{\mathbb{P}_Z} := \mathbb{E}_Z[k(Z,\cdot)] = \int_{\mathcal{Z}} k(\mathbf{z},\cdot)\mathrm{d}\mathbb{P}_Z$. Kernel embeddings of probability measures provide a distance between distributions between their embeddings in the Hilbert Space $\mathcal{H}$, named Maximum Mean Discrepancy (MMD, (Gretton et al., 2012)):

$$\mathrm{MMD}(\mathbb{P}_Z, \mathbb{P}_Y) = \|\mu_{\mathbb{P}_Z} - \mu_{\mathbb{P}_Y}\|^2_{\mathcal{H}}\,. \tag{S1}$$

For two random variables $Z \sim \mathbb{P}_Z$ on $\mathcal{H}$ and $Y \sim \mathbb{P}_Y$ on $\mathcal{G}$, the HSIC is the squared MMD between the product distribution $\mathbb{P}_{ZY}$ and the product of its marginals $\mathbb{P}_Z\mathbb{P}_Y$,

$$\begin{aligned}
\mathrm{HSIC}(Z,Y) &= \mathrm{MMD}^2(\mathbb{P}_{ZY}, \mathbb{P}_Z\mathbb{P}_Y) && \text{(S2)}\\
&= \|\mu_{\mathbb{P}_{ZY}} - \mu_{\mathbb{P}_Z\mathbb{P}_Y}\|^2_{\mathcal{H}\otimes\mathcal{G}} && \text{(S3)}\\
&= \mathbb{E}_{Z,Y}\mathbb{E}_{Z',Y'}[k(Z,Z')l(Y,Y')] && \text{(S4)}\\
&\quad + \mathbb{E}_Z\mathbb{E}_Y\mathbb{E}_{Z'}\mathbb{E}_{Y'}[k(Z,Z')l(Y,Y')]\\
&\quad - 2\mathbb{E}_{Z,Y}\mathbb{E}_{Z'}\mathbb{E}_{Y'}[k(Z,Z')l(Y,Y')].
\end{aligned}$$

To determine the relevance of a variable $Z^{(i)}$, Spagnol et al. (2019) introduce

$$S^{\mathrm{HSIC}}(Z^{(i)}) = \mathrm{HSIC}(Z^{(i)}, \mathbb{I}(Z \in \mathcal{L}_\gamma)), \tag{S5}$$

with $\mathcal{L}_\gamma$ a region of interest: the locations where the objective function value is above a threshold γ. This measure reflects how important $Z^{(i)}$ is to reach $\mathcal{L}_\gamma$.

We implemented this measure, substituting expectations for empirical means over the dataset $\mathcal{D}$. We use $\gamma = 0.8$, a threshold identical to the one used for `SADCBO` in Equation (3). The kernel k is chosen to be a RBF kernel, and l is a linear kernel $l(y, y') = yy'$, a common choice for binary data. We create two baselines: `MMDCBO`, the analog of `SADCBO`, featuring both the observational and interventional phases, but using the MMD-based variable relevance measure, and likewise, `MMDBO`, the counterpart of `SADBO`, involving only the interventional phase.

Appendix C. Experiment details

C.1 Real-world datasets

Portfolio optimization dataset. This dataset was first introduced in (Cakmak et al., 2020). The goal is to tune the hyper-parameters of a trading strategy so as to maximize return under risk-aversion to random environmental conditions. A software is used to simulate and optimize the evolution of a portfolio over a period of four years using open-source market data. Each evaluation of this simulator returns the average daily return over this period of time under the given combination of hyper-parameters and environmental conditions. Since the simulator is expensive to evaluate, we do not use it directly but perform pool-based Bayesian Optimization using a pool of 3000 points generated according to a Sobol sampling design. The hyper-parameters to be optimized are the risk and trade aversion parameters and the holding cost multiplier These variables constitute the design variables. The contextual variables are the bid-ask spread and the borrowing cost.

Table S1: Dimensionality of the experiments carried out. For synthetic experiments, n.v. stands for artificial noise variables, added on top of the design and contextual variables.

Experiment	All dimensions	Design variables	Contextual variables
Portfolio	5	3	2
Yacht	6	4	2
Alanine	21	3	18
EggHolder	2 + 4 n.v.	1	1
Hartmann4D	4 + 3 n.v.	2	2
Michalewicz	8 + 6 n.v.	4	4
Hartmann6D	6 + 6 n.v.	3	3
Ackley	5 + 8 n.v.	2	3

Yacht hydrodynamics dataset. This dataset comes from the UCI Machine Learning Repository (Gerritsma et al., 2013). The optimization problem is to maximize the residuary resistance per unit weight of displacement of a yacht by controlling its 5-dimensional hull geometry coefficients. Another optimization variable is the 1-dimensional Froude number. We chose as design variables the first four dimensions of the hull geometry coefficients. The contextual variables are the last hull geometry dimension and the Froude number. Like the Portfolio optimization dataset, we have access to a limited number of samples (≈ 300) and thus perform pool-based Bayesian optimization.

Alanine conformer optimization. This case is a real-time computational physics optimization problem. Molecules can adopt different shapes, conformers, defined by the atomic bond lengths and angles within the molecule. Finding the lowest energy conformers for specific molecules is a relevant problem because the molecules typically take these shapes in nature. Here, alanine — a molecule with structure $C_3H_7NO_2$ — is optimized utilizing AMBER biomolecular molecule simulation toolkit (Salomon Ferrer et al., 2013; Case et al., 2023). The possible alanine structure variables to be optimized include the dihedral angles (ten angles), atom angles (eleven angles), and bond lengths (twelve lengths). The full problem is challenging to optimize with traditional gradient descent methods. Recently, the process has been significantly facilitated with Bayesian optimization of the major variables (Fang et al., 2021). For the purposes of this demonstration, two major dihedral angles are chosen as the design variables, the rest of the dihedral and atomic angles (18 angles) are chosen as the contextual variables, and bond lengths are omitted to facilitate faster simulations. The search space is selected by utilizing physics domain knowledge.

