C3BO: COUNT-CONDITIONED COMBINATORIAL BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluation-limited subset optimization must choose both how many items to select and which ones. Across four constrained benchmarks with additive objectives, the nine surrogate-based methods we evaluate reach an optimal count in at most 3.2% of proposals. We analyze a prior–archive mismatch in linear and Gaussian-process surrogates. For kernels depending only on which items differ, including RBF and Matérn kernels with ARD, we prove that the linear size direction receives only the average prior variance of individual-item directions. The benchmarks have a dominant size effect but narrow count coverage in their repaired archives, and posterior refits severely underestimate that effect. We propose C3BO, which corrects the surrogate's prior on the size direction and lets the corrected posterior choose the count. When the joint solve is not certified in time, it compares candidates from fixed-count subproblems, where size-only errors leave candidate rankings unchanged. With 100 evaluations, C3BO and count sampling, a baseline that keeps the standard prior and samples counts, both outperform all thirteen external methods across the four benchmarks. C3BO improves hypervolume AUC over count sampling by 10.4–17.8% on held-out instances. Ablations link the main gains to size-aware modeling combined with posterior-based count selection. Sampling counts independently of surrogate predictions also improves optimization with a tabular foundation model on all four benchmarks. On two nonlinear optimal-design problems, duplicate proposals expose a separate failure mode. Rejecting repeats brings C3BO's mean hypervolume AUC within 1% of count sampling on fresh scenarios.

1 INTRODUCTION

Evaluation-limited subset optimization must choose how many items to select and which ones, under feasibility constraints that are known in advance and cheap to check. Applications include sensor placement (Krause et al., 2008), feature selection (Guyon & Elisseeff, 2003) and facility selection. We call the two decisions the *count* and the *composition*. Surrogate-based methods, whether Bayesian optimization (Shahriari et al., 2015; Baptista & Poloczek, 2018; Oh et al., 2019; Wan et al., 2021; Deshwal et al., 2023; Papenmeier et al., 2023) or a learned model maximized by an integer program (Papalexopoulos et al., 2022), choose both at once, leaving the count implicit in the acquisition step. Across four additive benchmark families we find that choice systematically miscalibrated, trace it to a mismatch between a dominant size effect, the surrogate's prior and a narrow repaired archive, and address it with size-aware modelling and count-conditioned search.

Surrogate-based search gets the count wrong. We ran thirteen methods on four standard constrained problems with 250 items and two objectives, and computed the best size exactly for each trade-off. Figure 1 shows the pattern: search stays near its starting size, too small when items add value and too large when they cost. Across the nine surrogate-based methods, from Gaussian processes (Daulton et al., 2021; Ament et al., 2023) to foundation models (Hollmann et al., 2025; Yu et al., 2026), at most 3.2% of proposals land on a best size, and even a linear model with an exact solver never reaches one in three of the four problems. This matters because the good subsets in these problems all have nearly the same size, so no choice of items makes up for a wrong count.

Two things keep search at the wrong size. The first is the data: initial designs are random draws repaired to satisfy the constraints (Salcedo-Sanz, 2009), and repair moves their size away from 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.

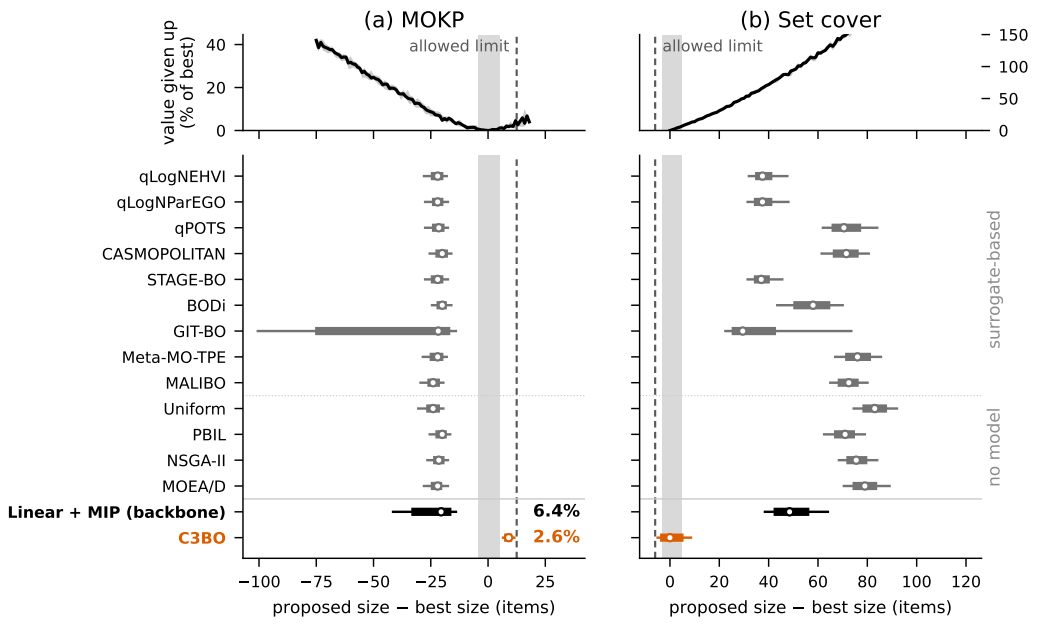


Figure 1: Surrogate-based search proposes the wrong number of items. Proposed minus best size over 20 instances and 8 trade-offs (dot: median; bar: interquartile range; line: 10–90%). Grey: best sizes; dashed: size limit

best one, so that in MWIS a request of 96 items comes back with 54. The second is the prior. We prove that every kernel that depends only on which items differ, ARD included (Rasmussen, 2003), gives the number of items only the average prior variance of a single item, whereas here it matters most. An archive of nearly one size cannot overrule such a prior. Refitted to archives from our runs, the linear backbone recovers at most 15.7% of the size effect and a Gaussian process at most 20.1%. Restoring it alone brings proposals within 10 items of the best size in 63–91% of states, against 0–9%.

We propose C3BO, which corrects the surrogate's belief about size and lets the posterior choose the count. It corrects the prior by adding $|x|$ as a feature with a diffuse prior, which amounts to a standard hierarchical prior on item effects (Gelman et al., 1995). It then solves the whole problem and, when that solve is not finished in time, compares subproblems of fixed count instead. Inside a fixed count, an error about size shifts every candidate equally, so it cannot change which one wins.

C3BO beats all 13 external methods in all four problems, in all 52 paired comparisons after Holm correction, and so does count sampling, which keeps the backbone's prior and draws the count from a broad law. On held-out instances C3BO beats count sampling by 10.4–17.8% in hypervolume AUC. The gain comes from letting the corrected posterior choose the count: at a 5 s limit, taking the count the solver picks under the corrected prior does as well as C3BO, while the corrected prior with a sampled count gains far less. The fixed-count subproblems matter when the solver is cut off: they keep the result stable where the unrestricted solve does not finish in time. When the prior cannot be set, as for a tabular foundation model, a sampled count still improves all four problems.

On two nonlinear optimal-design problems from PyOED (Chowdhary et al., 2025), repeated proposals expose a separate failure of batch diversity: about half of what a learned count evaluates repeats a design already measured. Removing repeats largely closes the gap to count sampling on fresh scenarios, though C3BO does not lead on sensor placement.

Our key contributions are as follows:

1. A diagnosis of why surrogate-based search picks the wrong size, with an exact prior-allocation result and a recovery formula linking archive information to size-effect regularization, confirmed on a linear and a Gaussian-process surrogate.

2. A method, C3BO, that relaxes shrinkage of the shared size effect and lets the corrected posterior select counts, with fixed-count subproblems as a budgeted fallback.
3. An evaluation against thirteen methods with registered predictions, failures included, and a verdict on held-out instances.

2 Related Work

Combinatorial Bayesian optimization and kernels on the hypercube. Surrogates that model binary or categorical designs directly (Baptista & Poloczek, 2018; Oh et al., 2019; Deshwal et al., 2021) give way at larger n to embeddings, trust regions and tabular foundation models (Deshwal et al., 2023; Papenmeier et al., 2023; Wan et al., 2021; Hollmann et al., 2025; Yu et al., 2026), while others keep a continuous acquisition step and relax or reparameterize the discrete variables (Daxberger et al., 2019; Daulton et al., 2022). A recent benchmark collects the family (Dreczkowski et al., 2023). All leave the size to the acquisition step. Kernels that depend on x and x' only through $x \oplus x'$ are diagonal in the Walsh basis (O'Donnell, 2014), with diffusion (Kondor & Lafferty, 2002) and Hamming kernels the permutation-symmetric special case, and ARD kernels on binary inputs (Rasmussen, 2003) in the same class because $(x_i - x'_i)^2 = (x \oplus x')_i$. The diagonalization is standard. What we add is what it implies for the size decision. BOCS (Baptista & Poloczek, 2018) is the closest ancestor of our backbone – a Bayesian linear model over binary variables, maximized by combinatorial optimization – and it is also where we expect the diagnosis to bite hardest, because it replaces the Gaussian prior on item coefficients with a horseshoe. The size effect is dense, loading equally on every coordinate, and a sparsity prior shrinks a dense effect harder than a Gaussian one does. Proposition 2 is proved for the Gaussian case and we did not run BOCS, so this is a prediction rather than a result.

Cardinality, subset selection and repair. Evolutionary and estimation-of-distribution methods search subsets without a surrogate (Deb et al., 2002; Zhang & Li, 2007; Baluja, 1994), and the multi-objective formulations among them treat the size as a second objective rather than as a decision to be made first (Qian et al., 2015; Friedrich & Neumann, 2015). Repair operators restore feasibility after an unconstrained proposal (Salcedo-Sanz, 2009). We show that repair also concentrates the observed size band, and that the band we observe excludes the best size in all four of our families.

Multi-objective acquisition and integer programs. Expected hypervolume improvement and its numerically stable forms (Daulton et al., 2021; Ament et al., 2023) keep the size inside one acquisition step, and where that step fits a Gaussian process on binary inputs, Proposition 1 applies to it. Thompson sampling over the Pareto set (Renganathan & Carlson, 2025) and adaptive ϵ-constraint decompositions (Yang et al., 2026) change how the trade-offs are traversed, and meta-learned surrogates transfer across tasks (Pan et al., 2024; Watanabe et al., 2023). None of them decides the count. Mixed-integer formulations of acquisition maximize a learned model exactly (Papalexopoulos et al., 2022). We show that what the surrogate learns changes how hard that program is. General treatments of Bayesian optimization (Shahriari et al., 2015; Garnett, 2023) do not separate the two decisions.

