RAMMOBO: RANDOM MAX-MARGIN MULTI-OBJECTIVE BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-objective Bayesian optimization (MOBO) maximizes M expensive black-box objectives with the goal of approximating the *Pareto set*, whose objective vectors cannot be improved in one component without sacrificing another. A common quality measure is *hypervolume*, the volume in objective space dominated by these vectors relative to a reference point. *Expected hypervolume improvement* (EHVI) directly targets this metric and is one-step Bayes-optimal, but its standard computation scales exponentially with M, making it prohibitively expensive for many-objective problems. We introduce *random max-margin multi-objective Bayesian optimization* (RaMMOBO), whose acquisition function lower-bounds EHVI while requiring only linear cost in M to evaluate. RaMMOBO measures the volume of a max-margin box contained in a candidate's hypervolume-improvement region and averages over random reweightings of the objective space, yielding a utility that is strictly Pareto-monotone on hypervolume-improving outcomes. It naturally handles noisy observations and different probabilistic surrogates and supports efficient gradient-based optimization. Across synthetic and real-world benchmarks, RaMMOBO matches or outperforms state-of-the-art methods, with largest gains at higher M, with acquisition optimization up to two orders of magnitude faster than qLogNEHVI. Our implementation is available at `https://anonymous.4open.science/r/rammobo`.

1 INTRODUCTION

Optimizing expensive scalar-valued black-box functions is central to scientific discovery, engineering design, and machine learning. Bayesian optimization (BO) addresses this problem by fitting a probabilistic surrogate to the observed data and using an acquisition function to decide where to query the function next (Shahriari et al., 2016). Multi-objective Bayesian optimization (MOBO) extends BO to problems with M competing objectives. In this setting, there is generally no single optimal solution: improving one objective may require worsening another. The goal is therefore to efficiently identify a set of Pareto-optimal solutions that represents different trade-offs between the objectives. Such problems arise, for example, in drug discovery and engineering design, and many practical applications may involve substantially more than three objectives (Fleming et al., 2005; Li et al., 2015; Fromer & Coley, 2023; Tanabe & Ishibuchi, 2020).

As M grows, however, selecting the next evaluation can itself become a computational bottleneck. Effective MOBO therefore requires acquisition functions that guide the search effectively while remaining computationally efficient to evaluate and optimize. Hypervolume-based acquisitions such as (noisy) expected hypervolume improvement ((N)EHVI) directly quantify improvement in dominated volume and are known to be one-step Bayes-optimal for hypervolume maximization (Daulton et al., 2021). However, standard implementations rely on geometric decompositions of the objective space (Daulton et al., 2021; Ament et al., 2023), whose computational cost grows exponentially in M (Yang et al., 2019). Hence, for problems with many objectives, hypervolume-based acquisition functions can quickly become computationally prohibitive to evaluate. Many alternative acquisition functions replace hypervolume computation with other costly subroutines: information-theoretic and posterior-sampling methods often require auxiliary multi-objective optimization (Hernández-Lobato et al., 2016; Belakaria et al., 2019; Tu et al., 2022; Renganathan & Carlson, 2025), while BOtied adds multivariate CDF fitting and gradient-free acquisition optimization (Park et al., 2024). Scalarization is computationally cheaper, but its Pareto-front coverage depends strongly on the

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

choice of scalarization and weight distribution (Knowles, 2006; Zhang & Golovin, 2020). These trade-offs motivate an acquisition function that retains a direct connection to multi-objective improvement while avoiding box decompositions and costly auxiliary optimization.

We introduce **RaMMOBO**, *random max-margin multi-objective Bayesian optimization*, a simple alternative to hypervolume-based acquisition functions that avoids box decompositions altogether. Consider Figure 1, which shows the objective space of an $M = 2$-dimensional maximization problem with current Pareto-front approximation $R = \{r_1, r_2\}$, which dominates the space $D_z(R)$ below it, up to a reference point z. Rather than explicitly computing *hypervolume improvement* (HVI) for a candidate objective vector y, RaMMOBO intuitively measures the shortest axis-aligned distance to the region dominated by R, called the margin $\Delta(y)$. RaMMOBO's utility $L(y)$ then measures the volume of a hypercube ending at y with side length $\Delta(y)$—while clipping the cube to a box for objective vectors like y', where the cube would include vectors that are worse than the reference z in some dimension. With this construction, L gives a direct lower bound on HVI without costly partitions of the objective space.

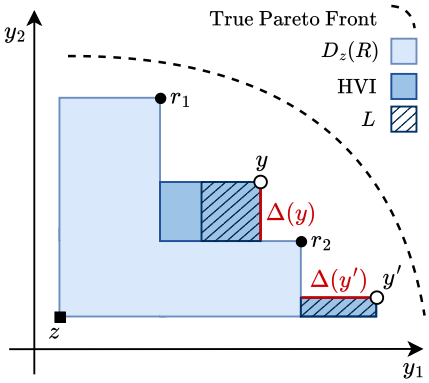


Figure 1: RaMMOBO's utility L lower-bounds hypervolume improvement (HVI); see text for details.

Furthermore, akin to random scalarization (Knowles, 2006), RaMMOBO randomizes the relative axis scalings of objective space and averages the resulting volumes, which yields a utility that is strictly Pareto-monotone on objective vectors that improve hypervolume. Crucially, for $N = |R|$ nondominated points, evaluating RaMMOBO's utility requires only $\mathcal{O}(NM)$ operations, while hypervolume-based methods scale exponentially in M. RaMMOBO naturally incorporates noisy observations, can be used with any surrogate model that provides differentiable posterior samples, and its acquisition function can be efficiently optimized via gradient-based methods.

We evaluate RaMMOBO on synthetic and real-world benchmarks spanning $M = 2$ to 14 objectives, including noisy and batch settings. RaMMOBO is competitive at lower objective counts and becomes increasingly advantageous as M grows, with acquisition optimization roughly two orders of magnitude faster than common methods based on hypervolume.

In summary, our contributions are threefold:

- We introduce RaMMOBO, whose evaluation cost scales linearly in the number of objectives M, avoiding the expensive box decompositions required by hypervolume-based methods. RaMMOBO naturally supports noisy observations and different probabilistic surrogates and admits efficient gradient-based acquisition optimization.
- We show that RaMMOBO retains key properties of hypervolume improvement: its utility is strictly Pareto-monotone on hypervolume-improving outcomes, and its acquisition lower-bounds noisy expected hypervolume improvement.
- We demonstrate strong empirical performance on synthetic and real-world benchmarks in sequential and batch settings. RaMMOBO is competitive at $M = 8$ and achieves state-of-the-art performance for $M \geq 10$, while reducing acquisition evaluation time by more than two orders of magnitude compared with hypervolume-based methods.

2 Background

2.1 Bayesian optimization

Bayesian optimization (BO) optimizes an expensive scalar-valued black-box function $f\colon \mathcal{X} \to \mathbb{R}$ under a limited evaluation budget by maintaining a probabilistic surrogate, often a Gaussian process (GP) (Rasmussen & Williams, 2006). Given noiseless observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with $y_i = f(x_i)$, the surrogate represents uncertainty about $f(x)$ through a random objective value $Y(x)$

with posterior predictive distribution $p(Y(x) \mid \mathcal{D})$. BO selects new evaluations by maximizing an acquisition function $\alpha\colon \mathcal{X} \to \mathbb{R}$,

$$x' \in \arg\max_{x \in \mathcal{X}} \alpha(x). \tag{1}$$

Many improvement-based acquisitions are posterior expectations of a utility $u : \mathbb{R} \to \mathbb{R}$,

$$\alpha(x) = \mathbb{E}_{p(Y(x)|\mathcal{D})}\left[u(Y(x))\right]. \tag{2}$$

Expected improvement (EI) (Jones et al., 1998) is the canonical example: for incumbent $f^\star = \max_{i \le n} f(x_i)$, it uses the improvement utility $u(y) = (y - f^\star)_+$,

$$\mathrm{EI}(x) = \mathbb{E}_{p(Y(x)|\mathcal{D})}\left[(Y(x) - f^\star)_+\right]. \tag{3}$$

When evaluations can be performed in parallel, batch BO instead selects $q > 1$ inputs that complement one another.

2.2 Multi-objective Bayesian optimization

In the multi-objective setting, the scalar objective is replaced by a deterministic vector-valued function $f = (f_1, \ldots, f_M) : \mathcal{X} \to \mathbb{R}^M$, with all M objectives maximized. Accordingly, the surrogate represents uncertainty at x through the random vector $Y(x) = (Y_1(x), \ldots, Y_M(x)) \in \mathbb{R}^M$ with posterior predictive distribution $p(Y(x) \mid \mathcal{D})$. Objective vectors are only partially ordered: $y \in \mathbb{R}^M$ weakly dominates $y' \in \mathbb{R}^M$, written $y \succeq y'$, if $y_m \ge y'_m$ for every m, and dominates it, written $y \succ y'$, if at least one inequality is strict. An input is Pareto optimal if its objective vector is not dominated by that of any other input, and the corresponding objective vectors form the Pareto front. MOBO seeks a finite approximation of this front, which we denote by the nondominated set $R = \{r_1, \ldots, r_N\} \subset \mathbb{R}^M$.

A standard scalar measure of the quality of R is its *hypervolume*. Given a reference point $z \in \mathbb{R}^M$, let

$$D_z(R) := \left\{y \in \mathbb{R}^M : z \preceq y \text{ and } y \preceq r \text{ for some } r \in R\right\} \tag{4}$$

denote the region dominated by R and bounded below by z, as illustrated in Figure 1. Its hypervolume is

$$\mathrm{HV}_z(R) := \lambda(D_z(R)), \tag{5}$$

where λ denotes Lebesgue measure. The hypervolume improvement (HVI) of a candidate objective vector y is the additional dominated volume obtained by adding it to R,

$$\mathrm{HVI}_z(y; R) := \mathrm{HV}_z(R \cup \{y\}) - \mathrm{HV}_z(R) = \lambda(D_z(R \cup \{y\}) \setminus D_z(R)). \tag{6}$$

HVI respects the Pareto order: if $y \succeq y'$, then $\mathrm{HVI}_z(y; R) \ge \mathrm{HVI}_z(y'; R)$. Moreover, it is *strictly Pareto-monotone* on hypervolume-improving outcomes,

$$y \succ y', \qquad \mathrm{HVI}_z(y'; R) > 0 \quad \Longrightarrow \quad \mathrm{HVI}_z(y; R) > \mathrm{HVI}_z(y'; R). \tag{7}$$

This strictness is important when using a scalar utility to guide multi-objective search: if $y \succ y'$, assigning the same utility to both makes the acquisition insensitive to an objective improvement that comes at no cost in any other objective. Some existing multi-objective utilities are only weakly Pareto-monotone and can therefore exhibit this behavior. Our construction will instead preserve the strict Pareto monotonicity of HVI on hypervolume-improving outcomes.

