

ARTICLE

Adaptive Candidate Restriction for Scalable Random-Forest Bayesian Optimisation: Matched Accuracy at a Fraction of the Acquisition Cost

Irfan Khan¹, Saqib Alam², Ramesh Kumar Ayyasamy^{3,*} and Miao Zhang^{1,*}

¹School of Software, Quanzhou University of Information Engineering, Quanzhou, China

²School of Information Engineering, Nanchang Hangkong University, Nanchang, China

³Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar, Malaysia

*Corresponding Authors: Ramesh Kumar Ayyasamy. Email: rameshkumar@utar.edu.my; Miao Zhang. Email: zm@qzuie.edu.cn

Received: 13 July 2026; Accepted: 11 September 2026; Published: 21 October 2026

ABSTRACT: Random-forest sequential model-based optimisation (RF-SMBO) is a robust, CPU-friendly method for hyperparameter optimisation (HPO), but its conventional implementation scores every unevaluated configuration with the acquisition function at each iteration. On modern search grids containing tens of thousands of configurations—such as the transfer-HPO benchmarks widely used in automated machine learning (AutoML) research—this exhaustive scoring, not the objective evaluations, dominates the per-iteration cost. Practical solvers such as SMAC3 already restrict the candidate set heuristically, but the candidate count is fixed by convention and no existing analysis characterises when restriction is safe or how to size the candidate set automatically. We provide that analysis. We introduce randomised candidate restriction, in which the acquisition function is maximised over a small random subset of candidates rather than the full pool. The subset is sized online by ADAPTIVE-K, a schedule driven by the forest’s own predictive dispersion: it scores few candidates while the surrogate is unreliable and more as the surrogate becomes reliable, and it reverts to exact full scoring whenever the pool is small. The rule is a one-line change to any RF-SMBO implementation, and its three constants are frozen once on a single development space and never retuned. A coverage analysis formalises why restriction is safe. We prove that a random subset of K candidates misses every near-optimal configuration with probability at most $(1 - p)^K$, where p is the fraction of near-optimal configurations in the pool. This bound decays exponentially in K , so a few thousand random candidates contain a near-optimal configuration with high probability and the candidate budget is not an accuracy-sensitive hyperparameter. We verify this prediction by direct measurement: the restricted subset retains a candidate within 5% of the full-pool acquisition maximum on 85.5% of steps. An optimiser’s-curse measurement rules out selection-bias reduction as the operative mechanism. Empirically, across 14 HPO-B search spaces, 157 held-out tasks and 20 seeds, adaptive-K matches full-candidate RF-SMBO within a pre-specified margin of 0.01 normalised regret (median per-task difference zero), while scoring 20–27× fewer candidates per step and accelerating the acquisition step by up to 4.7× on the largest grids. It improves on random search and tree-structured Parzen estimator (TPE) baselines by more than 25% ($p < 10^{-5}$). Two live-training studies, tuning XGBoost over a 22,680-configuration grid and a random forest over a 6048-configuration grid on OpenML classification datasets, confirm these results when every evaluation is a real model fit. Restriction also generalises beyond the random-forest surrogate: fixed- K restriction matches full scoring for TPE and Gaussian-process upper confidence bound (GP-UCB) surrogates. All code, frozen configurations, and analysis scripts needed to reproduce our results are publicly available at <https://doi.org/10.5281/zenodo.22886237>.

KEYWORDS: Hyperparameter optimisation; Bayesian optimisation; automated machine learning; random forest; acquisition function; computational efficiency

1 Introduction

The performance of a machine-learning model depends heavily on its hyperparameters, and finding a good configuration is one of the most computationally demanding steps in deploying such models [1,2]. Sequential model-based optimisation (SMBO) is the standard approach to this search. At each iteration, SMBO fits a surrogate model of the objective to the evaluations collected so far. An acquisition function then scores candidate configurations by balancing predicted quality against uncertainty, and the highest-scoring candidate is evaluated next. A widely used choice of surrogate is the random forest, as in SMAC and its descendants [3]. Random-forest SMBO (RF-SMBO) handles the mixed, conditional and non-smooth search spaces of practical machine-learning pipelines, requires neither gradients nor a GPU, and runs on commodity CPUs. These properties make it a natural choice for automated machine learning (AutoML) at scale.

In principle, the RF-SMBO acquisition step scores *every* unevaluated configuration in the search grid before selecting one. On the small grids of classical benchmarks, scoring the whole pool adds little overhead. Modern transfer-HPO grids, however, routinely contain tens of thousands of configurations [4], and at that scale the scoring itself becomes the dominant per-iteration cost. On the largest space we study, each step scores up to 52,448 configurations with a 64-tree forest, and a single 25-evaluation run costs on the order of 30 million tree predictions, while each objective evaluation is a table lookup. The acquisition step, not the objective, is the bottleneck.

Practical solvers already address this bottleneck heuristically. Trust-region methods such as TuRBO draw local candidate sets for tractability [5], and SMAC3's default acquisition maximiser scores a bounded set of randomly sampled challengers interleaved with local search rather than the whole space [6]. These heuristics restrict the candidate set by convention, with the candidate count fixed or tuned per system. What they leave open are the analytic questions that determine whether such a shortcut is sound and how to size it: how small the candidate set can be before accuracy degrades, whether the answer depends on the size of the pool or on the stage of the search, and whether the set size can be chosen automatically from the surrogate itself, without introducing a new hyperparameter that needs tuning. To our knowledge, no existing analysis answers these questions for RF-SMBO on finite configuration pools.

This paper provides that analysis. Our central finding is that the accuracy of RF-SMBO saturates long before the pool is exhausted, so the same final accuracy is available at a small fraction of the scoring cost. *Randomised candidate restriction* exploits this: it maximises the acquisition over a small random subset of the unevaluated pool instead of the full grid (Fig. 1), sized online by ADAPTIVE-K from the forest's own predictive dispersion. A coverage bound explains why the shortcut is safe, and extensive experiments—both on precomputed benchmarks and on live-trained models—establish that the candidate budget is not an accuracy-sensitive hyperparameter. For practitioners the practical implication is a simple default: on large grids, maximise the acquisition over a few thousand random candidates rather than the whole pool.

We summarise our contributions as follows.

1. **A restriction rule with no per-task tuning.** Randomised candidate restriction maximises the acquisition over a small random subset of the pool, sized online by ADAPTIVE-K from the forest's own predictive dispersion (Section 3). The rule reverts to exact full scoring on small pools, its three constants are frozen once on a single development space, and it is a one-line change to any RF-SMBO implementation.
2. **Matched accuracy at an order of magnitude lower acquisition cost.** Across 14 HPO-B spaces, 157 held-out tasks and 20 seeds, adaptive-K matches full-candidate RF-SMBO within a pre-specified margin of 0.01 normalised regret while scoring 20–27× fewer candidates per step, and improves on random search

and TPE by more than 25% ($p < 10^{-5}$). Two live-training studies (XGBoost and a random forest on OpenML datasets) confirm the result when every evaluation is a real model fit (Section 5).

3. **A coverage analysis of when restriction is safe.** A coverage bound, verified by direct measurement (the subset retains a candidate within 5% of the full-pool acquisition maximum on 85.5% of steps), explains why a small random subset loses essentially no acquisition value. Final regret is flat in the candidate budget from roughly one hundred candidates to the full pool, so the subset size is not a sensitive hyperparameter, and a direct measurement rules out selection-bias reduction (the optimiser's curse) as the operative mechanism.

To support reproducibility and independent verification, the complete implementation, the frozen configurations, and every analysis script used in this paper are publicly available at <https://doi.org/10.5281/zenodo.22886237>.