3 Problem Setup

Let $\mathcal{F} \subseteq \{0,1\}^n$ be a known feasible set, $f(x) \in \mathbb{R}^2$ the objectives, and $f_h = \lambda_h^\top f$ the scalarization of role h. For the slice $\mathcal{F}_K = \{x \in \mathcal{F} : |x| = K\}$ let $V_h(K) = \max_{\mathcal{F}_K} f_h$ and $K_h^\star \in \arg\max_K V_h(K)$. For feasible x with $|x| = K$,

$$V_h(K_h^\star) - f_h(x) = \underbrace{V_h(K_h^\star) - V_h(K)}_{\text{size loss}} + \underbrace{V_h(K) - f_h(x)}_{\text{composition loss}}. \tag{1}$$

No within-slice solver can reduce the size loss. A slow solver at the right size pays composition loss. A fast solver at the wrong size pays size loss. Section 5.1 chooses between counts by trading these two losses under the posterior. We compute $K_h^\star$ exactly by integer programming in the four benchmark families, and with a greedy oracle on the design problems of Section 6.4.

Notation. Write each coefficient vector as $w = \bar{o}(w)\mathbf{1} + w_\circ$ with $\bar{o}(w) = \mathbf{1}^\top w/n$ and $\mathbf{1}^\top w_\circ = 0$, so that $w^\top x = \bar{o}(w)\,|x| + w_\circ^\top x$ and the first term is size-only. We call $\bar{o}(w)$ the *size effect*. In our families it is large, because item values share one sign: coefficients are uniform on $\{10, \dots, 99\}$, whose mean-to-SD ratio is 2.10, so $|\bar{o}(w)|$ is 2.1 standard deviations of $w_\circ$ in every family. It is negative for set cover, where items cost, and positive in the other three families.

4 Why Search Settles at the Wrong Size

Two facts keep search at the wrong size. The prior strongly shrinks a common size effect, and an archive of nearly one size cannot overcome that shrinkage. Proofs are in the appendix.

4.1 The prior does not favour size

Equip functions on $\{0,1\}^n$ with $\langle u, v\rangle = 2^{-n}\sum_x u(x)v(x)$ and let $\chi_S(x) = (-1)^{\sum_{i\in S} x_i}$. The order-$j$ subspace $W_j = \operatorname{span}\{\chi_S : |S| = j\}$ has dimension $\binom{n}{j}$. A function is *size-only* if it depends on x only through $|x|$.

Proposition 1 (Size neutrality of the prior)**.** *Let $f \sim \mathcal{GP}(0, C)$ with $C(x, x') = g(x \oplus x')$, and let f_j be the projection of f onto W_j. Then W_j contains exactly one unit-norm size-only direction u_j, and if $\mathbb{E}\|f_j\|^2 > 0$, $\mathbb{E}\langle f, u_j\rangle^2/\mathbb{E}\|f_j\|^2 = \binom{n}{j}^{-1}$.*

A size effect linear in $|x|$ lies on u_1, so its prior share is $1/n$. Every kernel of the form $\phi\big(\sum_i (x_i - x_i')^2/\ell_i^2\big)$ meets the condition on binary inputs, because $(x_i - x_i')^2 = (x \oplus x')_i$. This includes RBF and Matérn kernels with ARD, and so the Gaussian process that qLogNEHVI, qLogNParEGO and qPOTS share, and CASMOPOLITAN's, whose overlap kernel qualifies for the same reason. ARD changes how variance is spread across the order-j directions, but the size-only direction always receives their average. The proposition does not cover embeddings, density estimators, neural bases or foundation models. For those it is only observed.

The kernel therefore does not single out size. The target does. Item values in our families share one sign. With a prior scale matched to the spread of the true coefficients, the size effect lies $2.1\sqrt{n} \approx 33$ prior standard deviations out along u_1 in every family, since both 2.1 and $n = 250$ are common to all four, while a typical item direction lies about one out. The backbone's linear surrogate is not of this form, but its order-one part is, so the share is again $1/n$.

4.2 A narrow archive cannot overrule the prior

Initialization draws random designs and repairs them to feasibility, and repair changes their size: in MWIS no design in the 20 initial archives exceeds 77 items. In every family the initial sizes exclude the best ones.

Proposition 2 (Prior-controlled size-effect recovery)**.** *Model the standardized outputs as $a_0 + w^\top x + \varepsilon$ with a flat prior on the intercept a_0, $w \sim \mathcal{N}\big(0, s^2(I + L^2\mathbf{1}\mathbf{1}^\top)\big)$ and noise $\mathcal{N}(0, \sigma^2)$. Fix (s, σ, L) and the archive $X \in \{0,1\}^{N\times n}$, with sizes $\mathbf{k} = X\mathbf{1}$ and $\mathbf{k}_c = \mathbf{k} - \bar{k}\mathbf{1}$. Then $\bar{o}(\hat{c}) = \frac{a}{a+\eta_L}\,\bar{o}(w) +$ (terms not involving $\bar{o}(w)$), where $\eta_L = n/(1 + nL^2)$, $a = \mathbf{k}_c^\top(\tilde{X}\tilde{X}^\top + \kappa I)^{-1}\mathbf{k}_c$, $\tilde{X} = X_c(I - \mathbf{1}\mathbf{1}^\top/n)$, X_c is the column-centred design and $\kappa = \sigma^2/s^2$. Moreover $a \le \|\mathbf{k}_c\|^2/\kappa$.*

Recovery is set by archive information a against the prior precision η_L of the size effect. The independent prior ($L = 0$) has $\eta_0 = n$, the inverse of the $1/n$ share above, and $a \le \|\mathbf{k}_c\|^2/\kappa$ stays small on a narrow archive. Raising L lowers the threshold without changing the archive: at $n = 250$ and $L = 50$, $\eta_L \approx 4 \times 10^{-4}$. A near-unit coefficient still requires $a \gg \eta_L$. The posterior variance of $\bar{o}(w)$ is $s^2/(a + \eta_L)$, approaching s^2/a for $a > 0$ as $L \to \infty$: relaxing the prior removes shrinkage but creates no information absent from the archive. A single-size archive has $a = 0$, and then no posterior quantity depends on $\bar{o}(w)$.

The remaining error has a direction: the size of a maximizer of $(c + e\mathbf{1})^\top x$ is non-decreasing in e, so understating a positive size effect selects too few items, and understating a negative one too many.

4.3 EVIDENCE

We test both claims on the backbone's stored campaigns, without new evaluations. At rounds 3, 6 and 10 we refit the posterior and re-solve each role twice, once as run and once with the true size effect restored. In MOKP, MDK, set cover and MWIS the fitted size effect is -0.6%, 0.0%, 15.7% and -4.9% of the true one. The backbone keeps an intercept with a proper prior, so the relevant coefficient is $\rho_{\mathcal{A}}$ of Proposition 5: the intercept must offset $\bar{o}(w)\bar{k}$ under the same narrow prior, a coupling that centring removes. The realized ratio also carries item and noise terms, and so can be negative. It is largest in set cover, whose initial sizes span 40 items against 21–26 elsewhere, as the ratio above predicts. The size error has the predicted sign in 84–100% of states. Restoring the size effect alone moves the solution to within 10 items of the best size in 63–91% of states, against 0–9% as run.

The failure does not depend on the surrogate. Refitted with default BoTorch priors to the archives that qLogNEHVI stored, its Gaussian process recovers 0.3%, 0.4%, 20.1% and 2.8% of the size effect, again most in set cover. A mean function linear in $|x|$ with a flat prior on its slope recovers 96.5–104.4%.

5 C3BO: SIZE-AWARE MODELLING AND COUNT-CONDITIONED SEARCH

C3BO's main ingredient is the size-aware posterior of Section 5.1. From it, each role h forms its proposal in three steps.

1. Fit the size-aware posterior $\hat{c}_h$.
2. Solve $\max\{\hat{c}_h^\top x : x \in \mathcal{F}\}$. If the incumbent is certified in time, propose it, so the corrected posterior chooses the count.
3. Otherwise, as a fallback, also solve fixed-count slices $\max\{\hat{c}_h^\top x : x \in \mathcal{F}_K\}$ and propose the best candidate.

If a slice is empty or the solver returns no incumbent, a conditional Bernoulli design of size K_h is repaired to feasibility. A design already in the archive, or already chosen in the current batch, is not proposed again. The role takes its next candidate instead.

5.1 THE SIZE-AWARE POSTERIOR AND THE FALLBACK

A size-aware posterior. C3BO implements the prior family of Proposition 2. We centre the design columns, drop the intercept, and append one feature proportional to $|x| - \overline{|x|}$ with a prior scale $L = 50$ times that of an item coefficient. The fitted size coefficient is added back to every item, $\hat{c} = \hat{\beta} + \hat{\gamma}\mathbf{1}$. Since $|x|$ is a sum of item features, the model class is unchanged. Only the prior moves, to $w \sim \mathcal{N}\big(0,\, s^2(I + L^2\mathbf{1}\mathbf{1}^\top)\big)$, which lowers the prior precision of the size effect from n to η_L and leaves zero-sum directions untouched: the familiar hierarchical prior with a diffuse common mean. On the backbone's archives it recovers 99.5–100.6% of the size effect. Centring and intercept removal alone do not explain the improvement. Table 1 sets $L = 0$, which keeps both and removes only the diffuse prior on the size feature. Recovery of the size effect falls to 0.42–0.87 from approximately full, and in matched campaigns on five targets per family the same change costs 15.2–36.2% of hypervolume AUC, every interval clear of zero. The counts move as Lemma 6 says a shrunk size effect should: $L = 0$ selects fewer items than $L = 50$ in MOKP, MDK and MWIS, and more in set cover, which is upward-closed.

Choosing the count. Under this posterior the best count is the one whose slice has the highest predicted value. Within the solve limit T, C3BO first spends $T/2$ on the unrestricted problem $\max_{\mathcal{F}} \hat{c}^\top x$. If its incumbent x_U is certified within a relative gap ϵ, the proposal is x_U. Otherwise the fallback probes counts at offsets $\{2, 5, 10, 20, 40\}$ from the end of the feasible range that the learned size effect points to, from the interior outward and only while time remains. That end is estimated by one greedy pass over the constraints alone: a greedy cover, a greedy independent set, or items added cheapest-first. Each pass takes about a millisecond and errs toward the interior, so every probed count is feasible. The proposal is

$$\hat{x} = \arg\max\big\{\hat{c}^\top x :\ x \in \{x_U, x_{K_1}, \ldots, x_{K_M}\}\big\}. \tag{2}$$

Table 1: The diffuse prior on the size feature drives the AUC gain. $L = 0$ keeps centring and intercept removal, which already raise recovery, and drops only that prior. Recovery is offline on stored archives; AUC is the mean over 5 targets, with a paired bootstrap interval on the difference.

	MOKP	MDK	Set cover	MWIS
Size effect recovered, $L = 0$	0.82	0.87	0.42	0.66
Size effect recovered, $L = 50$	1.00	1.00	1.00	1.00
AUC, $L = 0$	0.529	0.504	0.639	0.244
AUC, $L = 50$	**0.648**	**0.594**	**1.001**	**0.321**
$L = 0$ against $L = 50$ (%)	-18.4 $[-19.9, -16.5]$	-15.2 $[-16.4, -14.1]$	-36.2 $[-37.3, -35.2]$	-24.0 $[-25.3, -22.6]$

Remark 3 (Invariance). Within a slice $\mathcal{F}_K$, adding any size-only function to $\hat{c}_h^\top x$ does not change the maximizer.