C.2 Synthetic test functions

Synthetic experiments. Five test functions are considered (Table S1 and appendix C.2). A min-max transformation is performed on the input data, scaling it to the unit cube: $\mathcal{X} \times \mathcal{Z} = [0,1]^{d+c}$. Similarly, the output is scaled between $[0,1]$ and a noise term $\varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$ is added with $\sigma^2_{\text{noise}} = 0.001$. The contextual variable distribution is $p(\mathbf{z}) = \mathcal{U}([0,1]^c)$.

Hartmann-6D function:

$$
\begin{aligned}
f(\mathbf{v}) &= -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(v^{(j)} - P_{ij})\right) \\
\alpha &= (1.0, 1.2, 3.0, 3.2)^T \\
\mathbf{A} &= \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix} \\
\mathbf{P} &= 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}
\end{aligned}
$$

defined over $\mathcal{V} = [0, 1]^6$. The second, fifth, and sixth variables were considered as design variables, while the first, third, and fourth variables were considered as contextual variables. 6 noise variables were added. Table S2 provides the results of a Sobol global sensitivity analysis performed using evaluations of the function collected over a grid of $N = 917504$ samples (Sobol, 2001). Adding up the first order indices for design and contextual variables separately leads to $S_{\mathbf{x}} \approx 0.124$ and $S_{\mathbf{z}} \approx 0.196$. This means that with respect to first-order interactions, contextual variables have more impact than design variables, in this synthetic example. One should notice however that these indices are computed across the whole search space and not specifically at the optimum.

Table S2: Sobol global sensitivity analysis for the Hartmann-6D function using $N = 917504$ samples.

Variable	First order sensitivity indices	Total order sensitivity indices
$z^{(1)}$	0.107	0.343
$x^{(2)}$	0.006	0.399
$z^{(3)}$	0.007	0.052
$z^{(4)}$	0.082	0.379
$x^{(5)}$	0.106	0.297
$x^{(6)}$	0.012	0.482

Hartmann-4D function:

$$f(\mathbf{v}) = \frac{1}{0.839}\left(1.1 - \sum_{i=1}^{4}\alpha_i \exp\left(-\sum_{j=1}^{4} A_{ij}(v^{(j)} - P_{ij})\right)\right)$$

$$\alpha = (1.0, 1.2, 3.0, 3.2)^T$$

$$\mathbf{A} = \begin{pmatrix} 10 & 3 & 17 & 3.5 \\ 0.05 & 10 & 17 & 0.1 \\ 3 & 3.5 & 1.7 & 10 \\ 17 & 8 & 0.05 & 10 \end{pmatrix}$$

$$\mathbf{P} = 10^{-4}\begin{pmatrix} 1312 & 1696 & 5569 & 124 \\ 2329 & 4135 & 8307 & 3736 \\ 2348 & 1451 & 3522 & 2883 \\ 4047 & 8828 & 8732 & 5743 \end{pmatrix}$$

defined over $\mathcal{V} = [0, 1]^4$. The first and fourth variables were considered as design variables, while the second and third variables were considered as contextual variables. 3 noise variables were added. Table S3 provides the results of a Sobol global sensitivity analysis performed using evaluations of the function collected over a grid of $N = 300000$ samples. Adding up the first order indices for design and contextual variables separately leads to $S_{\mathbf{x}} \approx 0.579$ and $S_{\mathbf{z}} \approx 0.091$. This means that with respect to first-order interactions, design variables have much more impact on the output than contextual variables. The gap slightly reduces when considering total order sensitivity indices. However, it is worth remembering that these indices are computed across the whole search space and not specifically at the optimum.

Table S3: Sobol global sensitivity analysis for the Hartmann-4D function using $N = 300000$ samples.

Variable	First order sensitivity indices	Total order sensitivity indices
$x^{(1)}$	0.307	0.477
$z^{(2)}$	0.037	0.279
$z^{(3)}$	0.054	0.103
$x^{(4)}$	0.272	0.526

Ackley 5D function:

$$f(\mathbf{v}) = -20\exp\left(-0.2\sqrt{\frac{1}{5}\sum_{j=1}^{5}(v^{(j)})^2}\right) - \exp\left(\frac{1}{5}\sum_{j=1}^{5}\cos(2\pi v^{(j)})\right) + 20 + e^1$$

defined over $\mathcal{V} = [-5, 5]^5$. 8 noise variables were added.

Michalewicz 8D function:

$$f(\mathbf{v}) = -\sum_{j=1}^{8} \sin(v^{(j)}) \sin^{20}\left(\frac{jv^{(j)}}{\pi}\right)$$

defined over $\mathcal{V} = [0, \pi]^8$. The first four variables were considered as design variables, while the four last were considered as contextual variables. 6 noise variables were added.

EggHolder 2D function:

$$f(\mathbf{v}) = -(v^{(2)} + 47) \sin\left(\sqrt{\left|v^{(2)} + \frac{v^{(1)}}{2} + 47\right|}\right) - v^{(1)} \sin\left(\sqrt{|v^{(1)} - (v^{(2)} + 47)|}\right)$$

defined over $\mathcal{V} = [-512, 512]^2$. The first variable was considered as a design variable, and the second one as a contextual variable. 4 noise variables were added. A Sobol global sensitivity analysis performed using evaluations of the function collected over a grid of $N = 3000000$ samples shows that both variables have a similar contribution to the output (Table S4).

Table S4: Sobol global sensitivity analysis for the EggHolder-2D function using $N = 3000000$ samples.