Averaging HVI under the posterior predictive distribution gives expected hypervolume improvement (EHVI) (Emmerich et al., 2006),

$$\mathrm{EHVI}_z(x) := \mathbb{E}_{p(Y(x)|\mathcal{D})}\left[\mathrm{HVI}_z(Y(x); R)\right]. \tag{8}$$

This mirrors EI, with hypervolume improvement replacing scalar improvement as the realized utility. With noisy observations $y_i = f(x_i) + \epsilon_i$, the latent objective vectors at previously evaluated inputs, and hence their dominance relationships, are uncertain. Noisy expected hypervolume improvement (NEHVI) accounts for this uncertainty by averaging HVI jointly over the candidate outcome and the latent outcomes at the evaluated inputs (Daulton et al., 2021). Writing $\mathcal{R} = \{Y(x_i)\}_{i=1}^n$,

$$\mathrm{NEHVI}_z(x) := \mathbb{E}_{p(Y(x),\mathcal{R}|\mathcal{D})}\left[\mathrm{HVI}_z(Y(x); \mathcal{R})\right]. \tag{9}$$

3 RANDOM MAX-MARGIN MULTI-OBJECTIVE BAYESIAN OPTIMIZATION

RaMMOBO is built around a simple idea: replace expensive hypervolume improvement with the volume of a cheap geometric object that is guaranteed to lie inside the hypervolume-improvement region. We obtain such an object from the maximin margin underlying expected maximin improvement (EMmI) (Svenson & Santner, 2016), whose evaluation scales linearly with the number of objectives M for a fixed reference-set size. The original margin, however, is reference-point-free and only weakly Pareto-monotone, so it does not retain key properties of hypervolume improvement. We address these limitations in two steps. First, we use the hypervolume reference point z to turn the margin into an axis-aligned box whose volume lower-bounds HVI. Second, we randomize the box geometry and average over these randomizations, recovering strict Pareto monotonicity on hypervolume-improving outcomes. Each weighted utility evaluation costs $\mathcal{O}(NM)$ for N reference outcomes, while the resulting acquisition provides a direct lower bound on (N)EHVI and supports efficient gradient-based acquisition optimization.

3.1 TURNING MAXIMIN IMPROVEMENT TO AN IMPROVEMENT VOLUME

We begin in the noiseless setting, with a fixed finite, nonempty nondominated set $R = \{r_1, \dots, r_N\} \subset \mathbb{R}^M$ and hypervolume reference point $z \in \mathbb{R}^M$. For maximization, the EMmI acquisition of Svenson & Santner (2016) is

$$\alpha_{\mathrm{EMmI}}(x) = \mathbb{E}_{p(Y(x)|\mathcal{D})}[\Delta(Y(x); R)] \quad \text{with} \quad \Delta(y; R) = \min_{i=1,\dots,N} \max_{m=1,\dots,M} (y_m - r_{i,m})_+ \tag{10}$$

The margin Δ measures how far y improves over every incumbent in at least one objective. It is positive precisely when no point in R weakly dominates y, but does not account for the hypervolume reference point: a positive margin need not imply positive HVI. In particular, $\Delta(y; R)$ can be positive even when $y_m \le z_m$ for some objective m, in which case y dominates no positive M-dimensional volume above z and therefore has zero HVI.

To connect this margin to hypervolume, we construct an axis-aligned hyperrectangle (box) ending at y. Each side is bounded by Δ to exclude the already-dominated region and clipped at z to respect the hypervolume reference point:

$$\ell_m(y; R) = \min\{\Delta(y; R), (y_m - z_m)_+\}, \qquad L(y; R) = \prod_{m=1}^{M} \ell_m(y; R). \tag{11}$$

For positive volume, clipping keeps this box above z, while the maximin margin ensures that no incumbent weakly dominates its interior. Its interior therefore lies in the hypervolume-improvement region, giving

$$L(y; R) \le \mathrm{HVI}_z(y; R). \tag{12}$$

The bound is trivial at zero volume and is the unweighted special case of the result proved in Appendix A.1.

3.2 RANDOM WEIGHTING YIELDS STRICT PARETO MONOTONICITY

The box utility L can miss strict Pareto improvements because its max-margin construction may be controlled by a different coordinate. For example, with $R = \{(0, 0)\}$ and $z = (-3, -3)$, both $u = (2, 1)$ and $v = (2, 2)$ satisfy $L(u; R) = L(v; R) = 4$ despite $v \succ u$. Randomly changing the box aspect ratio removes this insensitivity by making different coordinates determine the box geometry under different weightings.

Specifically, for a positive weight vector $w \in \mathbb{R}^M_{>0}$, define the weighted margin

$$\Delta_w(y; R) = \min_i \max_m w_m (y_m - r_{i,m})_+. \tag{13}$$

In the original objective coordinates, this margin corresponds to side length Δ_w / w_m in coordinate m. Clipping these sides at the hypervolume reference point gives

$$\ell_{w,m}(y; R) = \min\left\{ \frac{\Delta_w(y; R)}{w_m}, (y_m - z_m)_+ \right\}, \tag{14}$$

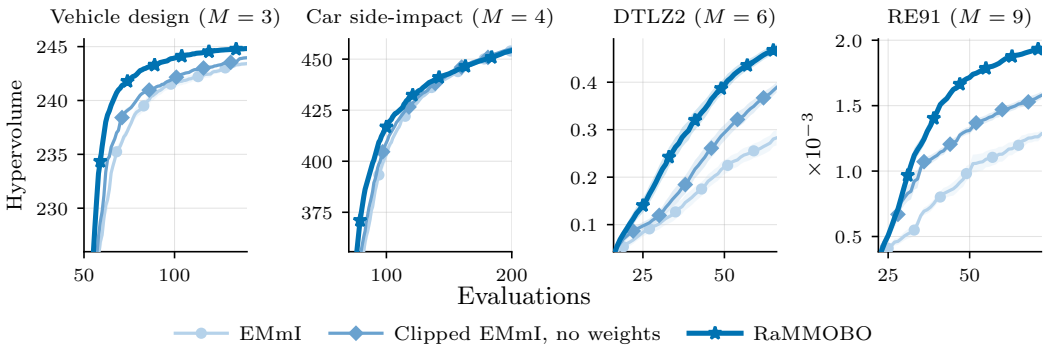


Figure 2: **Impact of improvements to EMmI on common benchmark problems.** Starting from EMmI (Eq. 10), we add clipping w.r.t. z, i.e., $\mathbb{E}_{p(Y(x)|\mathcal{D})}[L(Y(x); R)]$ (Eq. 11), and finally introduce random weights to yield RaMMOBO. All three variants use the observed nondominated outcomes as a fixed reference set (noiseless setting) and share the same estimator and optimizer (Appendix C.7). RaMMOBO consistently improves hypervolume, where the gap becomes especially substantial when the number of objectives M increases. Curves are the mean over 10 seeds with ± 1 standard-error band.

and the resulting volume remains a pointwise lower bound on HVI,

$$L_w(y; R) = \prod_{m=1}^{M} \ell_{w,m}(y; R) \leq \mathrm{HVI}_z(y; R). \tag{15}$$

Only the relative weights matter, since multiplying w by a common positive constant leaves the box unchanged. Setting $w = \mathbf{1}$ recovers the unweighted utility L, and Appendix A.1 proves the bound for general w.

We now average over random aspect ratios by drawing

$$W \sim M \cdot \mathrm{Dirichlet}(\mathbf{1}), \qquad \bar{L}(y; R) = \mathbb{E}_W[L_W(y; R)]. \tag{16}$$

This distribution treats objectives symmetrically and has support over all positive relative weightings. While each fixed L_w is only weakly Pareto-monotone, any strict Pareto improvement from an outcome with positive HVI is detected by a set of weightings with positive probability. Averaging therefore recovers strict Pareto monotonicity while preserving the HVI lower bound:

$$y \succ y', \qquad \mathrm{HVI}_z(y'; R) > 0 \quad \Longrightarrow \quad \bar{L}(y; R) > \bar{L}(y'; R). \tag{17}$$

The proof is given in Appendix A.2.

3.3 The RaMMOBO Acquisition Function

In the noiseless setting, taking the posterior expectation of the randomized utility yields

$$\mathbb{E}_{p(Y(x)|\mathcal{D})}\left[\bar{L}(Y(x); R)\right] \leq \mathrm{EHVI}_z(x).$$

With noisy observations, the objective values at previously evaluated inputs are themselves uncertain, so the reference set must also be treated as random. Following NEHVI, let

$$\mathcal{R} = \{Y(x_i)\}_{i=1}^{n}$$

denote the latent objective vectors at the evaluated inputs. Averaging L_W jointly over the posterior of the candidate outcome $Y(x)$, the random reference set $\mathcal{R}$, and an independent weight vector W defines the *Random Max-Margin Multi-objective Bayesian Optimization* (RaMMOBO) acquisition:

$$\boxed{\alpha_{\mathrm{RaMMOBO}}(x) = \mathbb{E}_{p(Y(x),\mathcal{R}|\mathcal{D})}\mathbb{E}_W\left[L_W(Y(x); \mathcal{R})\right] \leq \mathrm{NEHVI}_z(x)}. \tag{18}$$

The bound follows directly by averaging the pointwise inequality in Eq. (15) over the same joint posterior used by NEHVI (Daulton et al., 2021).

Algorithm 1 RaMMOBO's inner loop, sequential setting ($q = 1$)

Require: Data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$; surrogate with differentiable samples $Y(x) = g_{\mathcal{D}}(x, \xi)$; reference point z; number of Monte Carlo samples S; fatplus φ_τ

$\hat{\mathcal{X}} \leftarrow \{x_i : (x_i, y_i) \in \mathcal{D},\ \mathbb{E}[Y(x_i) \mid \mathcal{D}] \text{ is nondominated}\}$

Draw and fix $w^{(s)} \sim M \cdot \mathrm{Dirichlet}(\mathbf{1})$ and independent base samples $\xi^{(s)}, \xi_i^{(s)},\ s \le S, x_i \in \hat{\mathcal{X}}$

function $\hat{\alpha}(x)$ ▷ estimates smoothed version of $\log \alpha_{\mathrm{RaMMOBO}}(x)$

  $y^{(s)} \leftarrow g_{\mathcal{D}}(x, \xi^{(s)}),\ r_i^{(s)} \leftarrow g_{\mathcal{D}}(x_i, \xi_i^{(s)})$ ▷ for all s, i

  $\Delta_w^{(s)} \leftarrow \min_{i: x_i \in \hat{\mathcal{X}}} \max_m w_m^{(s)} \big(y_m^{(s)} - r_{i,m}^{(s)}\big)$

  $\ell^{(s)} \leftarrow \sum_m \min\big\{ \log \varphi_\tau(\Delta_w^{(s)}) - \log w_m^{(s)},\ \log(y_m^{(s)} - z_m)_+ \big\}$ ▷ $\log L_{w^{(s)}}$

  return $\mathrm{logsumexp}_s\, \ell^{(s)} - \log S$

end function

return $x' \in \arg\max_{x \in \mathcal{X}} \hat{\alpha}(x)$ ▷ multi-start L-BFGS-B, best quasi-random starts

Crucially, the bound need not be tight to be useful: Since an acquisition function is used only through its maximizer, it suffices that the two functions correlate well: Empirically, we measure rank correlation and find strong agreement between RaMMOBO and qLogNEHVI for random inputs on the benchmark problems we consider; see Figure 8 and Appendix C.8.

To form the reference set $\hat{\mathcal{X}}$ in practice, we use the subset of inputs x_i in $\mathcal{D}$ whose posterior means $\mathbb{E}[Y(x_i) \mid \mathcal{D}]$ are nondominated. As a second approximation, we sample $Y(x)$ and the selected reference outcomes independently from their posterior marginals for computational convenience. The qLogNEHVI baseline instead prunes inputs using posterior samples and samples candidate and baseline outcomes jointly.

For numerical stability, we follow Ament et al. (2023) and replace the ReLU function in $\Delta_w(y; R) = [\min_i \max_m w_m(y_m - r_{i,m})]_+$ with the smooth fatplus approximation. As shown in Algorithm 1, we optimize the resulting Monte Carlo estimate in log space using multi-start L-BFGS-B (Zhu et al., 1997), reusing the weight draws $w^{(s)}$ and the random inputs to the posterior sampler throughout optimization. The NEHVI lower bound concerns the unsmoothed utility and exact expectation under the same joint posterior and full reference set as NEHVI; reference-set reduction, independent sampling, and smoothing are implementation approximations.

