Enhancing Bayesian Optimization and Active Learning Through Kernel Diversity

Heng Zhang[1] Haotian Xiang[1] Qin Lu[1] Konstantinos D. Polyzos[2] Tara Javidi[2]
[1]University of Georgia, Athens, GA, USA
[2]University of California San Diego, La Jolla, CA, USA
hz39726@uga.edu haotian.xiang@uga.edu qin.lu@uga.edu
kpolyzos@ucsd.edu tjavidi@ucsd.edu

Abstract

Hyperparameter selection remains a key challenge in Bayesian optimization (BO) and Bayesian active learning (AL), as model misspecification can lead to suboptimal performance, while more accurate fully Bayesian treatments typically rely on computationally expensive MCMC sampling. This paper proposes a unified framework, KENDO (Kernel ENsemble Disagreement-aware Operator), that integrates Ensemble Gaussian Processes (EGP) with disagreement-aware acquisition strategies. The central idea is to replace hyperparameter sampling with a kernel ensemble and adaptive Bayesian weighting, combined with disagreement-aware acquisition strategies. Within this unified framework, we instantiate KENDO-BO for BO and KENDO-AL for Bayesian AL, demonstrating that both arise from a common self-correcting mechanism with task-specific acquisition objectives. We further extend the approach to multi-objective optimization via random scalarization that preserves the single-optimizer conditioning structure. Thorough numerical tests on synthetic and real-world benchmarks across single-objective optimization, multi-objective optimization, and active learning demonstrate that (i) KENDO-BO achieves competitive or superior optimization performance compared to state-of-the-art methods while reducing computational overhead by up to 5× and (ii) KENDO-AL achieves superior predictive calibration over MCMC-based active learning baselines with up to 27× speedup.

1 Introduction

Bayesian optimization (BO) and active learning (BAL) have become standard approaches for optimizing expensive black-box functions [9, 33] and for learning unknown functions in an uncertainty-aware, label-efficient fashion respectively. By constructing a probabilistic surrogate model, typically a Gaussian process (GP), both paradigms quantify uncertainty to guide efficient sequential sampling. However, the performance of GP-based BO and AL critically depends on hyperparameter selection, including kernel choice and lengthscale parameters [34]. Traditional approaches handle hyperparameters through point estimation via marginal likelihood maximization [30] or full Bayesian treatment via MCMC sampling [34]. Point estimation is computationally efficient but vulnerable to model misspecification and overfitting, especially in early optimization stages. Conversely, MCMC-based fully Bayesian methods account for hyperparameter uncertainty but incur prohibitive computational costs, particularly when sampling from multimodal or high-dimensional posteriors [35].

Beyond hyperparameter tuning, the choice of kernel family itself is a critical source of misspecification. For example, in the BO paradigm, as Figure 1 illustrates, no single kernel is universally optimal for BO: the best-performing kernel differs across problems, RBF on Rosenbrock (2D), Matérn-1.5 on

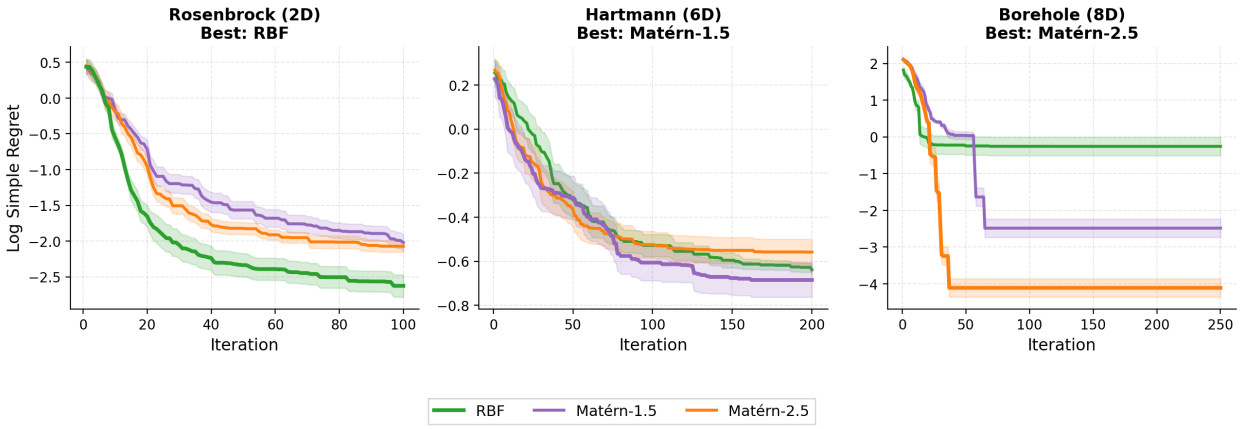


Figure 1: Single-kernel BO on three benchmarks for PES method, reported as $\log_{10}$ simple regret. No single kernel is universally optimal.

Hartmann (6D), and Matérn-2.5 on Borehole (8D) in $\log_{10}$ simple regret, motivating an ensemble that adaptively weights multiple kernels based on observed data. Recent work have explored ensemble-based alternatives. Self-Correcting Bayesian Optimization (SCoreBO) [18] introduces an acquisition function that conditions on sampled optimizers to balance hyperparameter learning with optimization, but still relies on hyperparameter sampling. Ensemble Gaussian Processes (EGP) [23, 24, 28] maintain multiple GPs with different kernels and adaptively weight them via Bayesian model averaging, avoiding sampling entirely. Nonetheless, these approaches adaptively select a single GP model and couple it with standard acquisition strategies, thereby overlooking the benefits from interaction and disagreement among GP models during the acquisition phase. To the best of our knowledge, using EGPs as surrogate models in place of a fully Bayesian treatment, and integrating them with appropriate related acquisition strategies, remains largely unexplored.

Our Contributions. We propose KENDO (Kernel ENsemble Disagreement-aware Operator), which unifies the strengths of ensemble modeling and self-correcting optimization:

- We replace SCoreBO's hyperparameter sampling with EGP's explicit kernel ensemble, eliminating MCMC overhead while preserving uncertainty quantification through kernel diversity and adaptive weighting.
- We derive a computationally efficient acquisition function that measures disagreement between marginal and optimizer-conditioned ensemble posteriors, balancing exploration in promising regions with model selection.
- We extend the framework to multi-objective optimization via random scalarization, which preserves the single-optimizer conditioning structure of SCoreBO while covering the Pareto front through diverse scalarization directions, avoiding the computational burden of multi-objective solvers.
- We introduce KENDO-AL, demonstrating that the ensemble mechanism generalizes beyond optimization to Bayesian active learning, achieving competitive results over MCMC-based approaches.
- Through extensive experiments and ablation studies, we demonstrate that the ensemble mechanism achieves substantial computational savings (up to 5× for BO and 27× for active learning) without sacrificing optimization performance.

2 Related Work

2.1 Gaussian Processes and Hyperparameter Learning

Gaussian processes provide a flexible non-parametric framework for probabilistic regression [30], in which the choice of kernel function and its hyperparameters fundamentally determines the inductive bias of the surrogate model. Standard practice optimizes these hyperparameters by maximizing the marginal likelihood [30], but the resulting point estimate neglects uncertainty and can produce overconfident predictions [34]. Fully Bayesian alternatives integrate over hyperparameter posteriors via MCMC [34] or variational inference [12, 37], and scalable variants such as slice sampling [26] and Hamiltonian Monte Carlo [15] improve robustness at the cost of substantial computational overhead where the surrogate must be refitted at every iteration. A complementary line of work addresses model uncertainty through ensembles rather than integration. Multi-kernel learning [6] constructs expressive composite kernels and deep kernel learning [40] parameterizes kernels via neural networks, while Ensemble Gaussian Processes [23] explicitly maintain a weighted collection of GPs with different kernels and update the weights via Bayesian model averaging based on predictive likelihood. This approach has shown promise but has not been integrated with modern BO or active learning acquisition functions that leverage optimizer information.

2.2 Acquisition Functions for Bayesian Optimization and Active Learning

A unifying theme connecting modern acquisition functions for both Bayesian optimization and Bayesian active learning is the use of *disagreement* as the exploration signal, where queries are placed at points on which competing posterior beliefs disagree most strongly about the prediction. SCoreBO [18] instantiates this idea in BO by conditioning on sampled optimizers and computing the Hellinger distance between the marginal posterior and each optimizer-conditional posterior in promising regions, although it inherits the heavy computational cost of hyperparameter sampling. Other BO methods follow alternative routes that do not exploit disagreement, including GP-UCB with fixed confidence bounds [36], entropy-search variants that integrate over functions [11, 14, 39], and meta-learning approaches that transfer hyperparameter knowledge across related tasks [8]. The same disagreement principle has long driven Bayesian active learning (BAL), where the goal shifts from finding optima to minimizing global prediction error [32]. Bayesian Active Learning by Disagreement (BALD) [16, 19] maximizes the mutual information between predictions and hyperparameters, Bayesian Query-by-Committee (BQBC) [31] measures variance in posterior means across hyperparameter samples, and Statistical distance-based Active Learning (SAL) [18] generalizes BQBC using Hellinger and Wasserstein distances to capture the full distributional disagreement induced by hyperparameter uncertainty. SAL thus plays the same role in active learning as SCoreBO does in optimization, and both rely on MCMC sampling to quantify hyperparameter uncertainty, an overhead that directly motivates our ensemble-based alternative.

2.3 Multi-Objective Bayesian Optimization

Multi-objective BO (MOBO) aims to approximate the Pareto front of multiple conflicting objectives [20], and standard approaches include expected hypervolume improvement [7, 3], ϵ-indicator-based methods [43], and information-theoretic acquisitions [13, 38, 1]. More recent work explores scalarization strategies [27] and uncertainty-aware acquisition functions [2], yet hyperparameter learning in multi-objective settings remains largely underexplored, as most existing methods either fix hyperparameters in advance or tune them independently for each objective without accounting for the coupled uncertainty across objectives.

3 Preliminaries

Both BO and BAL are sequential decision-making paradigms that deal with a *black-box* function f, which has no analytic expression and is also expensive to evaluate. While BO aims to seek the optimizer of f, namely,

$$\mathbf{x}_* = \arg\max_{\mathbf{x}\in\mathcal{X}} \quad f(\mathbf{x}) \tag{1}$$

BAL focuses on learning a good mapping for the target function f for every $\mathbf{x} \in \mathcal{X}$.

Specifically, BO and BAL rely on a probabilistic surrogate model that guides the judicious selection of query points sequentially, which are implemented iteratively in the two steps, **s1)** obtain $p(f(\mathbf{x})|\mathcal{D}_t)$ ($\mathcal{D}_t := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^t$) based on a chosen surrogate model; and **s2)** select $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}|\mathcal{D}_t)$, where the acquisition function (AF) α, usually available in closed form, is designed based on $p(f(\mathbf{x})|\mathcal{D}_t)$. For BO, the AF seeks to strike a balance between *exploration* and *exploitation*, while the AF in BAL is designed mainly for *exploration*.