Variable	First order sensitivity indices	Total order sensitivity indices
$x^{(1)}$	0.001	0.998
$z^{(2)}$	0.0004	0.999

Appendix D. Additional experimental results

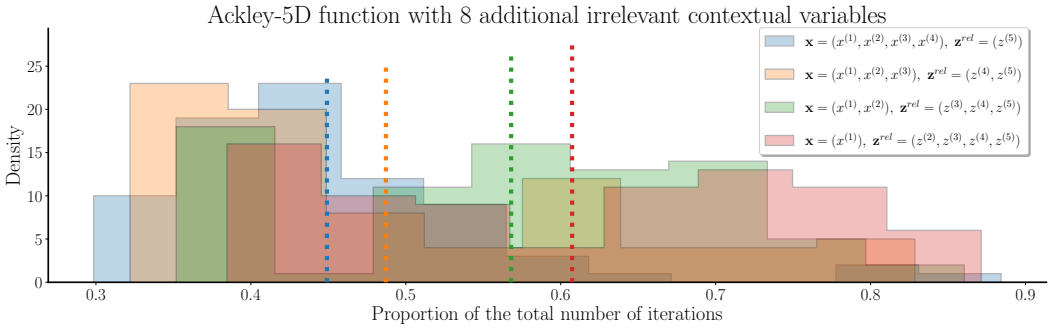

Figure S1: Distribution of early stopping time for `SADCBO` across 100 different BO trials. We consider the Ackley5D function with an increasingly larger ratio of relevant contextual variables over design variables, and 8 irrelevant contextual variables. $p(\mathbf{z}) = \mathcal{U}([0,1]^c)$. For any variable, the associated query cost is 1. As the impact of contextual variables on the output function grows, the proportion of iterations spent in the observational phase grows as well.

D.1 Number of irrelevant contextual variables.

We compare the performances reached by `SADCBO` when adding an increasingly larger number of noise variables. `SADCBO` is able to keep up with the oracle baseline `OBO` as dimensionality grows, except for Michalewicz8D, an high-dimensional case (Figure S2). Let us also mention a general tendency from `SADCBO` to better handle dimensionality compared to `MMDCBO` (orange curve).

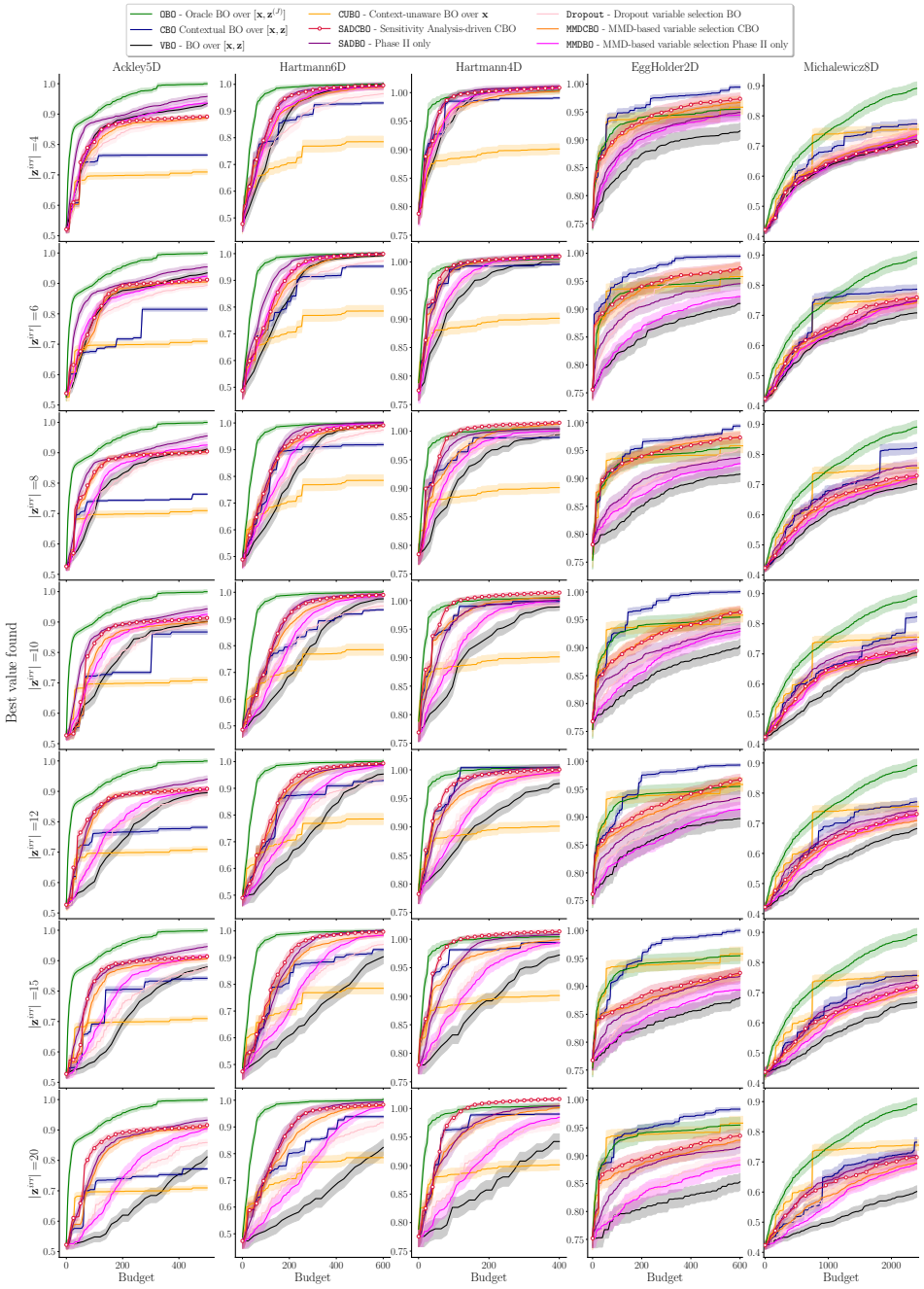

Figure S2: Varying the number of irrelevant contextual variables. For any variable, the associated query cost is 1. $p(\mathbf{z}) = \mathcal{U}([0,1]^c)$. On the three test functions Ackley5D, Hartmann6D and Hartmann4D, our approach outperforms other baselines and remains close to the oracle`OBO`, even in high dimensions.

D.2 Contextual variables intervention cost.

