SMOG: Scalable Meta-Learning for Multi-Objective Bayesian Optimization

Leonard Papenmeier
Department of Information Systems
University of Münster, Germany
leonard.papenmeier@uni-muenster.de

Petru Tighineanu
Bosch Center for Artificial Intelligence
Renningen, Germany

Abstract

Multi-objective optimization aims to solve problems with competing objectives. Evaluating such problems is often slow or expensive, limiting the budget of evaluations. In many applications, historical data from related optimization tasks is available and can be leveraged via *meta-learning* to accelerate optimization. Bayesian optimization, as a promising technique for expensive black-box problems, has been extended independently to meta-learning and multi-objective optimization, but methods that simultaneously address both settings remain largely unexplored. We propose SMOG—a scalable and modular meta-learning model based on a multi-output Gaussian process—that explicitly learns correlations between objectives. SMOG builds a structured joint Gaussian process prior across meta- and target tasks and, after conditioning on metadata, yields a closed-form prior for the target task. This construction propagates metadata uncertainty into the target surrogate in a principled way. SMOG supports hierarchical, parallel training, achieving linear scaling with the number of meta-tasks. The resulting surrogate integrates seamlessly with standard multi-objective Bayesian optimization acquisition functions. We demonstrate that our method is consistently competitive, delivering strong data efficiency across representative benchmarks and applications.

1 Introduction

Many high-impact optimization problems are intrinsically *multi-objective*: engineers and machine-learning practitioners rarely optimize a single scalar, but rather trade off competing objectives such as performance, cost, latency, or energy. At the same time, these objectives are often *expensive* to evaluate, naturally putting one in the low-data regime with a small total budget of tens of evaluations per task where Bayesian optimization (BO)/multi-objective Bayesian optimization (MOBO) is particularly effective [1–5]. Organizations rarely run non-recurring optimizations; instead, they accumulate logs from past optimization runs for related products, machines, datasets, workloads, or environments. This makes *meta-learning for expensive multi-objective optimization* an important scenario: rather than starting each new optimization from scratch, practitioners aim to save resources by leveraging prior tasks to achieve good solutions with only a few evaluations. Examples span industrial process tuning and calibration (where competing quality metrics must be balanced), scientific design (e.g., materials discovery and advanced manufacturing with multiple competing properties), and machine-learning system design (e.g., multi-objective hyperparameter and architecture tuning trading off accuracy, latency, and resource use) [6–11].

Despite its promise, meta-learning for multi-objective optimization is technically subtle in the low-data regime. First, in multi-objective settings, "what to transfer" is not a single optimum but information about a *Pareto set/front*, and decision-making typically depends on uncertainty-aware criteria (e.g., hypervolume-based utilities) [4, 5, 12]. Second, historical data are often scarce and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

heterogeneous across tasks; transfer must therefore account for *meta-task uncertainty* to avoid overconfident bias from weak or mismatched prior tasks [13–16]. Third, multi-objective problems add another layer: objectives can be correlated, so efficiently using evidence often requires non-trivial *probabilistic multi-output surrogates*. Treating objectives independently can waste information precisely when evaluations are precious. These challenges leave a key gap: we need meta-learning methods that are (i) uncertainty-aware, (ii) scalable across many meta-tasks, and (iii) able to exploit cross-objective correlations. Fully joint multi-task multi-output Gaussian process (GP) models are principled but quickly become infeasible when information from many related tasks is available [17, 18], while most scalable alternatives are developed for single-objective transfer and treat objectives independently when adapted to MOBO.

Contribution. We introduce `SMOG`[1], a multi-output GP surrogate that meta-learns a target-task prior with learned cross-objective covariance. `SMOG` is the first surrogate to simultaneously deliver three properties for meta-MOBO in the low-data regime: *fully Bayesian* uncertainty propagation over all meta-task data, *linear scaling* in the number of meta-tasks, and *learned correlations* between objectives. The single-objective `ScaML-GP` [16] is recovered as a specialization. `SMOG`'s posterior plugs directly into standard MOBO pipelines (e.g., hypervolume (HV)-based acquisition optimization) [4, 5, 12]. `SMOG` is a top contender across benchmark families and performs robustly.

2 Related work

Most work on meta-learning for BO focuses on learning a better surrogate for a single-objective target task. A principled approach is to build a *joint* Bayesian model across tasks (e.g., a multi-task GP), which yields coherent uncertainty estimates but is computationally prohibitive—scaling cubically in the total number of observations and, at best, quadratically in the task-correlation hyperparameters (HPs) [18–26]. To improve scalability, a number of methods rely either on heuristic combinations of per-task surrogates (e.g., GP ensembles) [13, 15, 27] or on building a parametric GP prior on the metadata [28–32]. These approaches scale better but sacrifice a joint Bayesian treatment and thus principled uncertainty propagation across tasks. A recent work, `ScaML-GP` [16], addresses this tension by introducing assumptions that lead to a modular GP model: conditioning on meta-data exposes a modular decomposition into M independent meta-task GP posteriors and a target-task GP prior, enabling scalable and fully Bayesian transfer.

In contrast to the rich single-objective literature, meta-learning methods that directly target multi-objective optimization remain scarce. A notable exception is the task-similarity extension of `MO-TPE` by Watanabe et al. [33], which transfers knowledge by reweighting the acquisition based on task similarity. While scalable, it is based on density ratios and lacks a unified probabilistic multi-output surrogate. Recent few-shot surrogate-assisted evolutionary methods for expensive multi-objective optimization meta-learn surrogates [34]. However, neither approach learns a correlated multi-output posterior and thus cannot exploit cross-objective dependencies or propagate meta-task uncertainty to the target surrogate in a principled way. Our work targets this underexplored regime by combining scalable GP meta-learning with a multi-output surrogate that models cross-objective dependencies.

Desiderata. Each of the methods above fails at least one of the requirements (i)–(iii) in Section 1: per-objective application of `ScaML-GP` [16] forgoes (iii); fully joint multi-task GPs over $(\text{task}, \text{objective})$ pairs [19] are cubic in the meta-observation count [18], violating (ii); ensemble combinations of per-task posteriors [27, 35] replace principled conditioning with heuristic weighting, sacrificing (i). `SMOG`, introduced next, is the specific design that satisfies all three.

3 Method

We aim to find a set of Pareto-optimal solutions of a *target* black-box function

$$\boldsymbol{f}_t : \mathcal{X} \to \mathbb{R}^O,$$

where $\mathcal{X} \subset \mathbb{R}^D$ is the search space of dimensionality D, and O is the number of objectives. Observations $\boldsymbol{y}_n = (y_{n,o})_{o \in \mathcal{O}}$ of $\boldsymbol{f}_t$ may be corrupted by independent zero-mean Gaussian noise, $\boldsymbol{y}_n = \boldsymbol{f}_t(\boldsymbol{x}_n) + \boldsymbol{\varepsilon}_n$ with $\boldsymbol{\varepsilon}_n \sim \mathcal{N}\big(\mathbf{0}, \operatorname{diag}(\sigma_1^2, \ldots, \sigma_O^2)\big)$. We use the target data $\mathcal{D}_t = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N_t}$

[1] Code is available under `https://github.com/boschresearch/scalable-meta-learning-for-mobo`.

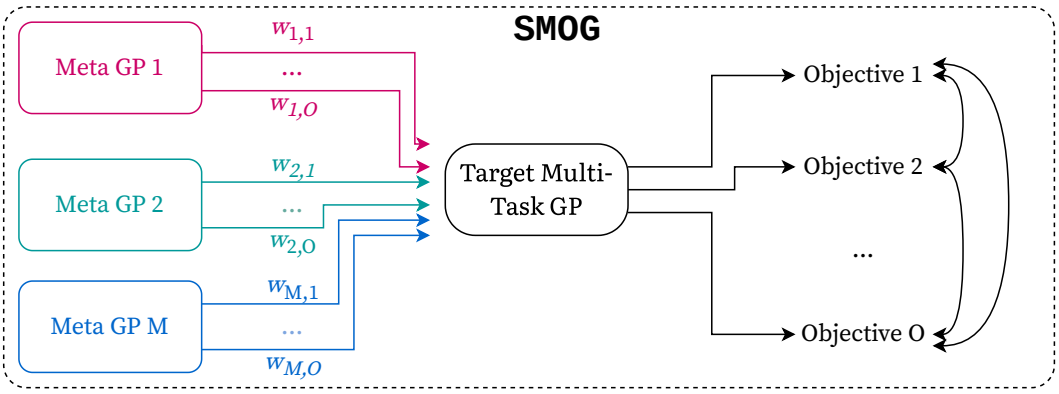


Figure 1: High-level view of SMOG: Meta tasks are modeled independently across tasks, with each source task represented by its own multi-output Kronecker-structured GP. The target GP combines the weighted means and covariance blocks of these source GPs with a residual multi-output kernel to form an informative target-task prior.

to build a probabilistic model. We model $\boldsymbol{f}_t$ with a multi-output GP prior with mean $m_o(\cdot)$ and kernel $k_{oo'}(\cdot,\cdot)$ for $o, o' \in \mathcal{O} = \{1, \dots, O\}$. Conditioned on $\mathcal{D}_t$, the posterior is a GP with mean and covariance [17]

$$\begin{aligned} \hat{m}_{t,o}(\boldsymbol{x}) &= m_o(\boldsymbol{x}) + \boldsymbol{k}((\boldsymbol{x}, o), \boldsymbol{X}_t) \left[\boldsymbol{K}(\boldsymbol{X}_t, \boldsymbol{X}_t) + \boldsymbol{\Sigma}_\varepsilon\right]^{-1} \left(\boldsymbol{Y}_t - \boldsymbol{m}(\boldsymbol{X}_t)\right), \\ \hat{k}_{t,oo'}(\boldsymbol{x}, \boldsymbol{x}') &= k((\boldsymbol{x}, o), (\boldsymbol{x}', o')) - \boldsymbol{k}((\boldsymbol{x}, o), \boldsymbol{X}_t) \left[\boldsymbol{K}(\boldsymbol{X}_t, \boldsymbol{X}_t) + \boldsymbol{\Sigma}_\varepsilon\right]^{-1} \boldsymbol{k}(\boldsymbol{X}_t, (\boldsymbol{x}', o')), \end{aligned} \tag{1}$$

where $\boldsymbol{X}_t = \big((\boldsymbol{x}_n, o)\big)_{n=1,\dots,N_t,\ o\in\mathcal{O}}$ stores points in the search space with their objective index and $\boldsymbol{Y}_t = \big(y_{n,o}\big)_{n,o}$ the corresponding observations. We assume per-objective Gaussian noise $\boldsymbol{\Sigma}_\varepsilon = \operatorname{diag}(\sigma_1^2 \boldsymbol{I}_{N_t}, \dots, \sigma_O^2 \boldsymbol{I}_{N_t})$. To leverage related tasks, we assume access to $M \geq 1$ meta-tasks with datasets $\mathcal{D}_{1:M} = \bigcup_{m\in\mathcal{M}} \mathcal{D}_m$, where $\mathcal{M} = [M]$. Each meta-task m provides a dataset $\mathcal{D}_m = \{(\boldsymbol{x}_{m,n}, \boldsymbol{y}_{m,n})\}_{n=1}^{N_m}$ with $\boldsymbol{y}_{m,n} \in \mathbb{R}^O$ and per-objective Gaussian noise $\boldsymbol{y}_{m,n} = \boldsymbol{f}_m(\boldsymbol{x}_{m,n}) + \boldsymbol{\varepsilon}_{m,n}$, $\boldsymbol{\varepsilon}_{m,n} \sim \mathcal{N}\big(\boldsymbol{0}, \operatorname{diag}(\sigma_{m,1}^2, \dots, \sigma_{m,O}^2)\big)$. For notational convenience, we collect all observations of meta-task m by concatenating points with their objective index and corresponding outputs as $\boldsymbol{X}_m = \big((\boldsymbol{x}_{m,n}, o)\big)_{n=1,\dots,N_m,\ o\in\mathcal{O}}$ and $\boldsymbol{Y}_m = \big(y_{m,n,o}\big)_{n=1,\dots,N_m,\ o\in\mathcal{O}}$.

SMOG extends ScaML-GP [16] to combine meta-learning with multi-objective optimization, inheriting its task scalability, modularity, and principled uncertainty propagation. SMOG satisfies requirements (i)–(iii) of Section 1 through three design choices: a per-objective additive decomposition with objective-specific weights w_{mo} leading to a closed-form prior after conditioning on metadata in Theorem 1, satisfying (i); a structured-coupling assumption that keeps the cross-task parameter count linear in M in Assumption 2, satisfying (ii); and propagating full objective-objective covariance blocks from Assumption 2 to Theorem 1, satisfying (iii). In the following, we focus the presentation on the novel components of SMOG (the multi-output extension of Assumption 2, objective-specific weights, and the target-task prior with full objective-objective covariance), referring the reader to Tighineanu et al. [16] for a detailed discussion on the shared components (Assumptions 1 and 3).

3.1 The SMOG kernel

We assume that meta- and target data are described by a multi-task kernel over all tasks and objectives

$$k\left[(\boldsymbol{x}, \nu, o), (\boldsymbol{x}', \nu', o')\right] = \sum_{v\in\mathcal{M}^*} \sum_{\theta\in\mathcal{O}} \left[c_{v\theta}\right]_{\nu o, \nu' o'} k_{v\theta}(\boldsymbol{x}, \boldsymbol{x}'), \tag{2}$$

where $\mathcal{M}^* = \mathcal{M} \cup \{t\}$ is the set of all tasks, t denotes the target index, $\nu, \nu' \in \mathcal{M}^*$ are task indices, $o, o' \in \mathcal{O}$ are indices of the objective, $k_{v\theta}$ are arbitrary kernel functions, and $\boldsymbol{C}_{v\theta}$ positive semi-definite (PSD) matrices called *coregionalization matrices*, since their entries $\left[c_{v\theta}\right]_{\nu o, \nu' o'}$ model the covariances between two objectives (ν, o) and (ν', o') [18]. We impose two assumptions on the multi-task GP in Equation (2) that focus learning on the most informative covariance terms.

Assumption 1. Distinct meta-task models are uncorrelated: $\mathrm{Cov}\left(f_{mo}, f_{m'o'}\right) = \delta_{m=m'} k_m\left[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')\right]$ for all $m, m' \in \mathcal{M}$ and $o, o' \in \mathcal{O}$.

SMOG inherits Assumption 1 from ScaML-GP. It gives scalability in M, but by itself does not yield a practical method, since the target-meta couplings can still involve all meta-task kernels and the number of HPs grows quadratically with M. We thus impose a second structural assumption to obtain a practical model in the low-data regime.

Assumption 2. The target-task model is given by a sum of scaled meta-task functions, $\sum_{m\in\mathcal{M}} \tilde{f}_{mo}$, and a residual function, $\tilde{f}_{to}$. Explicitly, $f_{to} = \tilde{f}_{to} + \sum_{m\in\mathcal{M}} \tilde{f}_{mo}$, with $\left|\mathrm{Corr}(\tilde{f}_{mo}, f_{mo})\right| = 1$, $\mathrm{Cov}(\tilde{f}_{to}, f_{mo'}) = 0$, and $\mathrm{Cov}(\tilde{f}_{to}, \tilde{f}_{to'}) = k_t\left[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')\right]$, $m \in \mathcal{M}$, $o, o' \in \mathcal{O}$.

We dissect Assumption 2 component by component, and discuss where SMOG's novelty lies. (i) The additive target decomposition, $f_{to} = \tilde{f}_{to} + \sum_{m\in\mathcal{M}} \tilde{f}_{mo}$, acknowledges that meta-learning may be imperfect and requires a residual target-specific component. Here, SMOG extends ScaML-GP's formulation to a per-objective decomposition. Cross-objective dependencies are neglected to enforce explainability: in practice, different objectives may correspond to distinct physical units, and combining them can lead to an unclear representation. (ii) The structured-coupling assumption, $\mathrm{Cov}(\tilde{f}_{to}, f_{mo'}) = 0$, follows ScaML-GP's approach and prevents target-meta couplings from mixing all meta-task kernels, keeping the HP count under control. (iii) The link to actual meta-task functions, $|\mathrm{Corr}(\tilde{f}_{mo}, f_{mo})| = 1$, implies $\tilde{f}_{mo} = w_{mo} f_{mo}$ and yields the practical form $f_{to} = \tilde{f}_{to} + \sum_m w_{mo} f_{mo}$ with $w_{mo} \in \mathbb{R}$. SMOG's generalization discards cross-objective contributions for the same reasons as in (i). This meta-function link is a specific choice of the SMOG/ScaML-GP family; other links are possible, but they typically lead to a less interpretable target-task prior in Theorem 1. We leave the exploration of such alternatives to future work.

Lemma 1. *Applying Assumptions 1 and 2 to Equation* (2) *yields a sparse structure with the following non-zero entries of the coregionalization matrices*

$$\begin{aligned} [c_{t\theta}]_{to,to'} &= [h_{t\theta}]_{oo'}, & [c_{m\theta}]_{mo,mo'} &= [h_{m\theta}]_{oo'}, \\ [c_{m\theta}]_{mo,to'} &= w_{mo'}[h_{m\theta}]_{oo'}, & [c_{m\theta}]_{to,to'} &= w_{mo} w_{mo'} [h_{m\theta}]_{oo'}, \end{aligned} \tag{3}$$

where $[c_{\cdot\theta}]_{a,b} = [c_{\cdot\theta}]_{b,a}$ *and* $[h_{v\theta}]_{oo'}$ *are PSD in the basis* (o, o').

See Appendix A.1 for a proof and an illustrative example. This sparse structure decreases the number of kernel parameters from $\mathcal{O}\left[M^3 O^3\right]$ in Equation (2) to $\mathcal{O}\left[M O^3\right]$ in Equation (3), which is linear in M. In other words, the coregionalization matrices are now populated by $O \times O$ blocks of $[h_{v\theta}]_{oo'}$.

Lemma 2. *Assumptions 1 and 2 with* $w_{mo} \in \mathbb{R}$ *for* $m \in \mathcal{M}$ *and* $o \in \mathcal{O}$ *yield a valid kernel given by*

$$k_{\texttt{SMOG}}\left[(\boldsymbol{x}, \nu, o), (\boldsymbol{x'}, \nu', o')\right] = \sum_{v\in\mathcal{M}^*} g_v(\nu, o) g_v(\nu', o') k_v\left[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')\right], \tag{4}$$

where $g_v(\nu, o)$ *is one if* $v = \nu$, w_{mo} *if* $(v = m) \wedge (\nu = t)$, *and zero otherwise.*

See Appendix A.2 for a proof. By Lemma 2, we have a valid joint kernel defining the prior distribution over all meta- and target functions *before* observing any data. The number of parameters in our kernel scales linearly with M. The modular and scalable nature of SMOG is revealed when conditioning Equation (4) on the metadata, yielding a valid GP, as we show next.

Theorem 1. *Under a zero-mean GP prior with the multi-task kernel given by Equation* (4)*, the distribution of the target-task objectives conditioned on the metadata is*

$$f_{to} \mid \mathcal{D}_{1:M}, \boldsymbol{x} \sim \mathcal{GP}(m_{t,\texttt{SMOG}}(\boldsymbol{x}, o), k_{t,\texttt{SMOG}}[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')]) \tag{5}$$

with

$$\begin{aligned} m_{t,\texttt{SMOG}}(\boldsymbol{x}, o) &= \sum_{m\in\mathcal{M}} w_{mo} \hat{m}_{mo}(\boldsymbol{x}), \\ k_{t,\texttt{SMOG}}[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')] &= k_t[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')] + \sum_{m\in\mathcal{M}} w_{mo} w_{mo'} \hat{k}_{moo'}(\boldsymbol{x}, \boldsymbol{x'}), \end{aligned} \tag{6}$$

where $\hat{m}_{mo}(\boldsymbol{x})$ *and* $\hat{k}_{moo'}(\boldsymbol{x}, \boldsymbol{x'})$ *are the posterior mean and covariance functions of the individual multi-output meta-task GPs conditioned only on their corresponding data,* $\mathcal{D}_m$.

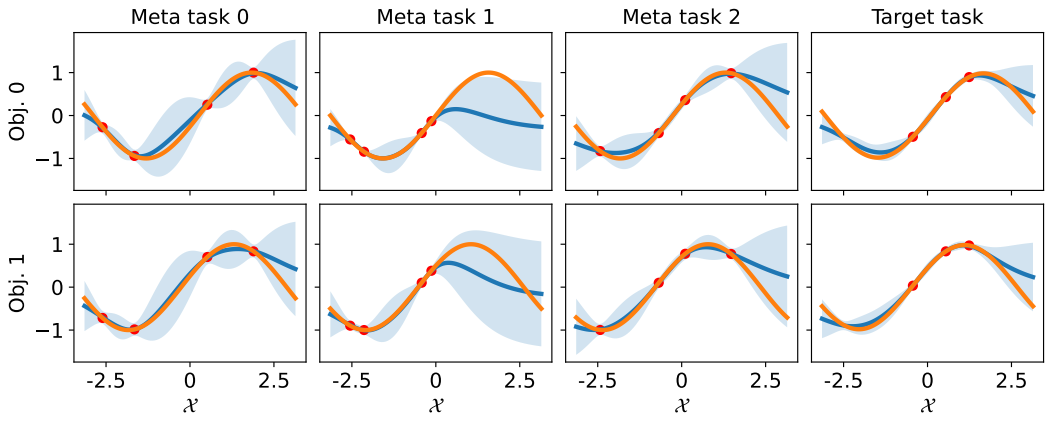


Figure 2: Example of a Sinusoidal function with two outputs (rows), three meta-tasks (columns 1–3), and one target task (column 4). SMOG learns a strong target-task posterior (solid blue line and shaded areas) of the objective function (solid orange line) by leveraging meta data (red dots for meta tasks 0–2) to learn an informative target-task prior, which is further refined by conditioning on the target data (red dots in the rightmost column). See Appendix C.3 for details on the benchmark.