3.1 GP-based surrogate models

Capable of learning function mappings with well-calibrated uncertainty values, GPs are the de facto choice for surrogate models in BO and BAL [30, 9]. In the GP context, the learning function f is assumed to be drawn from a GP prior, $f(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}'))$, where $\kappa_{\boldsymbol{\theta}}$ is the positive-definite *kernel* function with learnable hyperparameters $\boldsymbol{\theta}$ that measures pairwise similarity between any two distinct points $\mathbf{x}$, and $\mathbf{x}'$ and is used to capture the covariance between function values $f(\mathbf{x})$, $f(\mathbf{x}')$. Given $\mathcal{D}_t$, a typical process is to maximize the so-termed marginal likelihood $p(\mathcal{D}_t; \boldsymbol{\theta})$ to obtain a *point* estimate of $\boldsymbol{\theta}$. Going beyond the point estimation of the kernel hyperparameters, the *fully Bayesian* GP views $\boldsymbol{\theta}$ as random and maintains a posterior pdf for $\boldsymbol{\theta}$ via Bayes' rule as: $p(\boldsymbol{\theta}|\mathcal{D}_t) \propto p(\boldsymbol{\theta})p(\mathcal{D}_t; \boldsymbol{\theta})$. Although accounting for overfitting, such a fully Bayesian treatment entails computationally prohibitive Markov Chain Monte Carlo (MCMC) sampling.

The previous discussion adheres to GPs with a *pre-selected* kernel type. In practice though, this is a nontrivial design choice as different tasks may require different kernel functions. To automate this design, the ensemble (E) GP framework [23], given a prescribed kernel dictionary $\mathcal{K} := \{\kappa_1, \ldots, \kappa_M\}$, learns the function via a Gaussian mixture prior as:

$$f(\mathbf{x}) \sim \sum_{m=1}^{M} w_0^m \, \mathcal{GP}(0, \kappa_m(\mathbf{x}, \mathbf{x}')), \quad \sum_{m=1}^{M} w_0^m = 1 \tag{2}$$

where w_0^m is the weight corresponding to the mth GP model using kernel κ^m. After observing $\mathcal{D}_t$, the function posterior pdf is a GP mixture

$$p(f(\mathbf{x}) \mid \mathcal{D}_t) = \sum_{m=1}^{M} w_t^m \, \mathcal{N}(f(\mathbf{x}); \mu_{m,t}(\mathbf{x}), \sigma_{m,t}^2(\mathbf{x})) \tag{3}$$

where $p(f(\mathbf{x})|m, \mathcal{D}_t) := \mathcal{N}(f(\mathbf{x}); \mu_{m,t}(\mathbf{x}), \sigma_{m,t}^2(\mathbf{x}))$ is the posterior of GP model m with mean and variance given in closed-form [30]. The per-kernel weight $w_t^m = \mathbb{P}(m|\mathcal{D}_t)$ is proportional to the marginal likelihood

$$w_t^m \propto p(\mathcal{D}_t|m; \hat{\boldsymbol{\theta}}_m) \tag{4}$$

with $\hat{\boldsymbol{\theta}}_m$ being the estimated hyperparameters for kernel κ^m by maximizing the marginal likelihood.

3.2 Disagreement-based AF

Based on different design rules, a number of AFs have been designed for BO and BAL tailored to the chosen surrogate model. Most recently, leveraging the hyperparameter-induced disagreement conditioned on optimizer samples, a novel AF, termed as SCoreBO [18], is devised for the fully Bayesian GP model, defined as

$$\alpha_{\mathrm{SC}}(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\theta},*}[d(p(y_{\mathbf{x}} \mid \mathcal{D}), p(y_{\mathbf{x}} \mid \boldsymbol{\theta}, *, \mathcal{D}))] \tag{5}$$

where $\boldsymbol{\theta}$ are hyperparameters sampled from $p(\boldsymbol{\theta} \mid \mathcal{D})$, $* = (\mathbf{x}^*, f^*)$ is a sampled optimizer, and $d(\cdot, \cdot)$ is the Hellinger distance. This balances exploration in high-potential regions with hyperparameter learning. Unlike TS or EI that solely exploit the current posterior, α_{SC} also rewards queries that reduce hyperparameter uncertainty, which is critical when the surrogate is misspecified in early iterations.

4 Exploiting Kernel Diversity to Enhance BO and BAL

While SCoreBO relies on the diversity of kernel hyperparameters, it still presumes a single pre-selected kernel type and uses computationally heavy MCMC sampling. To further account for the effect of kernel family selection and to eliminate the computational burden of MCMC, we propose to replace hyperparameter sampling with an explicit kernel ensemble. The core idea lies on the disagreement across structurally different kernels instead of the disagreement across hyperparameter samples of a single kernel. We develop this idea in three stages: first establishing the ensemble-based surrogate (Section 4.1), then deriving the optimizer-conditioned acquisition function (Section 4.2) for (multi-objective) black-box optimization, and finally extending to active learning (Section 4.3).

4.1 From Hyperparameter Sampling to Kernel Ensembles

The central idea of KENDO is to replace SCoreBO's continuous hyperparameter sampling with EGP's discrete kernel ensemble. Each GP $m \in \{1, \ldots, M\}$ relies on a fixed kernel κ_m whose hyperparameters are fitted via marginal likelihood maximization, together with a dynamic weight w_t^m updated via (4). This substitution offers two advantages: (i) it eliminates MCMC sampling overhead; and (ii) it captures a qualitatively different and more consequential axis of model uncertainty, namely the choice of kernel family, without relying on a specific pre-defined kernel form. The Bayesian weight update requires only M predictive likelihood evaluations per iteration, compared to the $\mathcal{O}(100)$–$\mathcal{O}(1000)$ posterior samples typically needed for MCMC convergence.

To make the ensemble tractable so as to be readily used as input to acquisition functions, we approximate the Gaussian mixture posterior (3) via moment matching as

$$\bar{\mu}_t(\mathbf{x}) = \sum_{m=1}^{M} w_t^m \, \mu_{m,t}(\mathbf{x}) \tag{6}$$

$$\bar{\sigma}_t^2(\mathbf{x}) = \sum_{m=1}^{M} w_t^m \left[\sigma_{m,t}^2(\mathbf{x}) + (\mu_{m,t}(\mathbf{x}) - \bar{\mu}_t(\mathbf{x}))^2\right] \tag{7}$$

where $\bar{\mu}_t(\mathbf{x})$ is the ensemble mean for any $\mathbf{x}$ and $\sigma_{m,t}(\mathbf{x})$ is the associated variance. The variance expression decomposes into within-GP uncertainty (first term) and between-GP disagreement (second term), providing a single Gaussian summary that retains ensemble-level information.

Algorithm 1 KENDO-BO / KENDO-AL

Require: Kernel dict. $\mathcal{K}$, data $\mathcal{D}_0$, budget T *// For BO: #optimizers N; for AL: validation set $\mathcal{V}$*

for $m \in \mathcal{K}$ **do**
  Fit α_m via $\max p(\mathcal{D}_0 | i = m, \alpha)$; Init $w_0^m = 1/M$
end for
for $t = 0, \ldots, T-1$ **do**
  Compute $\bar{\mu}_t(\cdot), \bar{\sigma}_t^2(\cdot)$ via (6)–(7)
  // BO branch: sample optimizers and condition
  for $m \in \mathcal{K}$; $n = 1, \ldots, N$ **do**
    Sample $f_{m,n} \sim p(f | m, \mathcal{D}_t)$; Find $x^*_{m,n} = \arg\max f_{m,n}(\mathbf{x})$
    Condition GP m on $(x^*_{m,n}, f^*_{m,n})$ to get $\mu^*_{m,n}, \sigma^{2,*}_{m,n}$
  end for
  // BO: $\alpha(\mathbf{x}) = \sum_m \sum_n \frac{w_t^m}{N} d_{m,n}(\mathbf{x})$ via (8)–(9)
  // AL (skip lines 7–9): $\alpha(\mathbf{x}) = \sum_m w_t^m \cdot d_m(\mathbf{x})$ via (10)–(11)
  $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x})$
  Evaluate $f(\mathbf{x}_{t+1})$; $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\mathbf{x}_{t+1}, y_{t+1})\}$
  for $m \in \mathcal{K}$ **do**
    Update w_{t+1}^m via (4); Update GP posterior mean and variance via (23a)–(23b) (supplementary)
  end for
  // AL only: compute MLL on $\mathcal{V}$
end for
return *BO:* $\arg\max_{(\mathbf{x},y)\in\mathcal{D}_T} y$ *AL:* trained model with $\mathcal{D}_T$

4.2 Optimizer-Conditioned AF

With a tractable Gaussian summary of the ensemble posterior, we will subsequently utilize this ensemble to decide where to query next. We adapt the key idea of conditioning on sampled optimizers and disagreement across hyperparameter samples of a single kernel, to the ensemble setting. The intuition is that if knowing the location of the optimum would cause different kernels to revise their predictions differently at $\mathbf{x}$, then querying $\mathbf{x}$ is informative for both optimization and model selection. We define the acquisition function of KENDO-BO. For each kernel m and sample index n, we draw a function realization $f_{m,n}$ from the posterior of GP m, locate its optimizer $\mathbf{x}^*_{m,n} = \arg\max_x f_{m,n}(\mathbf{x})$ with value $f^*_{m,n}$, and condition GP m on $(\mathbf{x}^*_{m,n}, f^*_{m,n})$ to obtain a conditional posterior $\mathcal{N}(\mu^{*_{m,n}}_{m,t}(x), \sigma^{2,*_{m,n}}_{m,t}(\mathbf{x}))$. The acquisition function then aggregates the disagreement between the marginal ensemble posterior and each conditional posterior, yielding (see Appendix B for the full derivation)

$$\alpha(\mathbf{x}) = \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{w_t^m}{N} \cdot d_{m,n}(\mathbf{x}) \tag{8}$$

where $d_{m,n}(\mathbf{x})$ is the Hellinger distance between the moment-matched marginal ((6)–(7)) and the conditional posterior of kernel m given optimizer sample n. For two Gaussians, this admits a closed-form expression

$$d_{m,n}(\mathbf{x}) = 1 - \sqrt{\frac{2\bar{\sigma}_t(\mathbf{x})\sigma^{*_{m,n}}_{m,t}(\mathbf{x})}{\bar{\sigma}_t^2(\mathbf{x}) + \sigma^{2,*_{m,n}}_{m,t}(\mathbf{x})}} \exp\left[-\frac{(\bar{\mu}_t(\mathbf{x}) - \mu^{*_{m,n}}_{m,t}(\mathbf{x}))^2}{4(\bar{\sigma}_t^2(\mathbf{x}) + \sigma^{2,*_{m,n}}_{m,t}(\mathbf{x}))}\right] \tag{9}$$

Intuitively, $\alpha(\mathbf{x})$ is large at locations where knowing the optimizer would substantially revise the ensemble's prediction. Algorithm 1 summarizes the full procedure.