Many different scenarios will show up in a real-world setting. We now investigate four different query cost models, described as column headings in Figure S3, and show that `SADCBO` performs well for reasonably high enough contextual variable cost. For expensive contextual variables (third and fourth columns), `SADCBO` generally improves over `SADBO`. This highlights the importance of the contextual observational phase in the careful determination of which contextual variables justify the expense. Unsurprisingly however, when the cost to intervene on contextual variables is cheap relative to the cost to query design variables (first and second columns), `SADBO` (purple) catches up with `SADCBO` (red), even outperforming it on the Ackley5D and Hartmann6D functions. In other words, when the cost to intervene on contextual variables is sufficiently low, one should skip the observational phase and start by directly optimizing contextual variables.

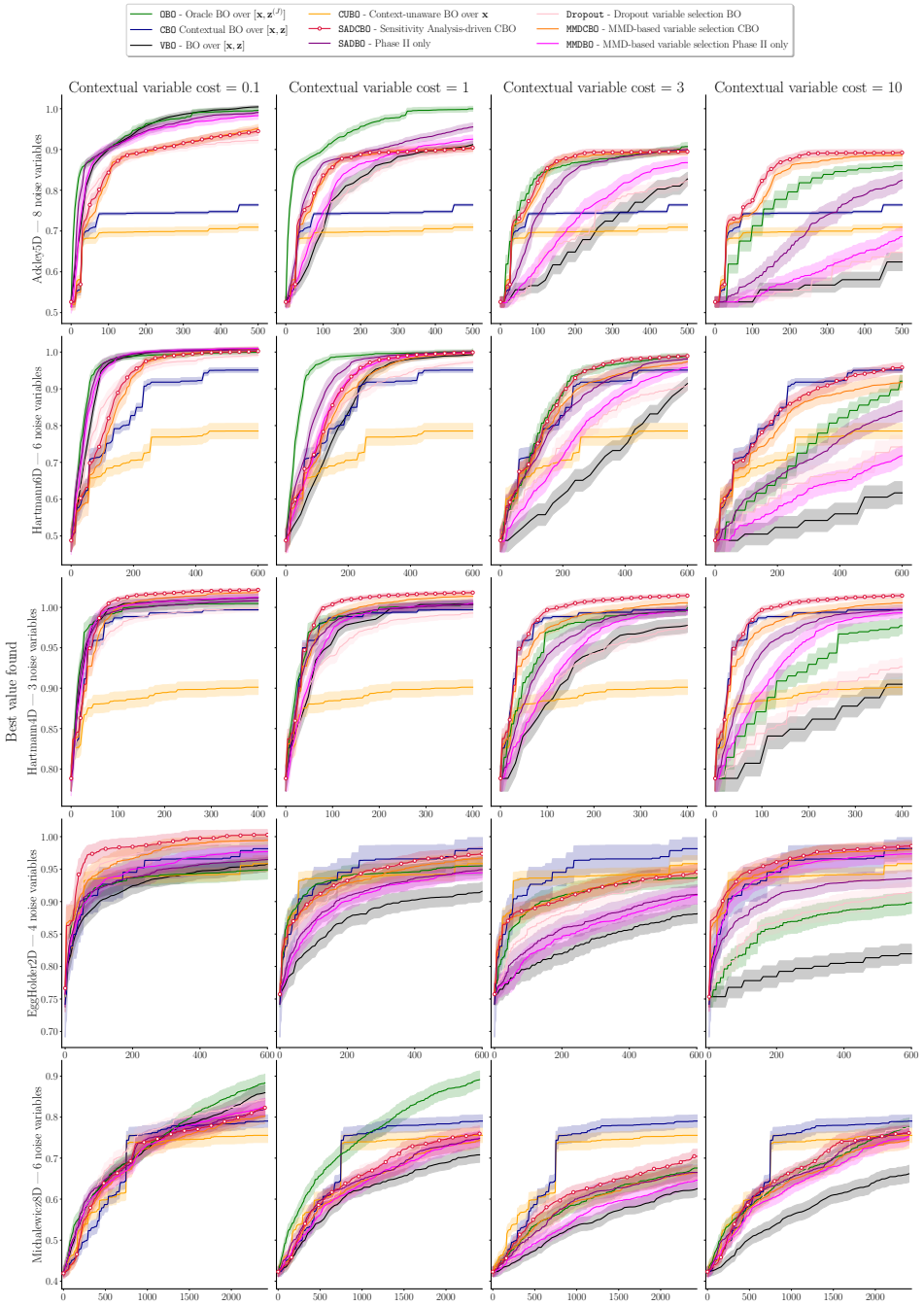

Figure S3: Ablation study on contextual variable query cost. Design variables have cost 1. $p(\mathbf{z}) = \mathcal{U}([0,1]^c)$. Low costs typically favor `SADBO` which can intervene on contextual variables from the start for a cheap price, whereas high costs lead to improved results for `SADCBO` , highlighting the importance of an observational phase.

D.3 Kernel structure.

So far, we considered a product kernel over design-context pairs:

$$k((\mathbf{x}, \mathbf{z}), (\mathbf{x}', \mathbf{z}')) = k_X(\mathbf{x}, \mathbf{x}')k_Z(\mathbf{z}, \mathbf{z}'). \tag{S6}$$

Many classical kernels satisfy this structure, e.g. the RBF and Matern kernels. Thus, two context-design pairs are similar if the contexts are similar *and* if the designs are similar. As mentioned by Krause and Ong (2011), one can also consider the additive combination $k((\mathbf{x}, \mathbf{z}), (\mathbf{x}', \mathbf{z}')) = k_X(\mathbf{x}, \mathbf{x}') + k_Z(\mathbf{z}, \mathbf{z}')$, such that context-design pairs can be found to be similar when the contexts are highly similar (even if the designs are not similar). We report performances for both kernels side-by-side in Figure S4, using RBF kernels both for k_X and k_Z. For functions where every variable has the same impact like Ackley5D, Michalewicz8D, and EggHolder2D, results are similar. On the contrary, a sharp decrease in the best value found occurs for all baselines on Hartmann6D and Hartmann4D, particularly in the case of `SADCBO` .