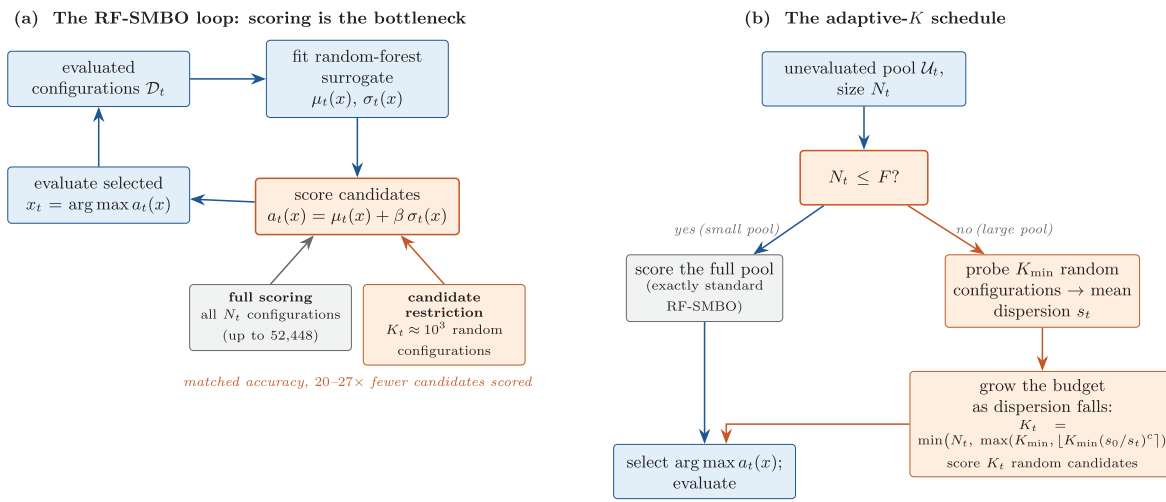


Figure 1: Overview of randomised candidate restriction in RF-SMBO. (a) The RF-SMBO loop fits a 64-tree random-forest surrogate to the evaluations made so far and scores candidate configurations with an upper confidence bound (UCB) acquisition function. Conventional RF-SMBO scores the entire unevaluated pool at every step—up to 52,448 configurations on the largest HPO-B grid—selecting the global acquisition maximiser. Candidate restriction instead draws a small random subset $\mathcal{C}_t$ and selects its maximiser, achieving matched final accuracy at a fraction of the scoring cost (Section 5.1). (b) The ADAPTIVE-K schedule determines the subset size K_t from the forest's own predictive dispersion s_t (Eq. (3)): when the pool is small ($N_t \leq F = 2500$), all configurations are scored in full, so the method coincides exactly with standard RF-SMBO; on large pools the candidate budget starts at $K_{\min} = 1000$ and grows as s_t falls, allocating more candidates as the surrogate becomes more reliable. The three constants ($F, K_{\min}, c$) are frozen once on a single development space and never retuned.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 presents the restriction rule, the ADAPTIVE-K schedule and the coverage analysis. Section 4 describes the experimental setup, and Section 5 reports the results. Section 6 discusses the limitations of the study, and Section 7 concludes.

2 Background and Related Work

2.1 Sequential Model-Based Hyperparameter Optimisation

Sequential model-based optimisation (SMBO) is a broad family of methods that iteratively fit a surrogate model to past evaluations and use it to guide the search; this family includes trust-region

algorithms, surrogate-assisted evolutionary optimisation, and—the variant we study—acquisition-function-based methods that score candidate configurations by balancing predicted quality against uncertainty [7,8]. When the surrogate is a random forest and the acquisition is maximised explicitly, as in SMAC and its descendants, this acquisition-function-based SMBO handles the conditional, mixed and non-smooth search spaces typical of machine-learning pipelines without gradients or a GPU [3,6]. This robustness, together with its low overhead, makes random-forest SMBO (RF-SMBO) a common choice for practical AutoML, where it underpins combined algorithm selection and hyperparameter optimisation [9–12]. The algorithm-selection half of that problem can also be addressed without search, by meta-learning a recommendation from dataset meta-features [13,14]; such methods choose *which* learner to tune, whereas we reduce the cost of tuning a learner already chosen. Our work leaves this optimiser unchanged and revisits only its acquisition step.

2.2 Acquisition Maximisation and Scalable Bayesian Optimisation

Bayesian optimisation balances exploration and exploitation through an acquisition function optimised at each step [15], with Gaussian-process upper-confidence-bound methods enjoying well-known regret guarantees [16]. Practical solvers extend this template with robust, scalable implementations [17–19]; surrogate models play the same role in expensive black-box and evolutionary optimisation more broadly [8]. The acquisition step itself is actively refined, for example by making priors robust to misspecification [20]. On the finite, tabular grids of transfer-HPO benchmarks, optimising the acquisition reduces to scoring candidate configurations and selecting the maximiser; this is the operation whose cost and statistical behaviour we study. Random search remains a strong baseline in this setting [21], a fact our coverage analysis explains: a modest random subset already contains near-optimal configurations with high probability. Restricting the acquisition to a sampled candidate set is itself established practice in scalable Bayesian optimisation, most visibly in trust-region methods that draw local candidates for tractability [5]. Production RF-SMBO implementations also approximate the maximisation: SMAC3’s default acquisition maximiser scores a bounded set of randomly sampled challengers interleaved with local search rather than the whole space [6]. These mechanisms restrict as a computational heuristic, with the candidate count fixed by convention; what has been missing is an analysis of when restriction is safe and how small the candidate set may be.

The idea of adapting optimiser parameters to the state of the search is well established in Bayesian optimisation. Adaptive acquisition strategies include adapting kernel hyperparameters during the run [22], portfolio methods that select among multiple acquisition functions per step [23], and cooling schedules that anneal the acquisition toward exploitation as confidence grows [7]. Our ADAPTIVE-K schedule follows this tradition: it adapts the *candidate-set size*, a computational parameter of the acquisition step, to the surrogate’s reliability as measured by its own predictive dispersion. Sections 3 and 5 supply the formal analysis and the schedule that exploits it.

2.3 Multi-Fidelity and Bandit-Based HPO

An orthogonal line of work reduces cost by spending less per evaluation rather than per acquisition step: Hyperband allocates budget through successive halving [24], and BOHB combines this with model-based proposals [25]; PriorBand further incorporates expert priors into multi-fidelity search [26]; and systems such as Auto-PyTorch integrate multi-fidelity metalearning end to end [27]. These methods address a different bottleneck (evaluation cost) from ours (acquisition cost) and are complementary; candidate restriction could be applied within the model-based component of such a hybrid.

2.4 Meta-Learning, Transfer HPO and Benchmarks

Because related tasks share structure, HPO can be warm-started from prior experience, initialising the search from configurations that performed well on similar datasets [28,29]. Transfer surrogates carry this further by conditioning the model itself on source tasks [30]. Reproducible HPO research relies on tabular benchmarks of precomputed evaluations; we use HPO-B [4], a transfer-HPO suite of thousands of tasks, as our evaluation suite. Our method itself is cold-start: it is initialised from a small random design and uses no meta-knowledge, so the contribution concerns the acquisition step rather than transfer across tasks. Within automated machine learning the emphasis is on methods that practitioners can adopt and audit [9]; candidate restriction is designed to meet that requirement.

2.5 The Optimiser's Curse and Selection Bias

Maximising a noisy estimate over many alternatives tends to select options whose estimated value is inflated by error rather than merit, a phenomenon known as the optimiser's curse and first characterised in decision analysis [31]. The severity of the effect depends on the scale and dependence structure of the estimation errors across the alternatives; random-forest predictions are averages of trees fit on shared data [32], so their errors are small and strongly correlated across configurations. We measure the curse directly on random-forest surrogates in Section 5.6.

3 Candidate Restriction and the Adaptive-K Schedule

3.1 Setup and Notation

We consider sequential model-based optimisation of a black-box objective $f : \mathcal{X} \rightarrow \mathbb{R}$ over a finite configuration pool $\mathcal{X} = \{x_1, \dots, x_M\}$, the standard tabular setting of transfer-HPO benchmarks. Values are normalised so that $f \in [0, 1]$ with $f(x^*) = 1$ at the pool optimum x^* , and we report simple regret $r = 1 - \max_{i \leq t} f(x_i)$ after a budget of t evaluations.

At step t a random-forest (RF) surrogate is fit on the observed set $\mathcal{D}_t$, yielding a predictive mean $\mu_t(x)$ and a tree-dispersion $\sigma_t(x)$, the standard deviation of the per-tree predictions. Acquisition uses the upper confidence bound $a_t(x) = \mu_t(x) + \beta \sigma_t(x)$ with fixed β . Writing $\mathcal{U}_t \subseteq \mathcal{X}$ for the unevaluated configurations and $N_t = |\mathcal{U}_t|$, *full-candidate* RF-SMBO selects

$$x_t^{\text{full}} = \arg \max_{x \in \mathcal{U}_t} a_t(x), \quad (1)$$

i.e., it scores every unevaluated configuration and selects the global argmax of the acquisition. *Randomised candidate restriction* instead draws a uniform random subset $\mathcal{C}_t \subseteq \mathcal{U}_t$ of size K and selects

$$x_t = \arg \max_{x \in \mathcal{C}_t} a_t(x), \quad |\mathcal{C}_t| = K \leq N_t. \quad (2)$$

Everything else (the RF with 64 trees, $\beta = 1.96$, and a five-step random warm-start) is held identical to (1), so the *only* degree of freedom under study is the size of the candidate set.

3.2 The Rule and the Adaptive Schedule

The simplest instantiation fixes the subset size: RF-CAND-2K sets $K = 2000$ at every restricted step, a value chosen once and frozen before evaluation. Because accuracy is flat in K across a wide plateau (Section 5.4), this fixed rule is already a complete, deployable method.