Whatever size error the posterior carries is a constant inside a slice, so it can only affect which count wins, not which items. A different prior also changes the item coefficients. Slices are easy: their median optimality gap is 0 in every family at both limits. Maximizing $\hat{c}^\top x$ over these candidates trades the two losses of Eq. (1) under the posterior. A count near the edge that the solver could not finish loses to an interior count exactly when its shortfall exceeds its predicted size advantage, and no parameter sets that trade.

The simplest variant proposes the incumbent of the size-aware unrestricted solve, run for the whole limit, so its count is whatever that incumbent has. It is the backbone with only its prior corrected, and we report it as C3BO w/o fallback. C3BO starts from the same solve and additionally probes count-conditioned subproblems when x_U is not certified, and skips repeated designs. How often the first case arises is where the two families of Section 5.2 part: at 5 s the unrestricted solve is certified and new in 88.5% and 89.9% of proposals in MOKP and MDK, against 0.4% and 1.1% in set cover and MWIS.

5.2 Where a learned count goes

$\mathcal{F}$ is *downward-closed* if removing items preserves feasibility, and *upward-closed* if adding them does. MOKP and MDK ($Ax \leq b$ with $A \geq 0$, $m = 2$ and 5 rows) and MWIS ($x_i + x_j \leq 1$ on 741 edges) are downward-closed, and set cover ($Ax \geq \mathbf{1}$) is upward-closed. The PyOED problems constrain only the count.

Proposition 4 (Edge). *Let $\mathcal{F}$ be downward-closed and let x^* maximize $\hat{c}^\top x$ over $\mathcal{F}$. (a) If $\hat{c}_i < 0$ then $x_i^* = 0$. (b) If $\hat{c} > 0$, every maximizer is inclusion-maximal, so $|x^*|$ lies between the smallest and largest sizes of inclusion-maximal elements of $\mathcal{F}$. For upward-closed $\mathcal{F}$ the signs reverse and maximizers are inclusion-minimal.*

Part (a) is why the uncorrected search is easy for the solver: a shrunk size effect leaves $\hat{c}$ with mixed signs, and the solver works on only 36–47% of the items. Part (b) is what the correction does when, as in all our runs, every item takes the same sign: nothing can be discarded, and the learned count lies in the band of inclusion-extremal sets. In our four families that band sits at an end of the feasible range, a fact about the families rather than part of the proposition.

The edge may be hard. The edge is the classical combinatorial problem itself, and its difficulty follows its linear relaxation. With m knapsack rows the relaxation is within $m \max_i \hat{c}_i$ of the integer optimum, so the knapsack edge is nearly as easy as its relaxation. For independent set and set cover the relaxation carries little information (Lovász, 1975; Chvatal, 1979). The median terminal gaps of the unrestricted solve at 1 s follow this order: 0, 0.0014, 0.68 and 7.7. When the edge is not finished in time, the selection rule prefers a count the solver can finish.

The batch may collapse. The posterior fixes where a proposal goes, not how many distinct proposals a batch contains. When it is confident, neighbouring trade-offs point to the same inclusion-extremal set, and several roles return the same design. Deduplication removes this within the same solve budget: a role whose best candidate repeats takes its next-best one under the same posterior, and a repeating x_U does not end the proposal early, since the stage goes on to probe counts inside the same limit. In our implementation that candidate lies at another offset from the same estimated

Table 2: Repeated proposals hurt C3BO only where the Pareto front spans the whole count range, as in the two optimal-design problems.

	MOKP	MDK	Set cover	MWIS	Sensor	Tomography
Front's count span (% of range)	1.3	1.4	1.4	1.5	100.0	100.0
C3BO proposals repeating (%)	6.1	4.1	32.8	7.5	46.9	58.0
C3BO over count sampling (%)	+16.5	+16.4	+9.8	+9.1	−0.2	+0.8

edge, so this trades count accuracy for a design not yet measured. Table 2 shows that what a collapse costs depends on how much of the count range the Pareto front spans.

6 Experiments

6.1 Setup

Problems. Four families of constrained bi-objective subset problems with $n = 250$ items: multi-objective and multi-dimensional knapsack (MOKP, 2 capacity rows; MDK, 5), set cover over 250 elements, and maximum-weight independent set on a random graph with 741 edges. Objectives are additive in the items. Instances are generated rather than taken from a benchmark suite: each of the 20 per family draws its own constraint matrix and objective coefficients from a seeded generator, which we release with the code. The two optimal-design problems of Section 6.4 come from PyOED (Chowdhary et al., 2025).

Baselines. We compare against nine surrogate-based and four non-surrogate methods, listed with their citations in Table 3, using the adaptations of Appendix B. All methods share the initial archives, the evaluation and batch budgets, and the repair. Because an adaptation also affects which counts a method can propose, Figure 1 is descriptive. The causal evidence for the diagnosis is the matched posterior interventions of Section 4.3.

Ablation and count sampling. Two arms isolate the parts of C3BO. C3BO w/o fallback proposes the incumbent of the size-aware unrestricted solve, with no probing and no deduplication. Count sampling keeps the backbone's own prior and the fixed-count solve, and draws the count from an equal-weight mixture of $\mathrm{Binomial}(n, p)$ laws, $p \in \{0.1, \ldots, 0.9\}$. It uses nothing the surrogate believes about size, and by invariance a size error cannot change which items it picks, so it also runs on a surrogate whose prior cannot be set. Deduplication belongs to C3BO only.

Measurement. Each run makes 100 evaluations in batches of 8, with a solve limit of 5 s per proposal. The appendix repeats it at 1 s and reports solver times. Designs were fixed on a development set, and the verdict is taken on a held-out set. The endpoint is hypervolume AUC on objectives normalized to $[0, 1]$, with reference point $(-1/7, -1/7)$, averaged over the ten batches after the initial archive. Comparisons use a paired bootstrap over targets with Holm correction. Registered predictions, failed ones included, are listed in Appendix K.

6.2 Comparison with existing methods

Table 3 gives hypervolume AUC on the main set. C3BO beats all 13 external methods in all four families. All 52 paired comparisons are positive and significant after Holm correction, with the largest adjusted two-sided p equal to 0.01. Count sampling does the same in its own family of 52 tests, and neither loses a single cell to any external method. On held-out instances C3BO beats count sampling, which differs in both prior and count, by 10.4–17.8%, and by 9.1–16.5% on the main set, with every interval clear of zero. The next section separates the two changes.

6.3 Where the gain comes from

The gain needs both the size-aware prior and a count taken from it. Figure 2 crosses the two, and its size-aware, posterior's-count corner is C3BO w/o fallback. With the count sampled, correcting the

Table 3: Hypervolume AUC on the main instance set, mean ± SD over 20 paired targets, external methods ordered by publication year. Neither C3BO nor count sampling loses a cell to any external method. The last row is the held-out margin of C3BO over count sampling, with 95% intervals.

Method	MOKP	MDK	SetCover	MWIS
Uniform	0.451 ± 0.006	0.438 ± 0.008	0.478 ± 0.017	0.195 ± 0.008
PBIL (Baluja, 1994)	0.472 ± 0.011	0.453 ± 0.005	0.521 ± 0.026	0.192 ± 0.008
NSGA-II (Deb et al., 2002)	0.455 ± 0.008	0.442 ± 0.009	0.486 ± 0.017	0.188 ± 0.007
MOEA/D (Zhang & Li, 2007)	0.451 ± 0.007	0.436 ± 0.008	0.477 ± 0.023	0.190 ± 0.006
CASMOPOLITAN (Wan et al., 2021)	0.473 ± 0.009	0.455 ± 0.008	0.534 ± 0.019	0.203 ± 0.008
qLogNEHVI (Daulton et al., 2021)	0.462 ± 0.009	0.446 ± 0.006	0.713 ± 0.010	0.247 ± 0.013
BODi (Deshwal et al., 2023)	0.467 ± 0.009	0.449 ± 0.005	0.612 ± 0.020	0.190 ± 0.008
Meta-MO-TPE (Watanabe et al., 2023)	0.462 ± 0.009	0.448 ± 0.010	0.505 ± 0.023	0.188 ± 0.006
qLogNParEGO (Ament et al., 2023)	0.463 ± 0.007	0.448 ± 0.005	0.721 ± 0.015	0.248 ± 0.013
MALIBO (Pan et al., 2024)	0.453 ± 0.007	0.440 ± 0.007	0.528 ± 0.016	0.197 ± 0.006
qPOTS (Renganathan & Carlson, 2025)	0.453 ± 0.008	0.441 ± 0.010	0.508 ± 0.020	0.192 ± 0.008
GIT-BO (Yu et al., 2026)	0.462 ± 0.008	0.443 ± 0.008	0.745 ± 0.018	0.261 ± 0.014
STAGE-BO (Yang et al., 2026)	0.463 ± 0.008	0.450 ± 0.006	0.722 ± 0.013	0.247 ± 0.013
Linear + MIP (backbone)	0.490 ± 0.014	0.475 ± 0.013	0.636 ± 0.021	0.229 ± 0.015
C3BO w/o fallback	0.647 ± 0.016	0.591 ± 0.013	0.996 ± 0.010	0.320 ± 0.014
Count sampling	0.559 ± 0.034	0.509 ± 0.023	0.911 ± 0.033	0.295 ± 0.016
C3BO	$\mathbf{0.652} \pm 0.015$	$\mathbf{0.593} \pm 0.015$	$\mathbf{1.001} \pm 0.010$	$\mathbf{0.321} \pm 0.014$
Held out: C3BO over count sampling (%)	$+14.6$ [$+13.0, +16.2$]	$+17.8$ [$+15.1, +20.5$]	$+10.4$ [$+8.5, +12.2$]	$+10.7$ [$+9.3, +12.1$]

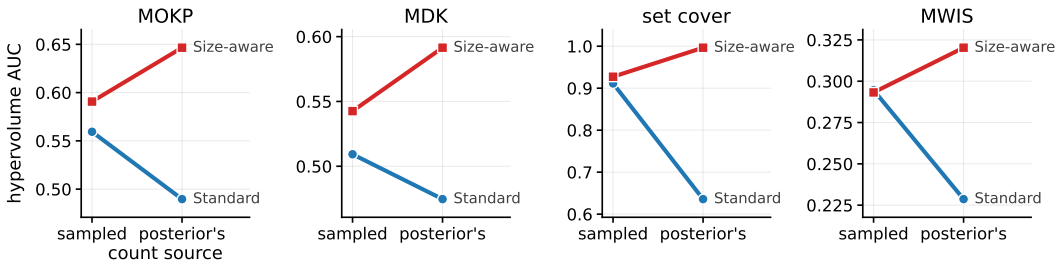


Figure 2: Hypervolume AUC for the prior, standard or size-aware, against the count, sampled or chosen by the posterior, one panel per family. In every family the posterior's count hurts under the standard prior and helps under the size-aware one.

prior changes AUC by -0.5 to $+6.5\%$. That gain cannot come from the size effect, which cancels inside a slice, so it comes from better item coefficients. With the standard prior, the posterior's count changes AUC by -30.2 to -6.8%. Together the two gain $+8.7$ to $+16.1\%$, an interaction of $+16$ to $+38$ points in every family. The gain therefore comes from the count the corrected prior chooses, as the diagnosis predicts. We do not split it into shares, because with an interaction this large any split depends on the order of the two changes.