4.3 Extensions to Active Learning

The optimizer conditioning in KENDO-BO focuses on data collection near promising optima. For active learning, where the goal is global function approximation, we simply remove this conditioning. The KENDO-AL acquisition function measures the disagreement between the ensemble marginal and each component directly

$$\alpha_{\text{SAL}}(x) = \sum_{m=1}^{M} w_t^m \cdot d(p(y_x \mid \mathcal{D}_t),\, p(y_x \mid i = m, \mathcal{D}_t)) \tag{10}$$

where the Hellinger distance between the moment-matched marginal and each kernel's posterior takes the form

$$d_m(x) = 1 - \sqrt{\frac{2\bar{\sigma}_t(x)\sigma_{m,t}(x)}{\bar{\sigma}_t^2(x) + \sigma_{m,t}^2(x)}} \exp\left[-\frac{(\bar{\mu}_t(x) - \mu_{m,t}(x))^2}{4(\bar{\sigma}_t^2(x) + \sigma_{m,t}^2(x))}\right] \tag{11}$$

This formulation prioritizes locations where kernel identity has maximal impact on predictions. Unlike MCMC-based SAL [18], KENDO-AL avoids sampling overhead while retaining the ability to measure full distributional disagreement. The method handles both outputscale uncertainty (driving global exploration) and lengthscale uncertainty (encouraging repeated queries for noise estimation), as each kernel's contribution is weighted by its predictive likelihood. The unified process is shown in Algorithm 1.

5 KENDO for MOBO

In this section, we extend the KENDO framework to the multi-objective Bayesian optimization (MOBO) setting, where K (possibly conflicting) objectives $\{f^k\}_{k=1}^K$ are optimized simultaneously. Unlike the single-objective case, the solution is no longer a single optimizer but the Pareto set $\mathcal{X}^* = \{\mathbf{x}^* \in \mathcal{X} : \nexists \mathbf{x} \in \mathcal{X} \text{ s.t. } \mathbf{f}(\mathbf{x}) \succ \mathbf{f}(\mathbf{x}^*)\}$, $\mathbf{f}(\mathbf{x}) := [f^1(\mathbf{x}), \ldots, f^K(\mathbf{x})]^\top$, formed under the Pareto dominance relation $\succ$, where $\mathbf{f}(\mathbf{x}) \succ \mathbf{f}(\mathbf{x}^*)$ denotes that $f^k(\mathbf{x}) \geq f^k(\mathbf{x}^*)$ for all k with strict inequality for at least one k in maximization problem. This relation induces the Pareto front $\mathcal{P}^* = \{\mathbf{f}(\mathbf{x}^*) : \mathbf{x}^* \in \mathcal{X}^*\}$ in objective space, and the discovered front is commonly evaluated against $\mathcal{P}^*$ via the hypervolume indicator with respect to a reference point [7]. Therefore, two design choices of KENDO-BO are key in the MOBO setting. The kernel ensemble needs to be instantiated per objective, allowing different f^k to prefer different kernels in $\mathcal{K}$, and the optimizer conditioning that drives the disagreement signal in (8) must be extended to the Pareto setting, where the solution is inherently a set rather than a single point. For K objectives $\{f^k\}_{k=1}^K$, we maintain independent EGPs per objective with weights $w_t^{k,m}$. Extending to multi-objective optimization, however, requires careful treatment, as the notion of a single "optimizer" no longer applies; the solution is a Pareto front rather than a single point. A direct extension would condition on sampled Pareto fronts, but this introduces $P_{m,n}$ phantom observations simultaneously, diffusing the conditioning signal across the input space and fundamentally departing from the single-optimizer mechanism that makes SCoreBO effective. Our experiments confirm that this yields suboptimal performance (Section 6.3).

Instead, we propose a random scalarization strategy that preserves the single-optimizer structure. At each iteration, we sample S weight vectors $\boldsymbol{\lambda}_s \sim \text{Dir}(\mathbf{1}_K)$ from a symmetric Dirichlet distribution. For each kernel m and scalarization s, we draw function paths $f_{m,s}^k$ for all objectives via Random Fourier features (RFF) [29, 41], form the scalarized objective $g_{m,s}(\mathbf{x}) = \sum_k \lambda_{s,k} f_{m,s}^k(\mathbf{x})$, and find its single optimizer $\mathbf{x}_{m,s}^*$. We then condition each per-objective GP on the phantom observation $(\mathbf{x}_{m,s}^*, f_{m,s}^{*,k})$ where $f_{m,s}^{*,k} = f_{m,s}^k(\mathbf{x}_{m,s}^*)$, and aggregate per-objective Hellinger distances

for the acquisition step as follows:

$$\alpha(\mathbf{x}) = \sum_{k=1}^{K} \sum_{m=1}^{M} w_t^{k,m} \cdot \frac{1}{S} \sum_{s=1}^{S} d\Big(\bar{p}(y_x^k \mid \mathcal{D}_t),\ p(y_x^k \mid i{=}m,\ (\mathbf{x}_{m,s}^*, f_{m,s}^{*,k}),\ \mathcal{D}_t)\Big) \tag{12}$$

where $\bar{p}(y_x^k \mid \mathcal{D}_t)$ is the moment-matched marginal for objective k and $d(\cdot,\cdot)$ is the Hellinger distance (9).

This design preserves the focused, single-point conditioning that makes KENDO-BO effective, while a sufficient diversity of scalarization directions $\{\boldsymbol{\lambda}_s\}$ provides coverage of the full Pareto front; Appendix C.5 confirms that the framework is robust to the choice of Dirichlet concentration controlling this diversity. Finding each scalarized optimizer requires only single-objective optimization via L-BFGS on the RFF path, avoiding multi-objective solvers such as NSGA-II [4]. The full process is summarized in Algorithm 2 in the supplementary file.

6 Experiments

6.1 Experimental Setup

All EGP-based methods use $\mathcal{K} = \{\text{RBF}, \text{Matérn-1.5}, \text{Matérn-2.5}\}$ with per-kernel marginal-likelihood fitting. To ensure fair comparison, every method (including baselines) uses the same GP priors: GammaPrior$(3.0, 6.0)$ on lengthscale and GammaPrior$(2.0, 0.15)$ on outputscale. For single-objective BO, we compare against NEI [21], PES [14], GIBBON (MES) [25], JES [17], the MCMC-based SCoreBO [18], and EGP-TS [24] that uses the same kernel ensemble as ours but replaces the disagreement-based acquisition with standard Thompson sampling. For multi-objective BO, baselines include JESMO [38], MESMO [1], PES(MO), qNEHVI [3], EGP-TS-MOBO, and KENDO-MO-PF (our Pareto-front-conditioning variant from Section 5, included to validate the scalarization design). For active learning, we compare against BALD [16], BQBC [31], QBMGP [31], and SAL [18]; all SAL variants use Hellinger distance.

The benchmarks used for evaluation are categorized as follows. *Single-objective BO:* Branin (2D), Rosenbrock (2D, 4D), Hartmann (3D, 4D, 6D), and three additional real world problems, RobotPush (3D), RF-HPO on Adult (4D), and Borehole (8D), to cover synthetic, simulator, and HPO problems. *Multi-objective BO:* two synthetic functions: ZDT2 [44] (d=6, K=2), DTLZ2 [5] (d=6, K=3), and four real world problems: VehicleSafety (d=5, K=3), CarSideImpact (d=7, K=3), Penicillin [22] (d=7, K=3), and LCBench [42] (d=7, K=2), reported via $\log_{10}$ hypervolume difference. *Active learning:* Higdon (1D), Branin (2D), Ishigami (3D) and Hartmann (6D), along with three real benchmarks, RobotPush (3D), GBT-HPO (4D) and Airfoil (5D), reported via negative MLL on held-out validation sets, with relative ranking as an aggregate view. All curves are averaged over 25 seeds; results for single-kernel baselines are additionally averaged over the three kernel variants in $\mathcal{K}$.

6.2 Single-Objective Optimization Results

For performance evaluation, the central question is whether substituting MCMC with a discrete kernel ensemble degrades optimization quality. Figure 2 shows that this well-motivated replacement does not degrade optimization performance. On the contrary, across all nine benchmarks, spanning synthetic functions to real problems and low to high dimensional regimes, KENDO-BO matches or outperforms all baselines. The relative-ranking panel confirms that it maintains the top average rank as the budget grows. The comparison against EGP-TS isolates the contribution of the acquisition

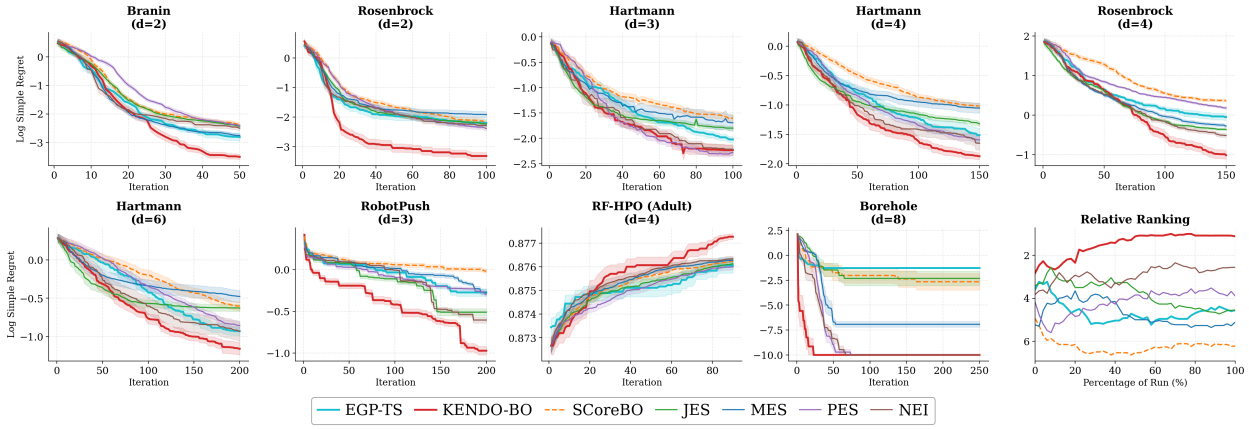


Figure 2: Average $\log_{10}$ simple regret on nine single-objective benchmarks plus the relative ranking panel, averaged over 25 random seeds. Results are averaged over 3 kernel variants. On Borehole, curves saturate at -10 due to a hard floor on $\log_{10}$ regret; KENDO-BO reaches the floor earliest.

function, since both methods share the same kernel ensemble and weighting scheme. KENDO-BO consistently outperforms EGP-TS, indicating that the gain comes from disagreement-based, optimizer-conditioned querying rather than from the ensemble alone.