The adaptive form, ADAPTIVE-K, refines it with a design principle: *score few candidates while the surrogate is unreliable, and more as it becomes reliable*. The surrogate's reliability is estimated directly from the forest itself, with no external information: let s_t be the mean tree-dispersion over a probe of $K_{\min}$ random unevaluated configurations, and s_0 its value at the first restricted step of the run. The schedule is

$$K_t = \min\left(N_t, \max\left(K_{\min}, \lfloor K_{\min}(s_0/s_t)^c \rfloor\right)\right), \quad (3)$$

with $K_t = N_t$ whenever $N_t \leq F$, where F is a full-scoring floor. Early in the run $s_t \approx s_0$ and $K_t = K_{\min}$; as the surrogate becomes more reliable, s_t falls and K_t grows, capped at N_t . (Vanishing dispersion is handled by the cap: as $s_t \rightarrow 0$, K_t reaches N_t and the schedule reduces to full scoring.) Whenever the pool is small ($N_t \leq F$) the rule scores it in full, so it is identical to full-candidate RF-SMBO there; Corollary 1 makes this small-pool identity exact. The three constants ($F, K_{\min}, c$) are the method's only hyperparameters; they are frozen once on a single development space and never retuned, exactly as $K = 2000$ was frozen for RF-CAND-2K. Algorithm 1 gives the full procedure.

Algorithm 1: Adaptive-K RF-SMBO (one task)

Require: finite pool $\mathcal{X}$; evaluation budget B ; frozen constants: full-scoring floor F , candidate floor $K_{\min}$, growth exponent c , UCB weight β

Ensure: best configuration found, $\hat{x}$

- 1: $\mathcal{D} \leftarrow$ evaluate 5 uniform draws from $\mathcal{X}$ $\triangleright$ warm-start
 - 2: $s_0 \leftarrow \text{NULL}$ $\triangleright$ reference dispersion
 - 3: **for** $t = 6, \dots, B$ **do**
 - 4: fit RF on $\mathcal{D}$, giving $\mu_t(\cdot), \sigma_t(\cdot)$
 - 5: $\mathcal{U}_t \leftarrow \mathcal{X} \setminus \mathcal{D}; N_t \leftarrow |\mathcal{U}_t|$
 - 6: **if** $N_t \leq F$ **then** $\triangleright$ small pool: exact full scoring (Cor. 1)
 - 7: $\mathcal{C}_t \leftarrow \mathcal{U}_t$
 - 8: **else** $\triangleright$ large pool: restricted scoring
 - 9: draw probe $P \subseteq \mathcal{U}_t, |P| = K_{\min}$, uniformly
 - 10: $s_t \leftarrow \frac{1}{K_{\min}} \sum_{x \in P} \sigma_t(x)$ $\triangleright$ surrogate reliability
 - 11: **if** $s_0 = \text{NULL}$ **then** $s_0 \leftarrow s_t$ $\triangleright$ calibrate once
 - 12: $K_t \leftarrow \min\left(N_t, \max\left(K_{\min}, \lfloor K_{\min}(s_0/s_t)^c \rfloor\right)\right)$
 - 13: $\mathcal{C}_t \leftarrow P \cup (K_t - K_{\min})$ further uniform draws from $\mathcal{U}_t \setminus P$
 - 14: **end if**
 - 15: $x_t \leftarrow \arg \max_{x \in \mathcal{C}_t} \mu_t(x) + \beta \sigma_t(x)$ $\triangleright$ UCB over $\mathcal{C}_t$ only
 - 16: evaluate $f(x_t)$; $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_t, f(x_t))\}$
 - 17: **end for**
 - 18: **return** $\hat{x} \leftarrow \arg \max_{x \in \mathcal{D}} f(x)$
-

The schedule's role is operational rather than accuracy-critical. Its full-scoring floor guarantees exact small-pool behaviour, and its confidence-driven contraction scores fewer candidates than a fixed K (a mean of 1210 vs. 2000) at unchanged accuracy. We therefore deploy ADAPTIVE-K as the method throughout and use the fixed rule as its ablation.

3.3 Why Restriction is Safe: A Coverage Analysis

Two competing effects govern the choice of candidate-set size. More candidates improve *coverage*, the probability that the set contains a high-acquisition configuration, but they also make the maximisation more susceptible to surrogate error. We bound both effects for the acquisition that the rule optimises.

Decompose the acquisition into merit and estimation error, $a_t(x) = g_t(x) + \varepsilon_t(x)$, where g_t is the noiseless acquisition a perfectly estimated surrogate would report and ε_t is the zero-mean estimation error. Write $x_t^\# = \arg \max_{x \in \mathcal{U}_t} g_t(x)$ for the highest-merit configuration in the full unevaluated pool, the target our bound compares against, while $x_t = \arg \max_{x \in \mathcal{C}_t} a_t(x)$ is what the restricted rule selects from the subset.

Lemma 1 (Coverage): *Let a fraction $p_t \in (0, 1]$ of the N_t configurations in $\mathcal{U}_t$ be η -good in acquisition, i.e., $g_t(x) \geq g_t(x_t^\#) - \eta$, and write $m = p_t N_t$. A uniform K -subset $\mathcal{C}_t$ drawn without replacement misses every η -good configuration with probability*

$$\Pr[\text{no } \eta\text{-good configuration in } \mathcal{C}_t] = \binom{N_t - m}{K} / \binom{N_t}{K} \leq (1 - p_t)^K \leq e^{-p_t K}, \quad (4)$$

and the expected merit gap to $x_t^\#$ therefore satisfies

$$\mathbb{E}\left[g_t(x_t^\#) - \max_{x \in \mathcal{C}_t} g_t(x)\right] \leq \eta + \Delta_t e^{-p_t K}, \quad (5)$$

where $\Delta_t = \max_x g_t(x) - \min_x g_t(x)$ is the acquisition range.

Proof: Sampling without replacement, the exact miss probability is the hypergeometric product

$$\binom{N_t - m}{K} / \binom{N_t}{K} = \prod_{i=0}^{K-1} \frac{N_t - m - i}{N_t - i},$$

in which each factor is at most $\frac{N_t - m}{N_t} = 1 - p_t$; this gives the bound $(1 - p_t)^K$ in (4), and the inequality $1 - u \leq e^{-u}$ gives the exponential form. If the subset contains an η -good configuration the gap is at most η ; otherwise it is at most Δ_t . Combining the two cases gives (5). $\square$

The miss probability in (4) decays exponentially in K : once K is a few multiples of $1/p_t$, the subset contains an η -good configuration with high probability, and scoring further candidates adds cost without adding coverage. Eq. (5) translates this into acquisition value: the loss from restriction is at most the tolerance η plus a term that vanishes exponentially with the candidate budget.

The second effect is *over-selection*: maximising a noisy estimate over a large set can select points whose acquisition is inflated by estimation error rather than merit, the optimiser's curse [31]. Under a standard tail condition its size is bounded as follows.

Assumption 1 (Sub-Gaussian surrogate error): *At step t the errors $\{\varepsilon_t(x)\}_{x \in \mathcal{U}_t}$ are zero-mean and σ_t -sub-Gaussian, with σ_t non-increasing in the number of observations $|\mathcal{D}_t|$.*

Assumption 1 is mild: RF predictions are averages of T de-correlated trees, so their error concentrates and shrinks as data accumulate. The tree-dispersion $\sigma_t(x)$ serves as an online proxy for this error scale, not as a calibrated estimate of it.

Lemma 2 (Over-selection bias): *Under Assumption 1, the surrogate error of the point x_t selected in (2) satisfies*

$$\mathbb{E}[\varepsilon_t(x_t)] \leq \mathbb{E}\left[\max_{x \in \mathcal{C}_t} \varepsilon_t(x)\right] \leq \sigma_t \sqrt{2 \ln K}, \quad (6)$$

and the two-sided deviation satisfies

$$\mathbb{E} \left[\max_{x \in \mathcal{C}_t} |\varepsilon_t(x)| \right] \leq \sigma_t \sqrt{2 \ln(2K)}. \quad (7)$$

Proof: The inequality $\varepsilon_t(x_t) \leq \max_{x \in \mathcal{C}_t} \varepsilon_t(x)$ holds pointwise. For $K\sigma_t$ -sub-Gaussian variables the maximal inequality [33] gives $\mathbb{E}[\max_{i \leq K} \varepsilon_i] \leq \sigma_t \sqrt{2 \ln K}$ (Jensen's inequality applied to the moment-generating function, optimised over the Chernoff parameter), which yields (6). The two-sided bound (7) applies the same argument to the $2K$ variables $\{\pm \varepsilon_t(x)\}$. $\square$

The bound (6) grows with the number of candidates scored: the more noisy estimates the maximisation is taken over, the larger the expected inflation of the selected score. Combining the two effects yields the main bound.