For batches ($q > 1$), we select inputs one at a time. The first input is selected using the single-candidate procedure above. Each selected input then remains *pending*: it belongs to the batch, but its outcome has not yet been observed. To choose the next input, we jointly sample possible outcomes at the pending inputs from the surrogate posterior and add them to the reference set for each Monte Carlo sample. Candidate outcomes are sampled conditionally on these pending outcomes, preserving their posterior dependence. The acquisition therefore rewards improvement beyond what the earlier batch members may already provide. We repeat this process without collecting new observations or refitting the surrogate. Once q inputs are selected, we evaluate them in parallel (Appendix B).

4 Related Work

MOBO acquisition functions differ in how directly they target hypervolume improvement and in the computational machinery required to do so. Scalarization-based methods such as ParEGO and its noisy batch variant qNParEGO apply improvement criteria to random augmented Chebyshev scalarizations (Knowles, 2006; Daulton et al., 2020; 2021), avoiding explicit hypervolume computation but conditioning each acquisition value on sampled preference directions. Relatedly, Zhang & Golovin (2020) express hypervolume as an expectation over random directions, which can be used as a partition-free Monte Carlo approximation to HVI. Although inexpensive per sample, this approximation performs substantially worse than RaMMOBO on most benchmarks (Figure 12), and its concentration guarantee has an exponential dependence on M; see Appendix D.1.

Hypervolume-based methods retain the most direct improvement-based interpretation. qNEHVI integrates hypervolume improvement over uncertainty in previously evaluated outcomes (Daulton

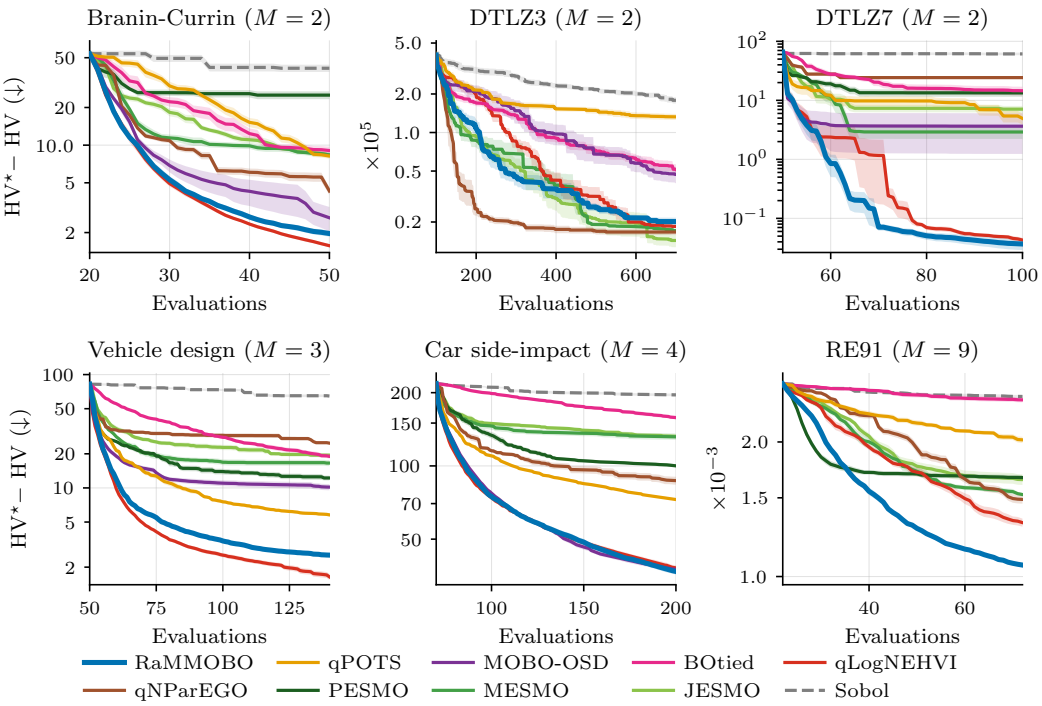


Figure 3: **Log hypervolume-difference against evaluations** on three synthetic and three real-world problems, $M = 2$ to 9, in the sequential ($q = 1$) setting. The x-axis counts function evaluations including the shared initial design. As exact box decompositions are computationally intractable at $M = 9$, qLogNEHVI uses approximate decompositions at $\alpha = 10^{-4}$.

et al., 2021), but exact hypervolume computation has super-polynomial worst-case scaling in M (Yang et al., 2019). qLogNEHVI improves the numerical behavior of this acquisition while retaining the same underlying geometric computation (Ament et al., 2023). Other approaches introduce different expensive subroutines: PESMO, MESMO, and multi-objective JES require auxiliary multi-objective optimization to construct posterior Pareto samples (Hernández-Lobato et al., 2016; Belakaria et al., 2019; Tu et al., 2022), qPOTS applies NSGA-II to sampled objectives (Deb et al., 2002; Renganathan & Carlson, 2025), MOBO-OSD combines local Pareto-front estimation with hypervolume-based batch selection (Ngo et al., 2025), and BOtied fits a multivariate CDF and optimizes its acquisition using CMA-ES (Park et al., 2024).

Expected maximin improvement (EMmI) (Svenson & Santner, 2016) is the construction most closely related to RaMMOBO because it also uses a partition-free margin that can be evaluated with cost linear in M. However, its reference-point-free margin utility is only weakly Pareto-monotone and is not generally a lower bound on EHVI, while the original method treats observed outcomes as deterministic and relies on derivative-free acquisition optimization. RaMMOBO retains the computational simplicity of this margin construction while turning it into a reference-point-aware volume utility, averaging over random objective weights, and integrating uncertainty in the reference outcomes. This yields a partition-free lower bound on NEHVI that is strictly Pareto-monotone on hypervolume-improving outcomes and can be optimized with gradient-based methods, without box decompositions or inner multi-objective optimization.

5 Experiments

In our experiments, we show that RaMMOBO becomes most advantageous for a large number of objectives, where hypervolume-based MOBO becomes computationally expensive. Across synthetic and real-world benchmarks with $M = 2$ to 14, we observe competitive performance at lower objective counts, increasing gains over qLogNEHVI as M grows, and acquisition optimization roughly

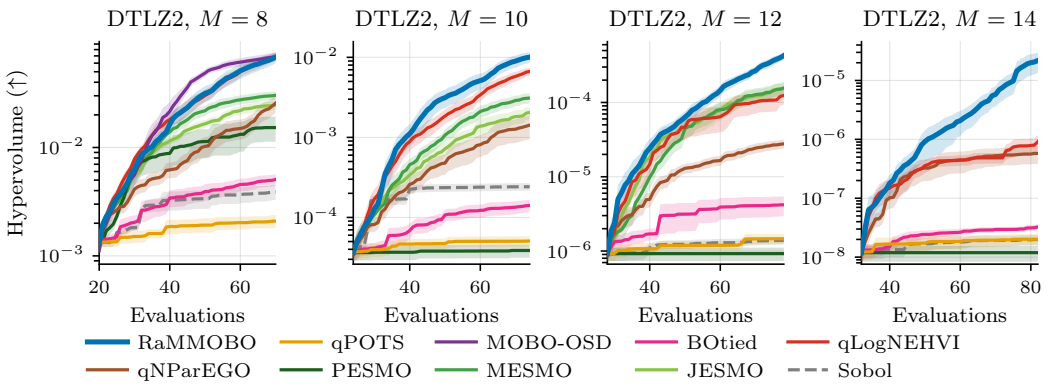


Figure 4: **Log hypervolume against evaluations for increasing number of objectives.** DTLZ2 at $M \in \{8, 10, 12, 14\}$, $2M + 4$ initial points and 50 rounds. As exact box decompositions are computationally intractable, qLogNEHVI uses approximate decomposition at $\alpha = 10^{-3}$.

two orders of magnitude faster than qLogNEHVI. We evaluate sample efficiency and computational cost in noiseless and noisy, sequential ($q = 1$) and batch ($q \in \{2, 4\}$) settings.

Benchmark problems. We evaluate RaMMOBO on a mix of standard synthetic test functions and engineering design problems, spanning low- to many-objective regimes. For the main comparison in Figure 3, we use three synthetic and three real-world problems with $M = 2$ to 9 objectives. The synthetic problems are Branin–Currin (Branin, 1972; Currin et al., 1991) ($d = 2$, $M = 2$), DTLZ3 ($d = 10$, $M = 2$), and DTLZ7 ($d = 5$, $M = 2$) (Deb et al., 2005). The real-world problems are Vehicle design (Liao et al., 2008) ($d = 5$, $M = 3$), Car side-impact (Jain & Deb, 2013) ($d = 7$, $M = 4$), and RE91 (RE9-7-1, car cab design; $d = 7$, $M = 9$) from the RE suite (Tanabe & Ishibuchi, 2020). To isolate how performance changes with the number of objectives, we additionally evaluate DTLZ2 (Deb et al., 2005) at $M \in \{8, 10, 12, 14\}$ with $d = M + 1$ (Figure 4).

Baselines. We compare against eight baselines spanning the main families of MOBO methods. These include qLogNEHVI (Ament et al., 2023), the numerically stabilized form of qNEHVI (Daulton et al., 2021); qNParEGO (Daulton et al., 2020), the Monte Carlo batch variant of ParEGO (Knowles, 2006); the information-theoretic methods PESMO (Hernández-Lobato et al., 2016), MESMO (Belakaria et al., 2019), and JESMO (Tu et al., 2022); the Thompson-sampling method qPOTS (Renganathan & Carlson, 2025); the search-direction-based MOBO-OSD (Ngo et al., 2025); and the rank-based BOtied (Park et al., 2024). For MESMO and JESMO, we use the lower-bound implementations in BoTorch. We additionally report scrambled Sobol sampling (Sobol', 1967; Owen, 1998) as a no-model reference.

Reliable hypervolume baselines are particularly important for this comparison. In our audit, the qNEHVI baseline reported by BOtied combines standardized outputs with an unstandardized reference point, while rerunning the released qPOTS code does not reproduce the reported stagnation of its qNEHVI baseline on Branin–Currin. We therefore use BoTorch's qLogNEHVI with consistent output and reference-point scaling throughout.

Experimental setup. All methods share the same surrogate, i.e., an independent GP per objective, with a Matérn-$5/2$ kernel, inputs in the unit cube, standardized outcomes and inferred observation noise. Therefore, differences between methods are differences between acquisition functions. Within a seed, every method starts from the same initial design. In every experiment, we run 10 independent seeds and report the mean with a ± 1 standard-error band throughout. RaMMOBO does not use any problem-specific or M-dependent parameters. Experimental details can be found in Appendix C. Methods that exceed the available time or memory budget are omitted from the corresponding panels; Appendix C.4 details these limits. For $M \geq 8$, exact hypervolume decomposition is intractable, so qLogNEHVI uses approximate decompositions: $\alpha = 10^{-3}$ on DTLZ2 and $\alpha = 10^{-4}$ on RE91; Appendix C.4 explains the approximation parameter α.