See Appendix B for a proof. According to Theorem 1, the prior distribution of the target-task is also a GP, given by the weighted sum of the meta-task posteriors. The prior mean function is analogous to ScaML-GP [16], with a per-objective weighting of meta-task posterior means. SMOG's covariance prior is a generalization of ScaML-GP's: while the latter weighs per-task posterior covariances with a scalar w_m^2, SMOG uses products $w_{mo}w_{mo'}$ that couple the full objective–objective covariance blocks. SMOG's residual kernel, k_t, learns correlations that cannot be captured by the meta-learned contribution alone. According to Equation (6), meta-tasks that align with the target receive large weights, while unrelated ones are downweighted. SMOG can therefore quickly learn in the presence of even a few similar meta-tasks. The prior in Equation (6) is conditioned on $\mathcal{D}_t$ via Equation (1) to obtain the target-task posterior. If the objectives are independent, i.e., all meta-task coregionalization blocks and the residual kernel k_t are diagonal in the objectives, the SMOG model class reduces to one ScaML-GP model per objective. Our implementation still shares the input kernel and the noise variance across objectives (Section 3.3 and Appendix C.4). Figure 2 visualizes how SMOG meta-learns on multi-objective problems.

Complexity analysis. A key feature of SMOG is that its computational complexity scales linearly in M. Consider the cost incurred by the different steps of Algorithm 1. Line 3 involves the inversion of meta-task data with an overhead $\mathcal{O}(O^3 \sum_{m\in\mathcal{M}} N_m^3)$. This happens only once during pre-training, the results are cached. To construct the prior in line 4, we evaluate the meta-task posterior at $\boldsymbol{X}_t$ with an overhead $\mathcal{O}(O^3 N_t^2 \sum_{m\in\mathcal{M}} N_m + O^3 N_t \sum_{m\in\mathcal{M}} N_m^2)$. Finally, line 5 involves inverting the target-task kernel matrix, taking $\mathcal{O}(O^3 N_t^3)$. All these terms scale linearly in M and are cheap to evaluate in the practically relevant regime of small N_t and O, and moderate data per meta-task. We empirically validate this analysis in Appendix D.

3.2 Hyperparameter inference

As for ScaML-GP, we assume that the HPs of each meta-task model, $\boldsymbol{\theta}_m$, are independent of the target data [36].

Assumption 3. For all meta-tasks $m \in \mathcal{M}$, $p(\boldsymbol{\theta}_m \mid \mathcal{D}_m, \mathcal{D}_t) = p(\boldsymbol{\theta}_m \mid \mathcal{D}_m)$.

Assumption 3 expresses a no-feedback modularization: meta-task HPs are inferred without feedback from the target data. In the regime we consider, where target data are scarce relative to the metadata, this reduces undesirable target-to-meta feedback and leads to a modular posterior

$$p(\boldsymbol{\theta}_{1:M}, \boldsymbol{\theta}_t \mid \mathcal{D}) = \left[\prod_{m=1}^{M} p(\boldsymbol{\theta}_m \mid \mathcal{D}_m)\right] p(\boldsymbol{\theta}_t \mid \mathcal{D}, \boldsymbol{\theta}_{1:M}), \tag{7}$$

Algorithm 1 SMOG

1: **Input:** metadata $\mathcal{D}_{1:M} = \cup_{m \in \mathcal{M}} \mathcal{D}_m$
2: **Output:** meta-learned multi-objective model $\mathcal{GP}(\mathcal{D}_t, \mathcal{D}_{1:M})$
3: Train individual GPs per meta-task and optimize $\boldsymbol{\theta}_m$
4: Construct the target-task prior as in Equation (6), and cache $\hat{\boldsymbol{m}}_{mo}(\boldsymbol{X}_t)$ and $\hat{\boldsymbol{K}}_{moo'}(\boldsymbol{X}_t, \boldsymbol{X}_t)$
5: Optimize the target-task HPs $\boldsymbol{\theta}_t$ as in Equation (9)
6: Condition the prior on $\mathcal{D}_t$ to obtain the posterior distribution for $\boldsymbol{f}_t$ as in Equation (1)

with $\mathcal{D} = \mathcal{D}_t \cup \mathcal{D}_{1:M}$. Being a GP, SMOG is amenable to fully Bayesian inference. In our experiments, however, we follow a modular plug-in MAP approximation [37]:

$$\boldsymbol{\theta}_m^\star = \arg\max_{\boldsymbol{\theta}_m} \log p(\boldsymbol{\theta}_m \mid \mathcal{D}_m), \quad \forall m \in \mathcal{M}, \tag{8}$$

followed by

$$\boldsymbol{\theta}_t^\star = \arg\max_{\boldsymbol{\theta}_t} \log p(\boldsymbol{\theta}_t \mid \mathcal{D}, \boldsymbol{\theta}_{1:M}^\star). \tag{9}$$

This two-stage optimization does not recover the joint posterior mode over $(\boldsymbol{\theta}_{1:M}, \boldsymbol{\theta}_t)$; rather, it is the optimization counterpart of the modularized posterior in Equation (7). We summarize SMOG in Algorithm 1 and show a conceptual diagram in Figure 1.

3.3 Equicorrelated residual target-task kernel

SMOG has a modular structure: conditioning on the metadata collapses the original monolithic GP into $M + 1$ interacting GPs in Theorem 1. The user has full flexibility to design these individual modules depending on prior knowledge or requirements. In our experiments, we use a conventional Kronecker structure for the meta-task GPs [19]. To model the correlations of residuals of the target task, we adopt a separable Kronecker form $\mathrm{Cov}(\tilde{f}_{to}, \tilde{f}_{to'}) = k_{\text{Obs}}(\boldsymbol{x}, \boldsymbol{x}')\, K_{\text{Obj}}[o, o']$ in which k_{Obs} is a within-task input kernel and $K_{\text{Obj}} \in \mathbb{R}^{O \times O}$ is an objective-objective coupling matrix. A general K_{Obj} has $\mathcal{O}(O^2)$ free entries. To improve identifiability with only tens of target observations, we restrict K_{Obj} to an *equicorrelated* form

$$K_{\text{Obj}}[i, j] = \begin{cases} \sigma_i^2, & i = j, \\ \rho\, \sigma_i\, \sigma_j, & i \neq j, \end{cases} \qquad \rho \in \left[-\tfrac{1}{O-1},\, 1 \right], \tag{10}$$

in which a single correlation parameter ρ is shared across all pairs of distinct objective residuals of the target task, and each objective retains its own marginal scale σ_i. This admits the rank-one-plus-diagonal decomposition $K_{\text{Obj}} = (1-\rho)\,\mathrm{diag}(\boldsymbol{\sigma}^2) + \rho\, \boldsymbol{\sigma}\boldsymbol{\sigma}^\top$, is PSD on the stated ρ interval, and reduces the free parameters of K_{Obj} from $\mathcal{O}(O^2)$ to $O + 1$. Compared to the generic low-rank form $K_{\text{Obj}} = BB^\top + \mathrm{diag}(\boldsymbol{v})$ with $B \in \mathbb{R}^{O \times O}$, equicorrelation imposes a single shared cross-objective correlation of the target task residuals while remaining identifiable on the data budgets that motivate meta-learning in the first place. We treat ρ as an HP with a Beta prior (Appendix C.1) and fit it jointly with the per-objective scales σ_i during target-task GP training.

4 Experimental evaluation

We study the performance of SMOG relative to a wide range of optimization algorithms on synthetic and real-world benchmarks with 2–4 objectives. No target-task observations are available initially: each run starts from a single configuration drawn uniformly at random, followed by 49 BO iterations, for a total budget of 50 target-task evaluations. We repeat each experiment 50 times with different random seeds unless stated otherwise. All experiments are in a low-data regime: 3–8 meta-tasks with 16 (synthetic, HPOBench) or 64 (Terrain) observation each. Every GP-based method except DGEMO (Appendix F.8) uses LogExpectedHypervolumeImprovement (LogEHVI) [38] as its acquisition function at every iteration following the initial sample; Appendix E.1 compares LogEHVI to qLogParEGO. These methods also share the same acquisition optimizer, so they differ only in the surrogate; for example, Ind. GP with LogEHVI corresponds to the standard MOBO pipeline with independent GPs per objective. MO-TPE uses its own tree-structured Parzen estimator (TPE)-based

acquisition. FCNetTabularBenchmark has only two objectives and was therefore excluded from all 4-objective experiments. Further details on the compute environment, reference point, hyperpriors, and marginal-likelihood optimization are provided in Appendix C.1; mixed-space acquisition function (AF) optimization is described in Appendix C.2.

For each group of runs sharing the same benchmark, number of objectives, and target task, we normalize the observed objective values to $[0, 1]$ per objective using group-wide minimum and maximum values computed from all pooled observations, and set the HV reference point to the origin in this normalized space (corresponding to the worst observed value per objective across all runs). The *HV gap* is $1 - \widehat{\mathrm{HV}}_t$, where $\widehat{\mathrm{HV}}_t \in [0, 1]$ is the normalized HV of the Pareto front built from all solutions found up to BO iteration t; lower values indicate a better Pareto front. The *cumulative HV gap* $\sum_{s=1}^{t}(1 - \widehat{\mathrm{HV}}_s)$ is the cumulative sum of per-iteration HV gaps and penalizes algorithms that discover good fronts late, rewarding fast convergence.

4.1 Benchmarks

We evaluate the performance of SMOG in controlled synthetic and real-world settings. We use 8 meta-tasks and 16 observations per meta-task in the main text, unless otherwise stated.

Adapted Hartmann6 benchmark. We define an adapted variant of the Hartmann6 benchmark to allow for meaningful multi-task, multi-objective optimization. To obtain a multi-task multi-objective setting, we (i) sample a separate coefficient vector $\boldsymbol{\alpha}_m$ per task, yielding related but distinct tasks, and (ii) apply small objective-specific input shifts $\boldsymbol{\varepsilon}_o$ so that different objectives attain their optima at different locations. We additionally study the behavior of SMOG for different numbers of meta tasks and observations per meta task in Appendix E. A complete specification is provided in Appendix C.3.

Adapted Branin-Currin benchmark. We consider an adapted variant of the 2D Branin-Currin multi-objective benchmark [39]. To obtain a multi-task multi-objective setting, we (i) perturb the underlying function parameters independently per task and objective, yielding related but distinct tasks, and (ii) apply small objective-specific input shifts so that objectives attain their optima at different locations. We report objectives in a maximization form (negated values) to match standard HV-based evaluation. A complete specification is provided in Appendix C.3.

Tabular HPO benchmarks. To evaluate SMOG's performance in real-world settings, we investigate it on HPOBench benchmarks [40]. The goal in HPOBench is to jointly optimize a neural network architecture and its HPs. The performance of a configuration can be observed on four different datasets: *Slice Localization*, *Protein Structure*, *Naval Propulsion*, and *Parkinson's Telemonitoring*. The benchmark has two objectives: validation MSE and runtime. We always run for the highest-fidelity setting, corresponding to the maximum number of epochs (100). We use one dataset as the target task, the other three as meta tasks, and 16 observations per meta task.

Terrain benchmark. In this benchmark, we study unmanned aerial vehicle (UAV) trajectory optimization problems [41]. The goal is to find a trajectory of 20 three-dimensional waypoints through one of 56 predefined landscapes that minimizes four target metrics: path-length cost, obstacle-avoidance cost, altitude cost, and smoothness cost. Other landscapes are used as meta tasks. We use the first three landscapes as target tasks. For each target task, we uniformly sample 8 meta tasks from the remaining landscapes and use 64 observations per meta task. See Appendix C.3 for details.

4.2 Models

We compare SMOG against the following baselines: MO-GP, a multi-task GP with Kronecker structure [19]; Ind. GP, a GP model without meta-learning and independent outputs; Ind. ScaML-GP, which uses one ScaML-GP model per objective [16]; MO-TPE, a TPE-based model that naturally handles multiple objectives and meta-learning [33]; Ind. RGPE and Ind. SGPT, meta-learning models using linear combinations of GPs [27, 35]; Ind. ABLR, adaptive Bayesian linear regression [29]; and Ind. GC3P, which uses Gaussian copulas to map observations from different tasks to comparable distributions [30]. Additionally, in Appendix F.8, we compare SMOG to NSGA-III [42] and DGEMO [43] and in Appendix F.7, we compare SMOG to SMOG (w/o cross-obj.), a variant without cross-objective correlations in the target-task model.

Models starting with "Ind." model distinct objectives independently; for these, LogEHVI operates on the product of the per-objective posteriors. We additionally evaluate each model's surrogate quality

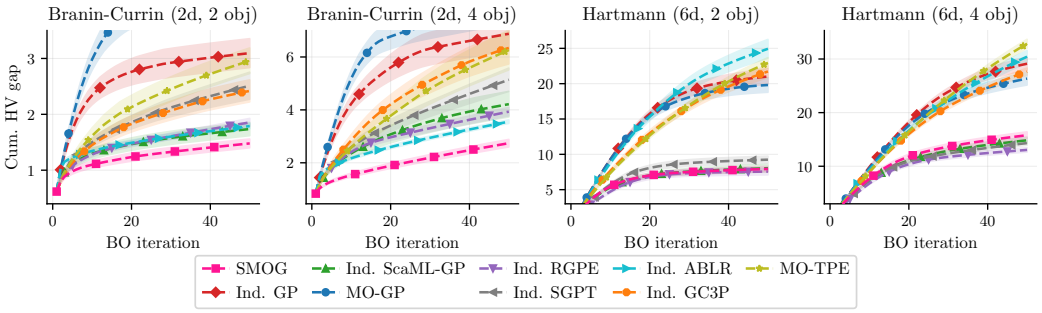


Figure 3: cumulative HV gap (lower is better) on the synthetic benchmarks. Columns correspond to benchmark and objective-count combinations: `Branin-Currin` (2 obj), `Branin-Currin` (4 obj), `Hartmann6` (2 obj), `Hartmann6` (4 obj). All results use 8 meta-tasks and 16 observations per meta-task. The corresponding HV gap is shown in Figure 11 in the Appendix. The first evaluation is the random initial point.

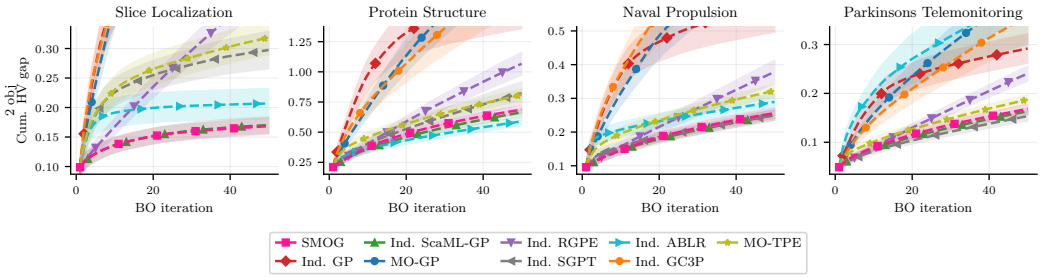


Figure 4: cumulative HV gap (lower is better) on the HPOBench benchmarks. Columns correspond to the four target datasets. All results use 3 meta-tasks and 16 observations per meta-task. The first evaluation is the random initial point.

(negative log predictive density (NLPD) and root mean squared error (RMSE)) in isolation, prior to any BO iterations, in Appendix F.6. Full implementation details are provided in Appendix C.4.

4.3 Results

Synthetic benchmarks. Figure 3 reports cumulative HV gap on `Branin-Currin` and `Hartmann6`, each with 2 and 4 objectives; the corresponding HV gap is in Figure 11. On `Branin-Currin`, the non-meta-learning GP baselines `Ind. GP` and `MO-GP` accumulate clearly more HV gap than the meta-learning methods. On `Hartmann6`, they are also separated from `SMOG`, but other baselines (`Ind. ABLR`, `MO-TPE` and `Ind. GC3P`) perform similarly. `SMOG` attains the lowest or near-lowest cumulative HV gap across all configurations, except for the 4-objective `Hartmann6` benchmark, where `Ind. RGPE` performs best. Table 1 compares `SMOG` with `Ind. ScaML-GP`, which removes all cross-objective modeling (Table 7). On `Branin-Currin`, `SMOG` significantly outperforms `Ind. ScaML-GP` with both 2 and 4 objectives (by approx. 15–35%); Appendix F.7 shows that most of this gain is retained without the target-task cross-objective terms. On `Hartmann6`, `SMOG` and `Ind. ScaML-GP` perform similarly; `Ind. RGPE` matches them with 2 objectives and outperforms them with 4, but is mediocre on `Branin-Currin`. `MO-TPE`, the only other method combining meta-learning with multi-objective optimization, beats the non-meta-learning baselines early but is otherwise not competitive. Appendix F.4 presents an analysis of the `Hartmann6` solutions in more detail; Appendix E.3 studies `SMOG`'s performance when varying the objective correlation strength.

HPOBench benchmarks. Figure 4 shows the performance of `SMOG` and competitors on the HPOBench benchmark. For space reasons, we defer the HV gap results on the individual datasets to Figure 12 in the Appendix. As observed on the synthetic benchmarks, `MO-GP` and `Ind. GP` are not competitive and accumulate substantially more HV gap than the metadata-leveraging methods on all four datasets. `Ind. GC3P` is worse than `Ind. GP` on all four datasets; all other meta-learning methods

Table 1: Final cumulative HV gap (mean $\pm$ SEM over 50 seeds; lower is better) of SMOG and Ind. ScaML-GP (one ScaML-GP model per objective). Both learn the target-task signal variance under matched priors (Appendix C.1). Δ is the relative reduction of mean HV gap achieved by SMOG (positive: SMOG better). Bold with *: significant at level 0.05 under a two-sided Wilcoxon signed-rank test paired by seed, with Holm correction over the 14 settings. Normalization as in Figure 3.

		Cumulative HV gap		
Problem	O	SMOG	Ind. ScaML-GP	Δ (%)
Branin-Currin	2	1.48 ± 0.08	1.74 ± 0.12	$\mathbf{+14.8}^*$
	4	2.74 ± 0.14	4.22 ± 0.49	$\mathbf{+35.0}^*$
Hartmann6	2	7.93 ± 0.37	8.04 ± 0.53	$+1.4$
	4	15.8 ± 0.6	14.8 ± 0.6	-6.4
HPOBench				
Slice	2	0.168 ± 0.013	0.170 ± 0.014	$+1.1$
Protein	2	0.687 ± 0.032	0.665 ± 0.031	-3.3
Naval	2	0.256 ± 0.015	0.250 ± 0.014	-2.4
Parkinsons	2	0.168 ± 0.009	0.164 ± 0.007	-2.9
Terrain				
task 0	2	10.1 ± 0.3	10.6 ± 0.3	$+4.5$
task 1	2	10.5 ± 0.3	10.9 ± 0.3	$+3.8$
task 2	2	9.11 ± 0.48	10.9 ± 0.6	$\mathbf{+16.5}^*$
task 0	4	20.6 ± 0.3	21.3 ± 0.4	$+3.3$
task 1	4	20.7 ± 0.4	21.0 ± 0.6	$+1.7$
task 2	4	20.1 ± 0.6	21.5 ± 0.5	$+6.5$
Significantly better / worse				3 / 0

outperform Ind. GP and MO-GP by a wide margin. SMOG and Ind. ScaML-GP perform comparably on all four problems (Table 1) and have the lowest HV gap on Slice Localization. SMOG is within 3% of the best method on Naval Propulsion; on Protein Structure and Parkinson's Telemonitoring, it is outperformed by Ind. ABLR (0.59 vs. 0.69) and Ind. SGPT (0.154 vs. 0.168), respectively. However, these methods struggle on other problems (see Figures 3 and 5). The Pareto fronts of the solutions found across all methods and datasets are shown in Figure 15 in the Appendix.

Terrain benchmark. Next, we study SMOG's performance on the Terrain benchmark. Figure 5 shows the cumulative HV gap for all three target tasks and both objective configurations (2- and 4-objective). Figure 13 in the Appendix shows the corresponding HV gap. The Terrain benchmark paints a similar picture to HPOBench, with stronger baselines: in five of the six settings, Ind. ABLR, Ind. SGPT or Ind. RGPE reach a lower mean HV gap than SMOG, but each of them is clearly worse in other settings (notably, Ind. ABLR with 4 objectives and Ind. RGPE and Ind. SGPT on task 1), while SMOG performs well overall. SMOG significantly outperforms Ind. ScaML-GP on task 2 (2 objectives).