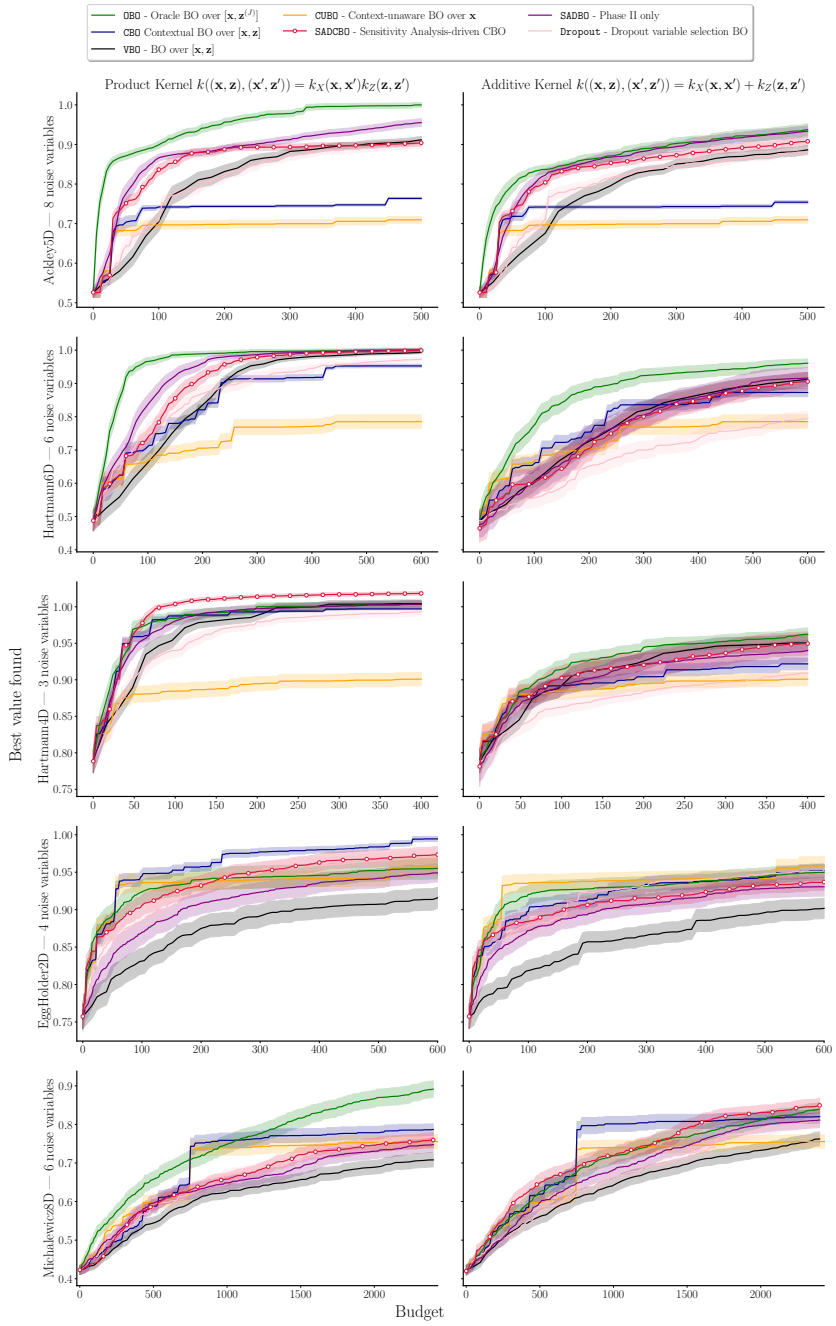

Figure S4: Using different kernel structures for the GP surrogate. Left: product structure over design and contextual variables. Right: additive kernel over design and contextual variables. For any variable, the associated query cost is 1. $p(\mathbf{z}) = \mathcal{U}([0, 1]^c)$. The additive kernel structure negatively impacts all baselines.

D.4 SADCBO hyperparameters.

We vary the 3 hyperparameters of SADCBO: $\eta \in [0, 1]$ the threshold based over the cumulative sum of sensitivity indices, which in turn regulates how many variables are selected every iteration; $\gamma \in [0, 1]$, a threshold upon which a value is considered high enough to have its input added to dataset $\mathcal{D}^{\gamma}$ Equation (3), used for sensitivity analysis; and Q the size of the dataset $\mathcal{D}^{Q}$.

Figure S5 reports the performances both for SADCBO (shades of red) and SADBO (shades of blue). Unsurprisingly, η stands out as the most stringent parameter: as its value decreases, fewer variables are included, at which point not all relevant ones are selected, leading to reduced performances. Note that in a setting where there are no relevant contextual variables, lower values of η will actually lead to better performances. This is investigated in Appendix E. Then, varying $\gamma \in [0, 1]$ slightly affects the results: γ increasing means that more samples are collected for sensitivity analysis, but these are less relevant for producing a reliable set of variables accounting for the fluctuations at the optimum. Finally, for the examples considered, Q has only a limited effect, close to that of varying γ. This might stem from the fact that batched acquisition functions are notoriously difficult to optimize and may sometimes struggle to enforce diversity.