At 5 s the fallback adds little to this. Against the better of C3BO w/o fallback and count sampling, the registered comparison, C3BO differs by -0.08, $+0.02$, $+0.25$ and $+0.53\%$ on held-out instances, every interval containing zero, and deduplication adds 0.2–0.8%. What the fallback buys is reproducibility when the solver is cut off, a measure we added after the registered runs. At 1 s, C3BO w/o fallback does not finish the root node of the set-cover relaxation, and its result depends on the run: across five runs the within-target standard deviation is 18.4% of the mean, against 0.5% for C3BO, whose slices finish in time.

6.4 Nonlinear objectives and other surrogates

Several trade-offs share one design. Two optimal-design problems from PyOED (Chowdhary et al., 2025) test a nonlinear objective: sensor placement ($n = 81$, at most 32 sensors) and tomography ($n = 250$, at most 40 angles). A greedy oracle puts the best count inside the range for 7 of 8 trade-offs in sensor placement and 2 of 8 in tomography. Before running we registered our predictions. As predicted, a learned count went to an end of the range in 97.5% and 97.3% of proposals, and sampling beat it by 0.90% and 2.20%. Against predic-

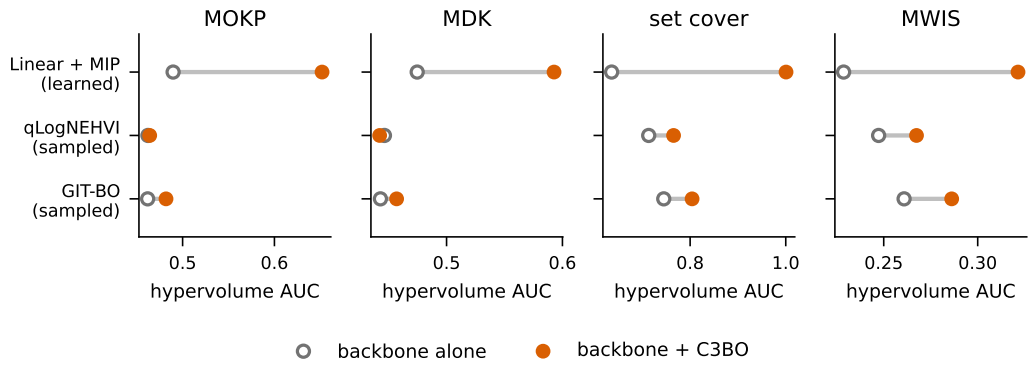


Figure 3: Effect of handling the count explicitly on three surrogates. Open: surrogate alone. Filled: C3BO on Linear + MIP; count sampling on qLogNEHVI and GIT-BO.

tion, learning did not beat the backbone, and not because of where the count went: a batch of eight trade-offs held only 2.3–2.6 distinct designs, and about half of what a learned count evaluated repeated a design already measured. We then added deduplication and tested it on 20 fresh scenarios per problem, with the criterion registered first: it may trail sampling by at most 1% on either problem. It passed in both, trailing by 0.22% $[0.07, 0.39]$ on sensor placement and leading by 0.84% $[0.42, 1.23]$ on tomography. Table 2 suggests why repeats cost more here: the Pareto front spans the whole count range, against 1.3–1.5% on the benchmarks.

We then ran the same twelve external methods on both problems, which sensor placement had no dispatch for, and registered beforehand that C3BO would lead in both. Table 4 shows that it *did not*. On tomography C3BO has the highest mean of all sixteen arms. On sensor placement it has a lower mean than four baselines that use no surrogate and than count sampling, while exceeding all eight external surrogate-based methods and the backbone. Sensor placement's best count is interior for seven of its eight trade-offs, and Proposition 4 sends a learned count to an end of the range. We state this as a scope condition: a learned count assumes the best count lies near an end of the feasible range, and where it is interior C3BO can trail. Five scenarios do not separate the arms at the top of that column.

Table 4: Nonlinear optimal-design problems from PyOED. Hypervolume AUC, mean ± SD over 5 scenarios.

Method	Sensor	Tomo.
Uniform	1.200 ± 0.005	0.504 ± 0.006
PBIL	$\mathbf{1.205} \pm 0.010$	0.510 ± 0.006
NSGA-II	1.201 ± 0.008	0.513 ± 0.009
MOEA/D	1.201 ± 0.007	0.506 ± 0.009
CASMOPOLITAN	1.185 ± 0.010	0.501 ± 0.007
qLogNEHVI	1.185 ± 0.009	0.498 ± 0.006
BODi	1.187 ± 0.008	0.500 ± 0.006
Bounce	1.192 ± 0.010	0.501 ± 0.008
qLogNParEGO	1.186 ± 0.008	0.498 ± 0.007
qPOTS	1.188 ± 0.008	0.500 ± 0.006
GIT-BO	1.191 ± 0.007	0.501 ± 0.004
STAGE-BO	1.185 ± 0.009	0.498 ± 0.007
Count sampling	1.198 ± 0.006	0.522 ± 0.011
C3BO	1.195 ± 0.006	$\mathbf{0.524} \pm 0.006$

When the prior cannot be set. On a Gaussian process and a tabular foundation model, whose priors we do not set, a sampled count alone moves AUC by -0.9 to $+9.7\%$. On the linear backbone C3BO adds 25–57% (Figure 3).

7 Conclusion

Surrogate-based subset search settles at the wrong size because repair starts the archive away from the best sizes and the prior strongly shrinks a shared item effect. C3BO corrects the prior so that the posterior chooses the count, with fixed-count subproblems as a fallback when the solver is cut off. Our benchmarks are affine, which favours a linear surrogate.

AI USE STATEMENT

An LLM-based assistant was used for coding, running experiments, formatting LaTeX tables, and editing the manuscript. The authors independently verified all experimental results and take full responsibility for the final content.

ETHICS STATEMENT

This work uses synthetically generated optimization benchmarks and published optimal-design problems and involves no human subjects, personal data, or sensitive attributes.

REPRODUCIBILITY STATEMENT

Proofs are in Appendix A, implementation and baseline details in Appendix B, solver timing and additional experimental results in Appendices D–J, and registered predictions and amendments in Appendix K. The experimental protocol and seeded instance generation are described in Section 6.1.

REFERENCES

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. *Advances in neural information processing systems*, 36:20577–20612, 2023.

Shumeet Baluja. *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning*. Carnegie Mellon University, 1994.

Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *International conference on machine learning*, pp. 462–471. PMLR, 2018.

Abhijit Chowdhary, Shady E Ahmed, and Ahmed Attia. Pyoed: An extensible suite for data assimilation and model-constrained optimal design of experiments. *ACM Transactions on Mathematical Software*, 50(2):1–22, 2025.

Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in neural information processing systems*, 34:2187–2200, 2021.

Samuel Daulton, Xingchen Wan, David Eriksson, Maximilian Balandat, Michael A Osborne, and Eytan Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. *Advances in Neural Information Processing Systems*, 35:12760–12774, 2022.

Erik Daxberger, Anastasia Makarova, Matteo Turchetta, and Andreas Krause. Mixed-variable bayesian optimization. *arXiv preprint arXiv:1907.01329*, 2019.

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.

Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Mercer features for efficient combinatorial bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7210–7218, 2021.

Aryan Deshwal, Sebastian Ament, Maximilian Balandat, Eytan Bakshy, Janardhan Rao Doppa, and David Eriksson. Bayesian optimization over high-dimensional combinatorial spaces via dictionary-based embeddings. In *International Conference on Artificial Intelligence and Statistics*, pp. 7021–7039. PMLR, 2023.

Kamil Dreczkowski, Antoine Grosnit, and Haitham Bou Ammar. Framework and benchmarks for combinatorial and mixed-variable bayesian optimization. *Advances in Neural Information Processing Systems*, 36:69464–69489, 2023.

Tobias Friedrich and Frank Neumann. Maximizing submodular functions under matroid constraints by evolutionary algorithms. *Evolutionary computation*, 23(4):543–558, 2015.

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

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. chapman and hall/cRc, 1995.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary computation*, 26(3): 411–440, 2018.

Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th international conference on machine learning*, volume 2002, pp. 315–322, 2002.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.

László Lovász. On the ratio of optimal integral and fractional covers. *Discrete mathematics*, 13(4): 383–390, 1975.

Ryan O'Donnell. *Analysis of boolean functions*, volume 2. Cambridge University Press Cambridge, 2014.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial bayesian optimization using the graph cartesian product. *Advances in Neural Information Processing Systems*, 32, 2019.

Jiarong Pan, Stefan Falkner, Felix Berkenkamp, and Joaquin Vanschoren. MALIBO: Meta-learning for likelihood-free Bayesian optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39102–39134. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/pan24b.html.

Theodore P Papalexopoulos, Christian Tjandraatmadja, Ross Anderson, Juan Pablo Vielma, and David Belanger. Constrained discrete black-box optimization using mixed-integer programming. In *International Conference on Machine Learning*, pp. 17295–17322. PMLR, 2022.

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

Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset Selection by Pareto Optimization. In *Neural Information Processing Systems*, pp. 1774–1782, 2015. URL https://mlanthology.org/neurips/2015/qian2015neurips-subset/.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Ashwin Renganathan and Kade Carlson. qpots: Efficient batch multiobjective bayesian optimization via pareto optimal thompson sampling. *Proceedings of Machine Learning Research*, 258: 4051–4059, 2025. ISSN 2640-3498. 28th International Conference on Artificial Intelligence and Statistics, AISTATS 2025.

Sancho Salcedo-Sanz. A survey of repair methods used as constraint handling techniques in evolutionary algorithms. *Computer science review*, 3(3):175–192, 2009.

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, 2015.

Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A. Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10663–10674. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wan21b.html.

Shuhei Watanabe, Noor Awad, Masaki Onishi, and Frank Hutter. Speeding up multi-objective hyperparameter optimization by task similarity-based meta-learning for the tree-structured Parzen estimator. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4380–4388, 2023. doi: 10.24963/ijcai.2023/487.

Yaohong Yang, Sammie Katt, and Samuel Kaski. Multi-objective bayesian optimization via adaptive ε-constraint decomposition. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tu7eFWXBiN.

Rosen Yu, Cyril Picard, and Faez Ahmed. Git-bo: High-dimensional bayesian optimization with tabular foundation models. In *International Conference on Learning Representations*, volume 2026, pp. 111636–111663, 2026.

Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

A PROOFS

Throughout, $\langle u, v\rangle = 2^{-n}\sum_x u(x)v(x)$, $\chi_S(x) = (-1)^{\sum_{i\in S} x_i}$, and $\hat{f}(S) = \langle f, \chi_S\rangle$.