5 Conclusion

Many impactful applications require optimizing competing objectives: in aerospace engineering, one seeks a lightweight structure with high strength, while in machine learning, one aims for accurate yet small models. In many cases, practitioners can use data from related tasks or experiments to quickly find a good solution for a new task. In this paper, we introduce SMOG—a scalable meta-learning algorithm for multi-objective black-box optimization problems. SMOG leverages observations from related tasks and models cross-objective correlations to construct an informative target-task prior, improving sample efficiency in the initial BO iterations. SMOG is principled, with a clear theoretical motivation, and performs robustly when studied in practice. As such, SMOG fills a gap by providing a practical and principled algorithm that combines meta-learning and multi-objective optimization.

Limitations. SMOG's limitations are dictated by its assumptions. Assumption 1 achieves scalability but ignores correlations across meta-tasks; Assumption 2 restricts the number of HPs but also limits the number of meta-learning channels; and Assumption 3 achieves a practical implementation but blocks the feedback from the target data to meta-task HPs. In addition, SMOG assumes that all tasks share the same search and objective spaces. While well-motivated in the regime of scarce and

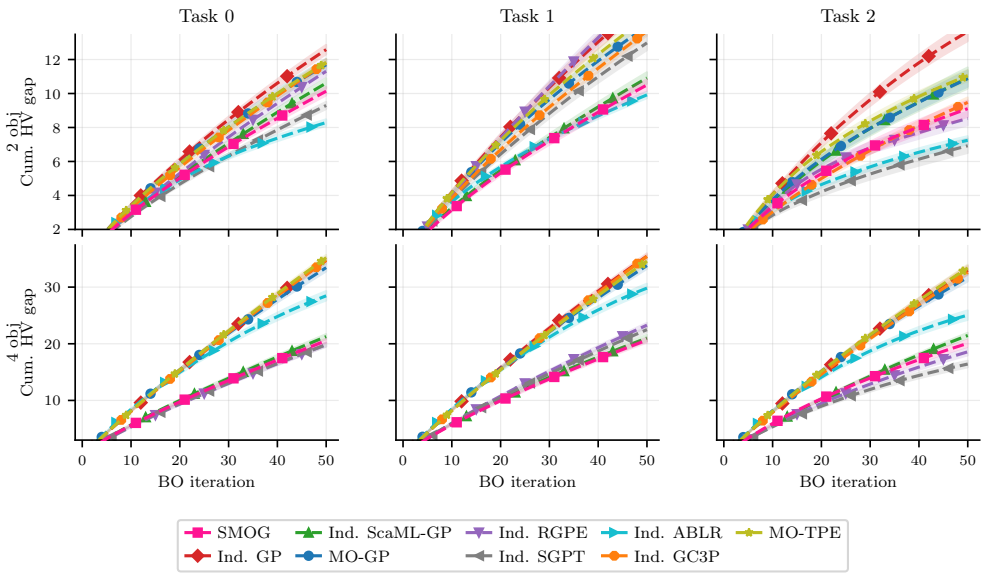


Figure 5: cumulative HV gap (lower is better) on the Terrain benchmark. Rows correspond to 2-objective (top) and 4-objective (bottom) variants; columns correspond to target tasks 0, 1, and 2. All results use 8 meta-tasks and 64 observations per meta-task. The first evaluation is the random initial point.

noisy data, more flexible neural-network-based models may be able to meta-learn more complex relationships when ample data is available. In addition, `SMOG` scales to a large number of tasks, but each task model remains an exact GP and scales cubically with the number of data points, so large datasets would require sparse or approximate GP extensions. While `SMOG` is mathematically defined for an arbitrary number of objectives, our experiments cover only 2–4 objectives. For larger numbers of objectives, exact GP inference and the optimization of HV-based AFs become more challenging. Beyond the two AFs studied here, combining `SMOG`'s correlated posterior with non-myopic [44] or information-theoretic [45, 46] strategies is a promising direction for future work.

Broader impact. This paper presents foundational research on sample-efficient optimization. On the positive side, more data-efficient optimization reduces the number of expensive experiments needed in engineering and science, thereby saving time, energy, and resources. On the negative side, we are not aware of direct pathways from this work to harmful applications.

Acknowledgments and Disclosure of Funding

Calculations (or parts of them) for this publication were performed on the HPC cluster PALMA II of the University of Münster, subsidized by the DFG (INST 211/667-1). The authors gratefully acknowledge the computing time granted by the Resource Allocation Board and provided on the supercomputer Emmy/Grete at NHR-Nord@Göttingen as part of the NHR infrastructure. The calculations for this research were conducted with computing resources under the project nhr_nw_test. The authors gratefully acknowledge the computing time provided to them at the NHR Center NHR4CES at RWTH Aachen University (project number p0026398). This is funded by the Federal Ministry of Education and Research, and the state governments participating on the basis of the resolutions of the GWK for national high performance computing at universities (`www.nhr-verein.de/unsere-partner`).

References

[1] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[2] 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.

[3] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2016.

[4] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.

[5] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, 34:2187–2200, 2021.

[6] Abhijith M Gopakumar, Prasanna V Balachandran, Dezhen Xue, James E Gubernatis, and Turab Lookman. Multi-objective optimization for materials discovery via adaptive design. *Scientific reports*, 8(1):3738, 2018.

[7] Jay I Myung, James R Deneault, Jorge Chang, Inhan Kang, Benji Maruyama, and Mark A Pitt. Multi-objective bayesian optimization: a case study in material extrusion. *Digital Discovery*, 4 (2):464–476, 2025.

[8] Florian Pfisterer, Lennart Schneider, Julia Moosbauer, Martin Binder, and Bernd Bischl. Yahpo gym-an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization. In *International Conference on Automated Machine Learning*, pages 3–1. PMLR, 2022.

[9] Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, Rene Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[10] Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1557–1563. IEEE, 2017.

[11] Henry C Herbol, Weici Hu, Peter Frazier, Paulette Clancy, and Matthias Poloczek. Efficient search of compositional space for hybrid organic–inorganic perovskites via bayesian optimization. *npj Computational Materials*, 4(1):51, 2018.

[12] Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation*, 10 (1):50–66, 2006.

[13] Zhongxiang Dai, Yizhou Chen, Haibin Yu, Bryan Kian Hsiang Low, and Patrick Jaillet. On provably robust meta-bayesian optimization. In *Uncertainty in Artificial Intelligence*, pages 475–485. PMLR, 2022.

[14] Michael Volpp, Lukas P. Fröhlich, Kirsten Fischer, Andreas Doerr, Stefan Falkner, Frank Hutter, and Christian Daniel. Meta-Learning Acquisition Functions for Transfer Learning in Bayesian Optimization. In *International Conference on Learning Representations*, 2020.

[15] Matthias Feurer, Benjamin Letham, Frank Hutter, and Eytan Bakshy. Practical transfer learning for bayesian optimization. *arXiv preprint arXiv:1802.02219v3*, 2022.

[16] Petru Tighineanu, Lukas Grossberger, Paul Baireuther, Kathrin Skubch, Stefan Falkner, Julia Vinogradska, and Felix Berkenkamp. Scalable meta-learning with gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1981–1989. PMLR, 2024.

[17] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, January 2006.

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

[19] Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20, 2007.

[20] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

[21] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-Task Bayesian Optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[22] Dani Yogatama and Gideon Mann. Efficient Transfer Learning Method for Automatic Hyperparameter Tuning. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 1077–1085, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

[23] Tinu Theckel Joy, Santu Rana, Sunil Kumar Gupta, and Svetha Venkatesh. Flexible Transfer Learning Framework for Bayesian Optimisation. In James Bailey, Latifur Khan, Takashi Washio, Gill Dobbie, Joshua Zhexue Huang, and Ruili Wang, editors, *Advances in Knowledge Discovery and Data Mining*. Springer International Publishing, 2016.

[24] Matthias Poloczek, Jialei Wang, and Peter Frazier. Multi-information source optimization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[25] Alistair Shilton, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Regret Bounds for Transfer Learning in Bayesian Optimisation. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 307–315, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[26] Petru Tighineanu, Kathrin Skubch, Paul Baireuther, Attila Reiss, Felix Berkenkamp, and Julia Vinogradska. Transfer learning with gaussian processes for bayesian optimization. In *International conference on artificial intelligence and statistics*, pages 6152–6181. PMLR, 2022.

[27] Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Scalable Gaussian process-based transfer surrogates for hyperparameter optimization. *Machine Learning*, 107(1):43–78, 2018.

[28] Matthias Poloczek, Jialei Wang, and Peter I Frazier. Warm starting bayesian optimization. In *2016 Winter simulation conference (WSC)*, pages 770–781. IEEE, 2016.

[29] Valerio Perrone, Rodolphe Jenatton, Matthias W Seeger, and Cedric Archambeau. Scalable hyperparameter transfer learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[30] David Salinas, Huibin Shen, and Valerio Perrone. A quantile-based approach for hyperparameter transfer learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8438–8448. PMLR, 13–18 Jul 2020.

[31] Martin Wistuba and Josif Grabocka. Few-shot bayesian optimization with deep kernel surrogates. In *International Conference on Learning Representations*, 2021.

[32] Zi Wang, George E. Dahl, Kevin Swersky, Chansoo Lee, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained Gaussian processes for Bayesian optimization. *arXiv preprint arXiv:2109.08215*, 2023.

[33] Shuhei Watanabe, Noor H Awad, Masaki Onishi, and Frank Hutter. Speeding up multi-objective hyperparameter optimization by task similarity-based meta-learning for the tree-structured parzen estimator. In *IJCAI*, 2023.

[34] Xunzhao Yu. Fseo: Few-shot evolutionary optimization via meta-learning for expensive multi-objective optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[35] Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable meta-learning for bayesian optimization using ranking-weighted gaussian process ensembles. In *AutoML Workshop at ICML*, volume 7, page 5, 2018.

[36] M. J. Bayarri, J. O. Berger, and F. Liu. Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119 – 150, 2009. doi: 10.1214/09-BA404.

[37] Kevin M Murphy and Robert H Topel. Estimation and inference in two-step econometric models. *Journal of Business & Economic Statistics*, 20(1):88–97, 2002.

[38] Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20577–20612, 2023.

[39] Faran Irshad, Stefan Karsch, and Andreas Döpp. Leveraging trust for joint multi-objective and multi-fidelity optimization. *Machine Learning: Science and Technology*, 5(1):015056, 2024.

[40] Aaron Klein and Frank Hutter. Tabular benchmarks for joint architecture and hyperparameter optimization. *arXiv preprint arXiv:1905.04970*, 2019.

[41] Mhd Ali Shehadeh and Jakub Kůdela. Benchmarking global optimization techniques for unmanned aerial vehicle path planning. *Expert Systems with Applications*, page 128645, 2025.

[42] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

[43] Mina Konakovic Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-guided multi-objective bayesian optimization with batch evaluations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17708–17720. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/cd3109c63bf4323e6b987a5923becb96-Paper.pdf`.

[44] Sam Daulton, Maximilian Balandat, and Eytan Bakshy. Hypervolume knowledge gradient: A lookahead approach for multi-objective bayesian optimization with partial information. In *International Conference on Machine Learning*, pages 7167–7204. PMLR, 2023.

[45] Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective bayesian optimization. In *International conference on machine learning*, pages 1492–1501. PMLR, 2016.

[46] Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 35: 9922–9938, 2022.

[47] Hisao Ishibuchi, Naoya Akedo, and Yusuke Nojima. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 649–656, 2011.

[48] Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *International Conference on Machine Learning*, pages 10663–10674. PMLR, 2021.

[49] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: Reliable high-dimensional bayesian optimization for combinatorial and mixed spaces. *Advances in Neural Information Processing Systems*, 36:1764–1793, 2023.

[50] Zeyuan Ma, Yue-Jiao Gong, Hongshu Guo, Wenjie Qiu, Sijie Ma, Hongqiao Lian, Jiajun Zhan, Kaixu Chen, Chen Wang, Zhiyang Huang, et al. Metabox-v2: A unified benchmark platform for meta-black-box optimization. *arXiv preprint arXiv:2505.17745*, 2025.

[51] Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.

[52] Julian Blank and Kalyanmoy Deb. pymoo: Multi-objective optimization in Python. *IEEE Access*, 8:89497–89509, 2020.

A Kernel properties

Here we prove the key properties of SMOG's kernel encoded in Lemmas 1 and 2.

A.1 Coregionalization matrices

Here we prove Lemma 1, which we restate below.

Lemma 1. *Applying Assumptions 1 and 2 to Equation* (2) *yields a sparse structure with the following non-zero entries of the coregionalization matrices*

$$\begin{aligned} [c_{t\theta}]_{to,to'} &= [h_{t\theta}]_{oo'}, & [c_{m\theta}]_{mo,mo'} &= [h_{m\theta}]_{oo'}, \\ [c_{m\theta}]_{mo,to'} &= w_{mo'}[h_{m\theta}]_{oo'}, & [c_{m\theta}]_{to,to'} &= w_{mo}w_{mo'}[h_{m\theta}]_{oo'}, \end{aligned} \tag{3}$$

where $[c_{\cdot\theta}]_{a,b} = [c_{\cdot\theta}]_{b,a}$ *and* $[h_{v\theta}]_{oo'}$ *are positive semi-definite (PSD) in the basis* (o, o').

Proof. We begin by showing that the coregionalization matrices in Equation (3) are uniquely defined by Assumptions 1 and 2. We collect all terms in Equation (2):

$$k[(\mathbf{x}, m, o), (\mathbf{x}', m, o')] = \text{Cov}\left[f_{mo}(\mathbf{x}), f_{mo'}(\mathbf{x}')\right] = k_m[(\mathbf{x}, o), (\mathbf{x}', o')] \quad \text{(Assumption 1)} \tag{11}$$

$$k[(\mathbf{x}, m, o), (\mathbf{x}', m' \neq m, o')] = 0 \quad \text{(Assumption 1)} \tag{12}$$

$$\begin{aligned} k[(\mathbf{x}, t, o), (\mathbf{x}', m, o')] &= \text{Cov}\left[f_{to}(\mathbf{x}), f_{mo'}(\mathbf{x}')\right] \\ &= \text{Cov}\left[\tilde{f}_{to} + \sum_{m' \in \mathcal{M}} \tilde{f}_{m'o}, f_{mo'}\right] \quad \text{(Assumption 2)} \\ &= \sum_{m' \in \mathcal{M}} w_{m'o}\text{Cov}\left[f_{m'o}, f_{mo'}\right] \quad \text{(Assumption 2)} \\ &= w_{mo}k_m[(\mathbf{x}, o), (\mathbf{x}', o')] \quad \text{(Assumption 1),} \end{aligned} \tag{13, 14}$$

where we have made use of the fact that the covariance is a bilinear function, and that the perfect correlation $\text{Corr}\left[f_{mo}(\mathbf{x}), \tilde{f}_{mo}(\mathbf{x}')\right] = \pm 1$ in Assumption 2 implies that $\tilde{f}_{mo} = w_{mo}f_{mo} + c$ with $w_{mo}, c \in \mathbb{R}$. Finally, we have

$$\begin{aligned} &k[(\mathbf{x}, t, o), (\mathbf{x}', t, o')] = \text{Cov}\left[f_{to}(\mathbf{x}), f_{to'}(\mathbf{x}')\right] \\ &= \text{Cov}\left[\tilde{f}_{to}(\mathbf{x}), \tilde{f}_{to'}(\mathbf{x}')\right] + \sum_{m,m' \in \mathcal{M}} w_{mo}w_{m'o'}\text{Cov}\left[f_{mo}(\mathbf{x}), f_{m'o'}(\mathbf{x}')\right] \quad \text{(Assumption 2)} \\ &= k_t[(\mathbf{x}, o), (\mathbf{x}', o')] + \sum_{m \in \mathcal{M}} w_{mo}w_{mo'}k_m[(\mathbf{x}, o), (\mathbf{x}', o')] \quad \text{(Assumptions 1 and 2)} \end{aligned} \tag{15}$$

By collecting the coefficients corresponding to k_m and k_t, we obtain the coregionalization matrices of SMOG in Equations (3) and (17).

These coregionalization matrices are elements of the original multi-task Gaussian process (MTGP), implying that they are PSD by definition. To see this, consider the expression for the meta-task block of the MTGP kernel in Equation (2) (excluding the target task and all meta-task-to-target-task couplings):

$$k\left[(\boldsymbol{x}, m, o), (\boldsymbol{x'}, m', o')\right] = \delta_{m=m'} \sum_{\theta \in \mathcal{O}} \left[h_{m\theta}\right]_{oo'} k_{m\theta}(\boldsymbol{x}, \boldsymbol{x'}) \equiv \delta_{m=m'} k_m\left[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')\right], \tag{16}$$

where $[h_{m\theta}]_{oo'} = [c_{m\theta}]_{mo,mo'}$. This formulation does not make any additional assumptions about the kernel across objectives, $k_m\left[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')\right] = \sum_{\theta \in \mathcal{O}} \left[h_{m\theta}\right]_{oo'} k_{m\theta}(\boldsymbol{x}, \boldsymbol{x'})$. The elements $[h_{m\theta}]_{oo'}$ appear directly in Equation (3) and are by definition PSD. Similar reasoning can be applied to the other elements, implying that all entries of SMOG's coregionalization matrices in Equation (3) are PSD. □

Example for two meta-tasks and objectives. We have the following coregionalization matrices in the basis $(m=1,o=1),(m=1,o=2),(m=2,o=1),(m=2,o=2),(t,o=1),(t,o=2)$:

$$\boldsymbol{C}_{1\theta} = \begin{pmatrix} \boldsymbol{H}_{1\theta} & \boldsymbol{0}_{2\times2} & W_1 \odot \boldsymbol{H}_{1\theta} \\ \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} \\ W_1^\top \odot \boldsymbol{H}_{1\theta} & \boldsymbol{0}_{2\times2} & W_1^\times \odot \boldsymbol{H}_{1\theta} \end{pmatrix}, \qquad \boldsymbol{C}_{2\theta} = \begin{pmatrix} \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} \\ \boldsymbol{0}_{2\times2} & \boldsymbol{H}_{2\theta} & W_2 \odot \boldsymbol{H}_{2\theta} \\ \boldsymbol{0}_{2\times2} & W_2^\top \odot \boldsymbol{H}_{2\theta} & W_2^\times \odot \boldsymbol{H}_{2\theta} \end{pmatrix},$$
$$\boldsymbol{C}_{t\theta} = \begin{pmatrix} \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} \\ \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} \\ \boldsymbol{0}_{2\times2} & \boldsymbol{0}_{2\times2} & \boldsymbol{H}_{t\theta} \end{pmatrix}, \qquad W_i = \begin{pmatrix} w_{i1} & w_{i2} \\ w_{i1} & w_{i2} \end{pmatrix}, \qquad W_i^\times = \begin{pmatrix} w_{i1}^2 & w_{i1}w_{i2} \\ w_{i1}w_{i2} & w_{i2}^2 \end{pmatrix}, \tag{17}$$

where $\odot$ is the elementwise (Hadamard) product. Here, $\boldsymbol{H}_{v\theta}$ is matrix notation for $[h_{v\theta}]_{oo'}$ and describes cross-objective correlations.

A.2 SMOG kernel

Next, we prove Lemma 2, which we restate below.

Lemma 2. *Assumptions 1 and 2 with $w_{mo} \in \mathbb{R}$ for $m \in \mathcal{M}$ and $o \in \mathcal{O}$ yield a valid kernel given by*

$$k_{\texttt{SMOG}}\left[(\boldsymbol{x},\nu,o),(\boldsymbol{x'},\nu',o')\right] = \sum_{v\in\mathcal{M}^*} g_v(\nu,o)g_v(\nu',o')k_v\left[(\boldsymbol{x},o),(\boldsymbol{x'},o')\right], \tag{4}$$

where $g_v(\nu,o)$ is one if $v=\nu$, w_{mo} if $(v=m)\wedge(\nu=t)$, and zero otherwise.