Theorem 1 (Acquisition-selection regret): *Under Assumption 1, the expected one-step acquisition-selection regret of (2), the merit gap between the highest-merit pool configuration $x_t^\#$ and the selected x_t , is bounded by*

$$\mathbb{E}[g_t(x_t^\#) - g_t(x_t)] \leq \underbrace{\Delta_t e^{-p_t K}}_{\text{coverage, } \downarrow K} + \underbrace{2\sigma_t \sqrt{2 \ln(2K)}}_{\text{over-selection, } \uparrow K} \quad (8)$$

up to the additive constant η ; call the right-hand side $B_t(K)$. B_t sums a term decreasing in K and one increasing in K ; wherever it admits an interior minimiser K_t^* , that minimiser increases as the surrogate error σ_t decreases.

Proof: Let $x_t^C = \arg \max_{x \in \mathcal{C}_t} g_t(x)$ be the highest-merit point in the subset, and split the selection regret as

$$g_t(x_t^\#) - g_t(x_t) = [g_t(x_t^\#) - g_t(x_t^C)] + [g_t(x_t^C) - g_t(x_t)].$$

The first bracket is the coverage gap of Lemma 1, at most $\eta + \Delta_t e^{-p_t K}$ in expectation by (5). For the second, optimality of x_t for $a_t = g_t + \varepsilon_t$ over $\mathcal{C}_t$ gives $g_t(x_t) + \varepsilon_t(x_t) \geq g_t(x_t^C) + \varepsilon_t(x_t^C)$, hence

$$g_t(x_t^C) - g_t(x_t) \leq \varepsilon_t(x_t) - \varepsilon_t(x_t^C) \leq 2 \max_{x \in \mathcal{C}_t} |\varepsilon_t(x)|,$$

whose expectation is at most $2\sigma_t \sqrt{2 \ln(2K)}$ by (7). Summing the two brackets gives (8). For the monotonicity claim, the coverage term is independent of σ_t while the over-selection term is increasing in σ_t at every K ; hence any interior minimiser increases as σ_t falls. $\square$

Remark 1 (Scope of the bound): *Theorem 1 bounds, at a single step, the acquisition value that the restricted rule loses relative to full scoring; it is not a cumulative-regret guarantee and does not directly imply that adaptive- K matches full RF-SMBO in terms of final normalised regret over an entire optimisation run. Converting the one-step bound into an objective-regret guarantee would additionally require the surrogate to be calibrated, so that acquisition value translates faithfully into objective value—a condition that the UCB satisfies only approximately.*

The role of the theory is therefore motivational and explanatory rather than predictive: the coverage/over-selection trade-off formalised in Eq. (8) motivates the design of the adaptive schedule (the optimal candidate budget grows as σ_t falls) and explains why accuracy is flat in the candidate budget once a coverage threshold is reached. The claim that adaptive- K matches full scoring over entire optimisation runs rests on the empirical evidence of the held-out and live-training evaluations in Section 5, not on the bound alone.

The dependence of $B_t(K)$ on σ_t recovers the design principle behind ADAPTIVE- K : the optimal candidate budget grows as the surrogate's error falls. Two further consequences matter for deployment.

Corollary 1 (Small-pool exact identity): *Because $C_t \subseteq \mathcal{U}_t$, whenever $K \geq N_t$ the restricted rule (2) scores the entire unevaluated pool and coincides exactly with full-candidate RF-SMBO (1). Consequently the schedule $K_t = \min(N_t, \widehat{K}_t^*)$ equals RF-SMBO on every step with $N_t \leq \widehat{K}_t^*$ and can differ from it only when $N_t > \widehat{K}_t^*$.*

Proposition 1 (Acquisition-cost reduction): *Scoring the candidate set costs $\Theta(K T)$ tree evaluations per step vs. $\Theta(N_t T)$ for (1). Total acquisition cost is therefore reduced by the factor $\bar{N}/\bar{K}$, the ratio of mean pool size to mean candidate-set size, independent of the objective.*

Together these results determine how the method behaves in deployment. On small pools the schedule scores the whole pool and coincides with full-candidate RF-SMBO; on large pools it restricts, and whether anything is lost there is settled empirically in Section 5. Proposition 1 quantifies the payoff: a deterministic reduction in acquisition cost that depends only on pool and candidate-set sizes, not on any statistical effect.

4 Experimental Setup

4.1 Benchmark and Tasks

We evaluate on HPO-B v3 [4], a standard transfer-HPO benchmark of precomputed evaluations. Of its 16 search spaces we use the 14 that pass a saturation screen computed from the precomputed data before any of the compared methods was run: a space is excluded if the benchmark’s informed default configuration, or an off-the-shelf transfer baseline, already attains mean normalised regret below 0.05 on its tasks. This removes the two-dimensional glmnet spaces 5860 and 5970; the remaining 14 span five learner families (XGBoost, Ranger, SVM, Rpart, Glmnet), dimensionalities 2–17, and configuration-pool sizes 254–52,448. All 14 spaces are finite, enumerable configuration pools, the regime in which exhaustive acquisition scoring is the standard default. Each space provides held-out tasks: the 77 official meta-test tasks form the primary split, and the 80 meta-validation tasks, never consulted for any tuning, serve as an independent robustness check; we also report the pooled 157-task set. Generalisation beyond precomputed benchmarks is tested with the two live-training studies described below.

4.2 Methods

We compare: RANDOM search; TPE, the tree-structured Parzen estimator (TPE), a kernel-density acquisition model [34]; RF-SMBO, full-candidate random-forest SMBO scoring every unevaluated configuration (Eq. (1)); RF-CAND-2K, the frozen fixed- K restriction with $K = 2000$; and ADAPTIVE-K, the uncertainty-adaptive schedule of Algorithm 1. All surrogate methods share the same RF (64 trees), UCB acquisition ($\beta = 1.96$), and 5-step random warm-start, so the only variable across RF-SMBO/RF-CAND-2K/ADAPTIVE-K is the candidate-set size.

4.3 Frozen Hyperparameters and Rationale

The three adaptive-K constants were selected once on a single development space (HPO-B 6767, an XGBoost space with 44,258 configurations and 17 dimensions) and then frozen; no constant is retuned on any evaluation task. Their values are: full-scoring floor $F = 2500$ (set above the largest small-pool space so that restriction is never active where the full pool is already cheap to score); early candidate floor $K_{\min} = 1000$ (chosen to lie well inside the saturation plateau established in Section 5.4, where regret is flat from roughly 100 to over 40,000 candidates); and growth exponent $c = 0.75$ (selected from a small grid of five $(F, K_{\min}, c)$ triples on 6767 alone, choosing the triple with lowest regret). The fixed variant’s $K = 2000$ was likewise frozen before evaluation; its position inside the plateau ensures that doubling or halving it would not change accuracy (Section 5.4).

The non-inferiority margin $\delta = 0.01$ normalised regret was fixed before analysis as a practically negligible gap on the $[0, 1]$ regret scale; it lies within seed-level variability and corresponds to less than 1 percentage point of objective performance. The 20 random seeds follow the HPO-B protocol and are sufficient to reduce the standard error of the mean to below δ on every task population. The 14 search spaces and 157 held-out tasks are the full HPO-B evaluation suite after the saturation screen described above.

4.4 Protocol and Metrics

Each method is run for 20 random seeds at budgets $B \in \{5, 10, 25\}$, with $B = 25$ the primary budget; we report normalised simple regret in $[0, 1]$ (oracle = 0). Acquisition cost is measured as the number of configurations scored by the surrogate per step: RF-SMBO scores all N_t unevaluated configurations, ADAPTIVE-K scores $K_t \leq N_t$, and we report the mean per-step ratio $\tilde{N}/\tilde{K}$ (Proposition 1). All method comparisons are *paired* at the task level (regret averaged over seeds first), tested with the Wilcoxon signed-rank test, the recommended protocol for comparing methods across datasets [35]. Because a few large-pool tasks dominate the arithmetic mean, we report primarily the median difference and win/loss/tie counts at a tie tolerance of 0.005 normalised regret, and report both two-sided and one-sided p -values. The primary comparison, ADAPTIVE-K against full-candidate RF-SMBO, is tested as formal non-inferiority against a pre-specified margin. Writing $d = R_{\text{adaptive}} - R_{\text{full}}$ for the per-task regret difference (negative favouring adaptive-K), we fixed the margin $\delta = 0.01$ normalised regret before analysis; on the $[0, 1]$ regret scale a gap this small is practically negligible and lies within seed-level variability. Adaptive-K is declared non-inferior on a population of tasks if the one-sided 95% upper confidence bound on $\mathbb{E}[d]$ (task bootstrap, 20,000 resamples) falls below δ .