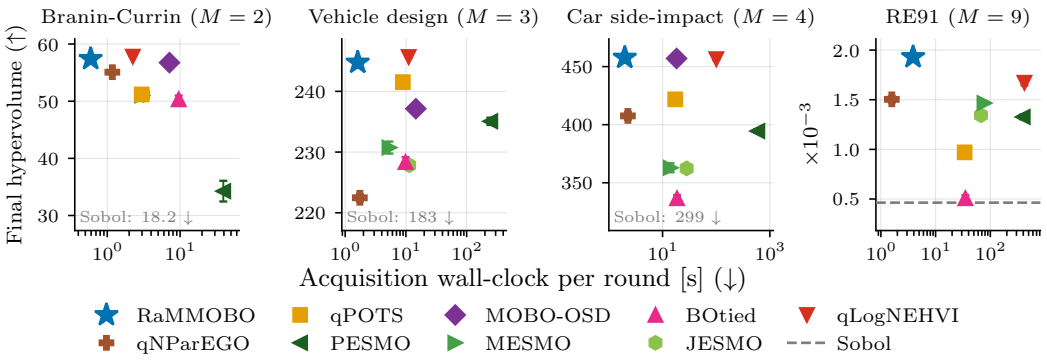


Figure 5: **Hypervolume against wall-clock.** Final hypervolume against wall-clock time (seconds) needed to optimize an acquisition function to propose the next input (log x; lower-right is worse, upper-left is better). Sobol's acquisition is practically free; its final hypervolume is shown as a dashed line, or as a text label with a downward arrow where it lies below the plotted range. On RE91, RaMMOBO's acquisition optimization is roughly two orders of magnitude faster than qLogNEHVI at $\alpha = 10^{-4}$. See Appendix D for more panels.

Metrics. We measure hypervolume (Zitzler & Thiele, 1999) at a reference point fixed before optimization and shared across methods. Figure 3 reports $\mathrm{HV}^\star - \mathrm{HV}$ on a log scale (lower is better), where $\mathrm{HV}^\star$ is an empirical reference value defined in Appendix C.5; Figure 4 reports hypervolume itself (higher is better). We also evaluate front coverage using the reference-point-free IGD^+ metric (Ishibuchi et al., 2015) in Figure 6.

Results and discussion. On the six benchmark problems (Figure 3), qNParEGO improves fastest on DTLZ3, while qLogNEHVI obtains the smallest final hypervolume differences on Branin–Currin and Vehicle design. RaMMOBO performs comparably to qLogNEHVI on Car side-impact and obtains the smallest final hypervolume differences on DTLZ7 and RE91. Figure 2 isolates the effect of the two modifications to EMmI: adding clipping at z and then random weighting improves hypervolume, most visibly at higher M. The IGD^+ results shown in Figure 6 also support RaMMOBO's strong Pareto front coverage, including an advantage over qLogNEHVI on Vehicle design.

As M grows, Figure 4 shows that RaMMOBO is competitive at $M = 8$ and achieves higher mean final hypervolume than approximate qLogNEHVI at $M \geq 10$, with a widening gap at larger M: at $M = 14$, it exceeds all shown competitors by more than an order of magnitude.

Figure 5 shows that this performance comes at substantially lower computational cost. RaMMOBO achieves competitive final hypervolume while optimizing its acquisition function orders of magnitude faster than several competing methods (see Appendix C.6 for details). These conclusions extend to Gaussian observation noise and to batch MOBO where $q \in \{2, 4\}$ (Figures 9, 10, and 11).

6 Conclusion

We introduced RaMMOBO, which uses a simple acquisition function that makes hypervolume-guided search computationally tractable as the number of objectives grows. Its central idea is to measure max-margin box volumes under random reweightings of the objective space. Averaging over weights and posterior uncertainty yields a lower bound on (N)EHVI with evaluation cost linear in M, avoiding expensive box decompositions. The utility is strictly Pareto-monotone on outcomes that improve hypervolume, supports noisy observations, and admits efficient gradient-based acquisition optimization. On synthetic and real-world benchmarks, RaMMOBO matches or outperforms state-of-the-art methods, with the largest gains at higher objective counts and acquisition optimization roughly two orders of magnitude faster than qLogNEHVI on RE91. These findings demonstrate that a tractable geometric lower bound can guide effective multi-objective search even where directly computing expected hypervolume improvement becomes prohibitive.

AI USE STATEMENT

The central idea of this work—lower-bounding hypervolume improvement by the volume of a margin box, which is easy to compute—was developed by the authors without generative AI. We used generative AI tools (large language models as coding and writing assistants) for the following tasks with required disclosure: refining the theoretical formulation, including the precise statements of the lower-bound and monotonicity results, and assisting in writing their proofs (Appendix A.1–A.2); implementing the method and the experimental pipeline; translating the reference implementations of qPOTS and MOBO-OSD into our code base; and giving feedback on the experimental design and supporting the analysis of results. We have not used generative AI tools to generate data sets, and qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure: searching and summarizing related literature, writing plotting code, and drafting and editing parts of the paper.

We have reviewed all AI-assisted work. Each proof was checked line by line by at least two authors, and the bounds were additionally verified numerically against exact hypervolume computations on randomly generated instances. AI-assisted code was reviewed by the authors, the transcribed baselines were compared against the authors' reference implementations, including reproductions of their published results where these were available. All reported numbers were recomputed from the stored raw experiment outputs, and all literature identified with AI assistance was checked against the original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is methodological: it proposes an acquisition function for multi-objective Bayesian optimization and evaluates it on standard synthetic test functions and publicly available engineering benchmark problems. It involves no human subjects, personal or sensitive data, or newly released data sets. As with machine learning and Bayesian optimization in general, more efficient optimization methods can be applied to harmful ends, for example to design materials, molecules, or systems with harmful properties; our method does not specifically enable such uses, and responsibility for the choice of objectives lies with those who apply it. We are not aware of conflicts of interest, and we do not see ethical concerns specific to this work that require further discussion.

REPRODUCIBILITY STATEMENT

RaMMOBO is specified in Section 3 and, as implemented, in Algorithm 1 (sequential) and Algorithm 2 (batch). The assumptions and complete proofs of the lower bound and of strict Pareto monotonicity are given in Appendix A.1 and Appendix A.2. Appendix C describes every experiment: the benchmark problems with their dimensions, budgets, reference points and noise levels, the shared surrogate model, the settings of RaMMOBO and of every baseline, the metrics, and the wall-clock measurement protocol. We release an anonymized code repository at `https://anonymous.4open.science/r/rammobo` that contains our implementation of RaMMOBO, all baselines as run in our experiments, a version-pinned environment specification of all software dependencies, and the configurations and scripts that reproduce every experiment and figure in this paper.

REFERENCES

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/hash/419f72cbd568ad62183f8132a3605a2a-Abstract-Conference.html`.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/82edc5c9e21035674d481640448049f3-Abstract.html.

Franklin H. Branin. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development*, 16(5):504–522, 1972.

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.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6fec24eac8f18ed793f5eaad3dd7977c-Abstract.html.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/11704817e347269b7254e744b5e22dac-Abstract.html.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pp. 105–145. Springer, 2005.

Michael T. M. Emmerich, Kyriakos C. Giannakoglou, and Boris Naujoks. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

Peter J. Fleming, Robin C. Purshouse, and Robert J. Lygoe. Many-objective optimization: An engineering design perspective. In *International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, volume 3410 of *Lecture Notes in Computer Science*, pp. 14–32, 2005.

Jenna C. Fromer and Connor W. Coley. Computer-aided multi-objective optimization in small molecule discovery. *Patterns*, 4(2):100678, 2023. doi: 10.1016/j.patter.2023.100678.

Daniel Hernández-Lobato, José Miguel Hernández-Lobato, Amar Shah, and Ryan P. Adams. Predictive entropy search for multi-objective Bayesian optimization. In *International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1492–1501. PMLR, 2016. URL https://proceedings.mlr.press/v48/hernandez-lobatoa16.html.

Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, pp. 110–125, 2015.

Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2013.

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

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. doi: 10.1109/TEVC.2005.851274.

Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys*, 48(1):13:1–13:35, 2015. doi: 10.1145/2792984.

Xingtao Liao, Qing Li, Xujing Yang, Weigang Zhang, and Wei Li. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization*, 35(6):561–569, 2008.

Lam Ngo, Huong Ha, Jeffrey Chan, and Hongyu Zhang. MOBO-OSD: Batch multi-objective Bayesian optimization via orthogonal search directions. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/18e780ccbd60697ea4a0184f1e680660-Abstract-Conference.html.

Art B. Owen. Scrambling Sobol' and Niederreiter–Xing points. *Journal of Complexity*, 14(4): 466–489, 1998.

Ji Won Park, Natasa Tagasovska, Michael Maser, Stephen Ra, and Kyunghyun Cho. BOtied: Multi-objective Bayesian optimization with tied multivariate ranks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39813–39833. PMLR, 2024. URL https://proceedings.mlr.press/v235/park24k.html.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Ashwin Renganathan and Kade Carlson. qPOTS: Efficient batch multiobjective Bayesian optimization via Pareto optimal Thompson sampling. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4051–4059. PMLR, 2025. URL https://proceedings.mlr.press/v258/renganathan25a.html.

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. doi: 10.1109/JPROC.2015.2494218.

Ilya M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

Joshua Svenson and Thomas Santner. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics & Data Analysis*, 94:250–264, 2016. doi: 10.1016/j.csda.2015.08.011.

Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9922–9938, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4086fe59dc3584708468fba0e459f6a7-Abstract.html.

Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 75(1):3–34, 2019. doi: 10.1007/s10898-019-00798-7.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11096–11105. PMLR, 2020. URL https://proceedings.mlr.press/v119/zhang20i.html.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997. doi: 10.1145/279232.279236.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

A PROOFS

A.1 HYPERVOLUME IMPROVEMENT LOWER BOUND

We prove the pointwise lower bound in Eq. (15). Throughout, all objectives are maximized, $\preceq$ denotes componentwise order, and λ denotes the M-dimensional Lebesgue measure.

Let $R = \{r_1, \dots, r_N\} \subset \mathbb{R}^M$ be a finite, nonempty reference set, let $z \in \mathbb{R}^M$ be the hypervolume reference point, and let $w \in \mathbb{R}^M_{>0}$. Recall the weighted margin and box lengths from Eqs. (13)–(14):

$$\Delta_w(y; R) = \min_i \max_m w_m(y_m - r_{i,m})_+, \qquad \ell_{w,m}(y; R) = \min\left\{ \frac{\Delta_w(y; R)}{w_m}, (y_m - z_m)_+ \right\}.$$

Then, for every $y \in \mathbb{R}^M$,

$$0 \le L_w(y; R) = \prod_{m=1}^{M} \ell_{w,m}(y; R) \le \mathrm{HVI}_z(y; R). \tag{19}$$

The bound does not require R to be nondominated or the weights to be normalized.

Proof. Fix y. If $L_w(y; R) = 0$, the claim follows from the nonnegativity of hypervolume improvement.

Suppose $L_w(y; R) > 0$. Then $\varepsilon := \Delta_w(y; R) > 0$ and $y_m > z_m$ for every m, so

$$0 < \ell_{w,m}(y; R) = \min\left\{ \frac{\varepsilon}{w_m}, y_m - z_m \right\}.$$

Consider the axis-aligned box

$$B = \prod_{m=1}^{M} \big[y_m - \ell_{w,m}(y; R), y_m \big]. \tag{20}$$

Since $\ell_{w,m}(y; R) \le y_m - z_m$, every $u \in B$ satisfies $z \preceq u \preceq y$. Hence $B \subseteq D_z(\{y\})$, with D_z defined in Eq. (4). For every $r_i \in R$, the definition of ε gives $\max_m w_m(y_m - r_{i,m})_+ \ge \varepsilon$. Choose a coordinate m_i with

$$w_{m_i}(y_{m_i} - r_{i,m_i})_+ \ge \varepsilon > 0.$$

Strict positivity makes the positive part inactive in this coordinate, so

$$y_{m_i} - r_{i,m_i} \ge \frac{\varepsilon}{w_{m_i}} \ge \ell_{w,m_i}(y; R).$$

Every $u \in \mathrm{int}(B)$ therefore satisfies $u_{m_i} > y_{m_i} - \ell_{w,m_i}(y; R) \ge r_{i,m_i}$, so r_i cannot weakly dominate u. Since this holds for every reference outcome,

$$\mathrm{int}(B) \subseteq D_z(\{y\}) \setminus D_z(R).$$

All sets above are measurable, since $D_z(R)$ is a finite union of boxes. The boundary of B has Lebesgue measure zero, so monotonicity of measure gives

$$\begin{aligned} L_w(y; R) &= \lambda(B) = \lambda(\mathrm{int}(B)) \\ &\le \lambda(D_z(\{y\}) \setminus D_z(R)) = \mathrm{HVI}_z(y; R). \end{aligned}$$

□

Setting $w = \mathbf{1}$ recovers the unweighted bound in Eq. (12). Averaging over W preserves the inequality and yields $\bar{L}(y; R) \le \mathrm{HVI}_z(y; R)$. Taking the posterior expectation over the candidate outcome gives the EHVI lower bound. In the noisy setting, the pointwise bound applies to each realization of the candidate and reference outcomes. Averaging over their joint posterior and an independent weight vector W yields

$$\mathbb{E}_{p(Y(x),\mathcal{R}|\mathcal{D})}\mathbb{E}_W[L_W(Y(x); \mathcal{R})] \le \mathbb{E}_{p(Y(x),\mathcal{R}|\mathcal{D})}[\mathrm{HVI}_z(Y(x); \mathcal{R})] = \mathrm{NEHVI}_z(x),$$

which establishes the acquisition bound in Eq. (18).