Proof. We start off by using the results of Lemma 1, Equation (17). The first notational simplification for the coregionalization matrices can be done by pulling the common $\boldsymbol{H}_{v\theta}$ term outside the big matrix. To do this, we introduce a custom matrix product $\boxtimes$ between matrices $\boldsymbol{A}$ of size $(\alpha+\beta)\times(\alpha+\beta)$ and $\boldsymbol{B}$ of size $\beta\times\beta$. Each $\beta\times\beta$ block of matrix $\boldsymbol{A}$ is multiplied with matrix $\boldsymbol{B}$ entrywise:

$$(\boldsymbol{A}\boxtimes\boldsymbol{B})_{\beta\times\beta} = \boldsymbol{A}_{\beta\times\beta}\odot\boldsymbol{B}, \tag{18}$$

where $\odot$ denotes the Hadamard product. Applying this to Equation (17) yields coregionalization matrices that are described by a $\boxtimes$-product between a $(M+1)O\times(M+1)O$ matrix containing only weights, $\boldsymbol{W}_v$, and a $O\times O$ matrix $\boldsymbol{H}_{v\theta}$ describing task-specific correlations across its objectives:

$$\boldsymbol{C}_{v\theta} = \boldsymbol{W}_v \boxtimes \boldsymbol{H}_{v\theta} \tag{19}$$

Careful inspection of $\boldsymbol{W}_v$ reveals that it can be written as an outer product, $\boldsymbol{w}_v\boldsymbol{w}_v^\top$, where $\boldsymbol{w}_v = (\boldsymbol{w}_m^\top, \boldsymbol{w}_t^\top)^\top$ and

$$\begin{aligned} \boldsymbol{w}_m &= (\ldots, \boldsymbol{0}_{1\times O}, \underbrace{\boldsymbol{1}_{1\times O}}_{\text{m-th entry}}, \boldsymbol{0}_{1\times O}, \ldots, \boldsymbol{0}_{1\times O}, w_{m1}, w_{m2}, \ldots, w_{mO})^\top \\ \boldsymbol{w}_t &= (\ldots, \boldsymbol{0}_{1\times O}, \boldsymbol{1}_{1\times O})^\top, \end{aligned} \tag{20}$$

implying that $\boldsymbol{W}_v$ is PSD for all $w_{mo}\in\mathbb{R}$. Inserting this into Equation (2) yields the final expression for our kernel, which we write in matrix form of size $(M+1)O\times(M+1)O$ for convenience:

$$\begin{aligned} \boldsymbol{K}_{\texttt{SMOG}}(\boldsymbol{x},\boldsymbol{x'}) &= \sum_{v\in\mathcal{M}\cup\{t\}} \boldsymbol{w}_v\boldsymbol{w}_v^\top \boxtimes \boldsymbol{K}_v(\boldsymbol{x},\boldsymbol{x'}), \\ \boldsymbol{K}_v(\boldsymbol{x},\boldsymbol{x'}) &= \sum_{\theta\in\mathcal{O}} \boldsymbol{H}_{v\theta}k_{v\theta}(\boldsymbol{x},\boldsymbol{x'}), \end{aligned} \tag{21}$$

where $\boldsymbol{K}_v(\boldsymbol{x},\boldsymbol{x'})$ is a kernel of size $O\times O$ that acts on the objectives of task v. Since $\boldsymbol{K}_v(\boldsymbol{x},\boldsymbol{x'})$ and $\boldsymbol{W}_v$ are both PSD, the kernel in Equation (21) is also PSD and is therefore a valid kernel. Expressing the kernel in Equation (21) in index notation concludes the proof of Lemma 2. □

B SMOG target-task prior

In the following we prove Theorem 1, which we restate below.

Theorem 1. *Under a zero-mean Gaussian process (GP) prior with the multi-task kernel given by Equation* (4)*, the distribution of the target-task objectives conditioned on the metadata is*

$$f_{to} \mid \mathcal{D}_{1:M}, \boldsymbol{x} \sim \mathcal{GP}(m_{t,\texttt{SMOG}}(\boldsymbol{x}, o), k_{t,\texttt{SMOG}}[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')]) \tag{5}$$

with

$$\begin{aligned} m_{t,\texttt{SMOG}}(\boldsymbol{x}, o) &= \sum_{m \in \mathcal{M}} w_{mo} \hat{m}_{mo}(\boldsymbol{x}), \\ k_{t,\texttt{SMOG}}[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')] &= k_t[(\boldsymbol{x}, o), (\boldsymbol{x'}, o')] + \sum_{m \in \mathcal{M}} w_{mo} w_{mo'} \hat{k}_{moo'}(\boldsymbol{x}, \boldsymbol{x'}), \end{aligned} \tag{6}$$

where $\hat{m}_{mo}(\boldsymbol{x})$ and $\hat{k}_{moo'}(\boldsymbol{x}, \boldsymbol{x'})$ are the posterior mean and covariance functions of the individual multi-output meta-task GPs conditioned only on their corresponding data, $\mathcal{D}_m$.

Proof. According to Equation (4), the joint prior model for the meta-observations $\boldsymbol{Y}_{\text{meta}} = \left(\boldsymbol{Y}_1^\top, \dots, \boldsymbol{Y}_M^\top\right)^\top$ and the target-task function value at a query point $(\mathbf{x}, o)$ given by

$$\begin{bmatrix} \boldsymbol{Y}_{\text{meta}} \\ f_{to}(\mathbf{x}) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{K}_{\text{meta}} & \boldsymbol{k}_{\text{meta},o} \\ \boldsymbol{k}_{\text{meta},o}^{\mathrm{T}} & k_{t,oo} \end{bmatrix}\right) \tag{22}$$

where $\boldsymbol{K}_{\text{meta}}$ is of size $N_{\text{meta}} \times N_{\text{meta}}$, $\boldsymbol{k}_{\text{meta},o}$ of size $N_{\text{meta}} \times 1$, and $k_{t,oo}$ a scalar, and are given by

$$\begin{aligned} \boldsymbol{K}_{\text{meta}} &= \operatorname{diag}\left(k_1[\mathbf{X}_1, \mathbf{X}_1] + \operatorname{diag}(\sigma_{11}^2, \dots, \sigma_{1O}^2) \otimes \boldsymbol{I}_{N_1}, \dots\right), \\ \boldsymbol{k}_{\text{meta},o} &= \left(w_{1o} k_1[\mathbf{X}_1, (\mathbf{x}, o)], \dots, w_{Mo} k_M[\mathbf{X}_M, (\mathbf{x}, o)]\right), \\ k_{t,oo} &= \left(k_t[(\mathbf{x}, o), (\mathbf{x}, o')] + \sum_{m \in \mathcal{M}} w_{mo} w_{mo'} k_m[(\mathbf{x}, o), (\mathbf{x}, o')]\right) \delta_{oo'} \equiv k_{t,oo'} \delta_{oo'}, \end{aligned}$$

and $N_{\text{meta}} = O \sum_{m=1}^{M} N_m$. Conditioning on the metadata $\mathcal{D}_{1:M}$ yields

$$p(f_{to} \mid \mathcal{D}_{1:M}, \mathbf{x}) = \mathcal{N}\left(m_{t,\texttt{SMOG}}(\mathbf{x}, o), k_{t,\texttt{SMOG}}[(\mathbf{x}, o), (\mathbf{x}, o')]\right), \tag{23}$$

where the target-task prior mean $m_{t,\texttt{SMOG}}$ and covariance $k_{t,\texttt{SMOG}}$ are given by the standard Gaussian conditioning rules

$$\begin{aligned} m_{t,\texttt{SMOG}}(\mathbf{x}, o) &= \boldsymbol{k}_{\text{meta},o}^{\mathrm{T}} \boldsymbol{K}_{\text{meta}}^{-1} \boldsymbol{Y}_{\text{meta}}, \\ k_{t,\texttt{SMOG}}[(\mathbf{x}, o), (\mathbf{x}, o')] &= \mathrm{k}_{t,oo'} - \boldsymbol{k}_{\text{meta},o}^{\mathrm{T}} \boldsymbol{K}_{\text{meta}}^{-1} \boldsymbol{k}_{\text{meta},o'}. \end{aligned}$$

Now since $\boldsymbol{K}_{\text{meta}}$ is block-diagonal in m, we have

$$\boldsymbol{K}_{\text{meta}}^{-1} = \operatorname{diag}\left((k_1[\mathbf{X}_1, \mathbf{X}_1] + \operatorname{diag}(\sigma_{11}^2, \dots, \sigma_{1O}^2) \otimes \boldsymbol{I}_{N_1})^{-1}, \dots\right),$$

so that

$$\begin{aligned} m_{t,\texttt{SMOG}}(\mathbf{x}, o) &= \sum_m w_{mo} k_m[(\mathbf{x}, o), \mathbf{X}_m] \left(k_m[\mathbf{X}_m, \mathbf{X}_m] + \operatorname{diag}(\sigma_{m1}^2, \dots, \sigma_{mO}^2) \otimes \boldsymbol{I}_{N_m}\right)^{-1} \boldsymbol{Y}_m \\ &= \sum_m w_{mo} \hat{m}_{mo}(\mathbf{x}), \end{aligned}$$

where $\hat{m}_m(\mathbf{x}, o)$ is the per meta-task posterior mean after conditioning on the corresponding data $\mathcal{D}_m$. Similarly, for the covariance, we have

$$\begin{aligned} &k_{t,\texttt{SMOG}}[(\mathbf{x}, o), (\mathbf{x}', o')] = k_{t,oo'} - \boldsymbol{k}_{\text{meta},o}^{\mathrm{T}} \boldsymbol{K}_{\text{meta}}^{-1} \boldsymbol{k}_{\text{meta},o'}, \\ &= k_t[(\mathbf{x}, o), (\mathbf{x}', o')] + \sum_m w_{mo} w_{mo'} k_m[(\mathbf{x}, o), (\mathbf{x}', o')] - \sum_m w_{mo} k_m[(\mathbf{x}, o), \mathbf{X}_m] \\ &\quad \times \left(k_m[\mathbf{X}_m, \mathbf{X}_m] + \operatorname{diag}(\sigma_{m1}^2, \dots, \sigma_{mO}^2) \otimes \boldsymbol{I}_{N_m}\right)^{-1} w_{mo'} k_m[\mathbf{X}_m, (\mathbf{x}', o')], \\ &= k_t[(\mathbf{x}, o), (\mathbf{x}', o')] + \sum_m w_{mo} w_{mo'} \left(k_m[(\mathbf{x}, o), (\mathbf{x}', o')] - k_m[(\mathbf{x}, o), \mathbf{X}_m]\right. \\ &\quad \times \left(k_m[\mathbf{X}_m, \mathbf{X}_m] + \operatorname{diag}(\sigma_{m1}^2, \dots, \sigma_{mO}^2) \otimes \boldsymbol{I}_{N_m}\right)^{-1} k_m[\mathbf{X}_m, (\mathbf{x}', o')]\left.\right), \\ &= k_t[(\mathbf{x}, o), (\mathbf{x}', o')] + \sum_m w_{mo} w_{mo'} \hat{k}_m[(\mathbf{x}, o), (\mathbf{x}', o')], \end{aligned}$$

where $\hat{k}_m$ is the corresponding per meta-task posterior covariance. □

C Implementation details

C.1 Experimental setup details

Compute environment. All experiments are launched via dedicated `SLURM` array scripts for both synthetic and real-world benchmarks and are executed within the same Apptainer-containerized software environment. The launch scripts submit jobs to a shared partition, using the array index as the run seed. By default, jobs run with a 6-hour wall-clock limit (16h for the Terrain benchmark); `SMOG` is allocated 8 CPU cores (≈16 GB RAM) and baseline methods are allocated 4 CPU cores (≈8 GB RAM). Experiments with 4 objectives required more memory than the default allocation: all baselines were raised to 8 CPU cores (≈16 GB RAM); additionally, the wall-clock limit for 4-objective real-world (terrain) jobs was extended to 16 hours to accommodate longer evaluation times.

Metrics. For each setting (benchmark, number of objectives, target task), we min–max normalize each objective using the minimum and maximum of the noiseless objective values over all runs of all methods, and compute the hypervolume (HV) $\widehat{\mathrm{HV}}_t \in [0,1]$ of the Pareto front of the first t evaluations in this normalized space, with the origin as reference point. We report the HV gap $1-\widehat{\mathrm{HV}}_t$ and the cumulative HV gap $\sum_{t=1}^{50}\big(1-\widehat{\mathrm{HV}}_t\big)$. Curves show the mean $\pm$ one standard error of the mean (SEM) over seeds; runs that ended early carry their last HV gap forward. Significance is assessed with paired two-sided Wilcoxon signed-rank tests, Holm-corrected across settings.

Reference point of the acquisition function. The reference point used inside the HV-based acquisition function (AF) is distinct from the one used for evaluation (the origin of the normalized objective space; see above). At every iteration, we standardize the target-task observations per objective to zero mean and unit variance, using their current mean and standard deviation, and optimize the AF in this standardized space. We then set the reference point with BoTorch's `infer_reference_point`[2] applied to all standardized observations: starting from the per-objective worst observed value, it moves the reference point back by a fixed fraction of the observed range in each objective (BoTorch's default of 0.1, following Ishibuchi et al. [47]).

Hyperpriors and hyperparameter optimization. For `SMOG`'s equicorrelated residual target kernel, we place a scaled $\mathrm{Beta}(2,2)$ hyperprior on the correlation parameter ρ over the full admissible interval $\left[-\frac{1}{O-1},1\right]$, matching the support induced by the equicorrelation constraint while still favoring interior correlation values. On the per-objective signal standard deviations σ_o of this kernel, we place a $\mathrm{LogNormal}(-1,0.5)$ hyperprior, initialized at its median e^{-1}. `SMOG (w/o cross-obj.)` (Appendix F.7) uses the same prior. For `Ind. ScaML-GP`, which learns a per-objective outputscale of the target GP instead, we use the implied prior on the variance, $\sigma_o^2 \sim \mathrm{LogNormal}(-2,1)$, likewise initialized at its median. `Ind. GP` keeps the signal variance fixed at 1 on standardized outputs, and `MO-GP` places a $\mathrm{LogNormal}(-0.5,0.5)$ prior on the task standard deviations of its multi-task kernel. The meta-task weights are constrained to be positive and receive a $\mathrm{Gamma}(1,1)$ hyperprior. For every hyperparameter (HP) optimization in Equations (8) and (9), we use SciPy's L-BFGS-B without random restarts, initializing ρ at its prior mean and all remaining HPs at their default values.

C.2 Mixed space acquisition function optimization

We employ an interleaving scheme to optimize the AF, similar to that used by Wan et al. [48] or Papenmeier et al. [49]. In particular, we optimize over the discrete variables by selecting the point with the maximum acquisition value from 2^{14} candidates uniformly sampled from the set of all possible discrete combinations. We then fix the discrete variables and optimize over the remaining variables using gradient-based AF maximization with two random restarts and 512 initial samples to select the two starting points. We repeat this interleaving scheme 5 times, starting with the discrete optimization and a uniformly random initialization of the continuous variables.

For categorical variables, we use one-hot encoding, representing n categories as n distinct problem dimensions. We exclude invalid configurations (e.g., two categories being set to "1") from the set of candidates of the AF maximizer.

[2] `https://botorch.readthedocs.io/en/v0.15.1/_modules/botorch/utils/multi_objective/hypervolume.html`, accessed on 01/26/2026

C.3 Benchmark details

Sinusoidal benchmark. As a simple sanity check, we evaluate `SMOG` on a one-dimensional benchmark function

$$f_{1,1}(x) = \sin(x - \delta) \qquad f_{2,1}(x) = \sin(x) \qquad f_{3,1}(x) = \sin(x + \delta) \qquad f_{m,2}(x) = f_{m,1}(x + \phi)$$

where $\delta = \frac{\pi}{12}$ and $\phi = \frac{\pi}{6}$. The target task is a weighted sum of the source tasks $f_{t,o}(x) = \boldsymbol{f}_o(x)^\intercal \boldsymbol{w}_o$, where $\boldsymbol{f}_o(x) = (f_{1,o}(x), \dots, f_{3,o}(x))$, $\boldsymbol{w}_1 = (0.5, 0.35, 0.15)$, and $\boldsymbol{w}_2 = (0.4, 0.4, 0.2)$. We do not benchmark on this problem, but it is instructive to visualize the intermediate posterior of `SMOG` (see Figure 2).

The adapted `Hartmann6` benchmark. The `Hartmann6` problem is defined as[3]

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

where $\boldsymbol{\alpha}$ is a vector of length 4 and $\boldsymbol{A}$ and $\boldsymbol{P}$ are known 4×6 matrices:

$$\boldsymbol{A} = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix} \tag{24}$$

$$\boldsymbol{P} = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix} \tag{25}$$

We adapt this benchmark to allow for meaningful multi-task, multi-objective optimization. First, we sample a separate coefficient vector, $\boldsymbol{\alpha}_m$, for each task. Second, we define an offset vector $\boldsymbol{\varepsilon}_o$ per objective and, for each combination of task and objective $(m, o) \in \mathcal{M}^* \times \mathcal{O}$, aim to minimize

$$f_{m,o}(\boldsymbol{x}) = -\sum_{i=1}^{4} \alpha_{m,i} \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij} - \varepsilon_{o,j})^2\right). \tag{26}$$

We sample $\boldsymbol{\alpha}_m$ and $\boldsymbol{\varepsilon}_o$ as follows:

$$\alpha_{m,1} \sim \mathcal{U}(1.0, 1.02) \tag{27}$$
$$\alpha_{m,2} \sim \mathcal{U}(1.18, 1.2) \tag{28}$$
$$\alpha_{m,3} \sim \mathcal{U}(2.0, 3.0) \tag{29}$$
$$\alpha_{m,4} \sim \mathcal{U}(3.2, 3.4) \tag{30}$$
$$\varepsilon_{o,j} \sim \mathcal{U}(0, 0.15) \tag{31}$$

The distribution for $\varepsilon_{o,j}$ is chosen such that the Gaussian blobs still overlap. The minimum full width at half maximum of any Gaussian blob is given by $2\sqrt{\frac{\ln 2}{17}} \approx 0.4$, where 17 is the largest element in $\boldsymbol{A}$. Hence, a maximum offset of 0.15 is sufficient to ensure correlation and a non-trivial Pareto front between the objectives.

The Terrain benchmark. We rely on the MetaBox implementation by Ma et al. [50], which implements the Terrain benchmark defined in Shehadeh and Kůdela [41]. While MetaBox provides an additional path-clearance cost, we use only the four defined in Shehadeh and Kůdela [41] (path-length cost, obstacle-avoidance cost, altitude cost, and smoothness cost). Furthermore, we define a variant with two objectives, optimizing both path length cost and obstacle avoidance cost. For the target task, we use the first three terrains and choose "meta" terrains uniformly at random with a fixed seed.

Adapted `Branin-Currin` benchmark. We define a multi-task, multi-objective variant of the classical `Branin-Currin` test problem on the unit box $\mathcal{X} = [0, 1]^2$. We consider a set of tasks

[3] `https://www.sfu.ca/~ssurjano/hart6.html`, accessed on: May 6, 2026

$\nu \in \mathcal{M}^* = \{0, 1, \ldots, M\}$, where $\nu = 0$ denotes the target task and $\nu \in \{1, \ldots, M\}$ are meta-tasks, and objectives $o \in \mathcal{O} = [O]$. For general O we alternate between Branin-type (o even) and Currin-type (o odd) objectives, recovering the structure of the standard bi-objective `Branin-Currin` benchmark when $O = 2$.

To ensure that different objectives attain their optima at different locations (and thus induce non-trivial Pareto fronts), we draw an input-shift vector $\boldsymbol{\varepsilon}_o \sim \mathcal{U}([-0.01, 0.01]^2)$ for each objective o and evaluate every objective at the shifted and clamped input:

$$\tilde{\boldsymbol{x}} = \Pi_{[0,1]^2}\big(\boldsymbol{x} - \boldsymbol{\varepsilon}_o\big), \tag{32}$$

where $\Pi_{[0,1]^2}$ denotes component-wise clipping to $[0, 1]$.

Each task–objective pair (ν, o) has its own parameter vector $\boldsymbol{\theta}_{\nu,o}$. Rather than sampling all parameters independently, we employ a two-stage scheme that controls inter-task similarity via a *perturbation scale* $\sigma_p \geq 0$ (default $\sigma_p = 0.05$).

1. **Target parameters.** For every objective o, a target parameterisation $\boldsymbol{\theta}_{0,o}$ is drawn from a *tight* range (specified below).
2. **Meta-task parameters.** For $\nu \geq 1$ each $\boldsymbol{\theta}_{\nu,o}$ is obtained by perturbing $\boldsymbol{\theta}_{0,o}$ with controlled noise whose magnitude is governed by σ_p, and setting $\sigma_p = 0$ makes every meta-task identical to the target.

Every pair (ν, o) uses a deterministic random seed derived from the global seed, so the first M tasks are invariant when the total number of meta-tasks is increased.