4.5 Live-Training Studies

Two studies test whether the findings hold beyond precomputed evaluations; in both, every configuration selected by a method is actually trained. The first tunes XGBoost [36] over a 22,680-configuration grid (seven hyperparameters; pool well above F) with 3-fold cross-validation (AUC) on 14 OpenML classification datasets [37], 10 seeds. The second tunes a different learner family, a random forest, over an independently constructed 6048-configuration grid (six hyperparameters) on 10 OpenML datasets, 10 seeds; four high-dimensional datasets were excluded beforehand on training-runtime grounds. In both studies we compare the best cross-validated AUC at budgets $B \in \{5, 10, 25\}$, paired at the *dataset* level (seeds averaged within each dataset), and record the measured candidate-scoring ratio.

5 Results

This section presents the results in four parts. Adaptive-K matches full-candidate RF-SMBO at an order of magnitude lower acquisition cost and clearly outperforms random search and TPE. We then show that the candidate budget is not a sensitive hyperparameter and identify the mechanism, coverage rather than selection-bias reduction. Finally, we examine the tasks on which restriction loses and its generalisation to live training and to other surrogates.

5.1 Adaptive-K Matches Exhaustive Scoring

The central result is that restriction sacrifices no measurable accuracy. We apply the non-inferiority protocol of Section 4 (Table 1, Fig. 2; per-space values in Appendix A). On the pooled 157-task set the mean per-task difference is -0.002 with a one-sided 95% upper bound of $+0.004$; on the primary meta-test split -0.010 (upper $+0.002$); and on the *active* large-pool subset where restriction operates ($N_t > F$, 100 tasks) -0.003 (upper $+0.006$); each is below the margin $\delta = 0.01$. Removing the development space 6767

entirely leaves the result unchanged (upper bound +0.005). The median per-task difference is zero on every population. On the meta-validation split taken alone, the upper bound is +0.011, so non-inferiority is not established on that split at $\delta = 0.01$; the excess is driven by a few large-pool tasks, and the pooled and active populations, which contain those tasks, remain within the margin.

Table 1: Non-inferiority of adaptive-K to full-candidate RF-SMBO at the pre-specified margin $\delta = 0.01$ ($B = 25$, 20 seeds; paired at the task level, seeds averaged first). $d = R_{\text{adaptive}} - R_{\text{full}}$, negative favouring adaptive-K; the median d is 0.000 on every population. W/L/T at a 0.005 tie tolerance; “95% upper” is the one-sided upper confidence bound on $\mathbb{E}[d]$ (task bootstrap).

Population	n	Mean d	95% Upper	W/L/T	NI ($\delta = 0.01$)
Meta-test	77	−0.010	+0.002	17/10/50	Yes
Meta-validation	80	+0.005	+0.011	4/10/66	Not met
Pooled	157	−0.002	+0.004	21/20/116	Yes
Active $N_t > F$	100	−0.003	+0.006	21/20/59	Yes
Pooled, excl. 6767	144	−0.002	+0.005	17/18/109	Yes

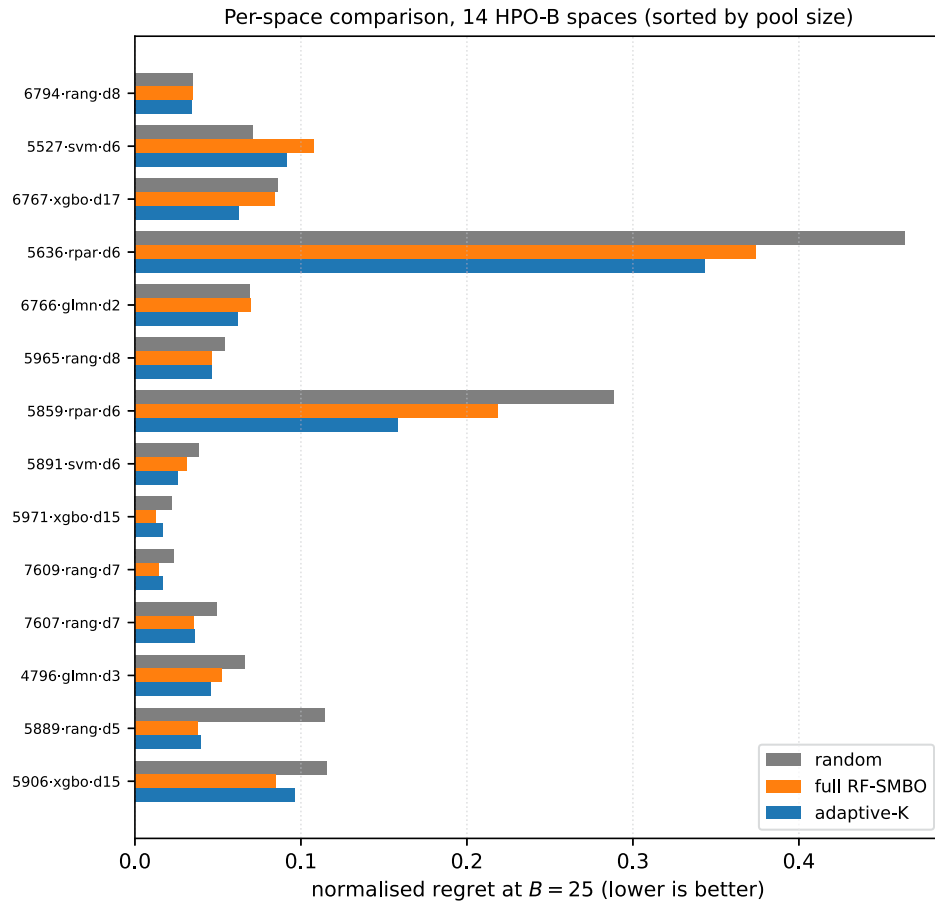


Figure 2: Per-space normalised regret at evaluation budget $B = 25$ across all 14 HPO-B search spaces, sorted by configuration-pool size N (leftmost bars: largest pools). Each bar shows the mean regret over the space’s held-out tasks and 20 seeds; lower is better. Adaptive-K (light blue) matches full-candidate RF-SMBO (dark blue) space by space, with both substantially outperforming random search (grey) and TPE (orange) on every space.

The result is unchanged at the level of the 14 search spaces, the coarser unit of generalisation: per-space mean regret (Table A1) favours adaptive-K in eight spaces and full scoring in five, with one tie, and no space-level difference favouring full scoring exceeds 0.011 normalised regret.

Two features of Table 1 matter for deployment. First, small pools are exact ties by construction (Corollary 1): 66 of 80 validation and 50 of 77 test tasks are exact ties, and the small-pool residual is at most $|\Delta| = 0.0022$ normalised regret, arising from tie-breaking among configurations with equal acquisition. Second, individual large-pool tasks can fall on either side of zero (21 wins, 20 losses at the 0.005 tolerance on the pooled set); Section 5.7 examines the losses.

5.2 An Order of Magnitude Less Acquisition Work

The gain from restriction is computational. We quantify the acquisition cost in four ways, from exact to empirical, separating the *workload* reduction from the *wall-clock* reduction. First, *complexity*: full-candidate RF-SMBO performs $\Theta(N_t T)$ tree evaluations per step to score the unevaluated pool of size N_t with a T -tree forest, whereas adaptive-K performs $\Theta(K_t T)$ with $K_t \leq N_t$ (Proposition 1). Second, *measured scoring counts* (Table 2): full scoring evaluates the entire unevaluated pool (up to $\sim 52,000$ configurations), adaptive-K between $K_{\min} = 1000$ and a schedule-determined cap; the mean per-step ratio is $\bar{N}/\bar{K} \approx 20\times$ overall (24,248 vs. 1210 configurations scored per step) and $27\times$ on the large-pool spaces, equivalently about 31M vs. 1.5M tree predictions per optimisation run. This saving is deterministic: it depends only on pool and candidate-set sizes, not on the objective. Third, the ratio is a *function of pool size* (Fig. 3): $1\times$ on small pools (small-pool identity, by construction), rising to $\approx 27\times$ on the large-pool spaces on average and up to $\approx 45\times$ on the largest single spaces, which is exactly the $O(N) \rightarrow O(K)$ dependence the complexity predicts. Fourth, *measured wall-clock* (Table 3): scoring time falls by up to $16\times$ on the largest grids; because the random-forest *fit* is a fixed per-step cost that restriction does not affect, the per-step wall-clock speedup is smaller, about $1\times$ on small pools rising to $\approx 4.7\times$ at 52,000 configurations (2.4 to 0.52 s per run). On tabular benchmarks each objective evaluation is a table lookup, so the per-step figure is also the end-to-end acquisition speedup. We reserve the order-of-magnitude figure for the candidate-scoring workload and report wall-clock separately.