A.2 STRICT PARETO MONOTONICITY

We prove the strict Pareto monotonicity claim in Eq. (17):

$$y \succ y', \qquad \mathrm{HVI}_z(y'; R) > 0 \quad \Longrightarrow \quad \bar{L}(y; R) > \bar{L}(y'; R). \tag{21}$$

We work in the multi-objective setting $M \geq 2$. Here $R = \{r_1, \ldots, r_N\} \subset \mathbb{R}^M$ is a finite, nonempty reference set, $z \in \mathbb{R}^M$ is the hypervolume reference point, and $\bar{L}$ is the averaged box utility

$$\bar{L}(y; R) = \mathbb{E}_W[L_W(y; R)], \qquad W \sim M \cdot \mathrm{Dirichlet}(\mathbf{1}),$$

with Δ_w, $\ell_{w,m}$, and L_w defined in Eqs. (13)–(15).

Proof. Assume $y \succ y'$ and $\mathrm{HVI}_z(y'; R) > 0$. Then $y'_m > z_m$ for every m: otherwise $D_z(\{y'\})$ has zero M-dimensional measure. Also, no $r_i \in R$ weakly dominates y', since that would imply $D_z(\{y'\}) \subseteq D_z(R)$. Thus each reference has a positive gap at y'. For every $w > 0$, its weighted maximum is positive; finiteness of R makes their minimum $\Delta_w(y'; R)$ positive as well. All box sides at y' are therefore positive.

Only weight ratios matter. Indeed, for every $c > 0$,

$$\Delta_{cw}(u; R) = c\,\Delta_w(u; R), \qquad \ell_{cw,m}(u; R) = \ell_{w,m}(u; R), \qquad L_{cw}(u; R) = L_w(u; R).$$

We may therefore construct convenient unnormalized weights and normalize them afterward without changing either box.

First, L_w is weakly Pareto-monotone for every fixed $w > 0$. Indeed, $y \succeq y'$ implies $(y_m - r_{i,m})_+ \geq (y'_m - r_{i,m})_+$ for all i, m. Multiplication by positive weights, followed by the maximum over m and the minimum over i, preserves this inequality. Hence $\Delta_w(y; R) \geq \Delta_w(y'; R)$. Since also $(y_m - z_m)_+ \geq (y'_m - z_m)_+$, every box side is nondecreasing:

$$\ell_{w,m}(y; R) \geq \ell_{w,m}(y'; R).$$

Hence

$$L_w(y; R) \geq L_w(y'; R) \qquad \text{for every } w > 0. \tag{22}$$

We next construct a weight vector for which the inequality is strict. Choose a coordinate k and set $\delta := y_k - y'_k > 0$. It suffices to make the kth side strictly larger. Partition the reference indices into

$$\mathcal{I} = \{i : \exists m \neq k,\ y'_m > r_{i,m}\}, \qquad \mathcal{J} = \{1, \ldots, N\} \setminus \mathcal{I}.$$

The references in $\mathcal{I}$ have a positive gap outside k; those in $\mathcal{J}$ are exceeded by y' only in coordinate k.

We choose weights that elongate the unclipped box along the improved coordinate k. Set $w_k = 1$ and $w_m = C$ for $m \neq k$. There are only finitely many positive gaps outside coordinate k, so we can choose $C > 1$ such that

$$C(y'_m - r_{i,m})_+ > y_k - z_k \qquad \text{whenever } m \neq k \text{ and } y'_m > r_{i,m}.$$

If no such gap exists, any $C > 1$ suffices.

For $i \in \mathcal{I}$, the weight choice and $y \succeq y'$ imply that its weighted maximum at y exceeds the clipping threshold $y_k - z_k$. For $i \in \mathcal{J}$, all gaps outside k vanish at y'. Since r_i does not weakly dominate y', we have $y'_k > r_{i,k}$, and its weighted maximum at y' equals $y'_k - r_{i,k}$. This gap increases by δ at y, giving

$$\max_m w_m(y_m - r_{i,m})_+ \geq y_k - r_{i,k} = y'_k - r_{i,k} + \delta \geq \Delta_w(y'; R) + \delta.$$

Taking the minimum over reference outcomes therefore gives

$$\Delta_w(y; R) \geq \min\{y_k - z_k, \Delta_w(y'; R) + \delta\}.$$

The clipping limit also increases by δ, since $y_k - z_k = y'_k - z_k + \delta$. Using $w_k = 1$, clipping therefore yields

$$\ell_{w,k}(y; R) \geq \min\{y_k - z_k, \Delta_w(y'; R) + \delta\} = \ell_{w,k}(y'; R) + \delta.$$

The kth side thus increases strictly. All other sides are nondecreasing and all sides at y' are positive, so

$$L_w(y; R) > L_w(y'; R).$$

Finally, normalize w to $w^0 = Mw/\sum_m w_m$ in the relative interior of the normalized simplex,

$$\mathcal{S}_M^\circ = \left\{ w \in \mathbb{R}_{>0}^M : \sum_m w_m = M \right\},$$

preserving $L_{w^0}(y; R) > L_{w^0}(y'; R)$ by scale invariance.

For fixed y and y', the difference $L_w(y; R) - L_w(y'; R)$ is continuous in w on $\mathcal{S}_M^\circ$: its definition uses finite minima and maxima, positive parts, products, and division by strictly positive coordinates. The strict inequality therefore holds on a relatively open neighborhood $U \subset \mathcal{S}_M^\circ$ of w^0. The distribution $M \cdot \mathrm{Dirichlet}(\mathbf{1})$ has strictly positive density throughout $\mathcal{S}_M^\circ$ with respect to the $(M-1)$-dimensional measure on its affine hull, so $\mathbb{P}(W \in U) > 0$. Both expectations are finite, since

$$0 \le L_w(u; R) \le \prod_{m=1}^{M} (u_m - z_m)_+ \qquad \text{for every } w > 0.$$

The difference is nonnegative for every weight vector by (22) and strictly positive on an event of positive probability. Consequently,

$$\bar{L}(y; R) - \bar{L}(y'; R) = \mathbb{E}_W[L_W(y; R) - L_W(y'; R)] > 0. \qquad \square$$

B BATCH-RAMMOBO

Algorithm 2 constructs a batch by selecting its members greedily. Starting from an empty pending set P, each step maximizes the acquisition against the evaluated reference inputs $\hat{\mathcal{X}}$ and the inputs already in P, then adds the selected input to P. The observed data, surrogate parameters, and reference inputs remain fixed, while conditional predictions account for the pending outcomes. The q selected inputs are then evaluated in parallel.

For each Monte Carlo sample s, we draw the latent outcomes at the pending inputs jointly and hold them fixed during the next acquisition maximization. Write these outcomes as $R_P^{(s)}$. We draw the candidate outcome conditional on $Y(P) = R_P^{(s)}$, giving the joint posterior law $p(Y(x), Y(P) \mid \mathcal{D})$ while keeping the pending outcomes fixed as x varies. In Algorithm 2, this is implemented by applying the posterior sampler $g_{\mathcal{D}}$ of Algorithm 1 jointly to (x, P) with fixed base randomness $\xi^{(s)}$. Outcomes at the evaluated reference inputs are sampled independently of one another and of the candidate and pending outcomes. The sampled pending outcomes augment the reference set, so each choice accounts for improvements the earlier choices may already provide.

The weight draws and base randomness remain fixed during each inner optimization. Pending outcomes are sampled for the updated set P before the next greedy step; this requires no new observations or surrogate refitting. When $P = \emptyset$, $R_P^{(s)}$ is empty and the conditional candidate distribution reduces to $p(Y(x) \mid \mathcal{D})$, recovering the single-candidate procedure in Algorithm 1.

Algorithm 2 RaMMOBO's inner loop, batch setting ($q > 1$)

Require: Data $\mathcal{D}$; surrogate with differentiable samples $Y(x) = g_{\mathcal{D}}(x, \xi)$; reference point z; batch size q; number of Monte Carlo samples S; fatplus φ_τ

$\hat{\mathcal{X}} \leftarrow \{x_i : (x_i, y_i) \in \mathcal{D},\ \mathbb{E}[Y(x_i) \mid \mathcal{D}] \text{ is nondominated}\}$
Draw and fix $w^{(s)} \sim M \cdot \mathrm{Dirichlet}(\mathbf{1})$ and independent base samples $\xi^{(s)}, \xi_i^{(s)},\ s \le S, x_i \in \hat{\mathcal{X}}$
function $\hat{\alpha}(x; P)$ ▷ estimates smoothed $\log \alpha_{\mathrm{RaMMOBO}}(x)$ given pending points P
    $\big(y^{(s)}, \{r_p^{(s)}\}_{p \in P}\big) \leftarrow g_{\mathcal{D}}\big((x, P), \xi^{(s)}\big)$ jointly; $r_i^{(s)} \leftarrow g_{\mathcal{D}}(x_i, \xi_i^{(s)}), x_i \in \hat{\mathcal{X}}$
    $\Delta_w^{(s)} \leftarrow \min_{r \in \{r_i^{(s)} : x_i \in \hat{\mathcal{X}}\} \cup \{r_p^{(s)} : p \in P\}} \max_m w_m^{(s)} \big(y_m^{(s)} - r_m\big)$
    $\ell^{(s)} \leftarrow \sum_m \min\big\{\log \varphi_\tau(\Delta_w^{(s)}) - \log w_m^{(s)},\ \log(y_m^{(s)} - z_m)_+\big\}$ ▷ $\log L_{w^{(s)}}$
    return $\mathrm{logsumexp}_s\, \ell^{(s)} - \log S$
end function
$P \leftarrow \emptyset$
for $j = 1, \ldots, q$ **do** ▷ greedy batch
    $x'_j \in \arg\max_{x \in \mathcal{X}} \hat{\alpha}(x; P)$ ▷ multi-start L-BFGS-B, best quasi-random starts
    $P \leftarrow P \cup \{x'_j\}$
end for
return P

C Experimental Details

C.1 Benchmark problems and protocol

All objectives are maximized; problems that BoTorch defines for minimization are negated, together with their reference points. Table 1 lists every problem. Branin–Currin, DTLZ3, DTLZ7, Vehicle design and Car side-impact are the sequential problems of Renganathan & Carlson (2025), used as implemented in BoTorch (Balandat et al., 2020), with an initial design of $n_0 = 10d$ points and their numbers of iterations; note that Renganathan & Carlson (2025) count iterations after the initial design, so their evaluation budgets equal ours plus n_0. For DTLZ3 we report the first 600 of the 700 iterations. RE91 (car cab design) is our port of the RE suite implementation (Tanabe & Ishibuchi, 2020), verified against it on 512 designs, with its four stochastic variables $x_8, \ldots, x_{11}$ held at their means $(0.345, 0.192, 0, 0)$, so that the problem is deterministic and observation noise is controlled by the protocol alone. The many-objective study uses DTLZ2 (Deb et al., 2005) with $d = M + 1$ (i.e. $k = 2$ distance variables) and follows Park et al. (2024) in *maximizing* its objectives (BoTorch's DTLZ2 without negation). Its Pareto front is therefore the part of the sphere of radius $1 + \max g = 1.5$ in the positive orthant, not the unit-sphere front of the minimization form. RE91 and DTLZ2 start from $n_0 = 2M + 4$ points and run 50 iterations.