Branin-type objectives. The shifted input $\tilde{\boldsymbol{x}} = (\tilde{x}_1, \tilde{x}_2)$ is mapped to the standard Branin domain via $z_1 = 15\,\tilde{x}_1 - 5$, $z_2 = 15\,\tilde{x}_2$:

$$B_{\nu,o}(\boldsymbol{x}) = \big(z_2 - b_{\nu,o}\, z_1^2 + c_{\nu,o}\, z_1 - r_{\nu,o}\big)^2 + s(1-t)\cos(z_1) + s, \qquad s = 10,\ t = \tfrac{1}{8\pi}. \tag{33}$$

The target parameters are sampled from tight ranges:

$$b_{0,o} \sim \mathcal{U}(0.125,\, 0.133), \quad c_{0,o} \sim \mathcal{U}(1.55,\, 1.63), \quad r_{0,o} \sim \mathcal{U}(5.9,\, 6.1), \tag{34}$$

These ranges are centered on the standard Branin values $b^* \approx 0.129$, $c^* \approx 1.59$, $r^* = 6$. Meta-task parameters are obtained by additive Gaussian perturbation, clipped to the same tight bounds $[l_k, u_k]$:

$$\theta_{\nu,o,k} = \mathrm{clip}\big(\theta_{0,o,k} + \delta_k,\ l_k,\ u_k\big), \qquad \delta_k \sim \mathcal{N}\big(0,\, [\sigma_p(u_k - l_k)]^2\big), \quad k \in \{b, c, r\}. \tag{35}$$

Currin-type objectives. The Currin rational form is [51]:

$$C_{\nu,o}(\boldsymbol{x}) = \Big(1 - \exp\big(-\tfrac{1}{2\tilde{x}_2}\big)\Big)\, \frac{p_{\nu,o}(\tilde{x}_1)}{q_{\nu,o}(\tilde{x}_1)}, \tag{36}$$

where $p_{\nu,o}$ and $q_{\nu,o}$ are order-three polynomials that are fixed in Currin et al. [51], but will be slightly varied for the purpose of multi-objective meta-learning. We set $\tilde{x}_2 \leftarrow \max(\tilde{x}_2, 10^{-6})$ for numerical stability. The numerator and denominator polynomials are

$$p_{\nu,o}(u) = a^{(n)}_{\nu,o,1}\, u^3 + a^{(n)}_{\nu,o,2}\, u^2 + a^{(n)}_{\nu,o,3}\, u + a^{(n)}_{\nu,o,4}, \tag{37}$$

$$q_{\nu,o}(u) = a^{(d)}_{\nu,o,1}\, u^3 + a^{(d)}_{\nu,o,2}\, u^2 + a^{(d)}_{\nu,o,3}\, u + a^{(d)}_{\nu,o,4}. \tag{38}$$

The target coefficients are obtained by mildly perturbing the standard Currin constants with i.i.d. uniform multiplicative noise:

$$\boldsymbol{a}^{(n)}_{0,o} = (2300,\, 1900,\, 2092,\, 60)\ \circ\ \boldsymbol{\eta}^{(n)}_{0,o}, \qquad \eta^{(n)}_{0,o,i} \sim \mathcal{U}(0.97,\, 1.03), \tag{39}$$

$$\boldsymbol{a}^{(d)}_{0,o} = (100,\, 500,\, 4,\, 20)\ \circ\ \boldsymbol{\eta}^{(d)}_{0,o}, \qquad \eta^{(d)}_{0,o,j} \sim \mathcal{U}(0.97,\, 1.03). \tag{40}$$

Meta-task coefficients are obtained by log-normal multiplicative perturbation of the target values:

$$a^{(\cdot)}_{\nu,o,i} = a^{(\cdot)}_{0,o,i}\, \exp(\xi_i), \qquad \xi_i \sim \mathcal{N}\big(0,\, (0.1\,\sigma_p)^2\big), \tag{41}$$

This perturbation is applied independently to every numerator and denominator coefficient.

Negation and normalization. We negate the raw function values so that the benchmark is framed as maximization:

$$f_{\nu,o}(\boldsymbol{x}) = -\begin{cases} B_{\nu,o}(\boldsymbol{x}), & \text{type}(o) = \text{Branin}, \\ C_{\nu,o}(\boldsymbol{x}), & \text{type}(o) = \text{Currin}. \end{cases} \tag{42}$$

By default, each objective is normalized to $[0, 1]$. We estimate the global minimum m_o and maximum M_o of $f_{\nu,o}$ across all tasks $\nu \in \{0, \dots, M\}$ via multi-start L-BFGS-B and return

$$\hat{f}_{\nu,o}(\boldsymbol{x}) = \text{clip}\left(\frac{f_{\nu,o}(\boldsymbol{x}) - m_o}{M_o - m_o},\ 0,\ 1\right). \tag{43}$$

C.4 Model details

In this section, we detail the implementations of the different models used in our experiments. Unless stated otherwise, all GPs use a Matérn-$\frac{5}{2}$ kernel with automated relevance determination (ARD) and a Gaussian likelihood with a $\text{LogNormal}(-4, 1)$ hyperprior and a lower bound of 10^{-6} on the noise variance. The lengthscale hyperprior depends on the role of the GP.

- **Meta-task GPs** of `SMOG`, `SMOG (w/o cross-obj.)` and `Ind. ScaML-GP`, as well as the base and target GPs of `Ind. RGPE` and `Ind. SGPT`, use a $\text{Gamma}(1.5, 1)$ lengthscale hyperprior.
- **Target-task GPs** of `SMOG`, `SMOG (w/o cross-obj.)`, `Ind. ScaML-GP` and `Ind. GP` use a $\text{LogNormal}(0.5, 1.5)$ lengthscale hyperprior. For the `ScaML-GP`-based models, this GP models only the difference between the meta-learned prior and the target function.
- `MO-GP` uses the defaults of BoTorch's `KroneckerMultiTaskGP`: an RBF kernel with ARD and BoTorch's dimension-scaled $\text{LogNormal}(\sqrt{2} + \frac{1}{2}\log D, \sqrt{3})$ lengthscale hyperprior, and an LKJ prior ($\eta = 1.5$) on the task correlations.
- `Ind. GC3P` uses an RBF kernel with ARD without hyperpriors and a lower bound of 10^{-3} on the noise variance.

The priors on the signal variances are given in Appendix C.1. We study the sensitivity to the choice of the Matérn-$\frac{5}{2}$ kernel in Appendix E.2. The models with a multi-task kernel (`SMOG`, `SMOG (w/o cross-obj.)` and `MO-GP`) learn a single noise variance shared across objectives, and model the per-objective function variance as part of the multi-task kernel.

Models starting with "`Ind.`" (`Ind. GP`, `Ind. ABLR`, `Ind. RGPE`, `Ind. ScaML-GP`, `Ind. SGPT`, `Ind. GC3P`) fit one model per objective; their target-task models have separate HPs per objective, and the posterior is independent across objectives. The meta-task GPs of `Ind. ScaML-GP` are multi-output GPs with independent outputs that share one lengthscale vector and one noise variance across objectives.

D Memory and runtime analysis

Runtime. We measure the wall-clock time of `SMOG` and its competitors and report the results in Table 2. The measurements use the same containerized setup as the main experiments, with a dedicated allocation of 4 CPU cores for the 2-objective and 8 CPU cores for the 4-objective case. The benchmark is the 6-dimensional Hartmann6 function with 2 and 4 objectives (Appendix C.3). Each optimizer starts with 10 target-task observations and runs 5 Bayesian optimization (BO) iterations; models that use metadata receive 8 meta-tasks, each with 64 observations. We report the one-time cost of fitting the meta-task GPs separately from the per-iteration cost of fitting the target-task HPs and optimizing the AF.

One-time cost. With 2 objectives, `SMOG` fits its meta-task GPs in 22.2 s. This is more than `Ind. ScaML-GP` (3.1 s) and `Ind. RGPE`/`Ind. SGPT` (about 6 s), because each meta-task is modeled with a multi-output Kronecker GP with a full-rank coregionalization matrix. With 4 objectives, this cost rises to 112.4 s. It is paid once per run, before the first BO iteration. Under Assumption 1, the meta-task GPs are independent, so this step can be parallelized across meta-tasks.

Per-iteration cost. With 2 objectives, one `SMOG` iteration takes 4.1 s: 2.1 s for the target-task HPs, 1.9 s for the AF, and 0.05 s to construct the prior. With 4 objectives, it takes 9.8 s: 6.6 s for the HPs

and 3.1 s for the AF. SMOG has the highest per-iteration cost of all methods. The closest competitors are SMOG (w/o cross-obj.) (2.8 s) (see Appendix F.7), Ind. ScaML-GP (2.2 s) with 2 objectives, Ind. ScaML-GP (6.5 s), and MO-GP (5.3 s) with 4 objectives. Most of the difference to SMOG (w/o cross-obj.) comes from fitting the target-task hyperparameters (2.1 vs. 1.0 s with 2 objectives; 6.6 vs. 1.6 s with 4 objectives): the cross-objective correlation and the cross-objective meta-task blocks enlarge the target-task model.

In total, a 50-evaluation run in this setting costs a few minutes of model computation: about $22 + 49 \times 4.1 \approx 220$ s with 2 objectives and $112 + 49 \times 9.8 \approx 590$ s with 4 objectives. This estimate is a lower bound, because the per-iteration cost grows with the number of target observations. This overhead is small compared with the expensive evaluations that motivate meta-learned BO.

Table 2: Wall-clock timing breakdown per optimizer (mean $\pm$ std across 50 runs). All times in seconds.

Optimizer	Meta-GP fit	Prior constr.	Target HP Opt / iter	AF Opt / iter	Total / iter
2 objectives					
SMOG	22.190 ± 2.859	0.051 ± 0.003	2.109 ± 0.497	1.905 ± 0.272	4.058 ± 0.627
SMOG (w/o cross-obj.)	22.781 ± 4.128	0.053 ± 0.005	1.018 ± 0.223	1.785 ± 0.309	2.847 ± 0.432
Ind. GP	—	—	0.305 ± 0.077	0.182 ± 0.043	0.493 ± 0.092
Ind. ABLR	—	0.183 ± 0.005	1.118 ± 0.129	0.143 ± 0.068	1.266 ± 0.190
Ind. GC3P	—	—	0.135 ± 0.011	0.628 ± 0.105	0.766 ± 0.109
MO-GP	—	—	1.278 ± 0.409	0.427 ± 0.072	1.710 ± 0.392
Ind. RGPE	6.048 ± 0.510	0.065 ± 0.004	0.341 ± 0.050	0.581 ± 0.089	0.926 ± 0.101
Ind. SGPT	5.896 ± 0.664	0.063 ± 0.002	0.303 ± 0.040	0.584 ± 0.091	0.891 ± 0.114
Ind. ScaML-GP	3.131 ± 0.529	0.059 ± 0.005	1.512 ± 0.347	0.724 ± 0.149	2.240 ± 0.432
MO-TPE	—	—	—	0.032 ± 0.005	0.038 ± 0.005
4 objectives					
SMOG	112.391 ± 17.778	0.054 ± 0.004	6.624 ± 1.737	3.113 ± 1.126	9.792 ± 2.157
SMOG (w/o cross-obj.)	113.842 ± 17.708	0.054 ± 0.005	1.593 ± 0.431	3.104 ± 1.219	4.754 ± 1.350
Ind. GP	—	—	0.545 ± 0.190	0.219 ± 0.044	0.772 ± 0.208
Ind. ABLR	—	0.189 ± 0.009	3.177 ± 3.288	0.378 ± 0.319	3.563 ± 3.511
Ind. GC3P	—	—	0.262 ± 0.025	1.128 ± 0.173	1.396 ± 0.183
MO-GP	—	—	4.762 ± 2.111	0.508 ± 0.094	5.276 ± 2.115
Ind. RGPE	11.560 ± 1.304	0.064 ± 0.002	0.679 ± 0.181	1.033 ± 0.145	1.719 ± 0.244
Ind. SGPT	11.670 ± 0.718	0.065 ± 0.004	0.620 ± 0.081	1.108 ± 0.162	1.735 ± 0.206
Ind. ScaML-GP	3.456 ± 0.461	0.074 ± 0.005	5.232 ± 0.846	1.234 ± 0.211	6.473 ± 0.921
MO-TPE	—	—	—	0.030 ± 0.004	0.036 ± 0.004

Memory. We also measure the memory consumption of SMOG on the same Hartmann6 benchmark. We initialize SMOG with $M \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ meta-tasks, $N_m = 64$ observations per meta-task, and 10 target-task observations, and run a single BO iteration. We record the resident set size (RSS) of the process after optimizer initialization and its peak over the entire job. All 50 runs of every configuration completed; the results are shown in Table 3.

The RSS after initialization is almost independent of M (about 1.6 GB, which is dominated by the Python and PyTorch runtime); the growth with M occurs during the BO iteration. Memory grows monotonically and sublinearly with M: doubling M increases the peak RSS by at most $1.33\times$ with 2 objectives and $1.65\times$ with 4 objectives. Up to $M = 8$, the peak RSS stays within 10% of the single-meta-task value. From $M = 1$ to $M = 128$, it grows from 1.8 to 3.7 GB ($2.05\times$) with 2 objectives and from 2.4 to 9.4 GB ($3.94\times$) with 4 objectives. Between $M = 64$ and $M = 128$, each additional meta-task adds about 14 MB with 2 objectives and about 58 MB with 4 objectives. This ratio of roughly four matches the $O \times O$ objective blocks per meta-task that SMOG propagates into the target-task prior (Equation (6)). Even with 128 meta-tasks and 4 objectives, SMOG requires less than 10 GB of memory, so memory is not a limiting factor in the regime we consider.

Table 3: Memory consumption as a function of the number of meta-tasks M (50 runs per configuration). Init RSS is the process's peak RSS after optimizer initialization. Job Peak RSS is the true peak over the entire job (via `resource.getrusage`). Scaling factors are relative to the job peak RSS.

# Meta Tasks	Init RSS (MB)	Job Peak RSS (MB)	Increase vs prev	Increase vs smallest M
2 objectives				
1	1585.993 ± 2.255	1814.253 ± 31.248	—	$1.00\times$
2	1586.187 ± 2.109	1926.328 ± 71.478	$1.06\times$	$1.06\times$
4	1586.594 ± 2.344	1965.159 ± 57.811	$1.02\times$	$1.08\times$
8	1586.684 ± 2.640	1999.707 ± 74.440	$1.02\times$	$1.10\times$
16	1587.684 ± 2.015	2085.789 ± 69.937	$1.04\times$	$1.15\times$
32	1589.581 ± 2.451	2278.858 ± 94.520	$1.09\times$	$1.26\times$
64	1594.472 ± 5.385	2801.514 ± 126.976	$1.23\times$	$1.54\times$
128	1607.019 ± 5.663	3716.342 ± 259.493	$1.33\times$	$2.05\times$
4 objectives				
1	1617.664 ± 4.593	2382.540 ± 22.906	—	$1.00\times$
2	1618.327 ± 3.445	2428.407 ± 11.986	$1.02\times$	$1.02\times$
4	1619.586 ± 4.459	2491.989 ± 16.551	$1.03\times$	$1.05\times$
8	1620.457 ± 3.804	2627.912 ± 25.516	$1.05\times$	$1.10\times$
16	1624.085 ± 7.626	2910.986 ± 67.627	$1.11\times$	$1.22\times$
32	1630.692 ± 8.114	3722.578 ± 110.854	$1.28\times$	$1.56\times$
64	1649.406 ± 7.595	5678.639 ± 337.544	$1.53\times$	$2.38\times$
128	1725.831 ± 25.216	9393.710 ± 621.011	$1.65\times$	$3.94\times$

E Ablation studies

We conduct ablation studies to study the sensitivity of `SMOG` and the other optimizers to different benchmark properties. On the `Hartmann6` benchmark, we set the default to 2 objectives, 8 meta-tasks, and 64 observations per meta-task, and vary one of these parameters per experiment. The ablations use one initial point and 29 BO iterations, totaling in 30 function evaluations.

We summarize each run by its cumulative HV gap over the 30 evaluations, normalized as in the corresponding figure, and test differences with paired two-sided Wilcoxon signed-rank tests, Holm-corrected over the values of the varied parameter.

In Figure 6, we vary the number of meta-tasks. More meta-tasks improve the performance of almost all methods, particularly in early iterations; `Ind. SGPT` improves only slightly. `SMOG`, `SMOG (w/o cross-obj.)`, `Ind. ScaML-GP` and `Ind. RGPE` form the leading group for every number of meta-tasks. With 2 meta-tasks, `SMOG` has the lowest cumulative HV gap, significantly lower than `Ind. RGPE` (by 9.0%); from 8 meta-tasks on, `Ind. RGPE` has the lowest HV gap, and `SMOG` accumulates 8–18% more (not significant). `SMOG` significantly outperforms `Ind. SGPT`, `Ind. ABLR`, `Ind. GC3P` and `MO-TPE` for every number of meta-tasks.

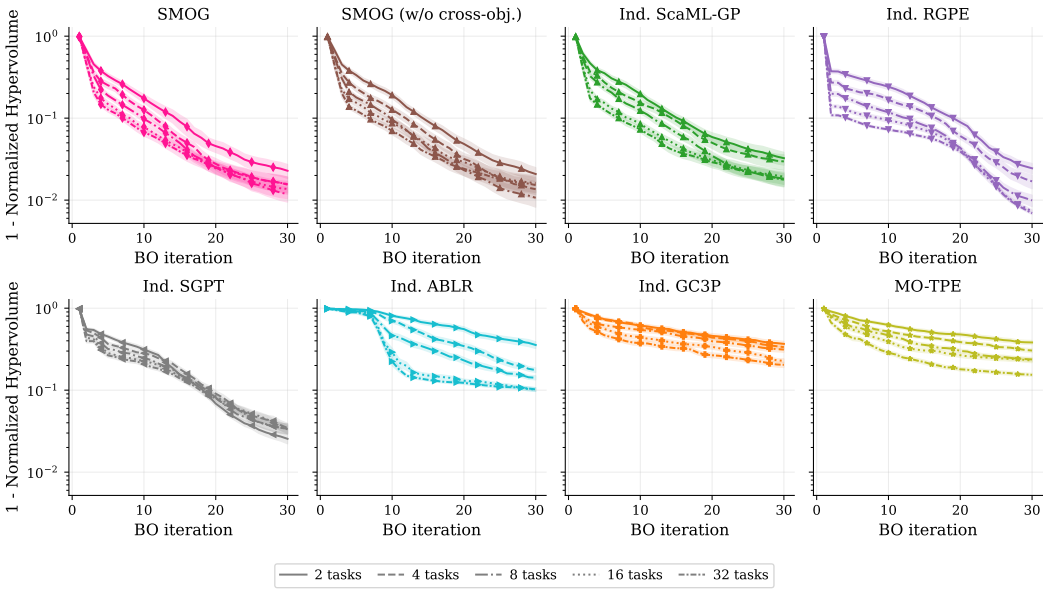


Figure 6: Ablation over the number of meta tasks.

Next, we vary the number of objectives (Figure 7). The relative ordering of the methods is largely preserved across 2, 3 and 4 objectives. `Ind. RGPE` has the lowest cumulative HV gap in all three settings, followed by `SMOG (w/o cross-obj.)`, `SMOG` and `Ind. ScaML-GP`; the differences among these four models are not significant. `SMOG` significantly outperforms `Ind. SGPT`, `Ind. ABLR`, `Ind. GC3P` and `MO-TPE` for every number of objectives, although its advantage shrinks as objectives are added (e.g., from 39.6% to 21.7% against `Ind. SGPT`).

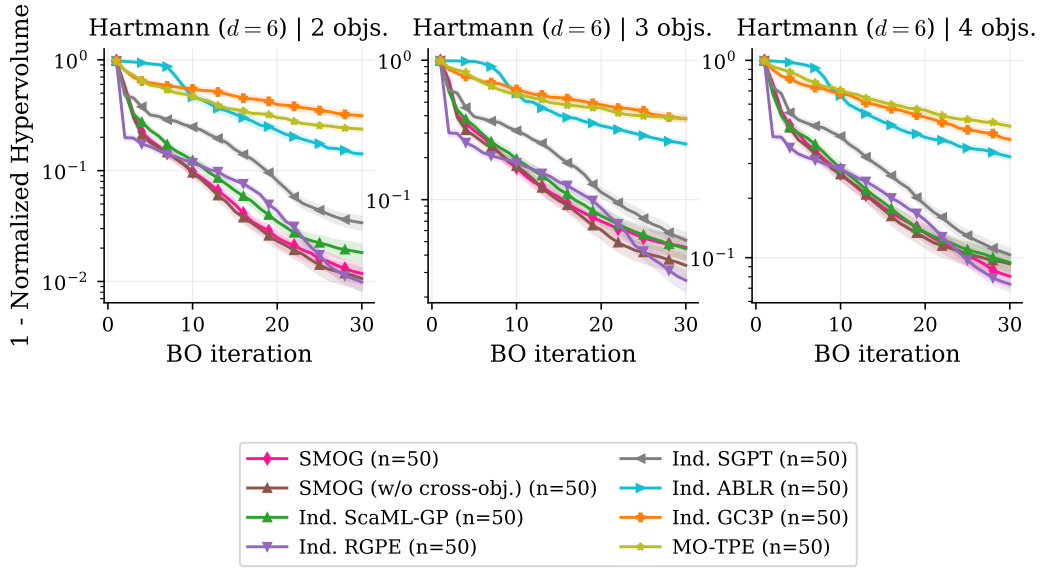


Figure 7: Ablation over the number of objectives.

Finally, we vary the number of observations per meta-task (Figure 8). More observations per meta-task improve the performance of nearly all methods. `Ind. RGPE` has the lowest cumulative HV gap for every observation count; the difference to `SMOG` is significant only with 32 observations, where `SMOG` accumulates 17.5% more HV gap. `SMOG` ranks second with 16 observations, and with 64 observations

it is within 4% of SMOG (w/o cross-obj.). SMOG significantly outperforms Ind. SGPT, Ind. ABLR, Ind. GC3P and MO-TPE for every observation count.