6.3 Multi-Objective Optimization Results

Figure 3 reports results on six MOBO benchmarks. KENDO-MO achieves the lowest $\log_{10}$ hypervolume difference on the majority of problems, with the largest margins on Penicillin and CarSideImpact, both real-world problems whose objectives have heterogeneous landscapes that benefit from per-objective adaptive weighting. The aggregate ranking panel further shows KENDO-MO on top across the full budget.

In comparison with KENDO-MO-PF, which conditions on sampled Pareto fronts instead of scalarized optimizers, KENDO-MO-PF underperforms KENDO-MO consistently across all six benchmarks. This corroborates what we discuss in Section 5: conditioning on an entire Pareto set introduces many phantom observations at once, and the conditioning signal gets diffused across the input space rather than concentrated where it is informative. Random scalarization keeps the focused, single-point conditioning that makes single-objective KENDO-BO work, while the diversity of $\boldsymbol{\lambda}_s$ ensures the Pareto front is still adequately covered.

6.4 Active Learning Results

Unlike BO, active learning calls for pure exploration, where the goal is global predictive accuracy, not finding an optimum. This objective allows us to test whether the ensemble's kernel diversity provides intrinsic value beyond its role in guiding optimization. Figure 4 evaluates KENDO-AL on seven benchmarks spanning synthetic functions (Higdon, Branin, Ishigami, Hartmann), a simulator task (RobotPush), an HPO problem (GBT-HPO), and an engineering surrogate (Airfoil). KENDO-AL achieves the lowest negative MLL on every benchmark and holds the best average rank in the aggregate panel. The disagreement-based methods (KENDO-AL and SAL) consistently perform well and KENDO-AL further improves on SAL by replacing MCMC over hyperparameters with the kernel ensemble, which bypasses the sensitivity to a pre-selected kernel that QBMGP and BQBC suffer from. These gains come with a 7.6×–26.9× speedup over SAL (Appendix C.2).

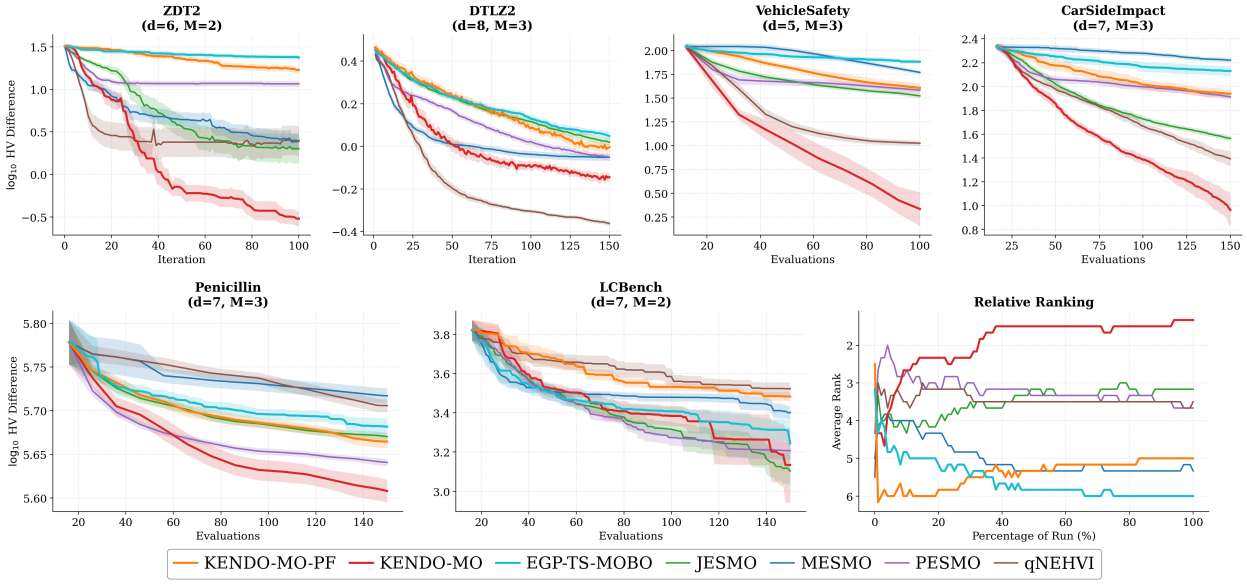


Figure 3: Multi-objective optimization results on six benchmarks plus the relative ranking panel. Reported as $\log_{10}$ hypervolume difference, averaged over 25 random seeds and 3 kernel variants.

6.5 Ablation Studies

We evaluate four design choices of KENDO: adaptive versus uniform weighting, computational efficiency, kernel weight dynamics, and the choice of scalarization for MOBO. Detailed results and discussion are deferred to Appendix C; we summarize the main findings here.

Adaptive vs. uniform weighting. The Bayesian weighting scheme in (4) outperforms uniform averaging across all three task families, with the gap pronounced on benchmarks where the preferred kernel varies across input regions (Appendix C.1).

Computational efficiency. KENDO runs 3.1×–5.4× faster than SCoreBO, and KENDO-AL runs 7.6×–26.9× faster than SAL; the weight update itself contributes less than 7 ms per iteration (Appendix C.2).

Kernel weight convergence. The ensemble exhibits winner-take-all behavior on benchmarks with a clearly superior kernel, and remains diversified when no single kernel dominates (Appendix C.3).

Sensitivity to scalarization design. The MOBO scalarization layer is robust to both the scalarization form and the Dirichlet concentration (Appendices C.4–C.5).

7 Conclusions

The central focus of this paper is kernel family diversity as a source of model uncertainty. We address this by replacing hyperparameter sampling from costly fully Bayesian MCMC based approaches with an explicit kernel ensemble and adaptive Bayesian weighting, that can be readily leveraged for both Bayesian optimization and active learning. Our KENDO-BO and KENDO-AL variants achieve superior optimization and predictive performance over MCMC-based approaches with substantial computational savings. The proposed approach naturally extends to multi-objective optimization via random scalarization, while ablation studies corroborate the effectiveness of adaptive weighting and the automatic model selection capabilities of the ensemble approach.

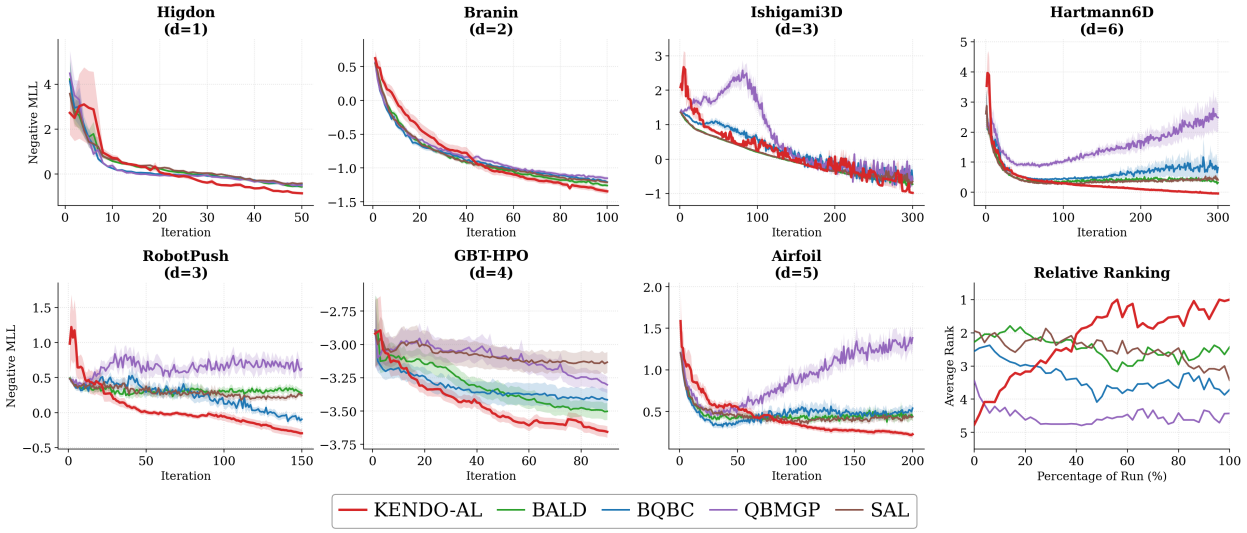


Figure 4: Active learning results on seven benchmarks plus the relative ranking panel. Negative MLL on held-out validation sets, averaged over 25 random seeds and three kernel variants.

Limitations. The current framework relies on moment matching to approximate Gaussian mixture posteriors, which may underestimate uncertainty in cases with highly disparate kernel predictions. Additionally, the kernel dictionary requires manual specification; though automatic construction via kernel composition [6] could be explored.

Future Work. Our future research agenda includes: (1) adaptive kernel dictionary construction during optimization; (2) integration with batch BO for parallel evaluations [10]; (3) extension to constrained optimization with feasibility modeling per kernel; (4) application to hyperparameter tuning of deep neural networks; and (5) investigation of the ensemble mechanism in deep active learning contexts where model capacity exceeds kernel expressiveness.

References

[1] S. Belakaria, A. Deshwal, and J. R. Doppa. Max-value entropy search for multi-objective Bayesian optimization. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 32, 2019.

[2] S. Daulton, M. Balandat, and E. Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 33, 2020.

[3] S. Daulton, M. Balandat, and E. Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 34, 2021.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197, 2002.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proc. Congr. Evol. Comput.*, pages 825–830, 2002.

[6] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum, and G. Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In *Proc. Intl. Conf. Mach. Learn.*, pages 1166–1174. PMLR, 2013.

[7] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Trans. Evol. Comput.*, 10 (4):421–439, 2006.

[8] M. Feurer, J. Springenberg, and F. Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proc. AAAI Conf. Artif. Intell.*, volume 29, 2015.

[9] P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[10] J. Gonzalez, Z. Dai, P. Hennig, and N. Lawrence. Batch Bayesian optimization via local penalization. In *Proc. Intl. Conf. Artif. Intell. Stat.*, volume 51, pages 648–657. PMLR, 2016.

[11] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *J. Machine Learning Res.*, 13:1809–1837, 2012.

[12] J. Hensman, A. G. de G. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *Proc. Intl. Conf. Artif. Intell. Stat.*, volume 38, pages 351–360. PMLR, 2015.

[13] D. Hernández-Lobato, J. M. Hernández-Lobato, A. Shah, and R. P. Adams. Predictive entropy search for multi-objective Bayesian optimization. In *Proc. Intl. Conf. Mach. Learn.*, volume 48, pages 1492–1501. PMLR, 2016.

[14] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 27, 2014.

[15] M. D. Hoffman, A. Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Machine Learning Res.*, 15(47):1593–1623, 2014.

[16] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[17] C. Hvarfner, F. Hutter, and L. Nardi. Joint entropy search for maximally-informed Bayesian optimization. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 35, 2022.