A.1 PROPOSITION 1 (SIZE BLINDNESS)

Diagonalization. $C(x, x') = g(x \oplus x')$ is a convolution kernel on the group $\mathbb{Z}_2^n$, whose characters are the χ_S. Since $\chi_S(x \oplus x') = \chi_S(x)\chi_S(x')$,

$$C(x, x') = \sum_S \omega_S\, \chi_S(x)\chi_S(x'), \qquad \omega_S = 2^{-n}\sum_z g(z)\chi_S(z),$$

and positive semidefiniteness of C gives $\omega_S \geq 0$. *Walsh coefficients.* $\mathrm{Cov}(\hat{f}(S), \hat{f}(S')) = 2^{-2n}\sum_{x,x'} C(x, x')\chi_S(x)\chi_{S'}(x') = \omega_S \mathbf{1}[S = S']$ by orthonormality, so $\mathbb{E}\|f_j\|^2 = \sum_{|S|=j}\omega_S$. *The size-only direction.* A function $\sum_{|S|=j} a_S\chi_S$ is invariant under every coordinate permutation π iff $a_{\pi S} = a_S$, since $\chi_S \circ \pi^{-1} = \chi_{\pi S}$. Permutations act transitively on j-subsets, so a_S is constant, and the invariant part of W_j is spanned by $e_j = \sum_{|S|=j}\chi_S$, a function of $|x|$ (a Krawtchouk polynomial). Every size-only function is permutation-invariant, so its order-j part is a multiple of e_j. Let $u_j = e_j/\|e_j\|$, $\|e_j\|^2 = \binom{n}{j}$. *The share.* $\langle f, u_j\rangle = \binom{n}{j}^{-1/2}\sum_{|S|=j}\hat{f}(S)$ is a sum of uncorrelated terms with variance $\binom{n}{j}^{-1}\sum_{|S|=j}\omega_S$; dividing by $\mathbb{E}\|f_j\|^2$ gives $1/\binom{n}{j}$. □

Which kernels qualify. For binary inputs $(x_i - x_i')^2 = (x \oplus x')_i$, so every kernel of the form $\phi(\sum_i (x_i - x_i')^2/\ell_i^2)$ is a function of $x \oplus x'$: RBF and Matérn with or without ARD, weighted overlap and diffusion kernels. A stationary kernel with a non-diagonal lengthscale matrix does not qualify. The Gaussian process shared by qLogNEHVI, qLogNParEGO and qPOTS – all three reach it through the same `_gp_acqf` – is ARD-RBF on inputs passed through a per-coordinate affine normalization whose learned scale is 1 and offset 0 on every coordinate of a binary design matrix (constant columns included) and whose bounds are fixed after fitting; it therefore leaves $(x_i - x_i')^2$ unchanged. CASMOPOLITAN does not share that model: in the MCBO build we use it is `gp_to`, a Gaussian process with a transformed overlap kernel. It is covered by the proposition all the same, since an overlap kernel on binary inputs depends on x and x' only through $x \oplus x'$.

What a fitted Gaussian process recovers. We refit the benchmark's own configuration (`SingleTaskGP`, ARD-RBF, standardized outcomes, BoTorch 0.17.2) on its stored qLogNEHVI archives, four families × five seeds × two objectives, and project the posterior mean onto u_1. The projection is closed-form in the fitted lengthscales and is checked at $n = 8$ against an explicit sum over all 256 designs. Medians: 0.3% (MOKP), 0.4% (MDK), 20.1% (set cover), 2.8% (MWIS). Set cover's ten per-objective values are 18.3, 18.6, 19.1, 19.4, 19.9, 20.2, 25.5, 26.1, 35.0 and 35.2%, so what misses the level registered in Appendix K is the spread and not one seed. Two checks guard against reading an implementation fault as a property of Gaussian processes. The covariance module is the one BoTorch builds: supplying `ScaleKernel(RBFKernel(ard_num_dims=n))` by hand drops the dimension-scaled lengthscale prior, after which the fit never leaves its initial point and every lengthscale returns $\mathrm{softplus}(0) = 0.693$. And the configuration is the adapter's: wrapping `fit_gpytorch_mll` during a short adapter campaign shows it does fit, at lengthscale 3.238 on MOKP and spread over 0.58–4.98 on set cover, matching this refit. The size-aware version replaces the constant mean with $c + \beta(|x|/n - \overline{|x|/n})$, both parameters free and unpenalized; since $\langle |x|/n, u_1\rangle = -1/(2\sqrt{n})$ its contribution to the projection is exact, and a truth $w^\top x$ is recovered at 100% when $\beta = n\,\overline{w}/\hat{\sigma}_y$. Its medians are 104.4, 96.5, 100.8 and 98.7%.

Numerical check. For $n = 8$ with random ARD lengthscales (RBF, Matérn-5/2 and weighted overlap), all eight orders give $1/\binom{n}{j}$ to machine precision, and the off-diagonal Walsh covariances are below 10^{-15}.

A.2 THE LINEAR SURROGATE

With $x_i = (1 - \chi_{\{i\}}(x))/2$, $a + w^\top x = \big(a + \frac{n}{2}\bar{o}(w)\big) - \frac{1}{2}\sum_i w_i\chi_{\{i\}}(x)$. So $\hat{f}(\emptyset) = a + \frac{n}{2}\bar{o}(w)$, $\hat{f}(\{i\}) = -w_i/2$, and all higher coefficients vanish. The $\hat{f}(\{i\})$ are i.i.d., so the argument of Appendix A.1 applies at order one and the share is $1/n$. Since $\sum_i \hat{f}(\{i\}) = -\frac{n}{2}\bar{o}(w)$ and $e_1 =$

$n - 2|x|$, the size-only order-one part of f is $\bar{o}(w)(|x| - n/2)$, with prior variance $\mathrm{Var}(\bar{o}(w))$. The intercept and the size effect both enter $\hat{f}(\emptyset)$, which is the only coupling between them.

A.3 A NARROW ARCHIVE: STATEMENTS AND PROOFS

Section 4.2 states Proposition 2 in the flat-intercept form, because that is the form in which the two causes meet in one ratio. Our backbone puts a proper prior on the intercept, and that form is here.

Proposition 5 (Size-effect gain, proper intercept)**.** *Let the standardized outputs be modelled as $a_0 + w^\top x + \varepsilon$ with the intercept and every item coefficient i.i.d. $\mathcal{N}(0, \tau^2/(n+1))$ and noise $\mathcal{N}(0, \sigma^2)$, and fix (τ, σ) and the archive $X \in \{0,1\}^{N \times n}$ with sizes $\mathbf{k} = X\mathbf{1}$. Then $\bar{o}(\hat{c}) = \rho_{\mathcal{A}}\, \bar{o}(w) +$ (terms not involving $\bar{o}(w)$), where $\rho_{\mathcal{A}} = \frac{1}{n}\mathbf{k}^\top(\mathbf{1}\mathbf{1}^\top + XX^\top + \kappa I)^{-1}\mathbf{k}_c$, $\kappa = (n+1)\sigma^2/\tau^2$ and $\mathbf{k}_c = \mathbf{k} - \bar{k}\mathbf{1}$.*

Lemma 6 (Monotone size)**.** *If $e < e'$, then $|x(e)| \le |x(e')|$ for all maximizers $x(e) \in \arg\max_{\mathcal{F}}(c + e\mathbf{1})^\top x$.*

Proposition 5. With $\Phi = [\mathbf{1}_N, X]/\sqrt{n+1}$ and standardized outputs $(y - \bar{y}\mathbf{1})/s$, the posterior mean of the item coefficients in objective units is

$$\hat{c} = \tfrac{s\tau^2}{n+1} X^\top \Big(\tfrac{\tau^2}{n+1}(\mathbf{1}\mathbf{1}^\top + XX^\top) + \sigma^2 I \Big)^{-1} \tfrac{y - \bar{y}\mathbf{1}}{s} = X^\top (\mathbf{1}\mathbf{1}^\top + XX^\top + \kappa I)^{-1}(y - \bar{y}\mathbf{1}).$$

Decompose $y - \bar{y}\mathbf{1} = \bar{o}(w)\mathbf{k}_c + (Xw_\circ - \overline{Xw_\circ}\mathbf{1}) + (\varepsilon - \bar{\varepsilon}\mathbf{1})$ and use $\mathbf{1}^\top X^\top = \mathbf{k}^\top$: $\bar{o}(\hat{c}) = \mathbf{1}^\top \hat{c}/n$ equals $\rho_{\mathcal{A}}\bar{o}(w)$ plus terms linear in $w_\circ$ and ε. □

Proposition 2. Let $\Sigma = s^2(I + L^2\mathbf{1}\mathbf{1}^\top)$ and $\beta = 1/n + L^2$. With a flat intercept, $\hat{c} = \Sigma X_c^\top (X_c \Sigma X_c^\top + \sigma^2 I)^{-1} y_c$. Since $\mathbf{1}^\top \Sigma = s^2(1 + nL^2)\mathbf{1}^\top = s^2 n\beta\, \mathbf{1}^\top$ and $X_c\mathbf{1} = \mathbf{k}_c$,

$$\bar{o}(\hat{c}) = \tfrac{1}{n}\mathbf{1}^\top \hat{c} = \beta\, \mathbf{k}_c^\top \big(X_c(I + L^2\mathbf{1}\mathbf{1}^\top)X_c^\top + \kappa I \big)^{-1} y_c, \qquad \kappa = \sigma^2/s^2.$$

Because $\tilde{X}\mathbf{1} = 0$, $X_c X_c^\top = \tilde{X}\tilde{X}^\top + \mathbf{k}_c\mathbf{k}_c^\top/n$, so $X_c(I + L^2\mathbf{1}\mathbf{1}^\top)X_c^\top = \tilde{X}\tilde{X}^\top + \beta\, \mathbf{k}_c\mathbf{k}_c^\top$. Write $y_c = \bar{o}(w)\mathbf{k}_c + X_c w_\circ + \varepsilon_c$ and $M = \tilde{X}\tilde{X}^\top + \kappa I$. Sherman–Morrison gives $\mathbf{k}_c^\top (M + \beta \mathbf{k}_c\mathbf{k}_c^\top)^{-1}\mathbf{k}_c = a/(1 + \beta a)$, so the coefficient of $\bar{o}(w)$ is $\beta a/(1+\beta a) = a/(a + 1/\beta) = a/(a + \eta_L)$; the other two terms of y_c do not involve $\bar{o}(w)$. Since $M \succeq \kappa I$, $a \le \|\mathbf{k}_c\|^2/\kappa$. At $L = 0$ and $s^2 = \tau^2/(n+1)$ the coefficient is $a/(n+a)$ with $\kappa = (n+1)\sigma^2/\tau^2$, the independent-prior case whose proper-intercept form is Proposition 5. □

The single-size case. If every archived design has size K_0 then $\mathbf{k} = K_0\mathbf{1}$, so $\mathbf{k}_c = 0$ and $a = 0$. For the stronger statement, $y_i = a_0 + \bar{o}(w)K_0 + w_\circ^\top x_i + \varepsilon_i$ and the first two terms are the same for every i. Output standardization is invariant to adding a constant, and every later quantity (the empirical-Bayes objective, $(\hat{\tau}, \hat{\sigma})$, the posterior mean and every draw for a fixed random state) is a function of the standardized outputs, X and the random state only. □

Lemma 6. Optimality gives $c^\top x(e) + e|x(e)| \ge c^\top x(e') + e|x(e')|$ and $c^\top x(e') + e'|x(e')| \ge c^\top x(e) + e'|x(e)|$; adding them yields $(e' - e)(|x(e')| - |x(e)|) \ge 0$. □

A.4 REMARK 3 AND PROPOSITION 4

Invariance. On $\mathcal{F}_K$ a size-only function is constant. □

Edge. (a) If $x_i^* = 1$ and $\hat{c}_i < 0$, then $x^* - e_i \in \mathcal{F}$ by downward closure and has larger value, a contradiction; items with $\hat{c}_i = 0$ can be set to 0 without loss. (b) If x^* were not inclusion-maximal, some $x^* + e_i \in \mathcal{F}$ would have larger value by $\hat{c}_i > 0$. For upward-closed $\mathcal{F}$ the same arguments apply to adding in (a) and removing in (b). The size band in (b) is immediate: an inclusion-maximal element has no feasible superset, so $|x^*|$ cannot fall outside the sizes such elements take. □

A.5 THE COUNT STAGE: STATEMENTS AND PROOFS

Section 6.1 and Section 5.1 state these in prose.