Table 2: Acquisition cost: mean configurations scored per step.

Setting	rf-smbo ($\bar{N}$)	Adaptive-K ($\bar{K}$)	Ratio
HPO-B (overall)	24,248	1210	20.0×
HPO-B (large pools)	37,364	1371	27.2×
Live-training grid (22,680)	22,666	1145	19.8×

The per-run figure understates the effect, because the acquisition cost is paid again on every step, seed, restart and task of a tuning campaign, and it is exactly in large campaigns that RF-SMBO is deployed. The scale of our own evaluation illustrates this: across 157 held-out tasks at 20 seeds (3140 optimisation runs), full-candidate scoring performs on the order of 9.7×10^{10} tree-level acquisition predictions, adaptive-K about 4.9×10^9 , which is more than 90 billion fewer surrogate-tree evaluations at matched accuracy. The saving matters most where the acquisition step is the bottleneck (benchmark-scale AutoML over many datasets, portfolio and repeated retuning under data drift, scheduled HPO pipelines) and least where a single objective evaluation is itself expensive. The reduction keeps RF-SMBO practical on grids of tens of thousands of configurations on commodity CPUs, with no GPU and no change to the objective.

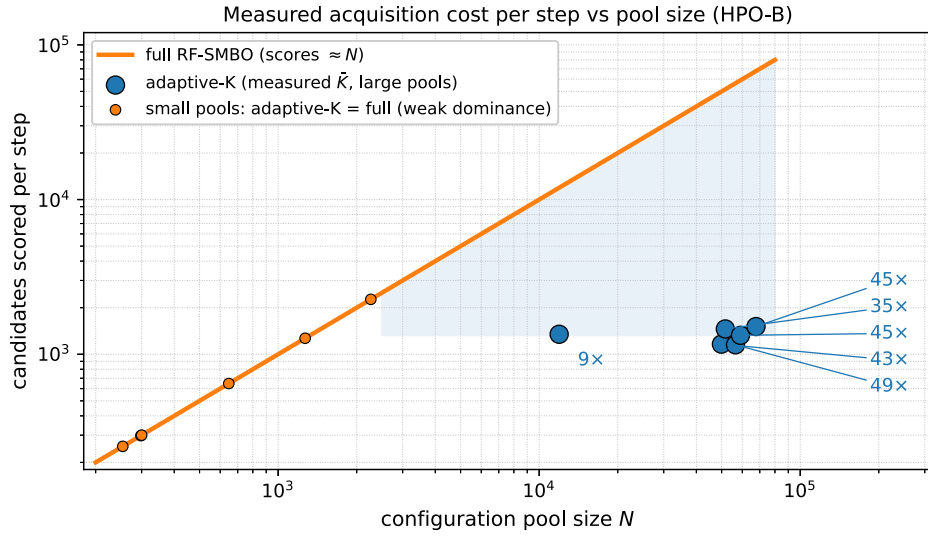


Figure 3: Measured acquisition cost per step against pool size (HPO-B). Full-candidate RF-SMBO scores the whole pool ($\approx N$); adaptive-K's measured mean candidate count $\hat{K}$ stays near 1300 regardless of pool size, so on small pools it coincides with full scoring and on large pools the per-space cost ratio grows from $\approx 9\times$ (11k configurations) to $\approx 45\times$ (52k). The shaded gap is the saving.

Table 3: Measured acquisition wall-clock (64-tree surrogate, $K = 2000$). The scoring-only speedup tracks N/K ; the per-step speedup is smaller because the random-forest fit (≈ 19 ms) is a fixed cost restriction does not reduce.

Pool N	Scoring Speedup	Per-Step (End-To-End) Speedup
300	1.0 $\times$	1.0 $\times$
5000	1.9 $\times$	1.2 $\times$
11,000	3.7 $\times$	1.6 $\times$
25,000	7.8 $\times$	2.6 $\times$
44,000	13.0 $\times$	3.7 $\times$
52,000	16.4 $\times$	4.7 $\times$

5.3 Comparison with Random Search and TPE

The advantage over random search and TPE is clear. On the meta-test split adaptive-K improves mean regret by 26.7% (from 0.106 to 0.077) over random search and 28.7% (from 0.109) over TPE, both at $p < 10^{-5}$ (paired Wilcoxon over the held-out tasks). Restriction therefore matches or exceeds the accuracy of the standard baselines while sharply reducing the cost of the surrogate step.

Fig. 4 shows anytime performance: adaptive-K tracks full RF-SMBO at every budget, and both achieve substantially lower regret than random search and TPE throughout the run.

5.4 The Candidate Budget Requires No Tuning

To determine how many candidates the surrogate actually needs, we sweep the candidate-set size from $K = 1$ to full scoring and measure final regret, normalised per space by that space's full-scoring regret (Fig. 5). At $K = 1$ selection is effectively random and regret is 2.1 $\times$ that of full scoring; as K grows, the surrogate's advantage appears almost immediately, with 87% of its entire benefit over random selection recovered by

$K = 20$ and 95% by $K = 100$. Beyond roughly a hundred candidates the curve is flat: from $K \approx 250$ to $K = 44,000$, more than two orders of magnitude, final regret does not measurably change.

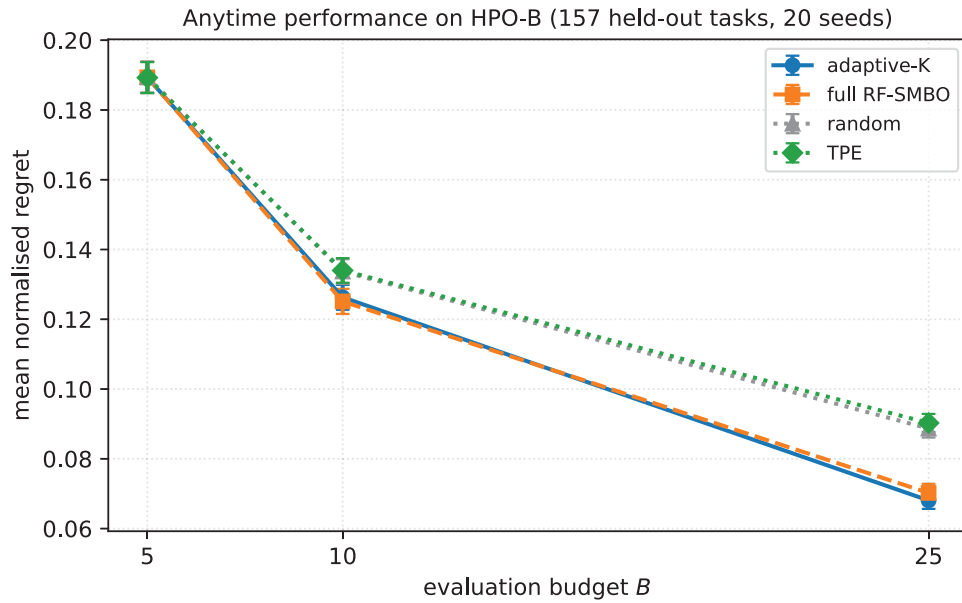


Figure 4: Anytime performance on HPO-B (157 held-out tasks, 20 seeds): mean normalised regret against evaluation budget. Adaptive-K is indistinguishable from full-candidate RF-SMBO at every budget, and both improve substantially on random search and TPE. Error bars are standard errors over tasks.

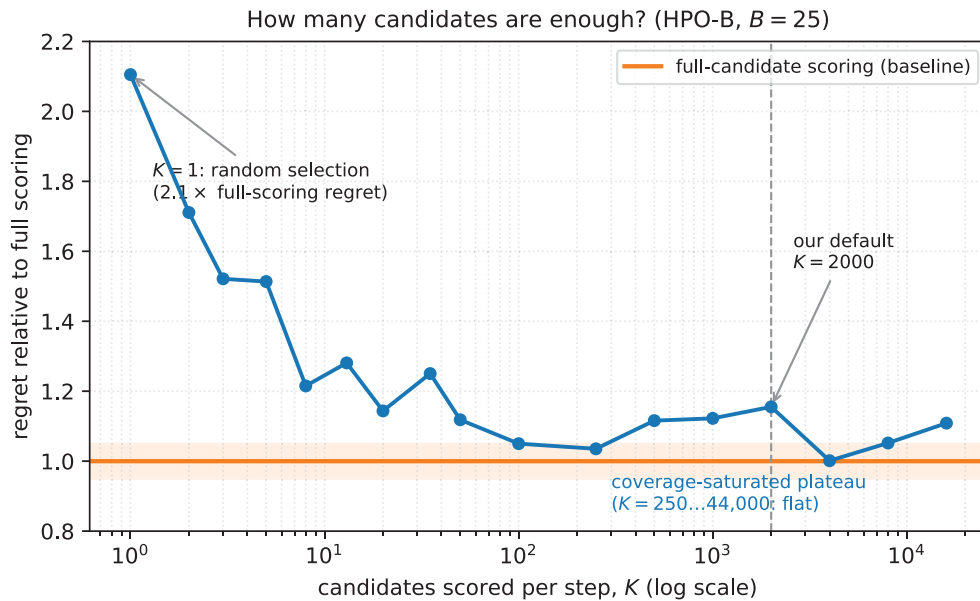


Figure 5: Final regret (normalised per space to full scoring) against the number of candidates scored per step, K , at $B = 25$. The surrogate recovers 87% of its benefit over random selection by $K = 20$ and 95% by $K = 100$; from $K \approx 250$ to $K = 44,000$ the curve is flat (shaded band: within 5% of full scoring). The default $K = 2000$ lies well inside the plateau.