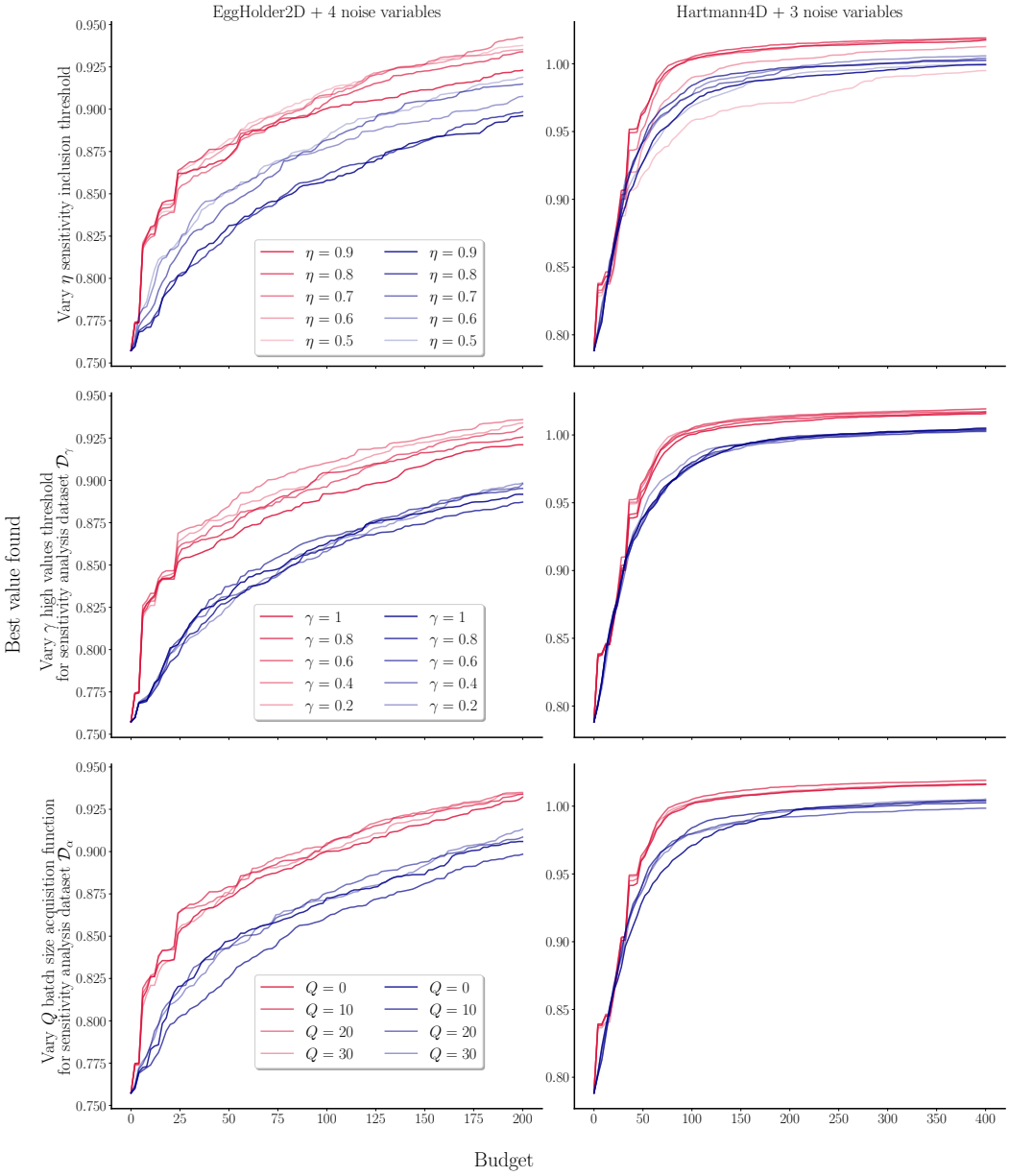

Figure S5: Varying hyperparameters for SADCBO and SADBO. For any variable, the associated query cost is 1. $p(\mathbf{z}) = \mathcal{U}([0,1]^c)$. Top: varying η, the contextual variable inclusion threshold over the cumulative sum of sensitivity indices. Middle: varying γ, the threshold used in the creation of the truncated dataset $\mathcal{D}^\gamma$ from Equation (3). Bottom: varying Q, the size of the dataset $\mathcal{D}^Q$ from **??**. η is the most sensitive hyperparameter here.

Appendix E. Forward variable selection further improves the performance of SADCBO

In the SADCBO method, once the FC variable relevance indices have been computed using Equation (2) and sorted in descending order, the contextual variables selected are those whose cumulative FC score is greater than $\eta \in [0, 1]$. This selected set explains the fraction η of the output sensitivity amongst all contextual variables. This can prove problematic in the extreme case where none of the contextual variables at hand have any impact on the function, as we will still have to select sufficiently enough variables to reach $100\eta\%$ of the output sensitivity.

An alternative approach described by Shen and Kingsford (2021) can be employed to tackle this issue. Using the sorted sensitivity indices, one performs forward variable selection by fitting GP surrogates which include an increasingly larger number of (highest ranked) contextual variables. The stopping criterion on the addition of contextual variables to the surrogate is computed based on a comparison of the negative Marginal Log-Likelihood (MLL) between nested models. This introduces a hyperparameter β, which we set to 10, similarly as Shen and Kingsford (2021). Note that the hyperparameter η is no longer necessary with this approach. Algorithm S2 summarizes the forward variable selection process performed at each BO iteration. We report the performance of this baseline, coined SADCBO + Forward selection on two experiments. In the first scenario (Figure S6), where the number of irrelevant contextual variables is varied, performing forward variable selection based on the sensitivity indices leads to faster convergence to the optimum, specifically as the number of irrelevant contexts grows large ($|\mathbf{z}^{irr}| \geq 12$). In the second scenario (Figure S7), we consider the Ackley5D (resp. Hartmann6D) function, but this time there are no relevant contextual variables: all relevant dimensions are associated with design variables, and there are 8 (resp. 6) additional irrelevant contextual variables. Performing forward variable selection leads to a marginally faster convergence to the optimum.