Proposition 7 (Coverage of requested counts)**.** *If each of B proposals draws its* requested *count independently from q, the probability that no request lands in a set of sizes G is at most $(1 - q(G))^B$.*

Proposition 8 (Anchor). *(a) $\hat{c}^\top \hat{x} \geq \hat{c}^\top x_U$ for every limit T. (b) If the solver's upper bound U satisfies $U - \hat{c}^\top x_U \leq \epsilon|\hat{c}^\top x_U|$, no feasible x exceeds $\hat{c}^\top x_U$ by more than $\epsilon|\hat{c}^\top x_U|$.*

Coverage. The requested count of each proposal is drawn from q independently of the others, so it lands in G with probability $q(G)$ and none of the B requests does with probability $(1 - q(G))^B$. The bound is on requested counts: the realized count can differ when a slice is empty or the solver returns no incumbent and the repaired backup is used. *Anchor.* (a) x_U is a candidate in Eq. (2). (b) Every feasible x satisfies $\hat{c}^\top x \leq U \leq \hat{c}^\top x_U + \epsilon|\hat{c}^\top x_U|$. □

Both parts are about the selection in Eq. (2), which ranks candidates before deduplication. If the candidate it chooses is a design already evaluated, that design is dropped and the next one is proposed, and the guarantee does not carry to it: the proposal is then the best of the candidates that remain, not of all of them. Where the arm reports a repeat rate above zero, the bound holds for the ranking and not for every evaluated design.

A.6 How tight the edge relaxation is

Lemma 9 (LP structure). *(i) If $\mathcal{F} = \{x : Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}_{\geq 0}$ and $\hat{c} > 0$, the LP relaxation exceeds the integer optimum by at most $m \max_i \hat{c}_i$. (ii) For independent set, $x = \frac{1}{2}\mathbf{1}$ is LP-feasible, so the relaxation is at least $\frac{1}{2}\sum_i \hat{c}_i$ while the integer optimum is at most $\alpha \max_i \hat{c}_i$, with α the independence number. (iii) For set cover the ratio of integer to LP optimum can grow logarithmically (Lovász, 1975; Chvatal, 1979).*

(i) A vertex of $\{Ax \leq b, 0 \leq x \leq 1\}$ has n linearly independent tight constraints, at most m of them from $Ax \leq b$, so at most m coordinates are fractional. Rounding them down keeps feasibility because $A \geq 0$ and loses at most $m \max_i \hat{c}_i$. (ii) Each edge constraint reads $\frac{1}{2} + \frac{1}{2} \leq 1$; an independent set has at most α items. (iii) See Lovász (1975) and Chvatal (1979). □

B External methods and how each was adapted

Nine of the thirteen were not proposed for constrained multi-objective search over binary variables, so each needed an adapter. We list what each one is and what the adapter does, with the identifier the runner uses, so that a reader can find the code path. Every method receives the same initial archives, the same 100 evaluations in batches of 8, and the same feasibility repair applied to its proposals.

Multi-objective Bayesian optimization with a Gaussian process. *qLogNEHVI* (`qnehvi_roundrepair`) and *qLogNParEGO* (`qlognparego_slotwise_roundrepair`) use the BoTorch acquisitions unchanged. Since the acquisition cannot be maximized by gradient steps over $\{0,1\}^{250}$ under the constraints, it is maximized over a finite pool of feasible designs drawn by the problem's own sampler and filtered to candidates not yet evaluated; the batch is then built one design at a time, each selection entered as pending so the next accounts for it. qLogNParEGO draws one scalarization weight per batch slot. *qPOTS* (`qpots_hls`) keeps the method's Thompson sampling over the posterior Pareto set – one coherent Matheron path per objective – and its archive-relative maximin batch rule, and replaces only the continuous NSGA-II inner search: Pareto-archive neighbours and Sobol binary starts seed a small population evolved by repaired 1–4 bit Hamming mutations. *STAGE-BO* (`stagebo_rr`) keeps the official Gaussian process and adaptive ϵ-constraint selection, and adds a finite posterior-front search with projection, repair and acquisition reranking.

Single-objective combinatorial Bayesian optimization. *CASMOPOLITAN* (`casmo_decomp`) and *BODi* (`bodi_decomp`) are single-objective methods, taken from the MCBO implementations (Dreczkowski et al., 2023). Each is run as a decomposition: one instance per scalarization arm, with the two objective weights spread evenly from 0.05 to 0.95 across arms, and the batch slots divided among the arms.

Foundation-model and meta-learned surrogates. *GIT-BO* (`gitbo_rr`) keeps the TabPFN surrogate and the gradient-informed active subspace, with ParEGO scalarization, finite-difference predictive gradients, binary projection, repair and quantile-UCB reranking. *MALIBO* (`malibo_s`) is

the official MetaBLOR model with a scalarization-conditioned adapter for binary multi-objective search. *Meta-MO-TPE* (`meta_motpe`) is the official MO-TPE with cross-task meta-learning; it estimates densities over the discrete variables directly.

Methods that need no surrogate adapter. *NSGA-II* (`nsga2`), *MOEA/D* (`moead`) and *PBIL* (`pbil`) are native binary multi-objective optimizers and run with binary variation operators; MOEA/D's ask/tell updates are matched to the evaluation budget. *Uniform* (`random`) draws feasible designs uniformly and is the non-Bayesian control.

Which implementation each wraps. Recorded by the runner that produced the results, not reconstructed afterwards.

Method	Implementation
qLogNEHVI, qLogNParEGO	BoTorch (Daulton et al., 2021; Ament et al., 2023)
qPOTS	`github.com/alonrot/qpots`
STAGE-BO	`github.com/YangYaohong1/STAGE-BO`
CASMOPOLITAN, BODi	MCBO, `github.com/huawei-noah/HEBO`
Bounce	`github.com/LeoIV/bounce`
GIT-BO	`github.com/rosenyu304/GITBO`
MALIBO	`github.com/boschresearch/meta-learning-likelihood-free-bayesian-optimization`
Meta-MO-TPE	`github.com/nabenabe0928/meta-learn-tpe`
NSGA-II, MOEA/D	`pymoo`
PBIL, uniform	ours, from the published description

What the adapters are not. No adapter changes a method's surrogate, its acquisition, or its batch rule. What they change is how the acquisition is searched, because a continuous inner optimizer has nothing to do on a constrained binary domain, and how a single-objective method is given two objectives. The pool and projection steps are a limit on these baselines and we do not claim otherwise: a method whose published strength is a continuous inner search is handicapped by a finite pool. This is one reason the paper's claim is about where the count is decided rather than about which surrogate is best, and why the backbone – a Bayesian linear model maximized exactly by a mixed-integer solver, with no pool and no projection – is reported alongside them.

C THE HYPERVOLUME REFERENCE POINT

Hypervolume is undefined without a reference point, and on these problems the choice is not cosmetic. The natural-looking choice, the origin, is the empty design: with objectives normalized to $[0, 1]$ a design that selects nothing scores zero on both. That places the reference point exactly on the boundary of the attainable region, so a point that is excellent in one objective and near zero in the other contributes almost no volume. Search that fills the middle of the front is then rewarded over search that reaches its ends, which is the opposite of what we want to measure here, since the best counts sit at one end of the feasible range in all four families.

We therefore place the reference point one stratum outside the normalized box in each objective, following Ishibuchi et al. (2018), who set it at $1 + 1/H$ in normalized minimization coordinates with H the number of divisions of the weight simplex and recommend $H = \mu - 1$ for μ compared solutions. A batch here holds eight scalarizations, so $\mu = 8$, $H = 7$, and with two objectives the $H + 1 = 8$ weight vectors are exactly the eight the batch uses. Mirrored to our maximized, normalized objectives this is $r = (-1/7, -1/7)$. Every non-dominated point then makes a bounded positive contribution regardless of where on the front it sits.

This is the more conservative choice for us, not the more flattering one. Under the origin the headline margin of a learned count over a sampled one on the main set at 5 s is $+19.5$, $+20.3$, $+11.4$ and $+11.7\%$; under the reference point we report it is $+15.7$, $+16.1$, $+9.7$ and $+8.7\%$. Every table and figure in the paper is computed at the reported reference point, with one exception, Table 9, whose caption states it: its first two rows are designs discarded before the reference point changed and their runs were not kept.

D SOLVE LIMITS AND MEASURED TIMES

The two per-solve limits are 1 s and 5 s. The solver honours them to within tens of milliseconds, and Table 5 reports the measured median solver seconds per proposal for every arm. With the schedule enforced, a learned count costs 0.80 and 0.81 s in the two hard families against a nominal 1 s.

A solve-limit defect, and what it changed. The slice solver is bound at module import, and the routine that raises the wall-clock cap rewrote the name in the two modules the benchmark's other arms use but not in that one, so slice solves silently kept a hardcoded 1 s cap and never received the requested limit. Re-solving the same recorded candidate with the limit actually delivered finishes in 80 ms against the 56 ms requested; the same solve with the module default takes 1.040 s, matching the 1.022 s recorded. Every recorded edge slice in that proposal took between 1.005 and 1.022 s irrespective of its requested limit or its size, which is the signature of a fixed cap and not of edge difficulty. The unrestricted solve in the same proposal, which goes through a module that was patched, honoured its limit (0.524 s against 0.500 s).

Table 5 lists every arm, whether the defect could bite it, and what was re-run. An arm is affected only if its slice solves would have run longer than the limit it was supposed to receive, so an arm whose nominal limit was already 1 s is unaffected even though it went through the same code path. The earlier claim that the main comparison was untouched was wrong: the long-limit sampled count is part of that comparison and was capped.