A controlled ablation confirms the flatness across the entire operating range. Holding K fixed at each value from 50 to full scoring on the 77 meta-test tasks (5 seeds, all 14 spaces), no fixed size differs significantly

from exhaustive scoring: every paired Wilcoxon test against full scoring is non-significant (Table 4), and a Friedman omnibus across all $K \geq 100$ finds no difference among them ($\chi^2 = 8.3$, $p = 0.31$). Median regret is essentially constant at ≈ 0.018 throughout.

Table 4: Accuracy is flat in the candidate budget. Fixed- K RF-SMBO on the 77 meta-test tasks (5 seeds); regret paired against full scoring per task–seed. No K differs significantly from full scoring (all $p > 0.05$; Friedman across $K \geq 100$: $p = 0.31$). Adaptive- K (bottom) matches this accuracy while scoring the fewest candidates.

Candidates K	Mean	Median	Δ	p
100	0.065	0.022	−0.009	0.97
500	0.074	0.018	+0.001	0.64
1000	0.069	0.018	−0.005	0.48
2000 (RF-CAND-2K)	0.079	0.018	+0.005	0.27
4000	0.065	0.020	−0.008	0.28
16,000	0.073	0.018	−0.001	0.88
Full scoring	0.074	0.017	—	—
Adaptive- K		Matches full scoring; $\bar{K}=1210$		

This flatness simplifies deployment. The candidate budget is not a sensitive hyperparameter: any value from about a hundred upward gives full-scoring accuracy, so the frozen default $K = 2000$ has a large safety margin, lying deep in the plateau so that the coverage bound (Lemma 1) holds even for landscapes with few near-optimal configurations. The plateau is a property of the surrogate, not of the budget: the rise to saturation is present at $B = 10$ as well as $B = 25$, with the saturation point in the same position. The steep left end shows the surrogate is essential, since selection from a single random candidate ($K = 1$) doubles regret; restriction is therefore not equivalent to random search, and the surrogate needs only a few dozen candidates.

Because regret is flat in K , the adaptive schedule cannot and does not improve accuracy: ADAPTIVE- K and the frozen fixed- K rule RF-CAND-2K differ by only +0.2% in mean regret. The schedule’s value is operational: by contracting the candidate set when the forest’s dispersion is low, it scores a mean of 1210 candidates per step, fewer than the fixed $K = 2000$, at indistinguishable accuracy, and its full-scoring floor restores the exact small-pool identity (Corollary 1).

5.5 Computational Cost at Different Pool Sizes

The computational saving of restriction grows with pool size. Table 5 reports wall-clock timings for a single RF-SMBO step (64-tree forest, one thread) at seven representative pool sizes spanning $N = 300$ to $N = 52,000$. Below $N = 5000$ the scoring phase is cheap relative to forest fitting (≤ 10 vs. ~ 18 ms) and restriction provides negligible speedup (1.0–1.2 $\times$). Beyond $N = 10,000$ scoring dominates: at $N = 25,000$ restriction scores only $K = 2000$ of 25,000 candidates, giving a 7.8 $\times$ scoring speedup; at $N = 52,000$ (the largest HPO-B space) the scoring speedup reaches 16.4 $\times$, and the end-to-end per-step speedup is 4.7 $\times$ (Table 3). The absolute per-step time with restriction stays nearly constant (≈ 24 ms) regardless of pool size, confirming that restriction converts a linear cost into a constant.

5.6 Why Restriction is Safe: Coverage, Not Selection Bias

We measure the mechanism behind this flatness directly. Coverage (Lemma 1) predicts that a random subset almost always retains the full pool’s best candidates. Instrumenting adaptive- K runs across all 14 spaces (5300 restricted steps), we compare, at each step, the highest acquisition value in the restricted subset against

the highest in the full unevaluated pool. The subset retains a candidate within 5% of the full-pool acquisition maximum on 85.5% of steps and within 10% on 95.3%, with a median loss of only 0.6% of the acquisition range. Restriction rarely loses meaningful acquisition value, exactly as the coverage bound predicts.

Table 5: Detailed wall-clock acquisition cost vs. pool size. Each row is a single RF-SMBO step with a 64-tree forest; “full” scores all N candidates, “restr.” scores $K = 2000$. Fit time (≈ 18 ms) is independent of N and not shown.

Pool N	Dim	Score Full (ms)	Score Restr. (ms)	Scoring Speedup
300	3	2.6	2.5	1.0×
1000	6	3.8	3.8	1.0×
5000	8	10.4	5.4	1.9×
11,000	9	20.5	5.5	3.7×
25,000	12	44.3	5.7	7.8×
44,000	6	70.9	5.4	13.0×
52,000	16	100.8	6.2	16.4×

The alternative explanation is that restriction helps by reducing *over-selection bias*, the optimiser’s curse of Lemma 2. Because HPO-B provides the true value $f(x)$ of every configuration, we measure this directly as the post-decision surprise $\mu_t(x_t) - f(x_t)$: the amount by which the forest over-predicts the value of the configuration it selects. On the random-forest surrogate the effect is essentially absent. Full-candidate RF-SMBO over-predicts its selected configuration by a mean of only +0.004 normalised regret (median -0.03 : the forest’s predictions regress toward observed values, so it tends to *under*-predict rather than over-predict the configurations it selects), and adaptive- K ’s figure is statistically identical (+0.006; paired $p = 0.38$). Nor does the bias grow with the number of candidates scored: measured counterfactually at the surrogate states an RF-SMBO run visits, over-prediction is flat in K (Table 6, Fig. 6a; a fit against $\sqrt{2 \ln K}$ gives $R^2 = 0.14$, slope ≈ 0).

Table 6: Optimiser’s curse on the RF surrogate: mean over-prediction (post-decision surprise) of the selected configuration as a function of the number of candidates scored, K . There is no systematic growth (fit against $\sqrt{2 \ln K}$: $R^2 = 0.14$), in contrast to the controlled sub-Gaussian simulation where $R^2 = 1.00$. Normalised-regret units.

K	50	250	1000	5000	N (full)
Mean over-prediction	+0.005	+0.005	+0.004	+0.012	+0.004

A controlled simulation confirms that the theory and the measurement are sound. With sub-Gaussian surrogate error the over-selection bias fits $\sigma\sqrt{2 \ln K}$ with $R^2 = 1.00$ (Fig. 6b), lies just below the upper bound of Lemma 2, falls by 35% when the candidate set is cut from 10^4 to 10^2 , and vanishes (≈ 0.006) once the surrogate is reliable. The curse is thus real when surrogate errors are near-independent across candidates with non-negligible scale, as in the simulation; the forest’s errors are small, strongly correlated across configurations, and mean-reverting, so the extreme-value growth that drives the curse never materialises. Coverage, not bias reduction, is what makes restriction safe here.

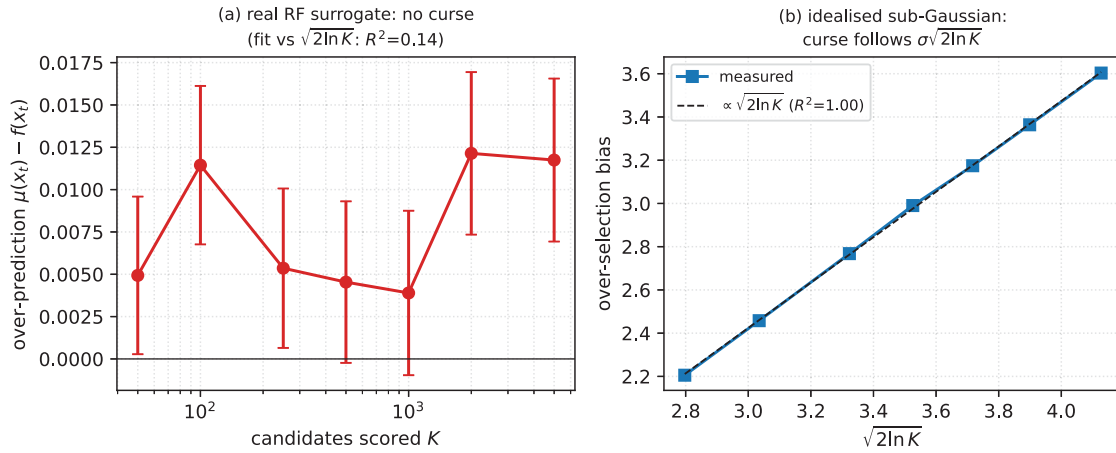


Figure 6: The optimiser's curse, measured. (a) On the random-forest surrogate the over-prediction of the selected configuration shows no systematic growth with the number of candidates scored (fit against $\sqrt{2\ln K}$: $R^2 = 0.14$). (b) In a controlled sub-Gaussian simulation the same measurement recovers the $\sigma\sqrt{2\ln K}$ law exactly ($R^2 = 1.00$): the curse is real for near-independent errors of non-negligible scale, a regime the forest's small, correlated, mean-reverting errors do not enter.