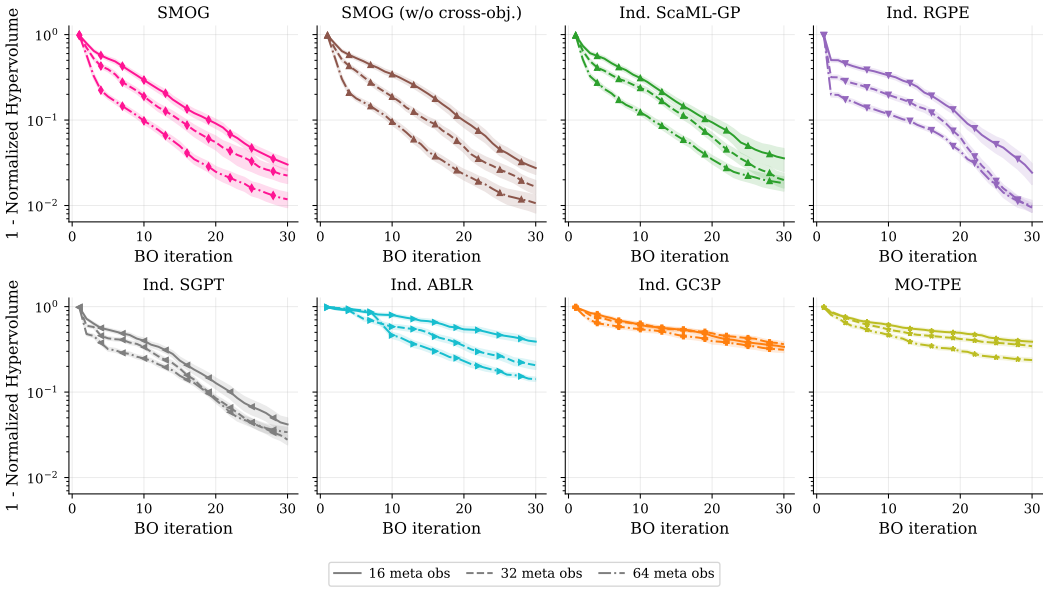


Figure 8: Ablation over the number of observations per meta task.

E.1 Acquisition function comparison

All GP-based optimizers except DGEMO use LogEHVI as the default AF. A natural question is whether the choice of AF decisively influences performance, or whether the meta-learning components are the main driver. To investigate this, we re-run all meta-learning optimizers on the Branin-Currin and Hartmann6 benchmarks (8 meta tasks, 64 observations per task, 2 objectives, 50 seeds) with two AFs: LogEHVI [38] (solid lines) and qLogParEGO [38] (dashed lines). MO-TPE is excluded from this comparison as it does not employ a GP-based AF. Figures 9 and 10 show the results.

Across both benchmarks and nearly all surrogates, LogEHVI reaches a lower final HV gap than qLogParEGO. This is consistent with the fact that LogEHVI directly optimizes an HV-based criterion, whereas qLogParEGO relies on random scalarizations of the objectives. The magnitude of this gap depends on both the surrogate and the benchmark. On Branin-Currin, the two AFs yield largely overlapping curves for several methods (e.g. Ind. ScaML-GP, Ind. RGPE, Ind. GC3P), while more pronounced differences appear for Ind. ABLR. On Hartmann6, LogEHVI provides a clearer advantage for most methods—including SMOG, Ind. ScaML-GP, and Ind. RGPE—suggesting that HV-aware acquisition is particularly beneficial in higher-dimensional settings.

Importantly, the relative ranking of meta-learning methods is largely preserved under both AFs, and the gap between LogEHVI and qLogParEGO on a given surrogate is typically smaller than the gap between different surrogates. SMOG remains among the top-performing methods on both benchmarks regardless of the AF used, confirming that its benefit stems from the structured task-covariance prior rather than from the choice of acquisition function, and supporting the broader conclusion that the meta-learning component—rather than the scalarization strategy—is the primary source of performance differences across optimizers.

Both AFs evaluated here are myopic. Lookahead and information-theoretic methods, including HVKG [44] and multi-objective entropy search [45, 46], provide complementary alternatives; studying how they can best exploit SMOG's correlated posterior is left for future work.

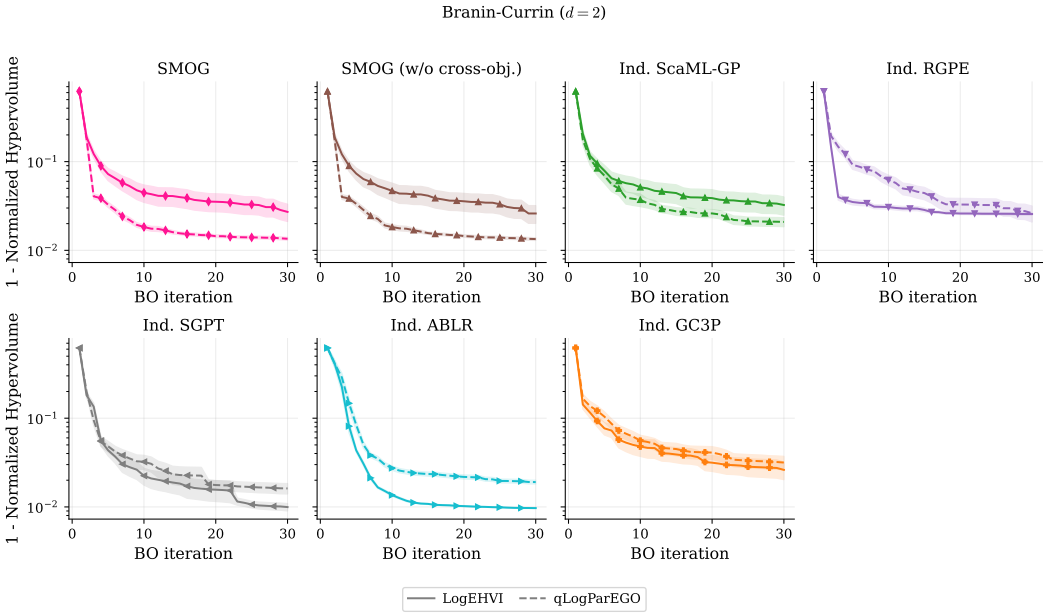


Figure 9: Acquisition function comparison on `Branin-Currin` ($d = 2$, 2 objectives). Solid lines: `LogEHVI`; dashed lines: `qLogParEGO`. Each curve shows the mean $\pm$ SEM over 50 seeds. Lower is better.

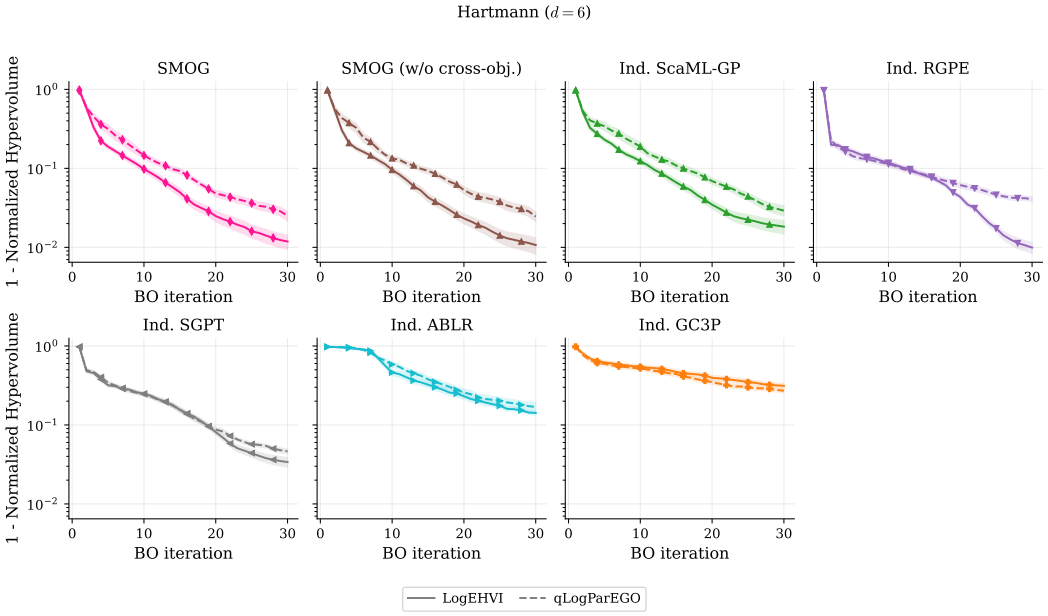


Figure 10: Acquisition function comparison on `Hartmann6` ($d = 6$, 2 objectives). Solid lines: `LogEHVI`; dashed lines: `qLogParEGO`. Each curve shows the mean $\pm$ SEM over 50 seeds. Lower is better.

E.2 Kernel sensitivity

All GP-based models in the main experiments use a Matérn-$\frac{5}{2}$ kernel. To test whether our conclusions depend on this choice, we replace the kernel of every GP (meta-task and target-task) by a Matérn-$\frac{3}{2}$ or an RBF kernel, keeping the ARD structure and the lengthscale priors (Appendix C.4), and rerun `SMOG`,

SMOG (w/o cross-obj.), and Ind. ScaML-GP on Branin-Currin (2 objectives), Hartmann6 (2 and 4 objectives), and Terrain task 0 (2 objectives) with the main text settings.

Table 4 shows that the comparison is largely insensitive to the kernel. On Branin-Currin, SMOG accumulates significantly less HV gap than Ind. ScaML-GP with every kernel ($+14.8\%$, $+8.6\%$, $+3.9\%$ for Matérn-$\frac{5}{2}$, Matérn-$\frac{3}{2}$, RBF); the advantage shrinks for the rougher kernels. On Hartmann6 and Terrain, the three models remain within a few percent of each other for every kernel; the only significant difference is an advantage over SMOG (w/o cross-obj.) on Hartmann6 with 2 objectives and the Matérn-$\frac{3}{2}$ kernel ($+9.6\%$).

Table 4: Sensitivity to the base kernel: final cumulative HV gap after 50 evaluations (mean $\pm$ SEM over 50 seeds; lower is better) when the input kernel of every GP (meta-task and target-task) is a Matérn-$\frac{5}{2}$ (default), Matérn-$\frac{3}{2}$, or RBF kernel (M-$\frac{5}{2}$, M-$\frac{3}{2}$, RBF), with the same ARD structure and lengthscale priors. Models: SMOG, SMOG (w/o cross-obj.) (w/o cross), Ind. ScaML-GP (Ind. ScaML). Normalization per setting pooled over all kernels and models. Δ: relative reduction of mean HV gap achieved by SMOG with the same kernel (positive: SMOG better). Bold with *: significant at level 0.05 under a two-sided Wilcoxon signed-rank test paired by seed, Holm-corrected over the four settings of each comparison and kernel.

			Cumulative HV gap			Δ (%) of SMOG vs	
Problem	O	Kernel	SMOG	w/o cross	Ind. ScaML	w/o cross	Ind. ScaML
Branin-C.	2	M-$\frac{5}{2}$	1.50 ± 0.08	1.53 ± 0.08	1.77 ± 0.12	$+1.8$	$\mathbf{+14.8}^*$
		M-$\frac{3}{2}$	1.71 ± 0.13	1.59 ± 0.09	1.87 ± 0.14	-7.4	$\mathbf{+8.6}^*$
		RBF	1.63 ± 0.12	1.54 ± 0.10	1.70 ± 0.10	-6.0	$\mathbf{+3.9}^*$
Hartmann6	2	M-$\frac{5}{2}$	7.93 ± 0.37	8.41 ± 0.44	8.04 ± 0.54	$+5.7$	$+1.4$
		M-$\frac{3}{2}$	7.91 ± 0.44	8.75 ± 0.46	8.72 ± 0.51	$\mathbf{+9.6}^*$	$+9.3$
		RBF	8.87 ± 0.41	8.79 ± 0.44	8.13 ± 0.38	-1.0	-9.2
Hartmann6	4	M-$\frac{5}{2}$	15.8 ± 0.6	15.2 ± 0.7	14.8 ± 0.6	-3.8	-6.4
		M-$\frac{3}{2}$	17.0 ± 0.8	16.2 ± 0.7	16.2 ± 0.7	-4.7	-5.0
		RBF	14.7 ± 0.6	14.4 ± 0.5	14.2 ± 0.5	-1.8	-3.1
Terrain t0	2	M-$\frac{5}{2}$	14.1 ± 0.3	14.0 ± 0.4	14.8 ± 0.4	-0.6	$+4.6$
		M-$\frac{3}{2}$	14.7 ± 0.4	14.3 ± 0.4	14.9 ± 0.5	-3.0	$+1.4$
		RBF	13.9 ± 0.3	13.8 ± 0.3	13.7 ± 0.3	-0.4	-1.6

E.3 Objective correlation strength

We study the effect of the correlation between objectives for SMOG, SMOG (w/o cross-obj.), and Ind. ScaML-GP. In particular, we vary the correlation ρ between $\{-0.9, -0.5, -0.2, 0, 0.2, 0.5, 0.9\}$. To allow varying the correlation strengths, we construct variants of the Hartmann6 and Branin-Currin benchmarks.

Correlated Hartmann Benchmark. We model each objective o of task ν as a convex combination of a private and a shared Hartmann landscape:

$$f_{\nu,o}(\boldsymbol{x}) = s_o\sqrt{\rho'}S_{\nu,o}(\circlearrowleft_0(\boldsymbol{x}); \boldsymbol{\alpha}_\nu, \boldsymbol{\varepsilon}_o) + \sqrt{1-\rho'}N_{\nu,o}(\circlearrowleft_o(\boldsymbol{x}); \boldsymbol{\alpha}_\nu, \boldsymbol{\varepsilon}_o). \tag{44}$$

Here, ρ controls the correlation strength, S and N are Hartmann functions as defined in Equation (26)—approximately normalized to zero mean and unit variances based on 8,192 uniform samples, $\boldsymbol{\alpha}_i$ and $\boldsymbol{\varepsilon}_j$ are the corresponding coefficient and offset vectors, and $x^{\circlearrowleft}$ indicates that a random rotation has been applied to x w.r.t. the search space center $\boldsymbol{c} = (0.5, \ldots, 0.5)$: $\circlearrowleft_i(\boldsymbol{x}) = R_i(\boldsymbol{x} - \boldsymbol{c}) + \boldsymbol{c}$, and R_i is a random orthonormal rotation matrix. We set $s_o \leftarrow \begin{cases} -1 & \rho < 0 \\ 1 & \text{otherwise} \end{cases}$. The mixing weight $\rho' \approx |\rho|$ is calibrated via bisection search to approximately match the desired correlation.

Correlated Branin-Currin Benchmark. Similar to above, we construct a correlated Branin-Currin variant. Different from Hartmann6, the two objectives of the Branin-Currin benchmark are negatively correlated with a Pearson correlation of approximately -0.45. For the

Table 5: Effect of the correlation ρ between objectives on the Correlated `Hartmann6` benchmark (2 objectives, 8 meta-tasks with 16 observations each; Appendix E.3): final cumulative HV gap after 50 evaluations (mean $\pm$ SEM over 50 seeds; lower is better), normalized per ρ. Models: `SMOG`, `SMOG (w/o cross-obj.)` (w/o cross), and `Ind. ScaML-GP` (Ind. ScaML). Δ: relative reduction of mean HV gap achieved by `SMOG` (positive: `SMOG` better). Bold with *: significant at level 0.05 under a two-sided Wilcoxon signed-rank test paired by seed, Holm-corrected over the seven values of ρ.

	Cumulative HV gap			Δ (%) of `SMOG` vs	
ρ	`SMOG`	w/o cross	Ind. ScaML	w/o cross	Ind. ScaML
-0.9	30.8 ± 0.5	29.5 ± 0.5	29.6 ± 0.4	$\mathbf{-4.6}^*$	-4.3
-0.5	21.6 ± 0.6	20.5 ± 0.4	21.1 ± 0.6	-5.5	-2.4
-0.2	23.8 ± 0.6	23.7 ± 0.5	24.0 ± 0.6	-0.1	$+1.1$
0.0	26.9 ± 0.5	26.9 ± 0.5	26.3 ± 0.6	0.0	-2.4
$+0.2$	24.4 ± 0.8	25.1 ± 0.9	24.2 ± 0.8	$+2.8$	-0.6
$+0.5$	19.2 ± 0.8	19.9 ± 0.8	19.4 ± 0.8	$\mathbf{+3.4}^*$	$+0.9$
$+0.9$	10.7 ± 0.5	11.1 ± 0.6	11.6 ± 0.6	$+4.0$	$+7.6$

Table 6: Effect of the correlation ρ between objectives on the Correlated `Branin-Currin` benchmark (2 objectives, 8 meta-tasks with 16 observations each; Appendix E.3); layout and statistics as in Table 5. At $\rho = -0.9$ the objectives are almost perfectly conflicting, so even the optimal front covers only a small part of the normalized objective space and all HV gaps are large.

	Cumulative HV gap			Δ (%) of `SMOG` vs	
ρ	`SMOG`	w/o cross	Ind. ScaML	w/o cross	Ind. ScaML
-0.9	28.9 ± 0.5	27.9 ± 0.6	27.1 ± 0.7	$\mathbf{-3.7}^*$	$\mathbf{-6.5}^*$
-0.5	1.83 ± 0.14	1.81 ± 0.12	2.09 ± 0.13	-1.0	$\mathbf{+12.7}^*$
-0.2	1.67 ± 0.14	1.58 ± 0.10	1.99 ± 0.13	-6.2	$\mathbf{+16.1}^*$
0.0	1.67 ± 0.11	1.67 ± 0.10	2.03 ± 0.11	0.0	$\mathbf{+17.8}^*$
$+0.2$	1.46 ± 0.07	1.48 ± 0.09	1.98 ± 0.13	$+1.7$	$\mathbf{+26.5}^*$
$+0.5$	1.28 ± 0.06	1.28 ± 0.06	1.81 ± 0.15	-0.5	$\mathbf{+29.3}^*$
$+0.9$	0.87 ± 0.04	0.91 ± 0.05	1.36 ± 0.12	$+5.1$	$\mathbf{+36.4}^*$

correlation analysis, we therefore change the second objective to a combination of Branin and Currin objectives to allow for positive correlations:

$$f_{\nu,1}(x) = \tilde{B}_\nu(x), \qquad f_{\nu,2}(x) = a\,\tilde{B}_\nu(x) + b\,\tilde{C}_\nu(x), \qquad b = \sqrt{\frac{1-\rho^2}{1-c^2}}, \qquad a = \rho - b\,c, \quad (45)$$

where the `Branin-Currin` benchmarks are constructed as described in Appendix C.3 and normalized w.r.t. the mean and variance of the target-task, estimated on 8,192 uniform samples. Again, we use $\rho \in \{-0.9, -0.5, -0.2, 0, 0.2, 0.5, 0.9\}$ and 8 meta-tasks and 16 observations per task.

Results. Table 5 and Table 6 show the cumulative HV gap of `SMOG` and `SMOG (w/o cross-obj.)` compared to `Ind. ScaML-GP` after 50 evaluations. We observe a general trend of `SMOG` to outperform the others with increasing objective correlation. However, these differences are mostly not significant when studying the `Hartmann6` benchmark or comparing to `SMOG (w/o cross-obj.)`. On `Branin-Currin`, `SMOG`'s headroom to `Ind. ScaML-GP` grows with ρ.

F Additional figures and experiments

F.1 Synthetic benchmarks

Figure 3 in the main text shows the cumulative HV gap on the synthetic benchmarks. Figure 11 below shows the corresponding HV gap.

F.2 HPOBench benchmarks

In this section, we present the HV gap results on the individual datasets of the HPOBench benchmark, complementing the cumulative HV gap shown in Figure 4 in the main text.

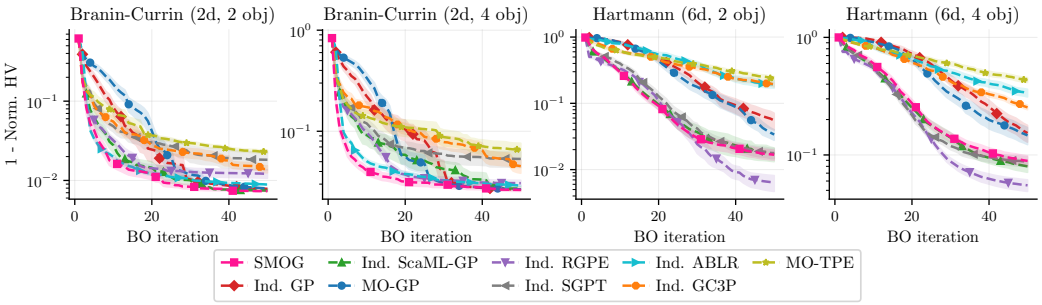


Figure 11: HV gap (1− normalized HV; lower is better) on the synthetic benchmarks. Columns correspond to benchmark and objective-count combinations: `Branin-Currin` (2 obj), `Branin-Currin` (4 obj), `Hartmann6` (2 obj), `Hartmann6` (4 obj). The y-axis is on a logarithmic scale. All results use 8 meta-tasks and 16 observations per meta-task.

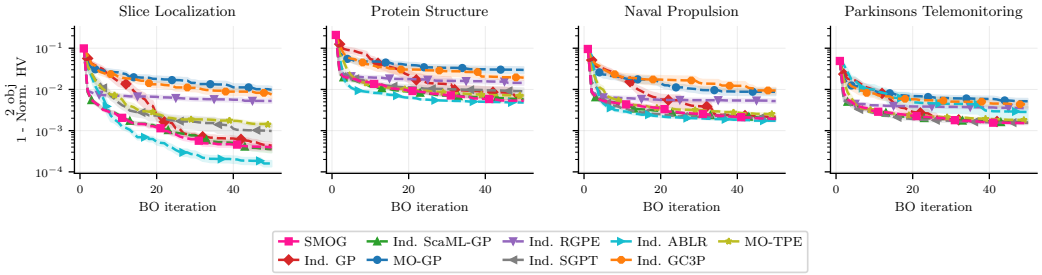


Figure 12: HV gap (lower is better) on the individual HPOBench datasets. Columns correspond to the four target datasets. All results use 3 meta-tasks and 16 observations per meta-task.