Table 5: Every arm against the solve-limit defect. "Bites" is whether the hardcoded 1 s cap was below the limit the arm should have received; the maximum wall time observed is the evidence. All affected arms were re-run with the limit delivered, and every number in this paper comes from the re-run.

Arm	Nominal	Max wall	Bites	Status
learned count, short and long	1/5 s	1.02 s	yes	re-run, 320 runs
sampled count, long	5 s	1.08 s	yes	re-run, 80 + 80 runs
corrected + sampled, long	5 s	1.07 s	yes	re-run, 80 runs
sampled count, short	1 s	1.66 s	no	unchanged
corrected + sampled, short	1 s	1.07 s	no	unchanged
C3BO w/o fallback	1/5 s	–	no	patched module
Linear + MIP (backbone)	–	0.003 s	no	unchanged

Two results changed. The held-out margin of the count stage over the better single remedy went from three of four families to four of four, because the earlier MOKP shortfall was an artifact of the cap. And the margin of the corrected search over the sampled count *fell*, from 14.7–20.6% to 11.0–20.3% on the main set, because the sampled count had been the arm most penalised by the cap in the two families whose edge is hard.

E SOLVE-LIMIT SWEEP

Five repeats per target, five targets, fresh process per repeat. At 1 s C3BO w/o fallback has a within-target standard deviation of 0.1274 against a mean of 0.6922 in set cover, 18.4% of its mean, and 0.0049 against 0.3017 in MWIS; a learned count has 0.0052 against 0.9528 and 0.0007 against 0.3080, 0.5% and 0.2%. Its mean over repeats is above the backbone's 0.636, so a single low run is a draw from the lower tail rather than a collapse. Under the registered rule, a difference is reported in the body only when it exceeds three times the within-target standard deviation of the difference; at 1 s the ratio is 2.0 in set cover and 1.2 in MWIS, so the comparison stays here. At 5 s the ratios are 25.5 and 23.7.

F SUPPORTING TABLES

The diagnosis, in full. Table 7 gives every row of the diagnosis of Section 4.3 per family, from offline re-solves of the backbone's stored campaigns at three rounds. Rows 1 and 2 are the two

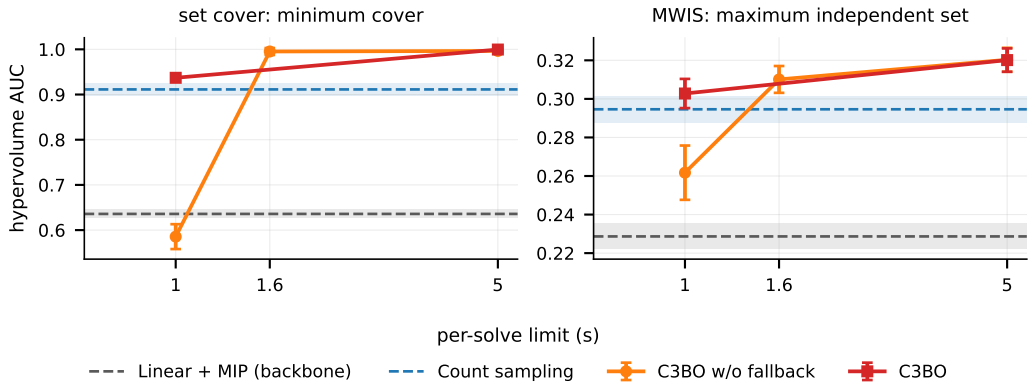


Figure 4: Hypervolume AUC against the nominal per-solve limit (95% intervals over 20 targets, one run per target). At 1 s C3BO w/o fallback has a within-target standard deviation of 18.4% of its mean, so its point there is one draw from a wide distribution and the apparent cliff is not a threshold; the repeated measurement below is the reliable comparison.

Table 6: Hypervolume AUC against measured solver seconds, on the two families whose edge is hard. Each cell is AUC (median solver seconds per proposal). With the solver's count the corrected search is far weaker at 1.0 s than at 1.6 s in set cover and recovers only once the limit is raised; with a count stage it does not depend on the limit that way, and it stays inside its nominal limit. The one-second row is reported here rather than in the body because repeats of it are unstable (Appendix K).

Arm	SetCover	MWIS
Linear + MIP (backbone)	0.636 (0.00)	0.229 (0.00)
sampled count, 1 s	0.843 (0.01)	0.288 (0.02)
sampled count, 5 s	0.911 (0.01)	0.295 (0.02)
solver's count, 1 s	0.586 (1.03)	0.262 (1.01)
solver's count, 1.6 s	0.995 (1.61)	0.310 (1.61)
solver's count, 5 s	0.996 (5.00)	0.320 (5.00)
learned count, 1 s	0.937 (0.80)	0.303 (0.81)
learned count, 5 s	1.000 (3.53)	0.320 (3.35)

causes side by side: the exact best counts, and the sizes the initial archives actually contain. The size-effect row is 2.1 in every family and is stated in Section 3.

What the correction does to the solver. Table 8 is the other side of Proposition 4: once every item carries the same sign the solver can discard nothing, so the share of items it must consider goes from a third or a half to all of them, and its median terminal gap grows with the difficulty of the edge problem. Correcting the prior makes the acquisition step harder, which is the cost the count stage is there to absorb.

G Development history and exploratory variants

These are the ways of choosing a learned count that we tried, in the order we ran them. They are a record of how the method was arrived at, not a controlled ablation of the final one: the sample size differs in row 3, and rows 1–4 ran before the solve-limit defect of Appendix D was fixed, so their budgets are not comparable with the final row.

H Held-out scores and the registered margin

At 1 s the held-out margin of a learned count over a sampled one is 14.4, 17.7, 10.6 and 6.1%, against 16.6, 16.6, 11.1 and 4.9% on the main set; the ordering of the families reproduces, including MWIS

Table 7: Diagnosis on the backbone's stored campaigns, from offline re-solves at rounds 3, 6 and 10 (20 instances and 8 roles per family, 480 states each). No new evaluations. Row 1 is the min–max of the exact $K_h^\star$ over the 20 targets and 8 roles. Row 2 is the min–max size *observed* in the 20 initial archives (20 uniform draws each, after repair); it is what the initialization produced, not the support of what it could produce. Rows 3–5: the posterior recovers almost none of a large size effect, and its error has the sign Lemma 6 predicts. Rows 6–7: re-solving the same states with the size effect corrected and nothing else moves the solution to the best size.

	MOKP	MDK	SetCover	MWIS
$K_h^\star$ band	136–152	129–138	42–53	88–103
Observed initialization count range	105–131	105–126	112–152	54–77
Size effect $\lvert\bar{o}(w)\rvert/\mathrm{SD}(w_\circ)$	2.1	2.1	2.1	2.1
Size effect recovered by the posterior	-0.6%	-0.0%	15.7%	-4.9%
Size error has the predicted sign	100%	100%	83.8%	100%
Within 10 of $K_h^\star$: as run	0.2%	8.5%	0%	0%
Within 10 of $K_h^\star$: size effect corrected	$\mathbf{63.1}\%$	$\mathbf{77.9}\%$	$\mathbf{89.4}\%$	$\mathbf{90.8}\%$
Campaigns entering the $K_h^\star$ band	0%	15%	0%	0%

Table 8: What the size correction does to the acquisition problem. $|P(\hat{c})|/n$ is the share of items the solver cannot discard (Proposition 4): those with $\hat{c}_i > 0$ when $\mathcal{F}$ is downward-closed, $\hat{c}_i < 0$ when it is upward-closed. Gaps are median relative optimality gaps at termination, read from the stored solves.

	MOKP	MDK	SetCover	MWIS
Closure	down	down	up	down
Edge problem	knapsack (m=2)	knapsack (m=5)	min. cover	max. indep. set
$\lvert P(\hat{c})\rvert/n$, uncorrected	47.0%	46.7%	36.5%	36.4%
$\lvert P(\hat{c})\rvert/n$, corrected	100%	100%	100%	100%
$K_h^\star$ distance to the edge*	10	9	4	3
Gap, corrected, 1 s	0	0.0014	0.68	7.7
Gap, corrected, 5 s	0	0.0010	0.06	0.10
Gap, fixed-count slices, 1 s	0	0	0	0

* For set cover and MWIS the edge is the minimum cover and the maximum independent set. Neither was proven optimal within the 60 s cap in any of the 40 targets, so the value used is the best incumbent found and the distance is measured against an estimate, not the true extreme. The incumbent is conservative in the direction that matters: it can only shrink the candidate range, never admit an infeasible count.

as the one where the advantage of learning shrinks most when the solver is hurried. Against the registered comparison, the better of the corrected search and the sampled count, the short-limit figures are -0.41% $[-0.87, +0.06]$ in MOKP, $+0.31\%$ $[-0.19, +0.82]$ in MDK, $+12.22\%$ $[+10.10, +14.41]$ in set cover and $+5.65\%$ $[+1.25, +10.86]$ in MWIS. Runs made before the solve-limit defect was fixed are listed in Appendix D.

I THE OPTIMAL-DESIGN PROBLEMS

The right panel is the state the registered predictions were judged in, before deduplication. Deduplication was designed on these twenty scenarios per domain, so any comparison on them is in-sample. It was therefore confirmed on twenty fresh scenarios per domain instead, drawn by jittered stratification over the same parameter box and sharing no coordinate value with the stored set, with the criterion fixed beforehand. The criterion passed in both domains. The sensor gap is a real loss and not a tie: the learned count trails sampling there by 0.22% with an interval that excludes zero, which is within the 1% the criterion allowed but is not the same as being level.

J ADDITIONAL RESULTS

This appendix collects further results that support Section 6.

Table 9: Ways of choosing a learned count, in the order run, at the 1 s limit, on 20 paired instances except row 3 (10). Fallback is the share of *posterior-branch* proposals that fell back to the backup design; row 1 also draws a quarter of its counts from the fixed law, whose own fallback rate is 13.8%, so its overall rate is lower than the 100% shown. Rows 3–5 were designed after main-set results. The AUC columns are computed against the origin rather than the reference point of Section 6.1: rows 1 and 2 are designs that were discarded before the reference point changed and their runs were not kept, so the table is internally consistent but not comparable cell by cell with Table 3.

	Count stage	Fallback		AUC	
		SetCover	MWIS	SetCover	MWIS
1	LP ranking	100%	100%	0.577	0.152
2	LP, integer-feasible range	24.2%	100%	–	–
3	seven counts, one solve each	0%	0%	0.661	0.165
4	anchor + 4 quantile counts	0%	0%	0.559	0.147
5	anchor + edge band	0%	0%	**0.681**	**0.166**
	solver's count, short / long	–	–	0.388 / 0.732	0.136 / 0.179
	sampled count	–	–	0.601	0.155

Table 10: Held-out hypervolume AUC, mean ± SD over 20 paired instances, Ishibuchi reference point. The body reports the margin between the last two rows of each block; these are the scores behind it. The backbone does not use the per-solve limit, so its two rows are the same runs.