[18] C. Hvarfner, E. O. Hellsten, F. Hutter, and L. Nardi. Self-correcting Bayesian optimization through Bayesian active learning. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 36, 2023.

[19] A. Kirsch, J. Van Amersfoort, and Y. Gal. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 32, 2019.

[20] J. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. Evol. Comput.*, 10(1):50–66, 2006.

[21] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy. Constrained bayesian optimization with noisy experiments. 2019.

[22] Q. Liang and L. Lai. Scalable Bayesian optimization accelerates process optimization of penicillin production. In *NeurIPS 2021 AI for Science Workshop*, 2021.

[23] Q. Lu, G. V. Karanikolas, Y. Shen, and G. B. Giannakis. Ensemble Gaussian processes with spectral features for online interactive learning with scalability. In *Proc. Intl. Conf. Artif. Intell. Stat.*, volume 108, pages 1910–1920. PMLR, 2020.

[24] Q. Lu, K. D. Polyzos, B. Li, and G. B. Giannakis. Surrogate modeling for bayesian optimization beyond a single gaussian process. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(9):11283–11296, 2023.

[25] H. B. Moss, D. S. Leslie, J. González, and P. Rayson. GIBBON: General-purpose information-based Bayesian optimisation. *J. Machine Learning Res.*, 22(235):1–49, 2021.

[26] I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 23, 2010.

[27] B. Paria, K. Kandasamy, and B. Póczos. A flexible framework for multi-objective Bayesian optimization using random scalarizations. In *Conf. on Uncertainty in Artif. Intell.*, volume 115, pages 766–776. PMLR, 2020.

[28] K. D. Polyzos, Q. Lu, and G. B. Giannakis. Bayesian optimization with ensemble learning models and adaptive expected improvement. In *Proc. of Intl. Conf. Acoust. Speech Signal Process.*, pages 1–5. IEEE, 2023.

[29] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 20, 2007.

[30] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[31] C. Riis, F. Antunes, F. B. Hüttel, C. L. Azevedo, and F. C. Pereira. Bayesian active learning with fully Bayesian Gaussian processes. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 35, 2022.

[32] B. Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, Department of Computer Sciences, 2009.

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

[34] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 25, 2012.

[35] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 29, 2016.

[36] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. Intl. Conf. Mach. Learn.*, pages 1015–1022, 2010.

[37] M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proc. Intl. Conf. Artif. Intell. Stat.*, volume 5, pages 567–574. PMLR, 2009.

[38] B. Tu, A. Gandy, N. Kantas, and B. Shafei. Joint entropy search for multi-objective Bayesian optimization. In *Proc. of Adv. Neural Inf. Process. Syst.*, volume 35, 2022.

[39] Z. Wang and S. Jegelka. Max-value entropy search for efficient Bayesian optimization. In *Proc. Intl. Conf. Mach. Learn.*, volume 70, pages 3627–3635. PMLR, 2017.

[40] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Proc. Intl. Conf. Artif. Intell. Stat.*, volume 51, pages 370–378. PMLR, 2016.

[41] J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *Proc. Intl. Conf. Mach. Learn.*, volume 119, pages 10292–10302. PMLR, 2020.

[42] L. Zimmer, M. Lindauer, and F. Hutter. Auto-PyTorch: Multi-fidelity metalearning for efficient and robust AutoDL. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(9):3079–3090, 2021.

[43] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Proc. Intl. Conf. Parallel Problem Solving from Nature*, pages 832–842. Springer, 2004.

[44] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, 2000.

Supplementary File

A KENDO-MO Algorithm for Multi-Objective Optimization

This appendix presents the full procedure of KENDO-MO described in Section 5.

Algorithm 2 KENDO-MO for Multi-Objective Optimization

Require: Kernel dict. $\mathcal{K}$, #scalarizations S, #objectives K, data $\mathcal{D}_0$, budget T

for $k = 1, \dots, K$; $m \in \mathcal{K}$ **do**
  Fit α_m^k via $\max p(\mathcal{D}_0^k | i = m, \alpha)$; Init $w_0^{k,m} = 1/M$
end for
for $t = 0, \dots, T-1$ **do**
  for $k = 1, \dots, K$ **do**
    Compute $\bar{\mu}_t^k(\cdot), \bar{\sigma}_t^{2,k}(\cdot)$ via (6)–(7) with weights $w_t^{k,m}$
  end for
  for $m \in \mathcal{K}$; $s = 1, \dots, S$ **do**
    Sample $\boldsymbol{\lambda}_s \sim \mathrm{Dir}(\mathbf{1}_K)$; Sample $f_{m,s}^k \sim p(f^k | i = m, \mathcal{D}_t)$ via RFF for all k
    $g_{m,s}(x) = \sum_k \lambda_{s,k} f_{m,s}^k(\mathbf{x})$; Find $x_{m,s}^* = \arg\max_x g_{m,s}(\mathbf{x})$
    for $k = 1, \dots, K$ **do**
      Condition GP m on $(\mathbf{x}_{m,s}^*, f_{m,s}^k(\mathbf{x}_{m,s}^*))$ to get $\mu_{m,s}^{*,k}, \sigma_{m,s}^{2,*,k}$
    end for
  end for
  $\mathbf{x}_{t+1} = \arg\max_x \sum_k \sum_m w_t^{k,m} \cdot \frac{1}{S} \sum_s d(\bar{p}^k, p_{m,s}^k)$ via (12)
  Evaluate $\mathbf{y}_{t+1} = (f^1(\mathbf{x}_{t+1}), \dots, f^K(\mathbf{x}_{t+1}))$; $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\mathbf{x}_{t+1}, \mathbf{y}_{t+1})\}$
  for $k = 1, \dots, K$; $m \in \mathcal{K}$ **do**
    Update $w_{t+1}^{k,m}$ via (4); Update GP posterior via (23a)–(23b)
  end for
end for
return Pareto front from $\mathcal{D}_T$

B Derivation of the KENDO-BO Acquisition Function

This appendix details how the KENDO-BO acquisition function in (8) arises from the disagreement principle of SCoreBO [18] once the source of model uncertainty is changed from continuous hyperparameters to a discrete kernel posterior. We also cover the moment-matching step that produces the Gaussian summary in (6) and (7), and the closed-form Hellinger distance in (9).

B.1 From SCoreBO's continuous expectation to KENDO's discrete sum

SCoreBO defines its acquisition as the expected Hellinger disagreement between the marginal predictive and an optimizer-conditioned predictive

$$\alpha_{\mathrm{SC}}(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\theta},*}[d(p(y_{\mathbf{x}} \mid \mathcal{D}_t),\ p(y_{\mathbf{x}} \mid \boldsymbol{\theta}, *, \mathcal{D}_t))]\,, \tag{13}$$

where the joint posterior factors as $p(\boldsymbol{\theta}, * \mid \mathcal{D}_t) = p(* \mid \boldsymbol{\theta}, \mathcal{D}_t)\, p(\boldsymbol{\theta} \mid \mathcal{D}_t)$. The outer expectation is intractable, and SCoreBO approximates it by drawing hyperparameter samples $\boldsymbol{\theta}_m$ from $p(\boldsymbol{\theta} \mid \mathcal{D}_t)$ via MCMC, then drawing N optimizers per sample, and applying the uniformly weighted estimator $\frac{1}{MN} \sum_m \sum_n d(\cdot, \cdot)$.

KENDO changes the random quantity itself. Instead of a continuous distribution over hyperparameters of a single kernel, the source of model uncertainty is the kernel *family*, whose posterior

is the discrete distribution $p(m \mid \mathcal{D}_t) = w_t^m$ defined by the Bayesian model average in (4). The acquisition function then takes the form

$$\alpha(\mathbf{x}) = \mathbb{E}_{m,*}[d(p(y_\mathbf{x} \mid \mathcal{D}_t),\ p(y_\mathbf{x} \mid m, *, \mathcal{D}_t))]\,. \tag{14}$$

Factoring the joint posterior as $p(m, * \mid \mathcal{D}_t) = p(* \mid m, \mathcal{D}_t)\, w_t^m$ and expanding the now discrete outer expectation gives

$$\alpha(\mathbf{x}) = \sum_{m=1}^{M} w_t^m \cdot \mathbb{E}_{*|m}[d(p(y_\mathbf{x} \mid \mathcal{D}_t),\ p(y_\mathbf{x} \mid m, *, \mathcal{D}_t))] \tag{15}$$

$$\approx \sum_{m=1}^{M} w_t^m \cdot \frac{1}{N} \sum_{n=1}^{N} d_{m,n}(\mathbf{x}) \ = \ \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{w_t^m}{N}\, d_{m,n}(\mathbf{x}), \tag{16}$$

which recovers (8). The inner step uses N optimizer samples $*_{m,n} = (\mathbf{x}^*_{m,n}, f^*_{m,n})$ obtained by optimizing function paths $f_{m,n} \sim p(f \mid m, \mathcal{D}_t)$ generated through random Fourier features [29, 41]. Conditioning the m-th GP on the phantom observation $(\mathbf{x}^*_{m,n}, f^*_{m,n})$ yields the conditional moments $\mu_{m,t}^{*_{m,n}}$ and $\sigma_{m,t}^{2,*_{m,n}}$.

B.2 Moment matching of the marginal mixture

The marginal predictive in (3) is a Gaussian mixture, so its Hellinger distance to a single Gaussian conditional has no closed form expression. We replace the mixture by a single Gaussian whose first and second moments match those of the mixture exactly. For the mixture $\sum_m w_t^m \mathcal{N}(\mu_{m,t}, \sigma^2_{m,t})$, these are

$$\bar{\mu}_t(\mathbf{x}) = \sum_{m=1}^{M} w_t^m \mu_{m,t}(\mathbf{x}), \tag{17}$$

$$\bar{\sigma}_t^2(\mathbf{x}) = \sum_{m=1}^{M} w_t^m \sigma^2_{m,t}(\mathbf{x}) + \sum_{m=1}^{M} w_t^m \big(\mu_{m,t}(\mathbf{x}) - \bar{\mu}_t(\mathbf{x})\big)^2, \tag{18}$$

which match (6) and (7). The variance decomposes additively into a within-component term (average individual variance) and a between-component term (spread of component means around the mixture mean).

B.3 Closed-form Hellinger distance between two Gaussians

For two univariate Gaussians $z_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $z_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$, the squared Hellinger distance admits the closed form

$$H^2(z_1, z_2) = 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2}} \exp\left[-\frac{1}{4}\frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}\right]. \tag{19}$$

Setting $z_1 \sim \mathcal{N}(\bar{\mu}_t(\mathbf{x}), \bar{\sigma}_t^2(\mathbf{x}))$ as the moment-matched marginal and $z_2 \sim \mathcal{N}(\mu_{m,t}^{*_{m,n}}(\mathbf{x}), \sigma_{m,t}^{2,*_{m,n}}(\mathbf{x}))$ as the conditional posterior of the m-th GP given optimizer sample n, and writing $d_{m,n}(\mathbf{x}) = H(z_1, z_2)$, we recover the per-pair Hellinger distance in (9).