F.3 Terrain benchmarks

Figure 13 shows the HV gap on the Terrain benchmark for all three target tasks and both objective configurations, complementing the cumulative HV gap shown in Figure 5 in the main text.

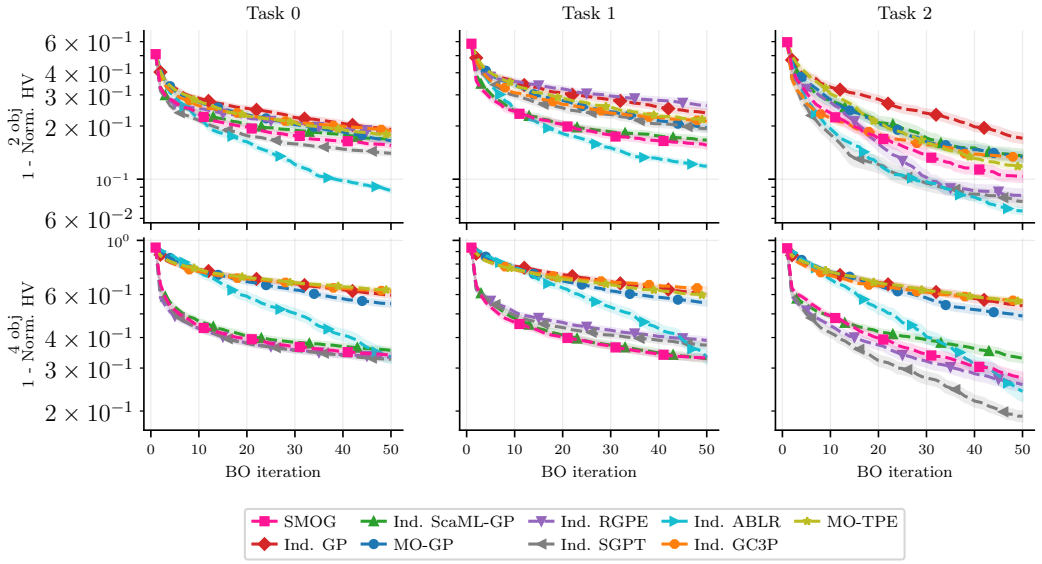


Figure 13: HV gap (lower is better) on the Terrain benchmark. Rows correspond to 2-objective (top) and 4-objective (bottom) variants; columns correspond to target tasks 0, 1, and 2.

F.4 Exemplary `Hartmann6` Pareto front

In this section, we highlight how `SMOG` learns well-behaved Pareto fronts on the two-objective `Hartmann6` benchmark. For each optimizer, we study the first 15 observations of the run of median HV at iteration 15. Based on these 15 observations, we plot the Pareto front and show the results in Figure 14. Additionally, we estimate a "true" Pareto front by observing 10^6 points sampled uniformly at random from the search space $\mathcal{X}$. Even though `Ind. SGPT` and `Ind. RGPE` can find well-performing solutions, the front found by `SMOG` lies closest to the "true" Pareto front after only 15 observations on the target task, closely followed by `Ind. ScaML-GP`. Note that some methods may not appear in Figure 14: the plot is clipped to $[2.5, \infty)$ on both axes, so methods whose Pareto-front solutions all fall outside this range are not shown. For details on `SMOG (w/o cross-obj.)`, see Appendix F.7; for details on `DGEMO` and `NSGA-III`, see Appendix F.8.

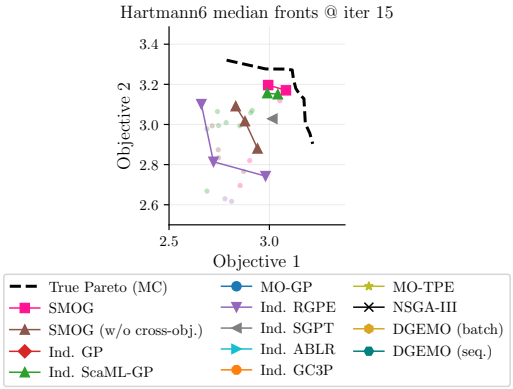


Figure 14: Pareto fronts of the first 15 observations of `SMOG` and competitors on the 6-dimensional two-objective `Hartmann6` benchmark for the run of median HV at iteration 15. The "true" Pareto front is approximated from 10^6 uniformly sampled points. Both axes are clipped to $[2.5, \infty)$; methods with no solutions in this range are not shown. `SMOG` learns a Pareto front close to the true Pareto front within 15 iterations.

F.5 HPOBench Pareto fronts

To assess the quality of solutions produced by the different optimizers, we plot the Pareto fronts based on observations across 50 repetitions (see Figure 15). Since HPOBench is a tabular benchmark, one can enumerate all possible solutions and compute the true Pareto front. However, since the HPOBench dataset provides four observations per configuration and we observe only once, we plot an estimated true Pareto front, which explains why some observations are better than the "true" Pareto front. For details on `SMOG (w/o cross-obj.)`, see Appendix F.7; for details on `DGEMO` and `NSGA-III`, see Appendix F.8.

Importantly, the solutions found by the optimizers are close to the "true" Pareto front, showing that the solutions actually are of good quality—a fact that is not evident from only studying the normalized HV.

Furthermore, the GP-based methods that do not leverage metadata (`Ind. GP` and `MO-GP`) tend to perform worse, confirming the intuition that metadata access improves solution quality.

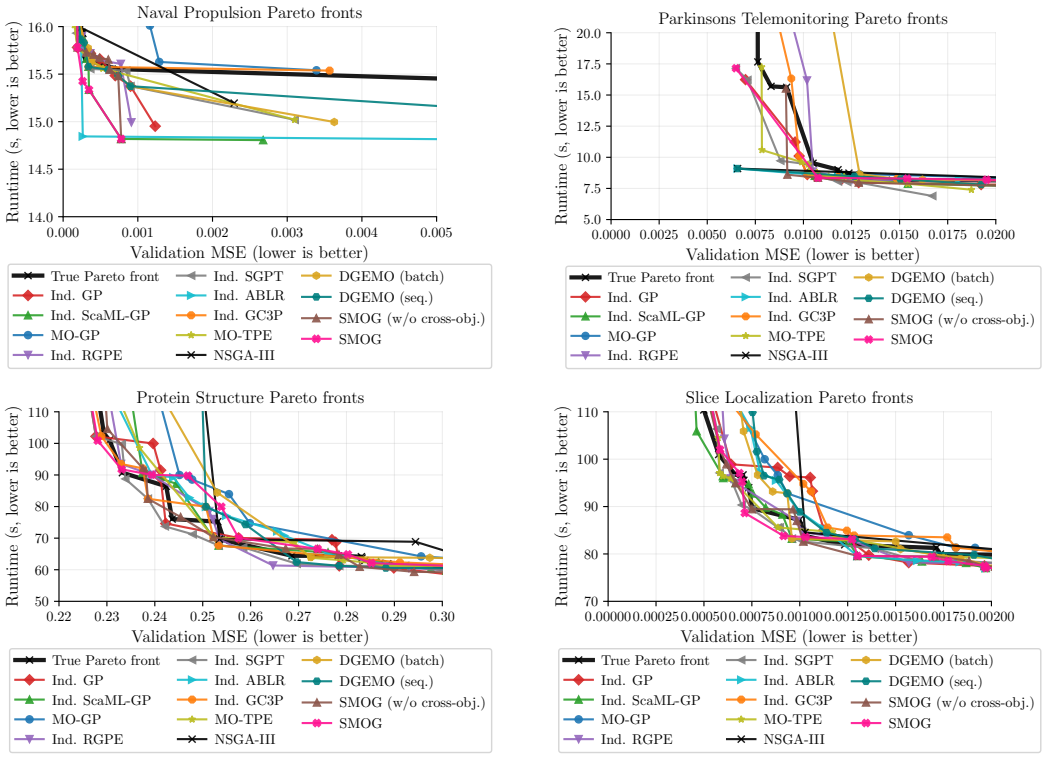


Figure 15: Pareto fronts of the HPOBench benchmark, pooled across the 50 repetitions. The black line shows an estimated true Pareto front.

F.6 Surrogate quality

To complement the optimization-performance results in the main text, we directly evaluate the *surrogate quality* of each model: After fitting on $n_{\text{init}} = 10$ target-task observations drawn from a Sobol sequence, we measure the negative log predictive density (NLPD) and root mean squared error (RMSE) against 500 held-out noiseless Sobol points. The experiment, therefore, isolates prior and meta-learning quality from the effect of the AF. Meta-data consists of $M = 8$ meta-tasks with 64 observations each; we run 50 independent seeds per setting on the synthetic benchmarks and 150 on Terrain. For Terrain, all runs use the same target terrain, selected with the benchmark seed, so the error bars reflect variation over seeds only. We evaluate the models of the main text and `SMOG (w/o cross-obj.)` (Appendix F.7); `MO-TPE` is excluded because it does not maintain a GP surrogate.

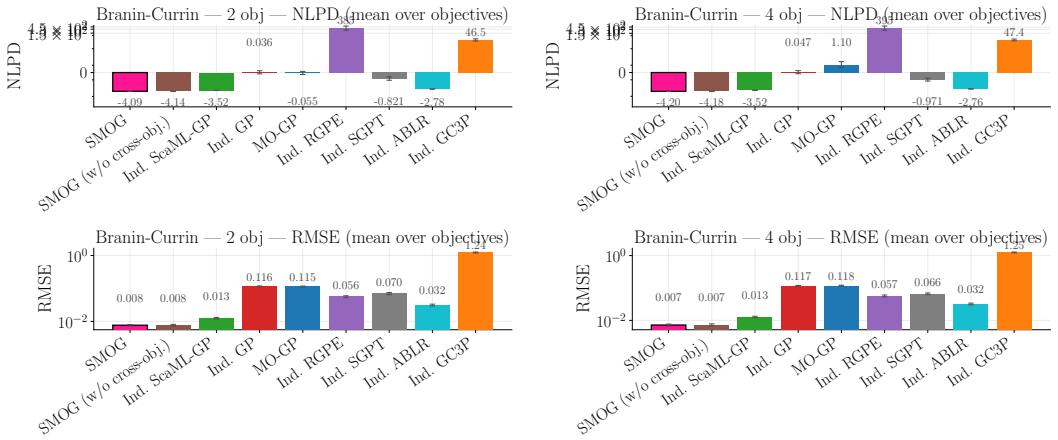
We report results across three benchmarks (`Branin-Currin`, `Hartmann6`, Terrain) and both 2- and 4-objective configurations. Lower NLPD and RMSE indicate a better-calibrated, more accurate surrogate. Figures 16–18 show the mean ± SEM over seeds.

RMSE. On `Branin-Currin`, `SMOG` and `SMOG (w/o cross-obj.)` are the most accurate surrogates with both 2 and 4 objectives (0.008 and 0.007), followed by `Ind. ScaML-GP` (0.013) and `Ind. ABLR` (0.032); `Ind. GP` and `MO-GP` have about 15 times the error of `SMOG` (0.12). On `Hartmann6`, the three `ScaML-GP`-based models `SMOG`, `SMOG (w/o cross-obj.)` and `Ind. ScaML-GP` have the lowest RMSE (0.25–0.26), ahead of `Ind. RGPE` (0.28–0.29) and `Ind. GP` and `MO-GP` (0.40–0.43). On Terrain, `SMOG`, `SMOG (w/o cross-obj.)`, `Ind. ScaML-GP` and `Ind. GP` are within 0.2% of each other; `MO-GP`, `Ind. RGPE` and `Ind. SGPT` are about 8% higher, `Ind. ABLR` about 75% higher, and `Ind. GC3P` more than three times higher. With 10 target observations, the metadata therefore improve accuracy considerably on the synthetic benchmarks but hardly on Terrain.

NLPD. On `Branin-Currin`, `SMOG` and `SMOG (w/o cross-obj.)` also have the best NLPD (-4.09 and -4.14 with 2 objectives, -4.20 and -4.18 with 4), followed by `Ind. ScaML-GP` (-3.52) and `Ind. ABLR` (-2.78 and -2.76); `Ind. RGPE` and `Ind. GC3P` are badly miscalibrated (mean NLPD

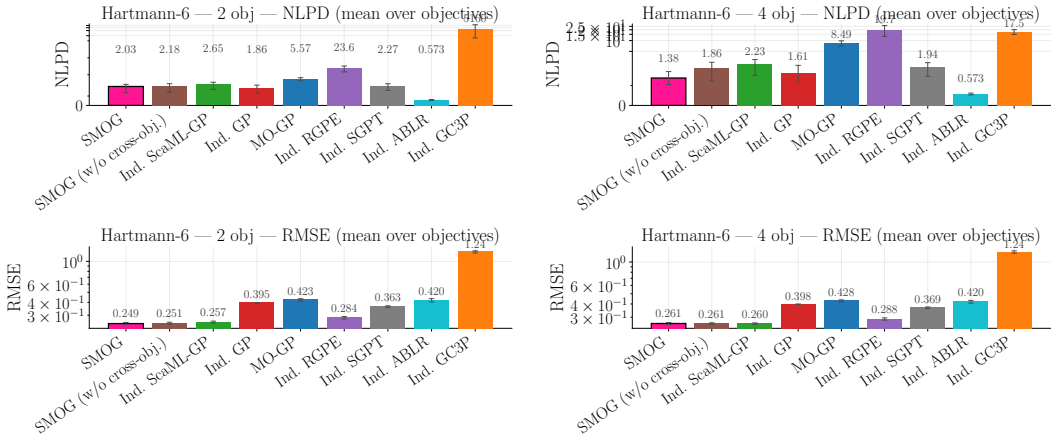
of 385–395 and 46–47). On `Hartmann6`, `Ind. ABLR` is the best-calibrated model (0.57 with 2 and 4 objectives), although its RMSE is 1.7 times that of `SMOG`. `SMOG` ranks third with 2 objectives (2.03, behind `Ind. GP` with 1.86) and second with 4 objectives (1.38), and is better calibrated than `SMOG (w/o cross-obj.)` (2.18 and 1.86) and `Ind. ScaML-GP` (2.65 and 2.23). The mean NLPD on `Hartmann6` is dominated by a few seeds with large values (SEM up to 0.9), so the differences among the leading models are not decisive. On Terrain, all models except `Ind. RGPE`, `Ind. ABLR` and `Ind. GC3P` lie within 2% of each other (9.94–10.10 with 2 objectives, 8.60–8.78 with 4), with `Ind. SGPT` and `Ind. GP` marginally ahead; `Ind. GC3P` reaches a mean NLPD on the order of 10^{18}.

Summary. `SMOG` is the most accurate surrogate, or tied for it, on both synthetic benchmarks, and the best calibrated on `Branin-Currin`. On `Hartmann6`, the parametric `Ind. ABLR` is better calibrated at a substantially higher RMSE. On Terrain, the metadata change neither accuracy nor calibration much for any well-behaved model. Removing the cross-objective terms (`SMOG (w/o cross-obj.)`) leaves the RMSE unchanged and increases the NLPD on `Hartmann6`.



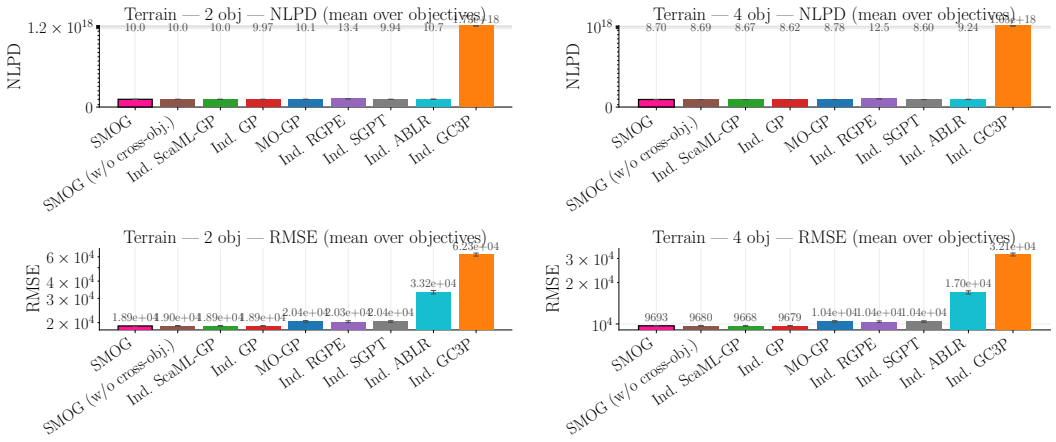
(a) 2 objectives (b) 4 objectives

Figure 16: Surrogate quality (NLPD and RMSE, lower is better) on `Branin-Currin` after 10 target observations. Error bars: ±1 SEM over 50 seeds. `SMOG` and `SMOG (w/o cross-obj.)` have the lowest RMSE and NLPD with both 2 and 4 objectives.



(a) 2 objectives (b) 4 objectives

Figure 17: Surrogate quality (NLPD and RMSE, lower is better) on `Hartmann6` after 10 target observations. Error bars: ±1 SEM over 50 seeds. `SMOG`, `SMOG (w/o cross-obj.)` and `Ind. ScaML-GP` have the lowest RMSE; `Ind. ABLR` has the lowest NLPD.



(a) 2 objectives

(b) 4 objectives

Figure 18: Surrogate quality (NLPD and RMSE, lower is better) on Terrain after 10 target observations (one target terrain, 150 seeds). Error bars: ±1 SEM over seeds. Apart from `Ind. RGPE`, `Ind. ABLR` and `Ind. GC3P`, all models are within a few percent of each other.

F.7 Impact of modeling target correlations

To quantify the effect of modeling target correlations, we evaluate a variant of `SMOG` called `SMOG (w/o cross-obj.)` that removes objective correlations in the target task. The differences to `SMOG` are twofold: 1) We enforce $\rho = 0$ (see Equation (10)) so that residual objective correlations are not modeled, and 2) we remove the contribution of meta-task cross-objective correlations in the target kernel, i.e., we set the sum in the RHS of Equation (6), $k_{t,\texttt{SMOG}}$, to 0 if $o \neq o'$.

Table 7: The `SMOG` variants and the closest baseline. All three use per-objective meta-task weights w_{mo}, the same meta-task data, and a learned target-task signal variance per objective. *Meta-task scale*: the per-objective signal variance of the fitted meta-task coregionalization matrix enters the target-task prior. *Cross-obj. meta term*: the term $\sum_m w_{mo} w_{mo'} \hat{k}_{moo'}$ for $o \neq o'$ in Equation (6). ρ: correlation of the equicorrelated target-task residual (Section 3.3). Rows 1–2 differ only in the cross-objective terms of the target-task model; row 3 models the target task with one `ScaML-GP` per objective, so its target-task posterior is independent across objectives. See also Section 4.2 and Appendix C.4.

Model	Meta-task GPs	Target-task prior			Target-task model
		Meta-task scale	Cross-obj. meta term	ρ	
`SMOG`	Kronecker	✓	✓	learned	joint[†]
`SMOG (w/o cross-obj.)`	Kronecker	✓	–	0	joint[†]
`Ind. ScaML-GP`	independent outputs[§]	–	–	–	per objective

[†]Input kernel and noise variance shared across objectives. [§]One lengthscale vector and one noise variance shared across objectives.

Table 7 shows an overview of the difference between `SMOG`, `SMOG (w/o cross-obj.)`, and `Ind. ScaML-GP`. Figures 19–21 compare `SMOG` to `SMOG (w/o cross-obj.)` on the synthetic, HPOBench, and Terrain benchmarks. We observe that `SMOG` and `SMOG (w/o cross-obj.)` perform very similarly, with the exception of the 4-objective variant of `Branin-Currin`, where `SMOG` accumulates 15.9% less HV gap; this is the only significant difference across the 14 settings (tested as in Table 1). `SMOG (w/o cross-obj.)` itself significantly outperforms `Ind. ScaML-GP` on `Branin-Currin` with 2 and 4 objectives (by 13.2% and 22.7%) and on Terrain task 2 with 2 objectives (by 17.9%). Overall, the results indicate that on the benchmarks studied in this work and the assumed Kronecker structure, the main advantage of `SMOG` comes from its ability to model correlations in the meta-task GPs.