Algorithm S2 Forward variable selection

Input: Dataset $\mathcal{D}$, contextual variables $[\mathbf{z}_{\text{sort}}^{(1)}, \dots, \mathbf{z}_{\text{sort}}^{(c)}]$ sorted in descending order of relevance according to Equation (2).
Let $L_{\mathbf{z}_{\text{sort}}^{(0)}}$ be the negative MLL of the GP fitted on $\mathcal{D}$ using only design variables $\mathbf{x}$
$j^\star \leftarrow c$
for $j = 1, \dots, c$ **do**
  Fit a GP on $\mathcal{D}$ using $[\mathbf{x}, \mathbf{z}_{\text{sort}}^{(1)}, \dots, \mathbf{z}_{\text{sort}}^{(j)}]$, with negative MLL $L_{\mathbf{z}_{\text{sort}}^{(j)}}$
  if $j = 1$ and $L_{\mathbf{z}_{\text{sort}}^{(j)}} < L_{\mathbf{z}_{\text{sort}}^{(j-1)}}$ **then**
    $j^\star \leftarrow 1$
    break
  else if $L_{\mathbf{z}_{\text{sort}}^{(j)}} < L_{\mathbf{z}_{\text{sort}}^{(j-1)}}$ or $L_{\mathbf{z}_{\text{sort}}^{(j)}} - L_{\mathbf{z}_{\text{sort}}^{(j-1)}} < (L_{\mathbf{z}_{\text{sort}}^{(j-1)}} - L_{\mathbf{z}_{\text{sort}}^{(j-2)}})/\beta$ **then**
    $j^\star \leftarrow j$
    break
  end if
end for
return $[\mathbf{z}_{\text{sort}}^{(1)}, \dots, \mathbf{z}_{\text{sort}}^{(j^\star)}]$

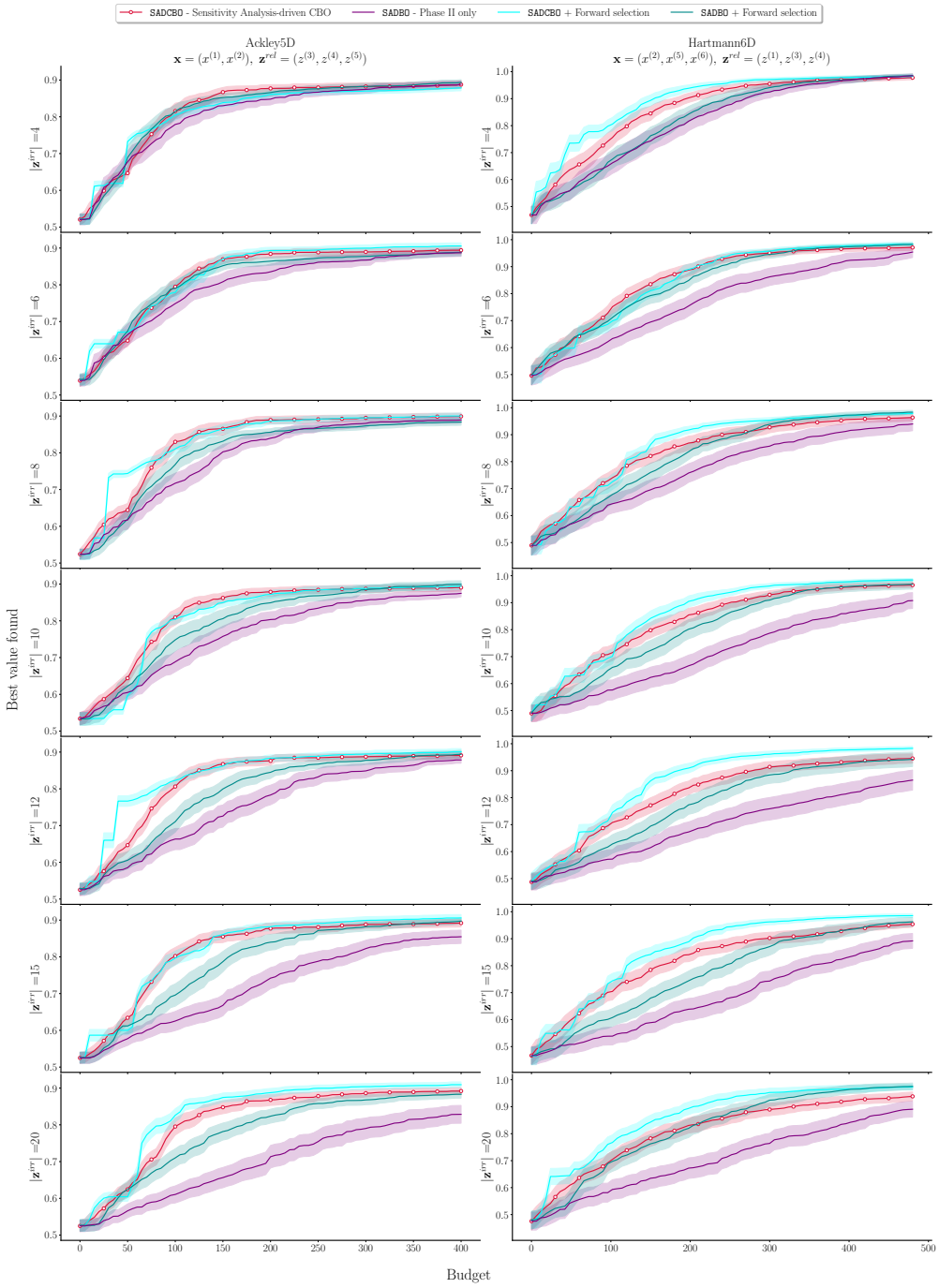

Figure S6: Comparison of `SADCBO` and `SADCBO` + forward selection, when increasing the number of irrelevant contextual variables. For any contextual (resp. design) variable, the associated query cost is 3 (resp. 1). $p(\mathbf{z}) = \mathcal{U}([0,1]^c)$. For both test functions, Sensitivity Analysis-driven CBO (red curve) remains competitive, even in high dimensions. Forward variable selection leads to faster convergence, specifically starting when the number of irrelevant variables $|\mathbf{z}^{irr}|$ reaches 12 or more.