Limit	Arm	MOKP	MDK	SetCover	MWIS
5 s	Linear + MIP (backbone)	0.467 ± 0.009	0.456 ± 0.008	0.499 ± 0.018	0.196 ± 0.006
	Count sampling	0.570 ± 0.021	0.501 ± 0.029	0.911 ± 0.038	0.289 ± 0.017
	C3BO	0.653 ± 0.017	0.590 ± 0.012	1.005 ± 0.017	0.319 ± 0.019
1 s	Count sampling	0.570 ± 0.021	0.500 ± 0.029	0.855 ± 0.037	0.286 ± 0.018
	C3BO	0.651 ± 0.016	0.588 ± 0.014	0.945 ± 0.027	0.303 ± 0.016

Completing proposals to the edge. In our families every item effect has one sign, so every Pareto-optimal design is inclusion-extremal. A natural alternative to correcting the prior is therefore to complete each proposal of a baseline greedily to an inclusion-extremal design. Table 11 applies this completion to qLogNEHVI. It helps most in set cover and MWIS. It still leaves qLogNEHVI 9–26% below C3BO in every family, and below count sampling in both knapsack families. Reaching the edge is not enough when inclusion-extremal designs span many counts. The count within that band, and the composition at that count, still have to be chosen, and the corrected posterior is what chooses them.

Table 11: Greedy completion to inclusion-extremal designs, main set. Hypervolume AUC, mean over 20 paired targets. The change row gives the relative gain of completion with 95% paired bootstrap intervals.

	MOKP	MDK	SetCover	MWIS
qLogNEHVI	0.462	0.448	0.713	0.245
qLogNEHVI + completion	0.482	0.459	0.907	0.288
Change (%)	+4.3 [+2.4, +6.8]	+2.5 [+1.6, +3.3]	+27.2 [+25.5, +29.3]	+17.5 [+13.4, +22.4]
Count sampling	0.559	0.509	0.911	0.295
C3BO	**0.652**	**0.593**	**1.001**	**0.321**

Replication set and a further adapter. The sampled count was also compared on the replication set, for 52 further comparisons with the same outcome. A Bounce-based decomposition adapter (Papenmeier et al., 2023), registered and run on a separate instance set, loses all four families to it in its own Holm family, largest adjusted p 4×10^{-4}.

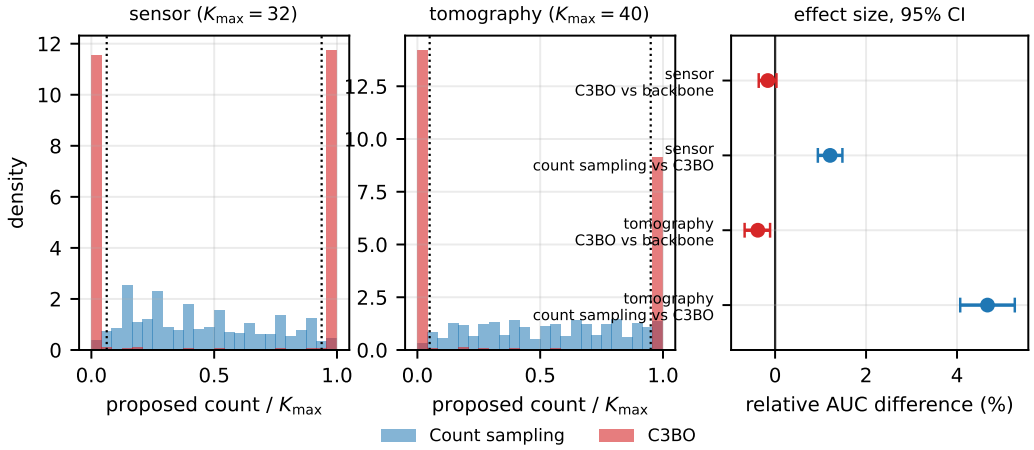


Figure 5: Two optimal-design problems from PyOED, whose only constraint is the count. Left and centre: proposed counts over $K_{\max}$, with the ± 2 of prediction P2 dotted. Right: relative difference in hypervolume AUC, 95% intervals, before deduplication.

Solver time at the same nominal budget. Both count stages run at the same nominal 5 s per solve. Their measured solver times differ by three orders of magnitude; Appendix D gives them.

How unstable a cut-off solve is. At 1 s, on one set-cover target, five runs of C3BO w/o fallback range from 0.5874 to 0.9973 in hypervolume AUC. That spread is why the one-second comparison is reported here and not in the body.

The oracle used on the optimal-design problems. The greedy curve is a lower bound on the slice optimum. Only in tomography, whose first objective is a log-determinant gain and so monotone submodular, does it carry a $(1-1/e)$ guarantee; sensor placement's first objective is an A-optimality criterion, and the sweep there is a heuristic.

Deduplication moves the count, it does not improve it. On tomography, deduplication moves the proposed count *further* from $K_h^\star$ – a median error of 16 items against 1 – and the arm still wins. What changed is that the batch stopped collapsing, not that the count became more accurate. Tomography has the most accurate learned count of the six problems here and the worst collapse, which is why the rule rests on the prior rather than on where $K_h^\star$ lies.

The counts behind Table 1. Median proposed count in the final round, against the exact best count $K^\star$, for the control of Section 5.1. With $L = 0$: 135, 127, 86 and 84 in MOKP, MDK, set cover and MWIS. With $L = 50$: 154, 141, 46 and 94. Best counts: 145, 134, 46 and 97. A shrunk size effect selects too few items where the feasible set is downward-closed and too many where it is upward-closed, which is set cover; both directions appear here. Lemma 6 gives the sign. These are not error bars: the median over a final round of eight roles is not a single design's count, and $L = 50$ is not exactly at $K^\star$ either.

Giving a foundation model the size directly. Instead of a count stage, the size can be handed to the surrogate as an extra input column. On GIT-BO this recovers 70% and 68% of the count stage's gain in the two knapsack families and 5% and 13% in the other two. With four families this is a pattern and not a test.

K Registered predictions and amendments

Every arm in Section 6 was registered before it ran. We list the predictions as made, including those that failed, and every amendment.

The count stage (benchmark families). Registered claim: at every solve limit the learned count stays within a stated margin of the better of the two single remedies in every family. Met in four of four at both limits on the held-out set. Three parameters were changed after main-set results and before the held-out verdict: candidate counts moved from quantiles of the count law to fixed offsets from the edge; the early-stop rule was tightened to require a finite reported gap; and the budget was counted on measured wall-clock rather than on a fixed schedule. The offset ladder $\{2, 5, 10, 20, 40\}$ is the most arbitrary choice in the method and is not derived from anything.

The optimal-design problems. Four predictions were fixed before any run, with the instruction to report P2 first because the rest of the reasoning depends on it. P2 (a learned count goes to an end of the feasible range in more than half of proposals) held, at 97.5% and 97.3%. P1 (learning beats the uncorrected control, the arm this paper now calls Linear + MIP) *failed* in both domains. P3 (sampling is at least as good as learning) held in both, and by most in tomography, which is the domain we had said we were least sure of. P4 (the count stage places C3BO between the two) is moot: the unrestricted solve certifies optimality in about 1 ms, so the count stage never places a candidate and C3BO with a learned count is identical to the corrected search. We report this as degeneracy, not as a confirmation.

Deduplication, on scenarios it was not designed on. Registered before the fresh scenarios were drawn: a learned count with deduplication trails a sampled count by at most 1% in *both* optimal-design domains. Met, at -0.22% $[-0.39, -0.07]$ on sensor placement and $+0.84\%$ $[+0.42, +1.23]$ on tomography. The stored scenarios gave -0.03% and $+1.12\%$, so the fresh numbers are close to the in-sample ones, which is the result the registration was written to test. The sensor interval excludes zero, so a learned count is reliably a little behind there; the criterion allowed that and we do not restate it as a tie.

External methods on the optimal-design problems. Two predictions were fixed before the twelve external methods were run on either domain, and before any of them had been run on sensor placement at all, which needed a dispatch that domain did not have.

Registered: C3BO with a learned count has the highest mean AUC of all sixteen arms in *both* domains. *Failed.* It has the highest mean on tomography, at 0.5238 against 0.5134 for the best external method. On sensor placement five arms have a higher mean: PBIL at 1.2045, NSGA-II and MOEA/D at 1.201, uniform sampling at 1.2000, and count sampling at 1.1980, against 1.1950. The four external ones use no surrogate; C3BO is ahead of all eight surrogate-based methods there and ahead of the backbone. The rows Table 4 leaves out, on the same five scenarios, are the backbone at 1.1918 and 0.5038, C3BO with a sampled count at 1.1980 and 0.5224, the learned count before deduplication at 1.1930 and 0.5039, and uniform sampling at 1.2000 and 0.5039, for sensor placement and tomography. Five scenarios separate neither the top five of that column nor the order within them, so the reading we take is that C3BO does not lead on sensor placement, not that PBIL beats it.

Registered: at least one external method beats the linear backbone on sensor placement, as a check that the comparison could go against us. *Met*, by five. That it could and did is what makes the tomography verdict worth something.

The two are consistent with the method's own condition rather than with noise. Proposition 4 sends a learned count to an end of the feasible range; sensor placement's best count is interior for seven of its eight trade-offs, so the count stage is aimed away from the answer there.

An unregistered control: is the gain the centring? The size-aware posterior centres the design columns, drops the intercept and gives the size feature a diffuse prior. Refitting the backbone's stored archives with the prior scale set to $L = 0$, which keeps the first two changes and removes the third, recovers 0.42–0.87 of the size effect against 0.997–1.005 at $L = 50$; in set cover the count it then selects is 81 against a best count of 46. Run as a campaign on five targets per family, $L = 0$ scores 0.529, 0.504, 0.639 and 0.244 against 0.648, 0.594, 1.002 and 0.321, a loss of 18.4, 15.2, 36.2 and 24.0% with every interval clear of zero, and it proposes fewer items in every family. Five targets give wide intervals and settle only that the cost is large. This was run after the fact, in response to the question it answers, and is not one of the predictions above.

The surrogate diagnostic. Registered: the standard GP recovers less than 20% of the size effect in every family. *Failed* narrowly in set cover (median 20.05%, range 18.3–35.2%), the family the ratio in Proposition 2 predicts to recover most. Registered: a size-aware prior recovers more than 90%. Met in four of four, at 96.5–104.4%. Both were fixed before the diagnostic ran. The diagnostic is offline – it refits stored archives and adds no evaluation – and no campaign reported in this paper was run or selected on its outcome.

An amendment made after measuring solver noise. The registered item is the comparison rule, and under it the one-second comparison belongs in the appendix (ratios 2.0 and 1.2 against a threshold of 3). Reproducibility – the within-target standard deviation of an arm across repeats – was not registered; it was measured afterwards, when the spread of the one-second results made the single-run comparison unreadable. We report it in the body in place of the comparison it replaced, and note here that it is an unregistered quantity, so that the substitution is visible.

A defect found after the first results, and what it changed. The routine that sets the solver's wall-clock limit rewrote the name in two of the three modules that hold it, so every fixed-count solve silently kept a hardcoded 1 s cap. Appendix D gives the evidence, the affected arms and what was re-run; 480 runs were repeated with the limit delivered.