Reference points. Hypervolume and every hypervolume-based acquisition depend on z, while several baselines (qPOTS, qNParEGO, BOtied, PESMO, MESMO, JESMO, Sobol) do not use one. To compare both groups fairly, we fix z per problem before any run, never tune it, and give every method that consumes a reference point (RaMMOBO, qLogNEHVI, MOBO-OSD) exactly the z that it is scored with. Following Daulton et al. (2021), z is BoTorch's built-in point wherever the problem defines one. For RE91 we take the per-objective worst value over 2^{17} Sobol designs, rounded up (a design-space bound that uses no knowledge of the Pareto front; its first coordinate lies slightly inside the front's range of f_1). For DTLZ2 we use 0.1 below the objectives' lower bound of 0. On Vehicle design and Car side-impact, BoTorch derives z from an approximation of the true Pareto front (Tanabe & Ishibuchi, 2020); this informs the methods that consume z about the extent of the front and can favor them over reference-point-free baselines on these two problems, which is why we also report a reference-free metric (IGD$^+$) in Figure 6.

Noise. In the noisy setting, every observation, including the initial design, is corrupted by independent Gaussian noise with the standard deviations of Table 1: 1% of each objective's range, estimated as the maximum minus the minimum over $2 \cdot 10^4$ Sobol designs. Hypervolume is always computed from the noiseless objective values of the evaluated inputs. The DTLZ2 study is noiseless.

Table 1: Benchmark problems. n_0: initial design; T: iterations at $q = 1$ (evaluations $n_0 + T$); z: reference point, used both inside every acquisition that takes one and for computing hypervolume (maximization signs); σ: per-objective noise standard deviation in the noisy setting.

Problem	d	M	n_0	T	reference point z	noise σ
Branin–Currin	2	2	20	30	$(-18, -6)$	(3.02, 0.126)
DTLZ3	10	2	100	600	$(-10^4, -10^4)$	(16.0, 16.7)
DTLZ7	5	2	50	50	$(-15, -15)$	(0.010, 0.174)
Vehicle design	5	3	50	90	$(-1864.72, -11.82, -0.290)$	(0.377, 0.051, 0.002)
Car side-impact	7	4	70	130	$(-45.49, -4.51, -13.34, -10.39)$	(0.222, 0.008, 0.021, 0.124)
RE91	7	9	22	50	$-(41.30, 1.03, 0.90, 0.78, 1.47, 1.13, 1.11, 1.03, 1.02)$	(0.236, 0.008, 0.005, 0.004, 0.008, 0.005, 0.002, 0.002, 0.002)
DTLZ2	M+1	6–14	$2M$+4	50	$-0.1 \cdot \mathbf{1}$	— (noiseless only)

Seeds and initial designs. Every configuration is run with 10 seeds. For each seed, the initial design is a scrambled Sobol sample drawn after seeding all random number generators, and is therefore identical across methods (including its noise). A curve is drawn over the seeds that reach the problem's horizon; a method with fewer than 5 such seeds is omitted from that panel.

Batch setting. For $q \in \{2, 4\}$ we keep the evaluation budget $n_0 + T$ of the sequential setting and run $\lceil T/q \rceil$ iterations of q points each; curves are shown up to the largest whole batch within the budget (e.g. 48 of 50 evaluations on Branin–Currin at $q = 4$). BoTorch-based acquisitions and RaMMOBO (Algorithm 2) construct batches greedily with the previously selected points as pending points; qNParEGO optimizes q independently drawn scalarizations sequentially; qPOTS returns the q points of its Pareto set approximation farthest from the observed inputs; MOBO-OSD uses its kriging-believer batch selection; BOtied runs one CMA-ES optimization per batch point.

C.2 Surrogate model

All methods use the same surrogate: BoTorch's `SingleTaskGP` with an independent GP per objective, a scaled Matérn-$5/2$ kernel with one lengthscale per input dimension, a constant mean, and a Gaussian likelihood with inferred noise (also in the noiseless setting). Hyperparameters use BoTorch's (v0.11.3) default priors: Gamma$(3, 6)$ on lengthscales, Gamma$(2, 0.15)$ on the output scale, and Gamma$(1.1, 0.05)$ on the noise variance, which is bounded below by 10^{-4}. Inputs are scaled to $[0, 1]^d$ and outcomes are standardized with the mean and standard deviation of the current observations. The model is refitted from scratch in every iteration by maximizing the marginal likelihood with L-BFGS-B (up to five attempts on fitting errors). The reference point z is expressed in the same standardized units in every iteration. All computations use double precision.

C.3 RaMMOBO

We implement RaMMOBO as in Algorithms 1 and 2 with the same settings on every problem, for every M, and in both noise settings: $S = 128$ Monte Carlo samples, fatplus temperature $\tau = 10^{-3}$ (in standardized units), and one weight vector $w^{(s)} \sim M \cdot \mathrm{Dirichlet}(\mathbf{1})$ per sample. The set $\hat{\mathcal{X}}$ contains the observed inputs whose posterior means are nondominated. Candidate and reference outcomes are sampled independently from their posterior marginals using scrambled-Sobol normal base samples, which are held fixed during acquisition optimization, as are the weights. The acquisition is optimized with BoTorch's `optimize_acqf`: 512 scrambled-Sobol candidates are evaluated, 10 starting points are selected by BoTorch's Boltzmann heuristic (always including the best candidate), and each is refined by L-BFGS-B (Zhu et al., 1997) for at most 200 iterations using automatic differentiation.

C.4 Baselines

qLogNEHVI, qNParEGO, PESMO, MESMO and JESMO are BoTorch's implementations; qPOTS and MOBO-OSD are transcribed from the authors' reference implementations, while BOtied uses the authors' released implementation. All run inside the same code base, on the surrogate of Appendix C.2 and the same initial designs. Unless stated otherwise, acquisitions are optimized with

the same `optimize_acqf` settings as RaMMOBO (512 candidates, 10 restarts, L-BFGS-B with at most 200 iterations) and Monte Carlo acquisitions use 128 quasi-random samples.

qLogNEHVI (Ament et al., 2023) uses baseline pruning and cached Cholesky sampling. For $M < 8$ it uses the exact box decomposition. For $M \geq 8$ the exact decomposition is intractable, and we use BoTorch's approximate decomposition with relative threshold α: $\alpha = 10^{-4}$ on RE91, and $\alpha = 10^{-3}$ in the DTLZ2 study (Figure 4). Here, α is a relative cell-volume threshold: cells overlapping the Pareto front are discarded once their volume is at most an α fraction of the bounding-box volume enclosing the Pareto set. Smaller α yields finer, more costly decompositions; $\alpha = 0$ recovers exact partitioning. BoTorch's default, $\alpha = 10^{M-8}$, yields no cells at all for large M. At high M the approximate decomposition of a sampled front can also be empty in individual iterations, in which case qLogNEHVI raises an error; we then assign zero improvement to all candidates, as BoTorch's qNEHVI does, and the iteration selects a random candidate. This happened in 0%, 13%, 46% and 58% of iterations at $M = 8, 10, 12, 14$ in the DTLZ2 study. In the batch setting on RE91 we use qLogNEHVI with $\alpha = 10^{-4}$ and the same safeguard.

qNParEGO (Daulton et al., 2020; Knowles, 2006) draws Chebyshev scalarization weights uniformly from the simplex (augmentation 0.05), and optimizes qNEI of the scalarized objective with baseline pruning.

PESMO, MESMO, JESMO (Hernández-Lobato et al., 2016; Belakaria et al., 2019; Tu et al., 2022) use 10 sampled Pareto sets of 10 points each, obtained by random search (population 2000, at most 50 attempts) on posterior sample paths. MESMO and JESMO use BoTorch's lower-bound variants with a box decomposition of each sampled front; PESMO uses expectation propagation with BoTorch's defaults and is optimized without gradients. If no valid set of Pareto samples is obtained within 100 attempts, the iteration falls back to a Sobol point. On RE91 the lower bound of MESMO and JESMO evaluates the log-determinant of a nearly singular matrix; we compute it from eigenvalues floored at 10^{-6} to keep it finite. MESMO and JESMO exceed the available memory (110 GB) at $M = 14$, and PESMO does not reach the 600 iterations of DTLZ3 within our compute budget; they are omitted there.

qPOTS (Renganathan & Carlson, 2025) runs NSGA-II (Deb et al., 2002) for 200 generations with population $100d$ on a posterior sample path and selects the point(s) of the resulting Pareto set farthest from the observed inputs. The authors' default of 10 generations did not reproduce their reported results; 200 generations did.

MOBO-OSD (Ngo et al., 2025) uses $n_\beta = 20$ subproblems and confidence multiplier $\delta = 1.96$; each subproblem is solved by SLSQP from four starts, and candidates are ranked by the hypervolume improvement of their posterior mean at z. We fix four defects of the released code that crash or stall the run and keep its selection of starting points as released. Its per-iteration cost grows rapidly with M (about 20 h per run at $M = 8$; more than 8 h for a single iteration at $M = 10$), so it is run on 5 seeds at $M = 8$ and omitted for $M \geq 9$.

BOtied (Park et al., 2024) is the variant that ranks candidates by a vine-copula CDF (nonparametric TLL family) of the posterior means, optimized by CMA-ES (population 1024, 32 iterations, initial step size 0.2).

Sobol evaluates one freshly scrambled Sobol point per iteration (Sobol', 1967; Owen, 1998).

C.5 Metrics

Hypervolume difference. We compute the exact hypervolume (Zitzler & Thiele, 1999) of the noiseless objective values of all evaluated points at the problem's z (Table 1). $\mathrm{HV}^\star$ is the hypervolume of the nondominated union of all points evaluated by any shown method, in both noise settings and at all batch sizes, at the same z; the plots show $\mathrm{HV}^\star - \mathrm{HV}$ on a log scale, with $\mathrm{HV}^\star$ shared by the sequential, noisy and batch figures. For RE91, computing this union exactly is infeasible; we use a deterministic subset of 1200 of its points that contains RaMMOBO's front, so its $\mathrm{HV}^\star$ is a lower

bound of the union's (yet above every method). The DTLZ2 study (Figure 4) shows hypervolume itself.

IGD$^+$ (Ishibuchi et al., 2015). For each problem, the reference front Z is the nondominated union of the noiseless values of all points evaluated by all methods in all runs (thinned to 6000 points where larger), and every objective is normalized by the range of Z. For an evaluated set A we report $\mathrm{IGD}^+(A) = \frac{1}{|Z|}\sum_{z'\in Z} \min_{a\in A} \|(z'-a)_+\|_2$ (maximization). IGD$^+$ uses no reference point.

Aggregation. Curves show the mean over seeds with a band of ± 1 standard error. The x-axis counts evaluations including the initial design.