B.4 Reduction to KENDO-AL

Removing the optimizer conditioning leaves

$$\alpha_{\mathrm{AL}}(\mathbf{x}) = \mathbb{E}_m[d(p(y_{\mathbf{x}} \mid \mathcal{D}_t),\ p(y_{\mathbf{x}} \mid m, \mathcal{D}_t))] = \sum_{m=1}^{M} w_t^m\, d_m(\mathbf{x}), \tag{20}$$

where $d_m(\mathbf{x})$ uses (19) between the moment-matched marginal and the m-th component. This recovers (10), and (11) follows from the same closed form.

C Ablation Studies

This appendix presents the full ablation studies summarized in Section 6.5 of the main text.

C.1 Adaptive vs. Uniform Weighting

Figure 5 compares KENDO with adaptive Bayesian weights (4) against a variant using fixed uniform weights ($w^m = 1/M$ throughout optimization). Across all three task categories, single-objective BO, active learning, and multi-objective optimization, adaptive weighting consistently achieves lower final regret.

On single-objective benchmarks, the advantage is most pronounced on Branin (2D) and Hartmann (6D), where adaptive weighting achieves roughly 0.5 and 0.3 lower $\log_{10}$ regret at convergence, respectively. For active learning, adaptive weighting leads to substantially better MLL on Hartmann (6D) and Ishigami (3D), where kernel preference varies significantly across regions of the input space. In multi-objective settings, the advantage is evident on DTLZ2 and Penicillin, where different objectives may favor different kernels, and adaptive per-objective weighting captures this structure more effectively than uniform averaging. These results confirm that the Bayesian weight update mechanism provides measurable performance gains by dynamically allocating model capacity to the most predictive kernels.

C.2 Computational Efficiency

A central claim of KENDO is that replacing MCMC hyperparameter sampling with an explicit kernel ensemble yields substantial computational savings. Figure 6 validates this claim by comparing per-iteration wall-clock time between EGP methods and their MCMC-based counterparts.

For Bayesian optimization, KENDO achieves 3.1×–5.4× speedup over SCoreBO across benchmarks, with the largest gain on Branin (2D) where MCMC overhead dominates relative to the simple acquisition landscape. The speedup is consistent across problem dimensions, ranging from 3.1× on Hartmann (4D) to 5.4× on Branin. The computational advantage is even more dramatic for active learning. KENDO-AL achieves 7.6×–26.9× speedup over SAL-Matérn-2.5, with the largest gains on Ishigami (26.9×) and Higdon (19.9×). This amplified speedup arises because active learning acquisition functions (which do not require RFF-based function sampling and optimizer search) have lower base cost, making the relative contribution of MCMC overhead proportionally larger.

Figure 7 provides a detailed breakdown of where computation time is spent within KENDO and KENDO-AL. For BO, acquisition function optimization dominates (50–80% of iteration time), with model fitting consuming most of the remainder (17–45%). RFF sampling and weight updates contribute negligibly. For active learning, the split between acquisition optimization and model fitting varies by benchmark, but weight update time remains consistently below 7 milliseconds, confirming that the Bayesian weight update (4) introduces negligible overhead.

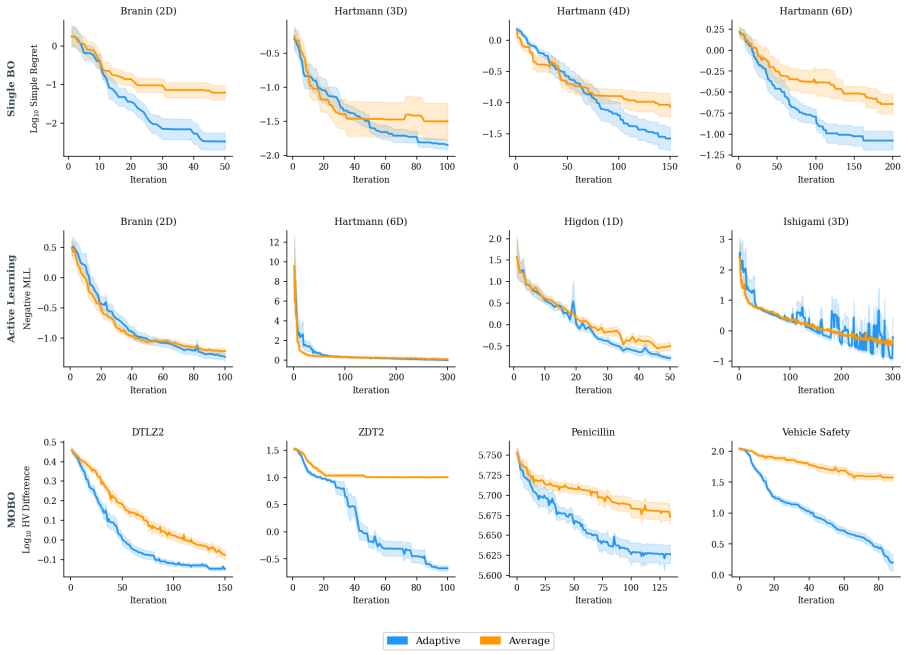


Figure 5: Convergence comparison between adaptive Bayesian weighting and uniform (average) weighting across all three task settings. **Top row:** $\log_{10}$ simple regret on single-objective benchmarks. **Middle row:** Negative MLL on active learning benchmarks. **Bottom row:** $\log_{10}$ hypervolume difference on multi-objective benchmarks. Adaptive weighting consistently achieves better performance.

C.3 Kernel Weight Convergence and Winner-Take-All Behavior

Figure 8 visualizes the kernel weight dynamics across all benchmarks and task settings by tracking the fraction of seeds for which each kernel holds the largest weight at each iteration.

On smooth, stationary benchmarks such as Branin, the ensemble weights consistently collapse to a single dominant kernel within approximately 10–20 iterations. This *winner-take-all* phenomenon is a direct consequence of the multiplicative Bayesian update rule (4): when one kernel achieves marginally higher predictive likelihood at each step, the weight ratio $w_t^m / w_t^{m'}$ evolves as a product of per-step likelihood ratios, amplifying small advantages exponentially. For single-objective BO on Branin, Matérn-2.5 wins in 5 out of 10 seeds with an average maximum weight of 0.950, reflecting its effective balance between the infinite smoothness of RBF and the limited differentiability of Matérn-1.5.

The summary tables in Figure 8 reveal that kernel preference is benchmark-dependent. On ZDT2 and Vehicle Safety (MOBO), RBF dominates universally (10/10 seeds, average max weight 1.000), while on Penicillin, Matérn-1.5 wins in 8/10 seeds, reflecting the less smooth objective landscape. For active learning, Matérn-2.5 is most frequently preferred (winning on 3 of 4 benchmarks), consistent with the observation that active learning objectives benefit from moderate smoothness assumptions for global function approximation.

Rather than harming performance, the winner-take-all behavior reflects successful automatic model selection. The ensemble efficiently identifies the most appropriate kernel for each target function without manual kernel selection or expensive cross-validation.

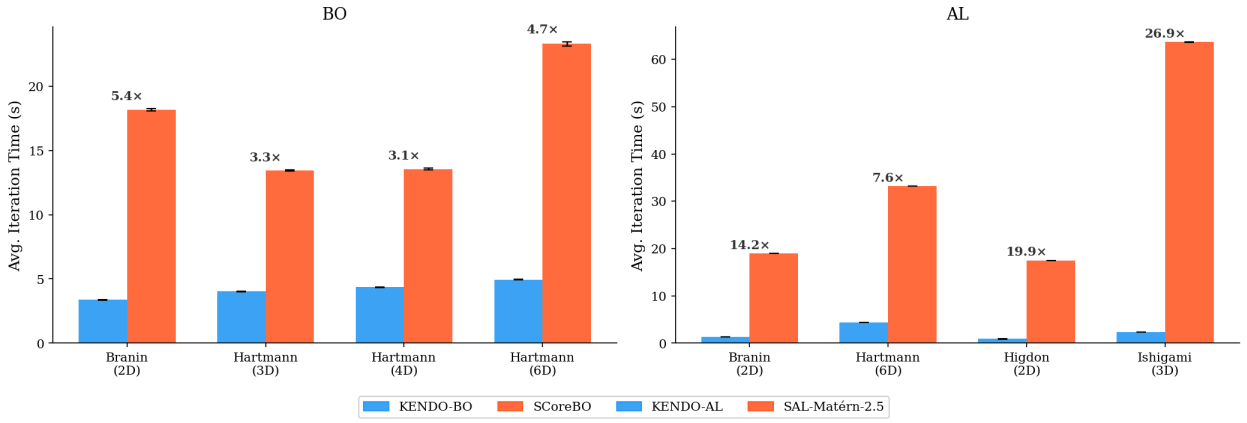


Figure 6: Per-iteration wall-clock time comparison. **Left:** KENDO vs. SCoreBO on single-objective benchmarks (3.1×–5.4× speedup). **Right:** KENDO-AL vs. SAL-Matérn-2.5 on active learning benchmarks (7.6×–26.9× speedup). The ensemble mechanism eliminates MCMC overhead while maintaining competitive optimization and learning performance (cf. Figures 2–4).

C.4 Sensitivity to Scalarization Strategy

KENDO-MO reduces multi-objective conditioning to single-optimizer conditioning via random scalarization. To assess whether its gains depend on this specific choice, we compare the default linear scalarization $g_{m,s}(\mathbf{x}) = \sum_k \lambda_{s,k} f^k_{m,s}(\mathbf{x})$ against Chebyshev scalarization $g_{m,s}(\mathbf{x}) = \min_k \lambda_{s,k} (f^k_{m,s}(\mathbf{x}) - z^*_k)$, keeping all other components (kernel ensemble, Bayesian weights, RFF sampling, Hellinger-based acquisition) fixed.

Robustness across benchmarks. Figure 9 reports hypervolume curves on four MOO benchmarks. A two-sided Wilcoxon rank-sum test on final hypervolume yields no statistically significant difference on ZDT2 ($p = 0.49$), Penicillin ($p = 0.16$), or Vehicle Safety ($p = 0.85$). This indicates that the gains of KENDO-MO arise from the kernel ensemble and disagreement-aware acquisition, *not* from the specific scalarization.

The DTLZ2 case. On DTLZ2, Chebyshev attains a small but significant advantage ($p = 0.014$). This is consistent with a well-known property in multi-objective optimization: linear scalarization is known to miss solutions on non-convex regions of the Pareto front, whereas Chebyshev scalarization can recover them [20]. DTLZ2's Pareto front is concave (a unit-sphere octant), so Chebyshev's edge here reflects a geometric property of the benchmark. To verify that the framework remains competitive on DTLZ2 once paired with the geometry-appropriate scalarization, Figure 10 plots KENDO-MO-Cheby alongside all multi-objective baselines: KENDO-MO-Cheby substantially outperforms all entropy-search (JESMO/MESMO/PESMO) and ensemble-TS (EGP-TS-MOBO) baselines; only qNEHVI, a hypervolume-specialized acquisition, achieves lower $\log_{10}$ HV difference. The scalarization is thus a tunable interface.