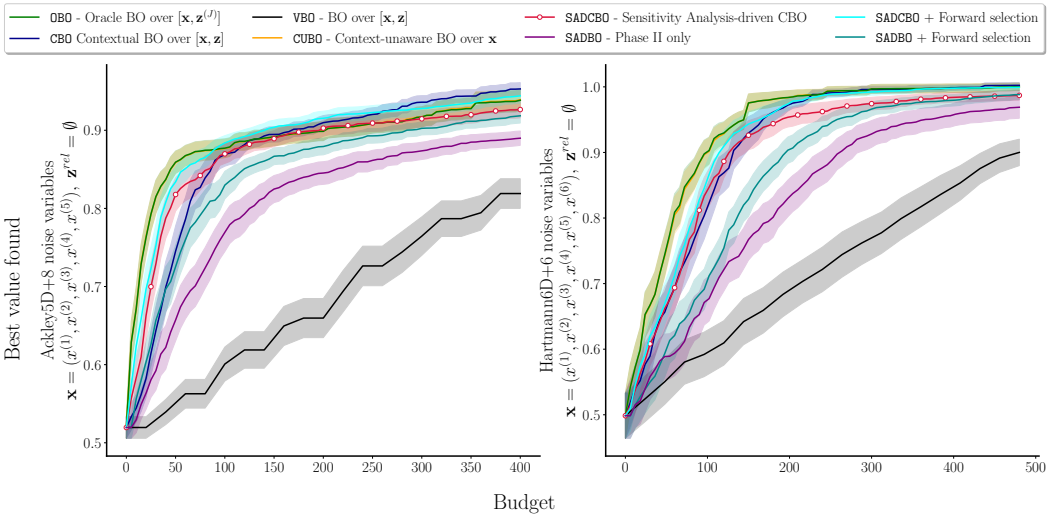

Figure S7: Synthetic examples where no relevant contextual variable is present. For any contextual (resp. design) variable, the associated query cost is 3 (resp. 1). $p(\mathbf{z}) = \mathcal{U}([0,1]^c)$. The baseline performing forward variable selection on top of `SADCBO` (cyan curve) provides a slight but consistent improvement over `SADCBO` (red curve), and likewise for `SADBO` (dark cyan versus purple curve).

References

Milad Abolhasani and Keith A. Brown. Role of AI in experiment materials science. *MRS Bulletin*, 2023.

Vahan Arsenyan, Antoine Grosnit, and Haitham Bou-Ammar. Contextual causal Bayesian optimisation. *arXiv preprint arXiv:2301.12412*, 2023.

Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka, and Volkan Cevher. Adversarially robust optimization with Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Sait Cakmak, Raul Astudillo Marban, Peter Frazier, and Enlu Zhou. Bayesian optimization of risk measures. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20130–20141. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/e8f2779682fd11fa2067beffc27a9192-Paper.pdf`.

David Case, H. Metin Aktulga, Kellon Belfon, Ido Ben-Shalom, Joshua Berryman, Scott Brozell, David Cerutti, Thomas Cheatham, Gerardo Andrés Cisneros, Vinícius Cruzeiro, Tom Darden, Negin Forouzesh, George Giambasu, Timothy Giese, Michael Gilson, Holger Gohlke, Andreas Götz, Julie Harris, Saeed Izadi, and Peter Kollman. Amber 2023, 04 2023.

David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

Lincan Fang, Esko Makkonen, Milica Todorović, Patrick Rinke, and xi Chen. Efficient amino acid conformer search with bayesian optimization. *Journal of Chemical Theory and Computation*, 17, 02 2021. doi: 10.1021/acs.jctc.0c00648.

J. Gerritsma, R. Onnink, and A. Versluis. Yacht Hydrodynamics. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5XG7R.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4 (2):268–276, 2018.

A Gretton, K Borgwardt, M J Rasch, and B Scholkopf. A kernel two-sample test. *J. of Mach. Learn. Res.*, 13:723–773, 2012.

Kate Higgins, Maxim Ziatdinov, Sergei Kalinin, and Mahshid Ahmadi. High-throughput study of antisolvents on the stability of multicomponent metal halide perovskites through robotics-based synthesis and machine learning approaches. *Journal of the American Chemical Society*, 2021.

Hideaki Ishibashi, Masayuki Karasuyama, Ichiro Takeuchi, and Hideitsu Hino. A stopping criterion for Bayesian optimization by the gap of expected minimum simple regrets. In *Proceedings of The International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13:455–492, 1998.

Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally Robust Bayesian Optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric Xing. ChemBO: Bayesian Optimization of Small Organic Molecules with Synthesizable Recommendations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Andreas Krause and Cheng Ong. Contextual Gaussian Process Bandit Optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. High Dimensional Bayesian Optimization Using Dropout. *arXiv preprint arXiv:1802.05400*, 2018.

Sulin Liu, Qing Feng, David Eriksson, Benjamin Letham, and Eytan Bakshy. Sparse Bayesian Optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Philip W. Nega, Zhi Li, Victor Ghosh, Janak Thapa, Shijing Sun, Noor Titan Putri Hartono, Mansoor Ani Najeeb Nellikkal, Alexander J. Norquist, Tonio Buonassisi, Emory M. Chan, and Joshua Schrier. Using automated serendipity to discover how trace water promotes and inhibits lead halide perovskite crystal formation. *Applied Physics Letters*, 119(4), 07 2021.

C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Romelia Salomon Ferrer, David Case, and Ross Walker. An overview of the amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3, 03 2013. doi: 10.1002/wcms.1121.

Isaac Sebenius, Topi Paananen, and Aki Vehtari. Feature collapsing for gaussian process variable ranking. In *Proceedings of The International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

Yihang Shen and Carl Kingsford. Computationally Efficient High-Dimensional Bayesian Optimization via Variable Selection. *arXiv preprint arXiv:2109.09264*, 2021.

I.M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 55(1):271–280, 2001.

Adrien Spagnol, Rodolphe Le Riche, and Sébastien Da Veiga. Bayesian optimization in effective dimensions via kernel-based sensitivity indices. In *Proceedings of the International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP)*, 2019.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, may 2012. doi: 10.1109/tit.2011.2182033.

James T. Wilson, Riccardo Moriconi, Frank Hutter, and Marc Peter Deisenroth. The reparameterization trick for acquisition functions. *arXiv preprint arXiv:1712.00424*, 2017.

Yichi Zhang, Daniel W Apley, and Wei Chen. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific Reports*, 10(1):1–13, 2020.

Juliusz Ziomek and Haitham Bou-Ammar. Are Random Decompositions all we need in High Dimensional Bayesian Optimisation?, 2023.