C.6 Wall-clock time

Figure 5 plots the final hypervolume (mean ± 1 standard error over 10 seeds, at the z of Table 1) against the time to optimize the acquisition function once. We measure it by replaying each method's recorded runs: for seeds 0 and 1 and six evenly spaced iterations (e.g. $1, 10, \ldots, 50$ for 50-iteration problems) we restore the observed data, refit the surrogate, and time the complete acquisition optimization, including its setup (e.g. box decompositions or Pareto-set sampling), the evaluation of the initial candidates and all optimizer restarts. Surrogate fitting and function evaluations are excluded. All timings were taken on identical compute nodes with two AMD EPYC 9354 CPUs (32 cores each, 2×32 cores / 128 hardware threads, 768 GB RAM), on the CPU only (no GPU), with each replay task limited to 4 threads and up to 15 tasks sharing a node; we report the mean over iterations and seeds.

C.7 Ablation

The ablation (Figure 2) runs three acquisitions in the noiseless setting on Vehicle design, Car side-impact, DTLZ2 with $M = 6$ ($d = 7$, $n_0 = 16$, 50 iterations, $z = -0.1 \cdot \mathbf{1}$) and RE91, with the protocol of Table 1. All three use the observed nondominated outcomes as a fixed reference set R (the noiseless setting of Section 3) and share RaMMOBO's estimator and optimizer: fatplus with $\tau = 10^{-3}$, averaging of $S = 128$ samples in log space, and the `optimize_acqf` settings of Appendix C.3. ***EMmI*** uses the margin $\Delta(y; R)$ of Eq. (10) as its utility; ***Clipped EMmI, no weights*** uses the box volume $L(y; R)$ of Eq. (11), i.e. $w = \mathbf{1}$; and RaMMOBO uses $L_W(y; R)$ with $W \sim M \cdot \mathrm{Dirichlet}(\mathbf{1})$.

C.8 Rank agreement

To test whether RaMMOBO orders candidates as qLogNEHVI does, we rerun RaMMOBO under the protocol of Table 1 (sequential, noiseless, 10 seeds) with a *shadow* qLogNEHVI attached: in every iteration, after RaMMOBO's acquisition has been optimized and its candidate chosen, qLogNEHVI is built on the same surrogate, the same observed data and the same reference point z (exact box decomposition for $M \le 4$; $\alpha = 10^{-4}$ on RE91, as in Figure 3), and both acquisitions score the same pool of 512 scrambled-Sobol inputs drawn afresh each iteration. We record Spearman's rank correlation between the two score vectors over the whole pool and over the 51 inputs (top decile) that qLogNEHVI ranks highest and show results in Figure 8. Note that when the Pareto front has already been approximated well in later BO epochs, it may happen that the top decile of points all score similarly in terms of $q$logNEHVI, and the ranks may disagree with RaMMOBO only due to small changes in magnitude which are irrelevant for the BO performance: For example, RaMMOBO outperforms exact $q$logNEHVI on DTLZ7 in terms of hypervolume difference, but achieves bad top-decile rank agreement in later rounds (Figure 8). The shadow is never optimized and never influences the run, so the trajectory is purely governed by RaMMOBO. Curves show the median over the 10 seeds with the interquartile band.

C.9 Software and hardware

All experiments use BoTorch 0.11.3, GPyTorch 1.12 and PyTorch 2.5.1; qPOTS uses pymoo 0.6.2 for NSGA-II, and BOtied uses pyvinecopulib 0.6.8 and cma 4.4.4. Runs were executed on CPU nodes with 2–4 threads per run.

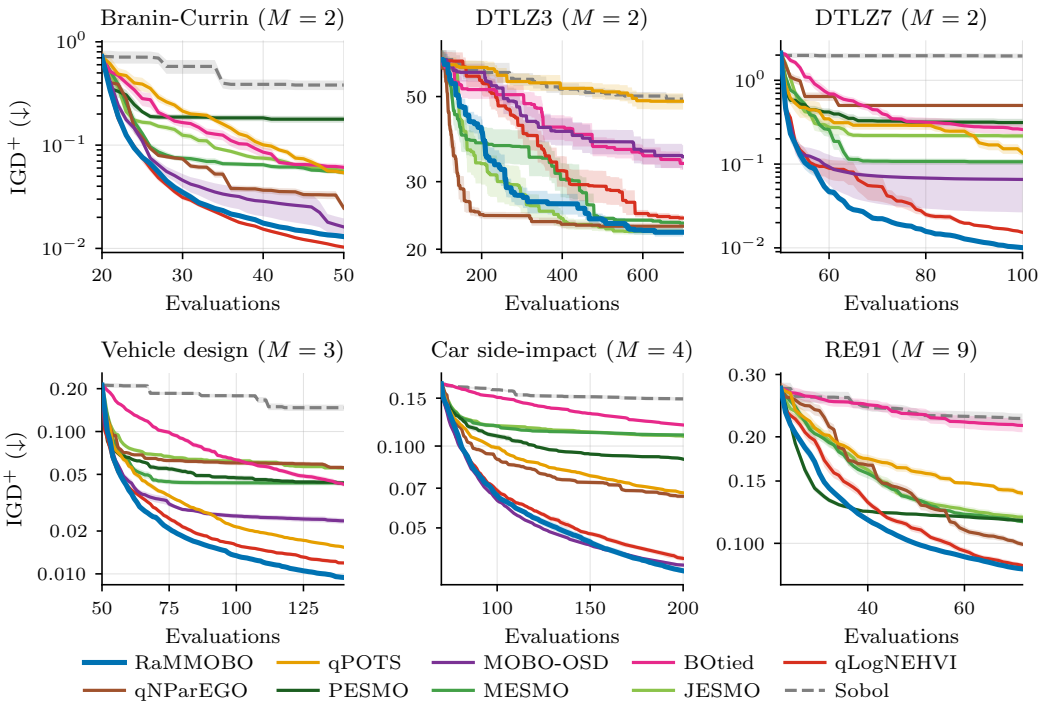


Figure 6: **IGD$^+$ against evaluations** in the sequential ($q = 1$) setting. Also in this reference-point-free metric, RaMMOBO performs on par with or outperforms state-of-the-art methods.

D Additional Experiments

Batch Setting. Figures 9 and 10 repeat the six problems of Figure 3 with batches of $q = 2$ and $q = 4$ points, keeping the evaluation budget of the sequential setting (Appendix C.1). RaMMOBO is extended to $q > 1$ with the same greedy scheme as the BoTorch-based baselines: each point of a batch is selected while treating sampled outcomes at the previously selected points as pending observations (Algorithm 2, Appendix B).

D.1 Monte Carlo hypervolume improvement

The hypervolume identity of Zhang & Golovin (2020, Lemma 5) expresses hypervolume as an expectation over random scalarizations. Taking the difference between the identities for $R \cup \{y\}$ and R yields an expectation for HVI, which can be estimated by averaging over sampled directions without a box decomposition. Figure 12 compares this naïve approximation with RaMMOBO under the same experimental protocol. It produces larger hypervolume differences on most problems, with especially poor progress on DTLZ7; on DTLZ3, it eventually reaches similar performance. An unbiased estimate at a fixed candidate therefore need not yield an effective acquisition optimizer at a finite sampling budget.

The concentration guarantee also retains an exponential dependence on the number of objectives. With coordinate extents bounded by $B \geq 1$, their Lemma 6 gives the sufficient sample budget

$$K = \mathcal{O}\left(B^{2M} M^{M} \frac{\log(1/\delta)}{\varepsilon^2} \right)$$

for additive error ε in normalized hypervolume HV/c_M with probability at least $1 - \delta$, where $c_M = \pi^{M/2}/(2^M \Gamma(M/2+1))$. Thus this bound does not guarantee accurate approximation with a sample budget polynomial in M. It is a sufficient bound, not a lower bound on the necessary number of samples.

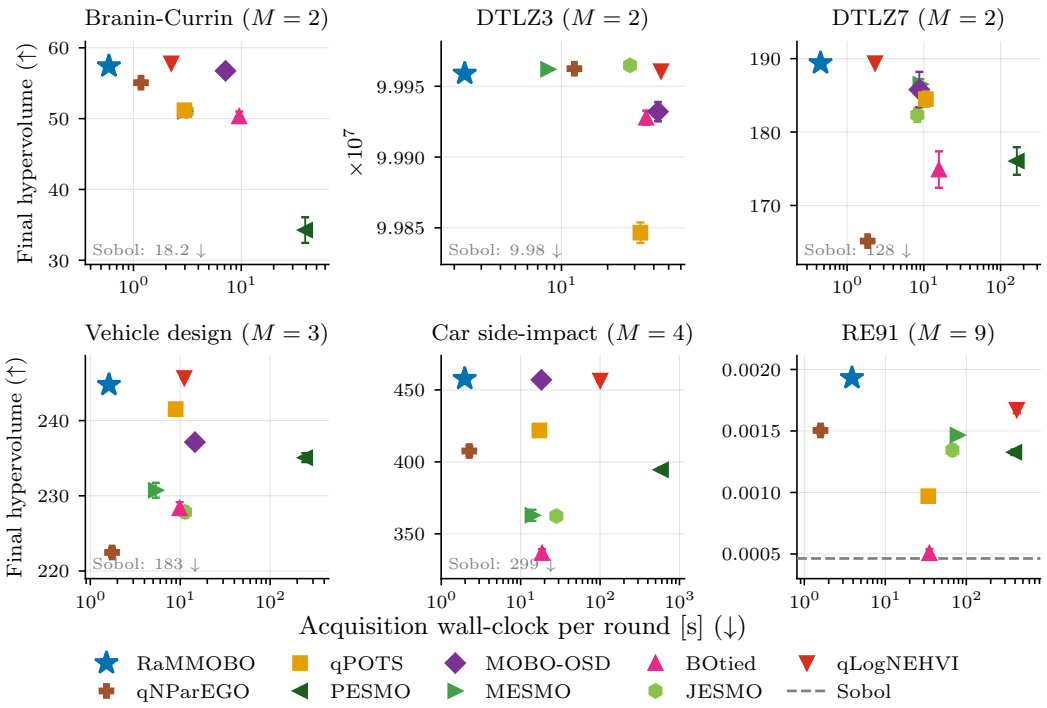


Figure 7: **Hypervolume against wall-clock** ($q = 1$). Final hypervolume against wall-clock time (seconds) needed to optimize an acquisition function to propose the next input. Same setup as in Figure 5, but showing all six panels from the main set of experiments.

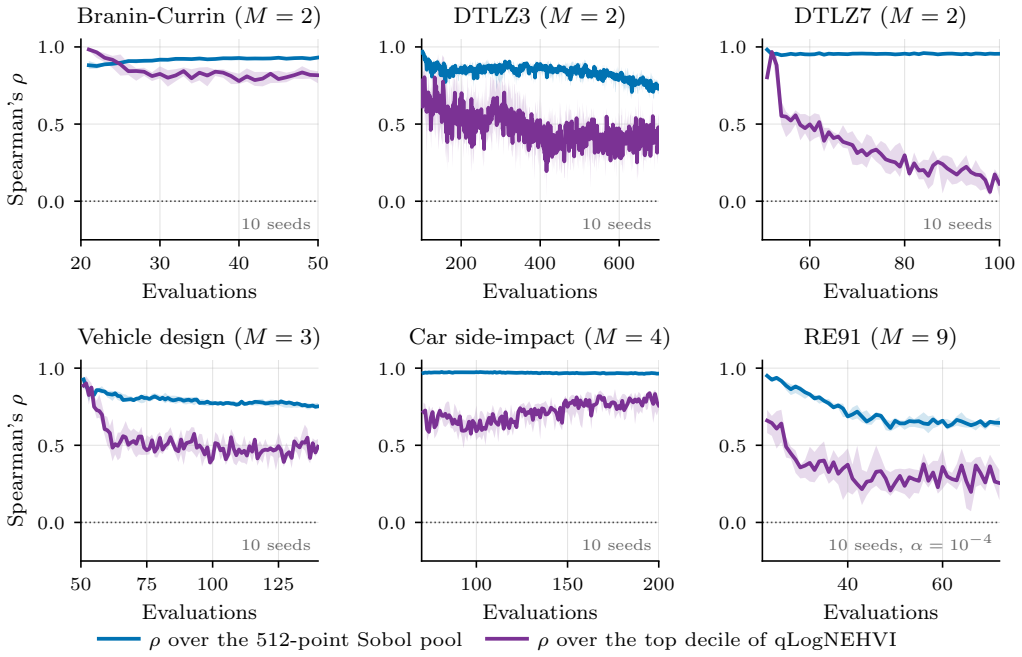


Figure 8: **Rank agreement between RaMMOBO and qLogNEHVI** in the sequential ($q = 1$) setting. Across the six benchmark problems, both acquisition functions rank a pool of 512 random inputs with consistently large Spearman correlation for $M \leq 4$ (blue curve), and their agreement weakens only on RE91 ($M = 9$), where qLogNEHVI itself relies on an approximate box decomposition and is outperformed by RaMMOBO. The purple curve shows rank correlation on the top 10% of points, ranked by qLogNEHVI; details in Appendix C.8.