Two reasons motivate linear scalarization as the default. First, the linear objective is smooth in $\mathbf{x}$, so the $S \times M$ inner arg max problems on the RFF path (Algorithm 2, line 10) are solved efficiently with L-BFGS; the min operator in Chebyshev is non-smooth and requires either subgradient methods or a smoothed surrogate, inflating per-iteration cost, which conflicts with our efficiency-oriented design (Appendix C.2). Second, linear scalarization matches the standard ParEGO formulation [20], facilitating fair comparison with prior multi-objective BO literature.

Takeaway. KENDO-MO is robust to the scalarization choice, and a practitioner may select either based on prior knowledge of the problem geometry, e.g., Chebyshev when the Pareto front is

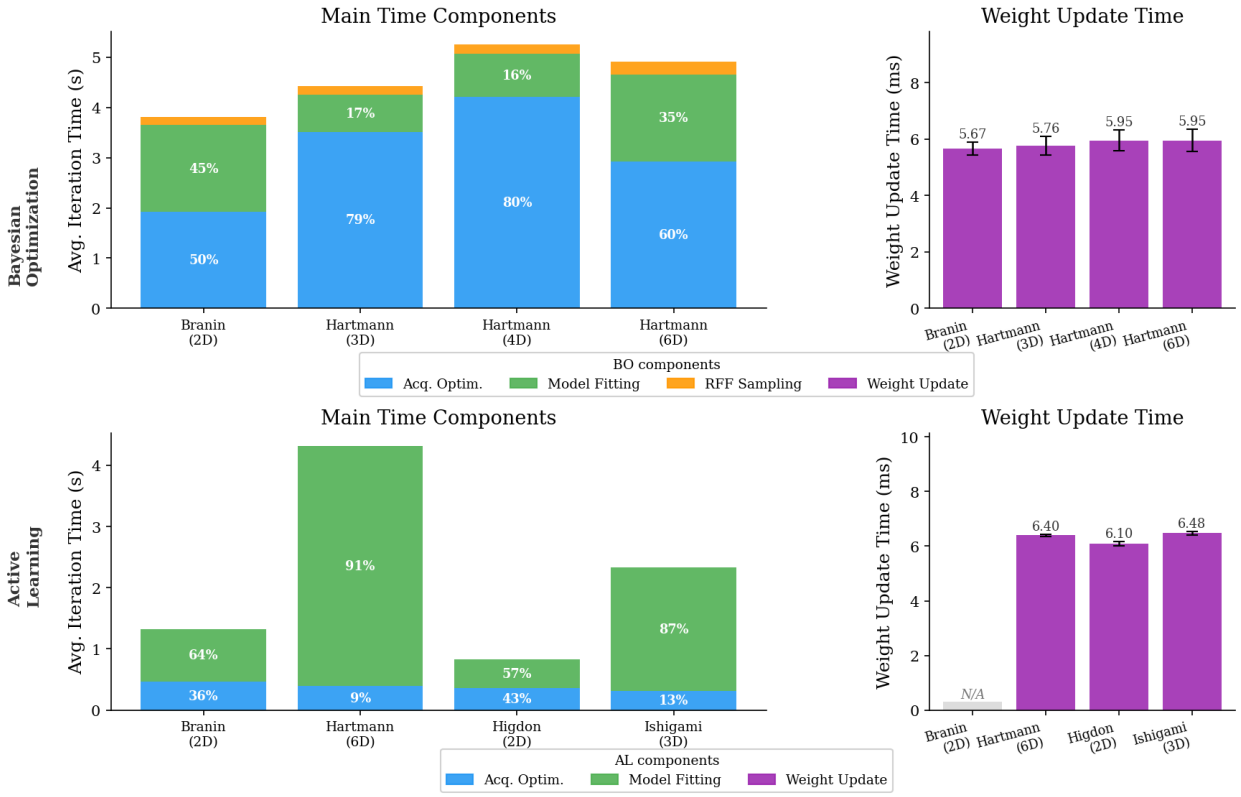


Figure 7: Time component breakdown for KENDO (**top**) and KENDO-AL (**bottom**). Left panels show main time components (acquisition optimization, model fitting, RFF sampling, weight update). Right panels isolate weight update time, confirming it contributes negligibly (< 7 ms per iteration). For BO, acquisition optimization dominates (50–80%); for AL, model fitting becomes the primary cost on higher-dimensional benchmarks.

suspected to be non-convex.

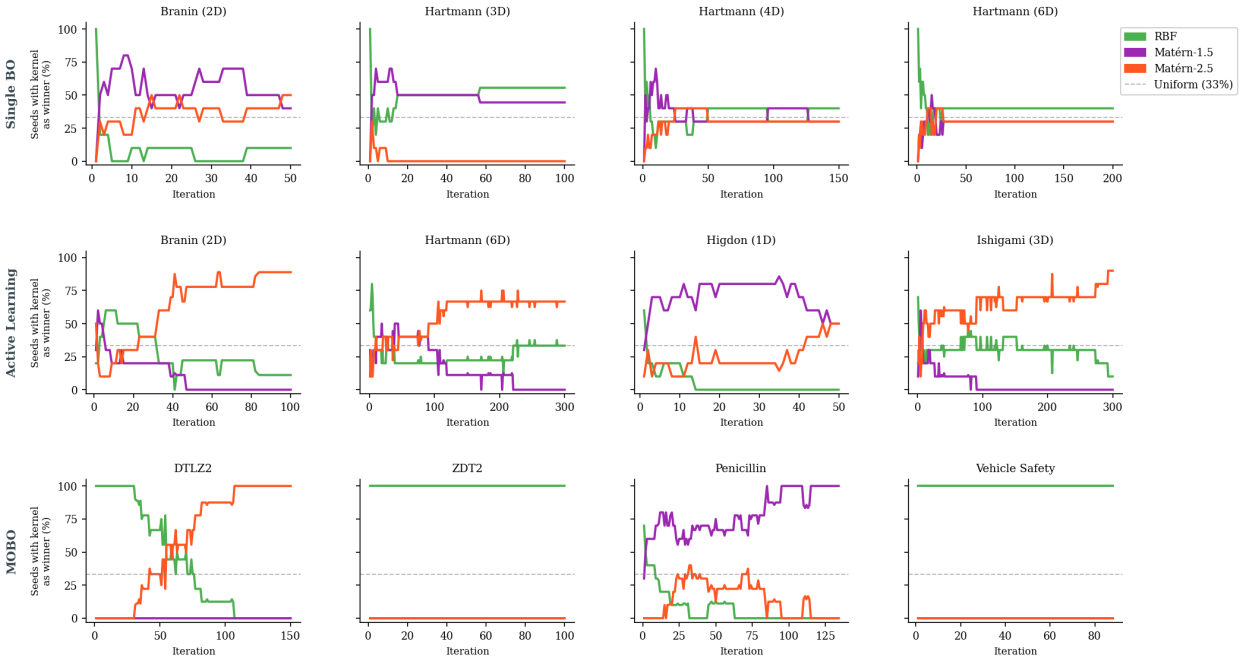
C.5 Sensitivity to Dirichlet Concentration α

KENDO-MO samples scalarization weights from a symmetric Dirichlet $\boldsymbol{\lambda}_s \sim \mathrm{Dir}(\alpha \mathbf{1}_K)$, whose density

$$p(\boldsymbol{\lambda}; \alpha) = \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \lambda_k^{\alpha-1}, \qquad \boldsymbol{\lambda} \in \Delta^{K-1}, \tag{21}$$

is governed by a single concentration parameter α that controls how mass is distributed on the simplex. The per-coordinate variance $\mathrm{Var}[\lambda_k] = (K-1)/(K^2(K\alpha+1))$ is monotonically decreasing in α, so $\alpha \to 0$ pushes $\boldsymbol{\lambda}$ toward the vertices (corner-heavy, near single-objective directions), $\alpha = 1$ recovers the uniform prior used in Section 5, and $\alpha \to \infty$ concentrates on the centroid $(1/K, \ldots, 1/K)$ (center-heavy, near equal-weight trade-offs). Through the inner arg max in Algorithm 2 (line 10), α therefore directly governs the diversity of the phantom optimizers $\{x^*_{m,s}\}$ that drive the disagreement signal in (12). We sweep $\alpha \in \{0.2, 1.0, 5.0\}$ on five MOO benchmarks (LCBench, VehicleSafety, DTLZ2, Penicillin, ZDT2).

Figure 11(b,c) verifies that α acts as designed. The mean sample entropy $\bar{H} = \mathbb{E}_{\boldsymbol{\lambda}}[-\sum_k \lambda_k \log \lambda_k]$ stratifies cleanly across the three values and approaches its ceiling $\log K$ at $\alpha = 5.0$ ($\bar{H} \approx 0.65$ for



Single BO

Benchmark	RBF wins	Matérn-1.5 wins	Matérn-2.5 wins	Avg. max weight
Branin (2D)	1/10	4/10	5/10	0.950
Hartmann (3D)	5/10	5/10	0/10	0.998
Hartmann (4D)	4/10	3/10	3/10	0.970
Hartmann (6D)	4/10	3/10	3/10	0.999

Active Learning

Benchmark	RBF wins	Matérn-1.5 wins	Matérn-2.5 wins	Avg. max weight
Branin (2D)	2/10	0/10	8/10	0.978
Hartmann (6D)	3/10	1/10	6/10	1.000
Higdon (1D)	0/10	5/10	5/10	0.831
Ishigami (3D)	1/10	0/10	9/10	1.000

MOBO

Benchmark	RBF wins	Matérn-1.5 wins	Matérn-2.5 wins	Avg. max weight
DTLZ2	2/10	0/10	8/10	0.760
ZDT2	10/10	0/10	0/10	1.000
Penicillin	0/10	8/10	2/10	0.780
Vehicle Safety	10/10	0/10	0/10	1.000

Figure 8: Kernel weight dynamics across all benchmarks. Line plots show the fraction of seeds (out of 10) for which each kernel holds the largest weight at each iteration. The dashed gray line indicates the uniform baseline (33%). Tables summarize the final kernel preference distribution and average maximum weight. The ensemble exhibits decisive model selection on benchmarks with a clear best kernel (e.g., RBF on ZDT2 with max weight 1.000), while maintaining diversity under ambiguity (e.g., Hartmann 3D in BO with a 5/5 split between RBF and Matérn-1.5).

$K = 2$ and 1.03 for $K = 3$), while the mean pairwise distance of phantom optimizers (column c) shows the expected inverse relationship, $\alpha = 0.2$ yields the highest phantom-X diversity and $\alpha = 5.0$ the lowest. The Dirichlet sampling layer therefore behaves exactly as the variance formula predicts.