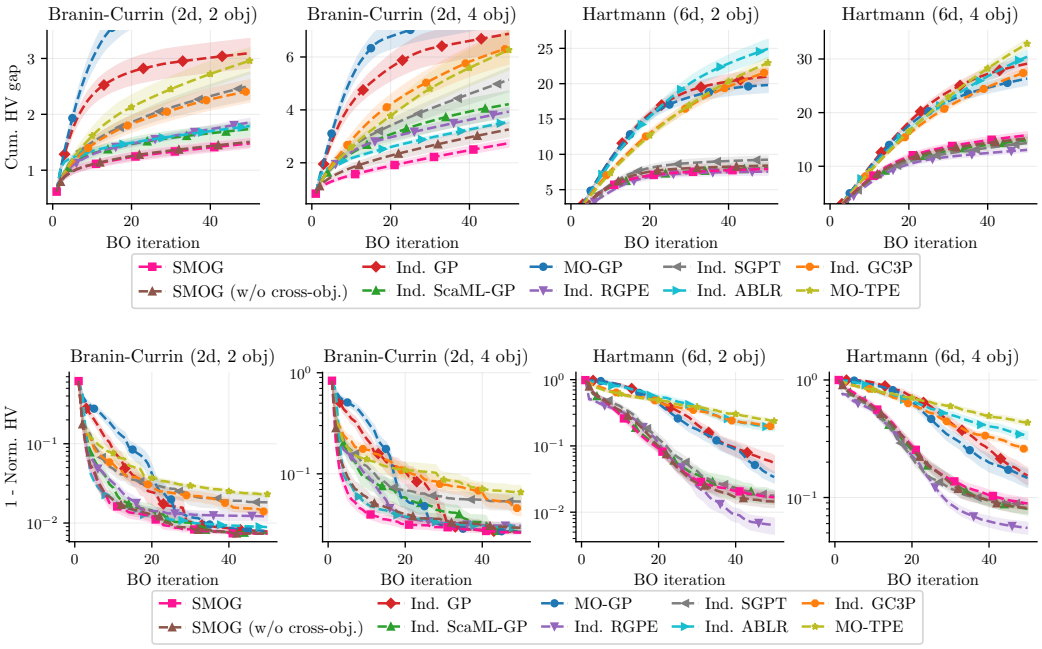


Figure 19: cumulative HV gap (top) and HV gap (bottom; lower is better for both) on the synthetic benchmarks, including `SMOG (w/o cross-obj.)`. Columns correspond to `Branin-Currin` (2 obj), `Branin-Currin` (4 obj), `Hartmann6` (2 obj), `Hartmann6` (4 obj). All results use 8 meta-tasks and 16 observations per meta-task.

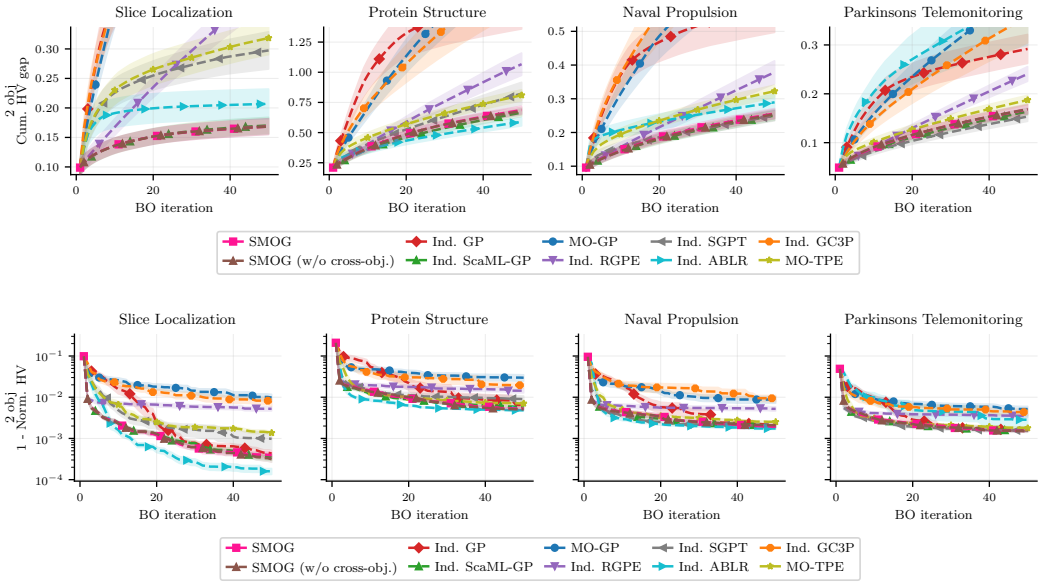


Figure 20: cumulative HV gap (top) and HV gap (bottom; lower is better for both) on the HPOBench benchmarks, including `SMOG (w/o cross-obj.)`. Columns correspond to the four target datasets. All results use 3 meta-tasks and 16 observations per meta-task.

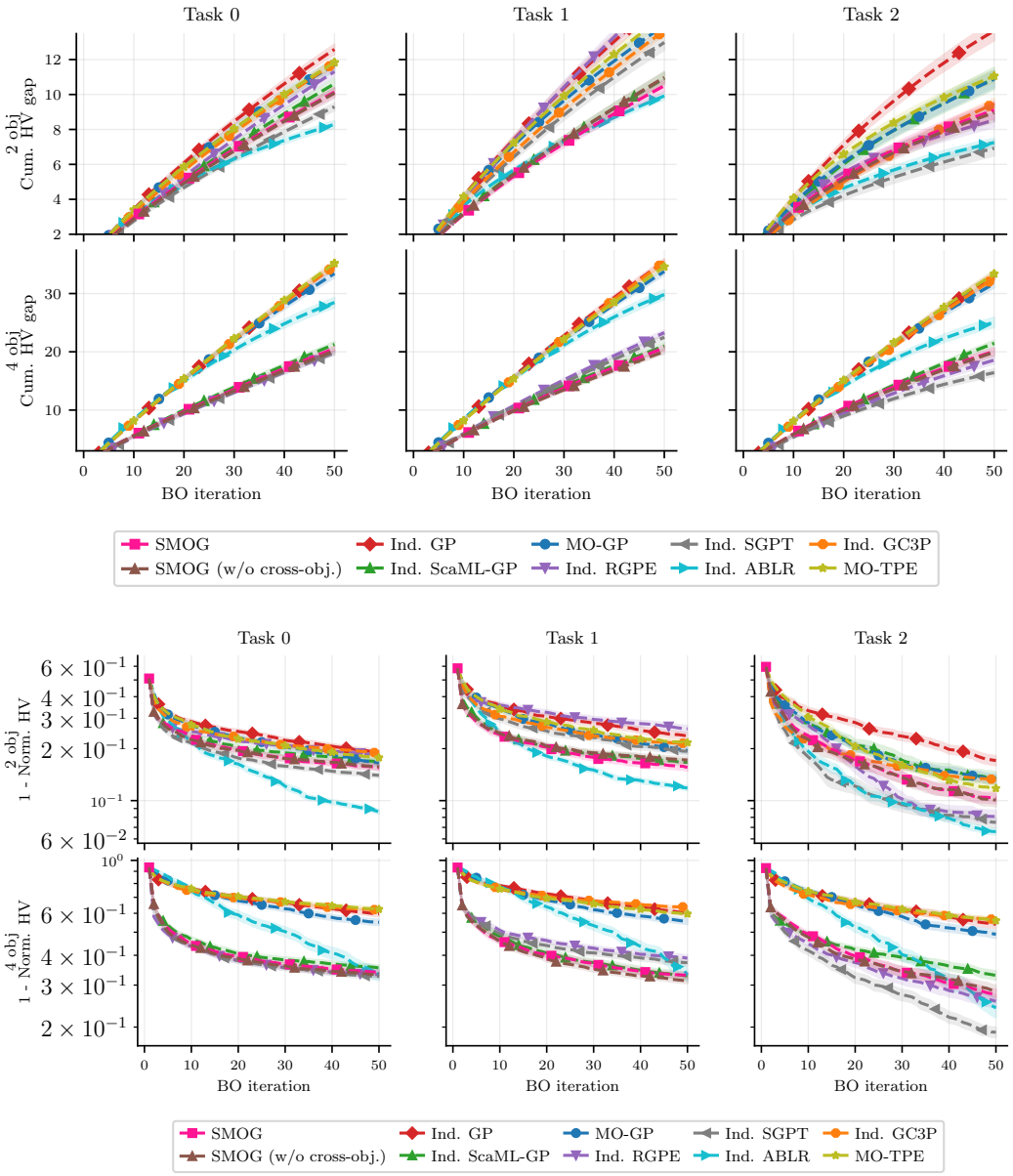


Figure 21: cumulative HV gap (top) and HV gap (bottom; lower is better for both) on the Terrain benchmark, including `SMOG (w/o cross-obj.)`. In each panel group, rows correspond to the 2-objective (top) and 4-objective (bottom) variants; columns correspond to target tasks 0, 1, and 2.

F.8 Comparison to NSGA-III and DGEMO

We compare `SMOG` to the reference-direction-based evolutionary algorithm `NSGA-III` [42] and the diversity-guided batch multi-objective Bayesian optimization (MOBO) method `DGEMO` [43], which we run with batch size five (`DGEMO (batch)`) and one (`DGEMO (seq.)`). Both are established multi-objective optimizers without meta-learning.

`NSGA-III` and `DGEMO` select points with their own mechanisms rather than `LogEHVI`, and they start from a larger random initial design, which counts toward their cumulative HV gap: one population of 11 (2 objectives) or 10 (4 objectives) points for `NSGA-III`, 10 points for `DGEMO (batch)`, and 2 points for `DGEMO (seq.)`. The public `DGEMO` implementation supports at most three objectives, so we run it only in the 2-objective settings.

We use the NSGA-III implementation of pymoo [52] with its default operators and Das–Dennis reference directions (10 partitions for 2 objectives, 2 for 4 objectives). We set the population size to the number of reference directions (11 and 10, respectively) so that several generations fit within the 50-evaluation budget, and we truncate the last generation to the remaining budget. For DGEMO, we use the authors' implementation[4]. DGEMO (batch) evaluates 10 initial points followed by 8 batches of 5; DGEMO (seq.) evaluates the minimum of 2 initial points followed by 48 batches of size 1. In the latter setting, the implementation raised errors in its graph-cut step for some seeds; we report the completed runs ($n = 38$–49 per setting).

Figures 22–24 add NSGA-III and DGEMO to the results on the synthetic, HPOBench, and Terrain benchmarks. SMOG accumulates significantly less cumulative HV gap than NSGA-III in all 14 settings and than DGEMO in all nine 2-objective settings (two-sided Wilcoxon signed-rank test, Holm-corrected).

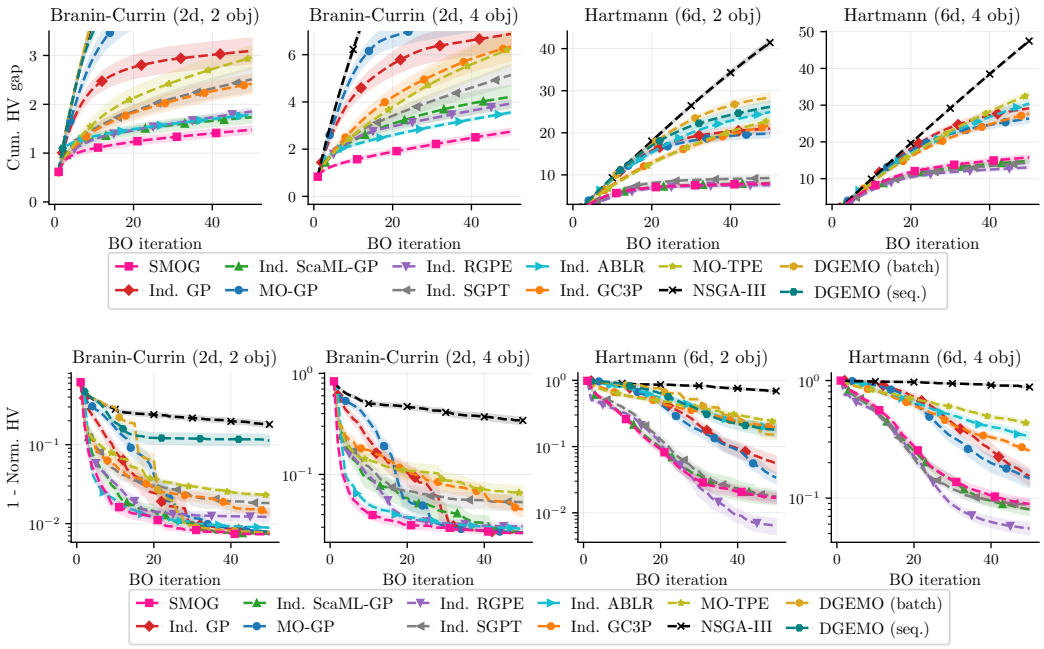


Figure 22: cumulative HV gap (top) and HV gap (bottom; lower is better for both) on the synthetic benchmarks, including NSGA-III, DGEMO (batch), and DGEMO (seq.). Columns correspond to Branin-Currin (2 obj), Branin-Currin (4 obj), Hartmann6 (2 obj), Hartmann6 (4 obj). DGEMO is run only in the 2-objective settings. All meta-learning results use 8 meta-tasks and 16 observations per meta-task.

[4] https://github.com/yunshengtian/DGEMO

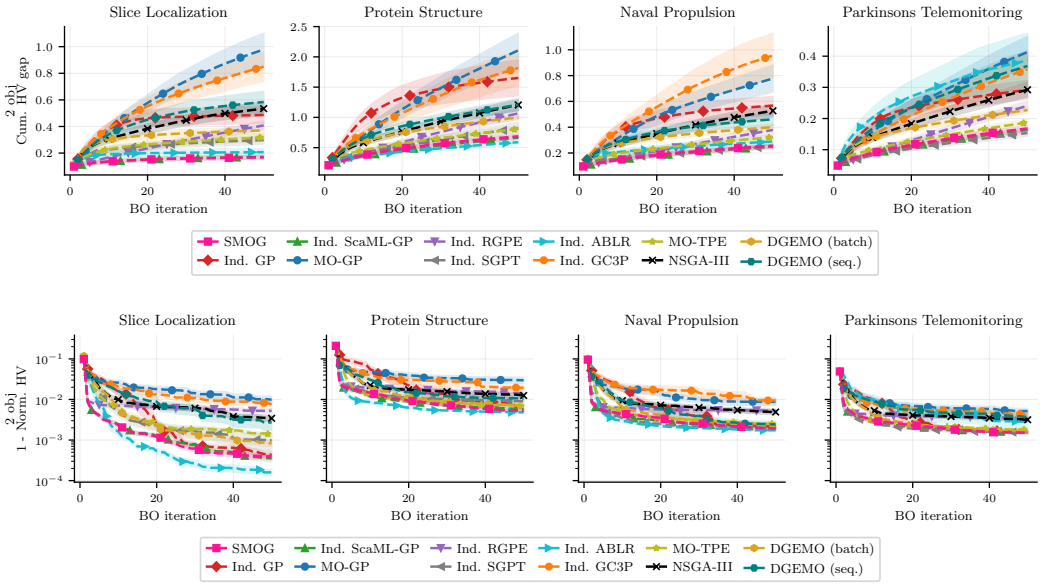


Figure 23: cumulative HV gap (top) and HV gap (bottom; lower is better for both) on the HPOBench benchmarks, including `NSGA-III`, `DGEMO (batch)`, and `DGEMO (seq.)`. Columns correspond to the four target datasets. `DGEMO` is run only in the 2-objective settings. All meta-learning results use 3 meta-tasks and 16 observations per meta-task.

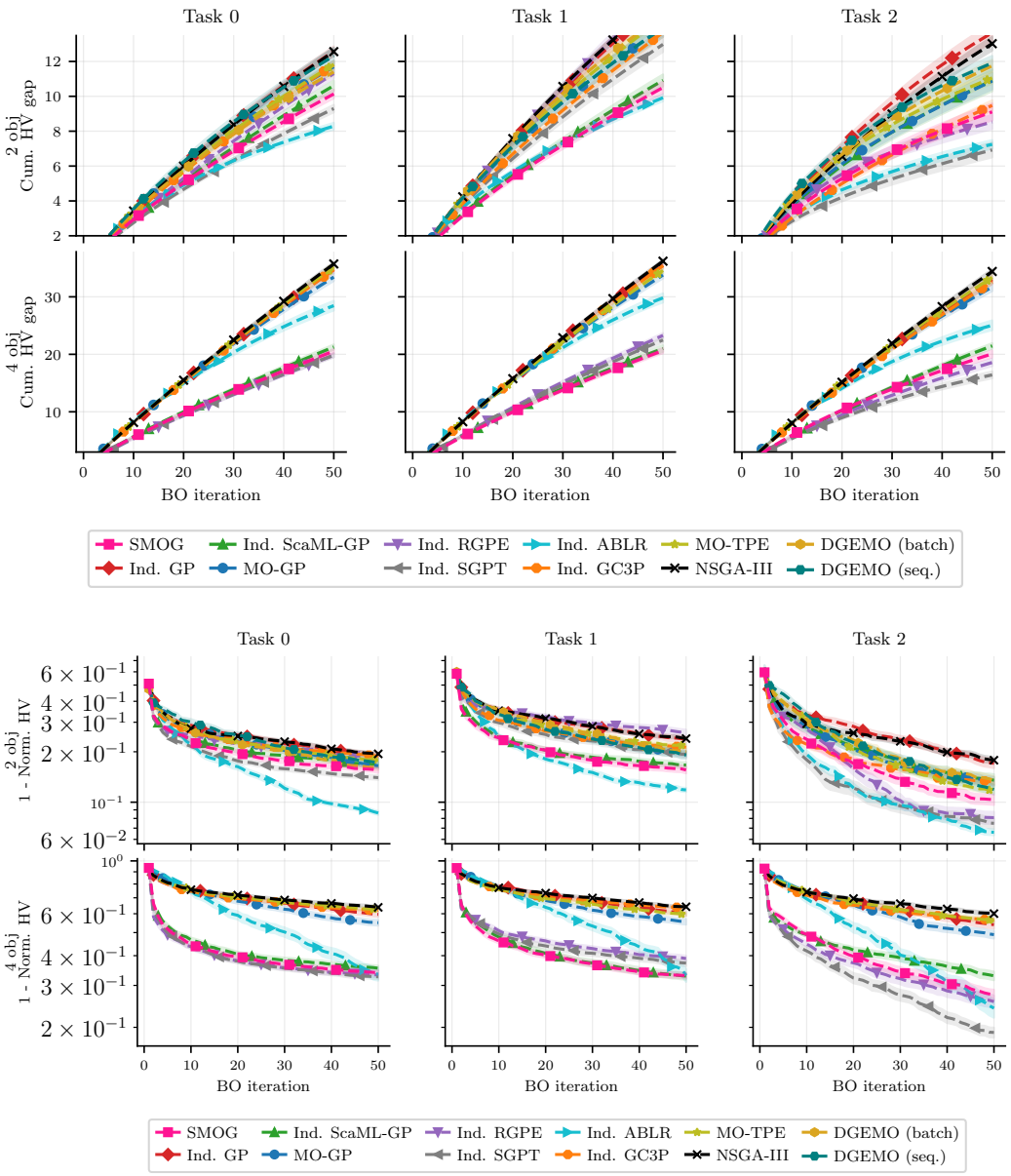


Figure 24: cumulative HV gap (top) and HV gap (bottom; lower is better for both) on the Terrain benchmark, including `NSGA-III`, `DGEMO (batch)`, and `DGEMO (seq.)`. In each panel group, rows correspond to the 2-objective (top) and 4-objective (bottom) variants; columns correspond to target tasks 0, 1, and 2. `DGEMO` is run only in the 2-objective settings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the three main contributions (fully Bayesian uncertainty propagation, learning cross-objective correlations, linear scaling in meta-tasks), all of which are substantiated by Theorem 1 and Lemmas 1 and 2 and the empirical evaluation in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations paragraph in the Conclusion section discusses the independence and no-feedback modularization assumptions, the choice of acquisition functions, number of objectives, cubic scaling in target observations, and the shared-objective-space requirement.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All three formal results (Lemma 1, Lemma 2, Theorem 1) are stated with their explicit assumptions (Assumptions 1–3) and proved in Appendix A and B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Algorithm 1 and Section 4 describe the experimental setup, and Appendix C provides full benchmark specifications, model hyperparameters, and implementation details sufficient to replicate the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available under `https://github.com/boschresearch/scalable-meta-learning-for-mobo`

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes the experimental protocol; Appendix C contains complete hyperparameter settings, meta-dataset construction, benchmark suite details, and the optimizer used for acquisition function optimization.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results are reported as mean $\pm$ standard error across 50 independent random seeds; this is stated in Section 4 and visible in all figures and tables.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D reports per-benchmark wall-clock runtimes and peak memory usage on Intel Xeon Cascade Lake CPUs; experiments were CPU-only.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This is foundational machine learning research on Bayesian optimization; it involves no human subjects, no sensitive data, and no high-risk applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: A dedicated Broader Impact paragraph in the Conclusion section discusses the positive impact of improved sample efficiency (reducing resource waste in scientific and engineering optimization) and notes the absence of foreseeable negative societal risks for this foundational algorithm research.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper presents a Bayesian optimization algorithm; it does not release pretrained generative models, scraped datasets, or any other assets with high misuse potential.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All benchmarks and datasets used (HPOBench, Terrain, MetaBox) are cited with their original papers; their publicly available licenses permit academic research use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new datasets, models, or other assets are released at submission time; code will be released upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or any research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects research; no IRB approval is required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as any important, original, or non-standard component of the proposed method or experiments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.