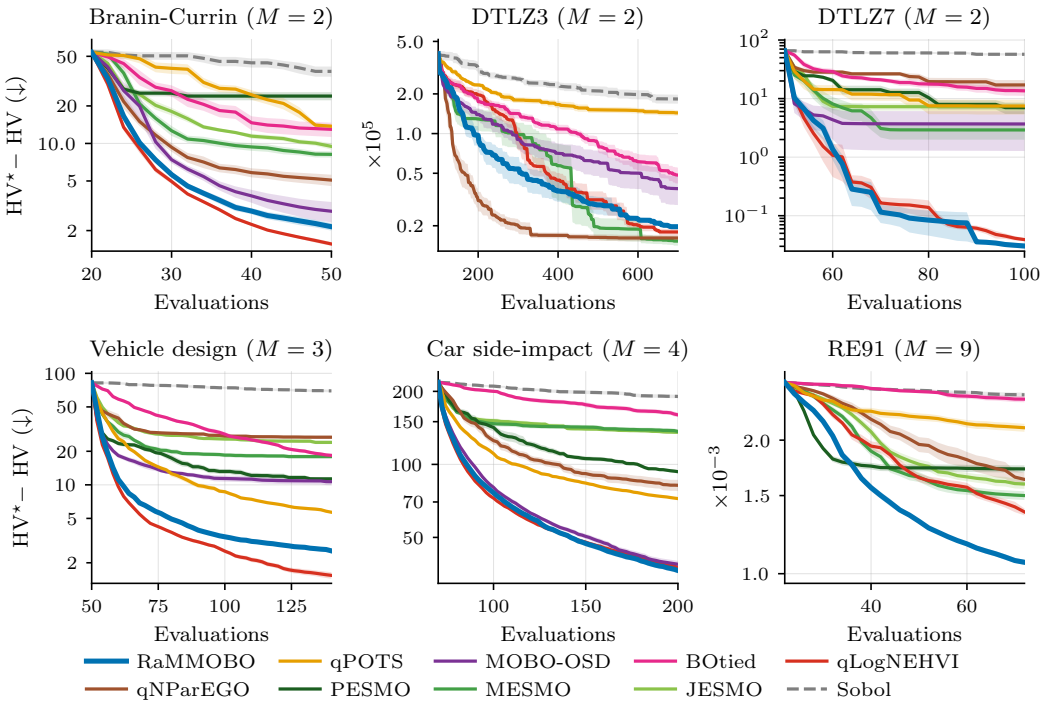


Figure 9: **Log hypervolume difference** for $q = 2$. We use batch RaMMOBO (Algorithm 2), which selects the two points greedily within each batch. Experimental details are given in Appendix C.

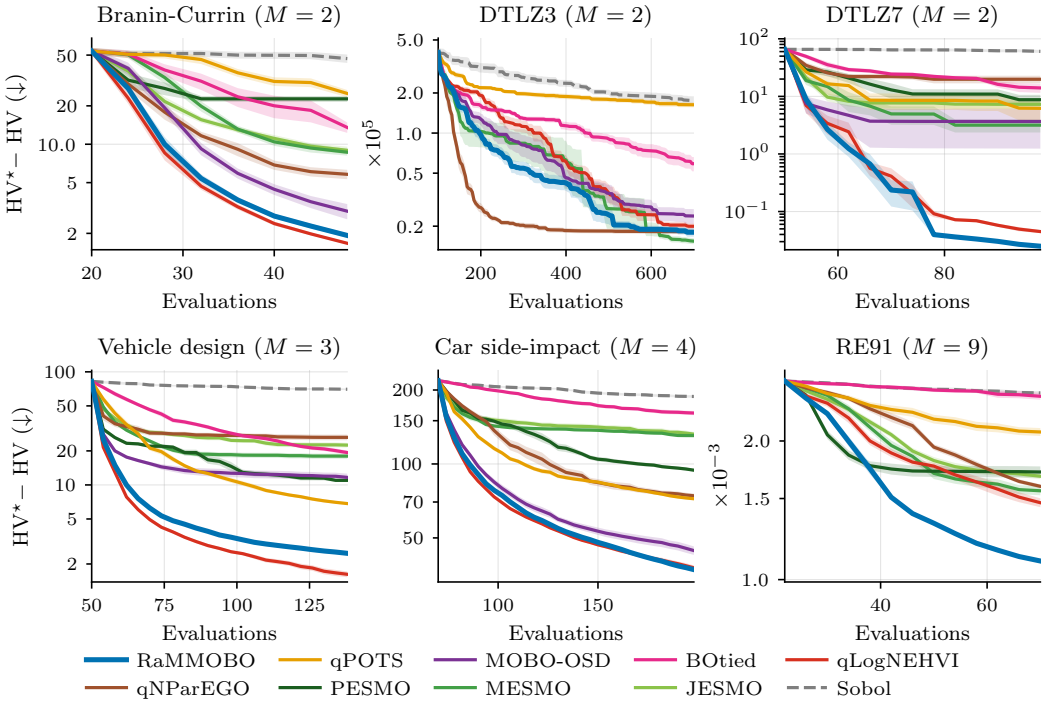


Figure 10: **Log hypervolume difference** for $q = 4$. We use batch RaMMOBO (Algorithm 2), which selects the four points greedily within each batch. Experimental details are given in Appendix C.

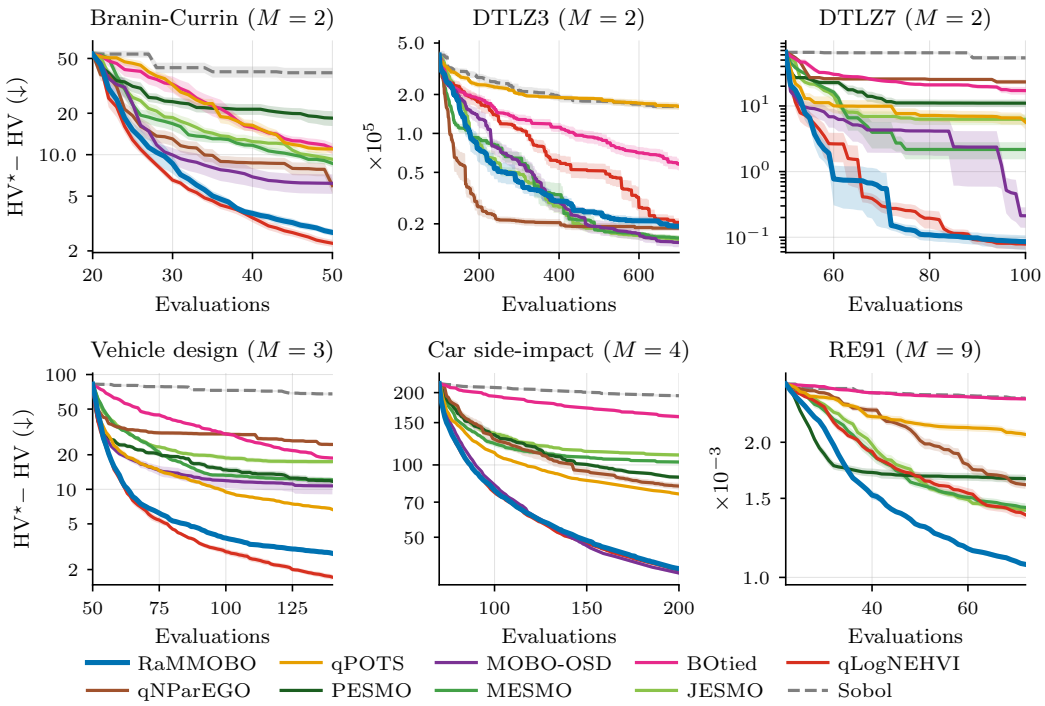


Figure 11: **Log hypervolume difference** ($q = 1$) in the noisy setting. Independent Gaussian noise with standard deviation equal to 1% of each objective's range is added to every observation, as described in Appendix C. Hypervolume is computed using the noiseless output of the black-box function.

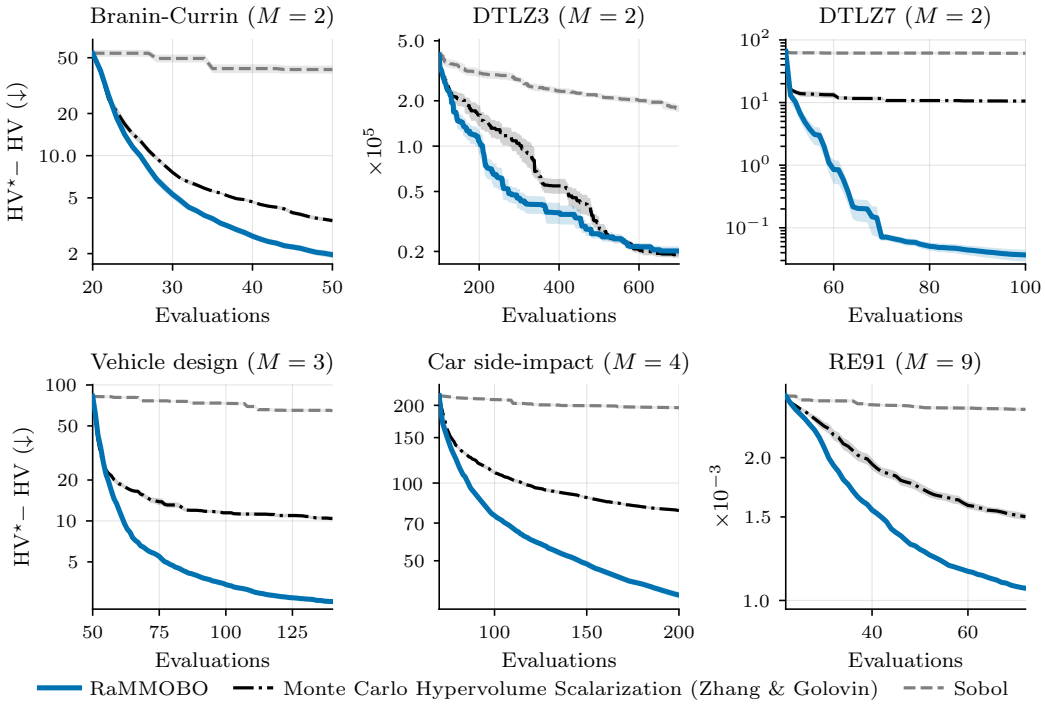


Figure 12: **Log hypervolume difference** comparing RaMMOBO to a naïve Monte Carlo estimate of hypervolume's formulation as an expectation in Zhang & Golovin (2020). We use identical experimental settings as in our original $q = 1$ run.