The hypervolume curves in column (a) show that the default $\alpha = 1.0$ is robust across all five benchmarks: it is never significantly outperformed by either extreme. The optimal α is, however, benchmark-dependent and tracks Pareto-front geometry. On benchmarks with concave fronts (ZDT2, DTLZ2) and heterogeneous landscapes (VehicleSafety), small α matches or exceeds the default, consistent with the classical result that linear scalarization on concave fronts benefits from corner-heavy directions to recover non-convex regions [20]. Conversely, on real-world problems whose trade-off region concentrates near the simplex centroid (LCBench, Penicillin), $\alpha = 5.0$ yields a small but consistent gain. Phantom-X diversity is *not* monotonically tied to hypervolume: sufficient diversity is necessary, but excess diversity wastes budget on extreme corners that lie far from the achievable front.

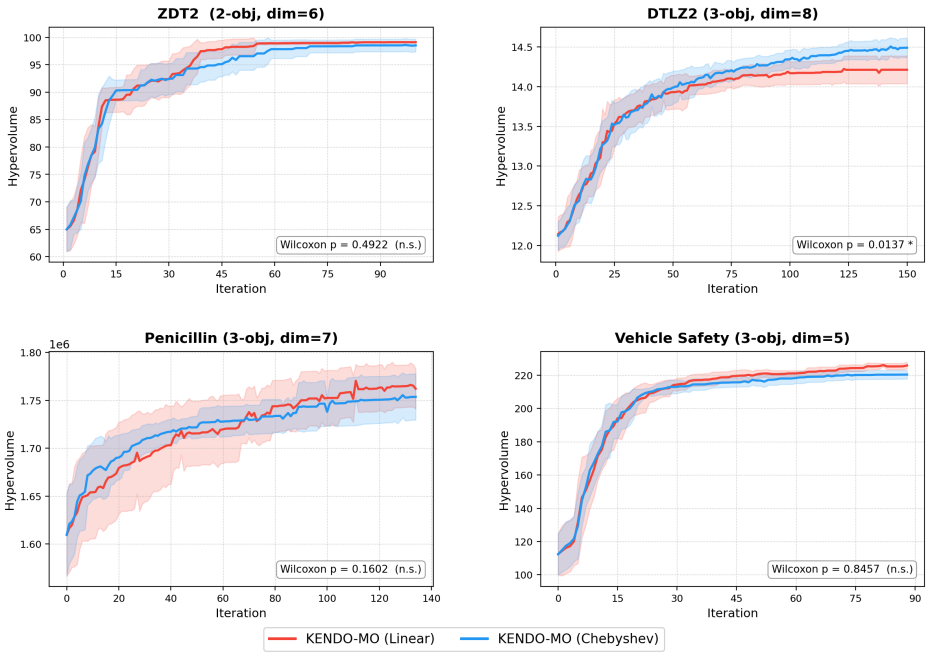


Figure 9: KENDO-MO with linear and Chebyshev scalarization on four MOO benchmarks. Three of four benchmarks show no statistically significant difference (two-sided Wilcoxon rank-sum test on final HV); DTLZ2 favors Chebyshev due to its concave Pareto front (see Appendix C.4).

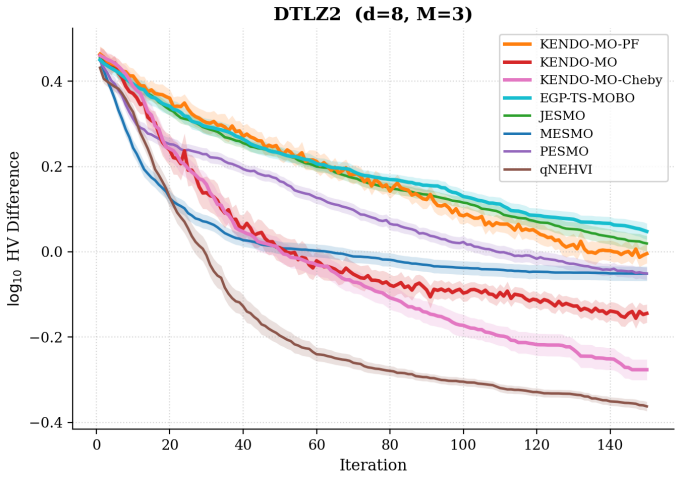


Figure 10: DTLZ2 with all multi-objective baselines plus KENDO-MO-Cheby. Both KENDO-MO variants outperform all entropy-search and ensemble-TS baselines; only qNEHVI (hypervolume-specialized) attains lower $\log_{10}$ HV difference, confirming that the scalarization choice is a tunable interface rather than a framework bottleneck.

Takeaway. KENDO-MO's gains arise from the kernel ensemble and disagreement-aware acquisition, not from a finely tuned Dirichlet concentration; the symmetric prior $\alpha = 1.0$ is a safe and competitive default, while practitioners with prior knowledge of front geometry may tune α accordingly. Together with the study in Appendix C.4, this confirms that the scalarization layer is a tunable interface rather than a sensitive bottleneck.

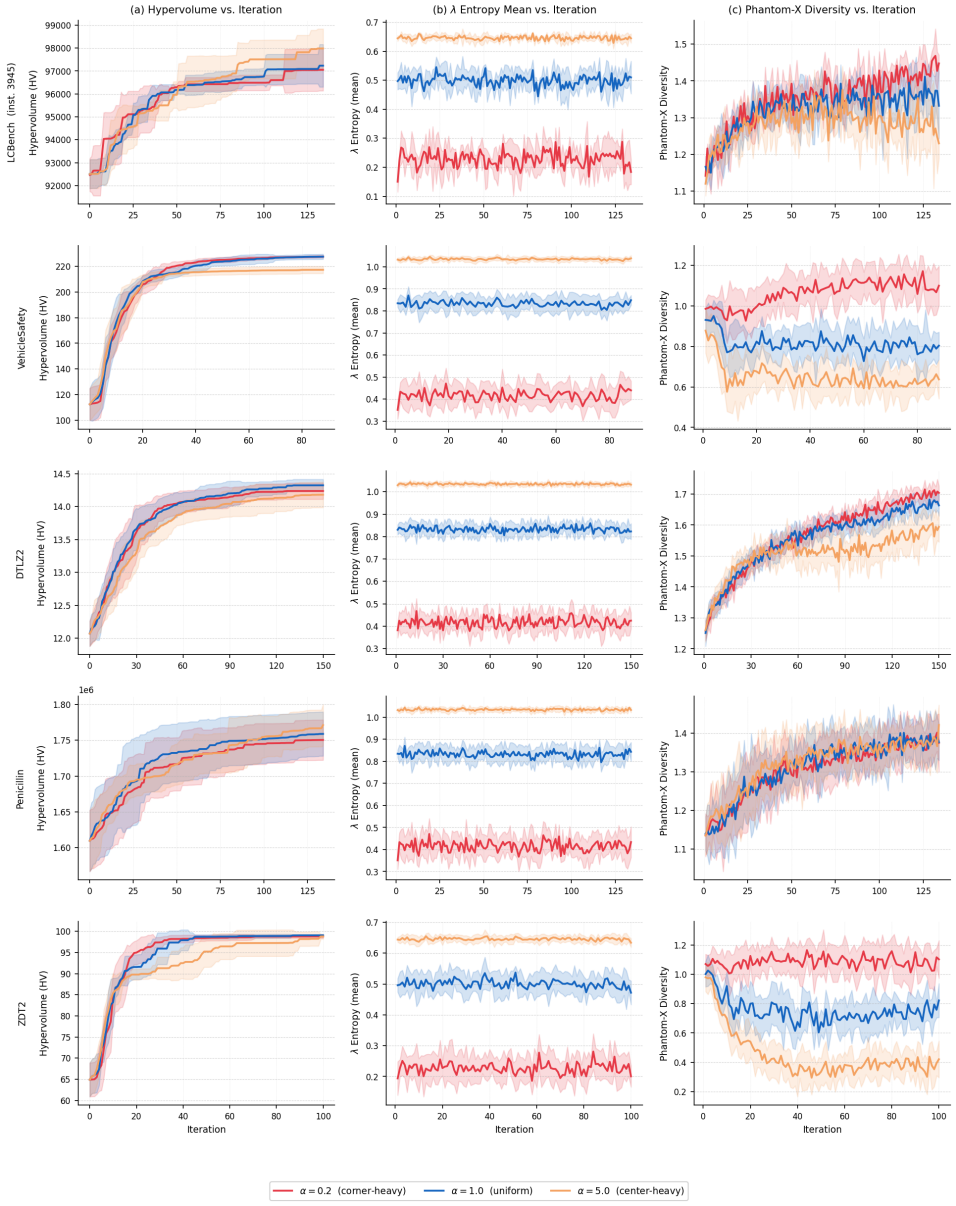


Figure 11: Sensitivity of KENDO-MO to Dirichlet concentration $\alpha \in \{0.2, 1.0, 5.0\}$ on five MOO benchmarks. **Column (a):** hypervolume vs. iteration. **Column (b):** mean Shannon entropy of sampled $\boldsymbol{\lambda}$, confirming that the Dirichlet sampling layer behaves as designed. **Column (c):** mean pairwise distance of phantom optimizers, showing α controls phantom-X diversity inversely. The default $\alpha = 1.0$ is never significantly outperformed; optimal α is benchmark-dependent and tracks Pareto-front geometry.

D Additional preliminaries

For each Gaussian process model m in the ensemble, using the data $\mathcal{D}_t := \{(\mathbf{x}_\tau, y_\tau)\}_{\tau=1}^t$ acquired at slot t, the corresponding posterior pdf $p(f(\mathbf{x})|m, \mathcal{D}_t)$ is updated via Bayes rule as [30]:

$$p(f(\mathbf{x})|m, \mathcal{D}_t) = \mathcal{N}(\mu_{m,t}(\mathbf{x}), \sigma^2_{m,t}(\mathbf{x})) \tag{22}$$

where

$$\mu_{m,t}(\mathbf{x}) = \mathbf{k}_t^\top(\mathbf{x})(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\mathbf{y}_t \tag{23a}$$

$$\sigma^2_{m,t}(\mathbf{x}) = \kappa(\mathbf{x},\mathbf{x}) - \mathbf{k}_t^\top(\mathbf{x})(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\mathbf{k}_t(\mathbf{x}). \tag{23b}$$

with $\mathbf{k}_t(\mathbf{x}) := [\kappa(\mathbf{x}_1,\mathbf{x}), \ldots, \kappa(\mathbf{x}_t,\mathbf{x})]^\top$, $\mathbf{K}_t$ denoting the $t \times t$ kernel/covariance matrix where $[\mathbf{K}]_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, and σ_n^2 denoting the observation variance.

Note that the mean in (23a) provides a point estimate of $f(\mathbf{x})$, while the variance in (23b) quantifies the associated uncertainty.