5.7 Where Restriction Loses

We report every held-out task on which adaptive-K underperforms full scoring by more than the tie tolerance: 20 tasks (10 test, 10 validation), listed in full in Table A2 of the supplement. All lie in large pools ($N > F$); none is a saturated or easy task. Fig. 7 plots every per-task difference against pool size: small-pool differences are exactly zero (small-pool identity), and large pools scatter in *both* directions, with adaptive-K's wins and its losses both occurring there. The worst cases are a single hard held-out SVM task (5527/3493, $\Delta = -0.195$) and two large Rpart tasks (5636/10101, $\Delta = -0.172$; 5859/31, $\Delta = -0.149$).

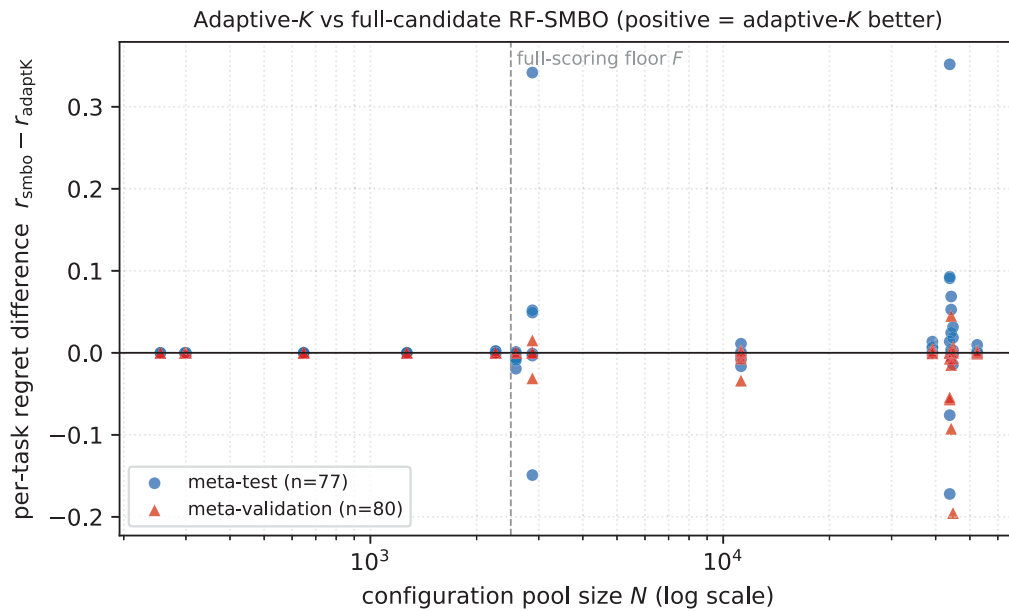


Figure 7: Per-task regret difference $r_{\text{smb0}} - r_{\text{adaptK}}$ (positive = adaptive-K better) vs. configuration-pool size, for all 157 held-out tasks. Small pools (left of the floor F) are ties by construction; large pools scatter in both directions.

The coverage instrumentation shows these are not coverage failures: the per-task coverage gap does not correlate with the per-task regret difference (Spearman -0.16 , $p = 0.26$). Large-pool comparisons are high-variance: restriction can under-cover a difficult landscape on one task, while full scoring can select noise-inflated configurations on another, and the two effects offset to give the matched accuracy reported above.

5.8 Live-Training Validation

In the XGBoost study, two results carry over from the precomputed benchmark to live training. First, the surrogate is informative in this setting: at $B = 25$ both full RF-SMBO and adaptive-K outperform random search (rf-smbo vs. random $+0.0018$ AUC, 11/14 datasets, $p = 0.003$; adaptive-K vs. random $+0.0026$, 14/14, $p < 0.001$), so this is not a saturated regime in which all methods perform equally. Second, given this baseline, adaptive-K matches full scoring (mean AUC difference $+0.0009$, 9/14 datasets, $p = 0.33$) while scoring $\approx 20\times$ fewer candidates per step (22,666 vs. 1145; Table 7, Fig. 8).

Table 7: XGBoost live-training study: mean cross-validated AUC over 14 OpenML datasets, by budget, paired at the dataset level ($n = 14$, seeds averaged within dataset). Adaptive-K matches full RF-SMBO; both outperform random search by $B = 25$. Paired Wilcoxon p against random in the last two columns.

B	Random	rf-smbo	Adaptive-K	$p_{\text{smbo-rand}}$	$p_{\text{adaptK-rand}}$
5	0.854	0.854	0.854	—	—
10	0.859	0.859	0.861	0.58	0.04
25	0.863	0.864	0.865	0.003	<0.001

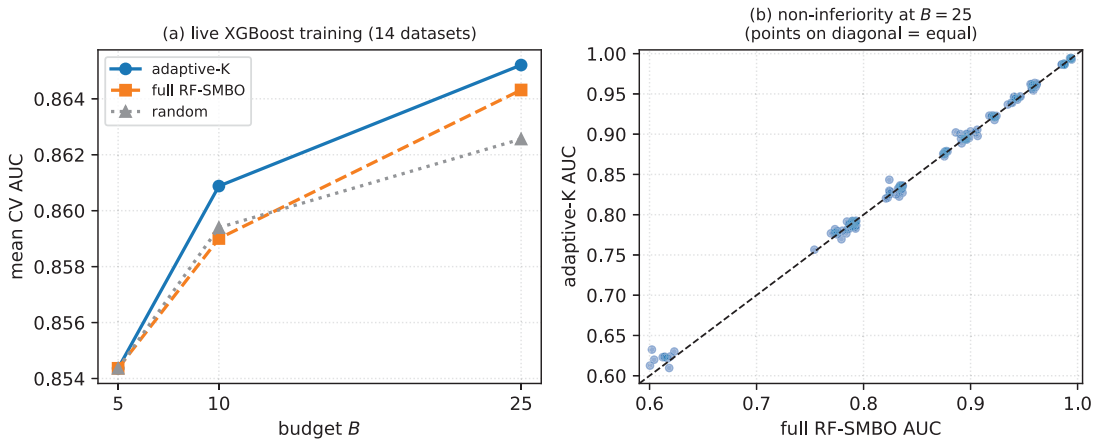


Figure 8: Live XGBoost training over a 22,680-configuration grid, 14 OpenML datasets. (a) Mean CV AUC against budget: adaptive-K and full RF-SMBO outperform random search, and adaptive-K follows full scoring closely, slightly exceeding it. (b) Per-run comparison at $B = 25$: adaptive-K vs. full RF-SMBO lies on the diagonal.

The random-forest study replicates the result on a second learner family and a second, independently constructed grid: at $B = 25$ adaptive-K matches full RF-SMBO (mean AUC difference $+0.0008$, 7/3 datasets, $p = 0.19$) while scoring $4.7\times$ fewer candidates per step (6034 vs. 1289; the ratio is smaller because the grid is smaller), and the surrogate again outperforms random search ($+0.0024$ AUC, 10/10 datasets, $p = 0.002$). The effect is a property of the candidate pool, not of any one learner, grid or benchmark.

5.9 Restriction across Surrogates

Coverage is a property of the candidate pool, not of the surrogate, so restriction should extend beyond the random forest. We repeat the restricted-vs.-full comparison ($K = 2000$, $B = 25$) with two other surrogates: tree-structured Parzen estimator (TPE, kernel density) [34] on all 14 spaces, and GP-UCB [16] on the eleven lower-dimensional spaces (dimension ≤ 8) where a Gaussian process fit on 5–25 points is well-conditioned (Table 8). For each surrogate we test both the fixed restriction ($K = 2000$) and the full adaptive-K schedule, using the surrogate’s own uncertainty proxy to set K_t : tree-dispersion standard deviation for the random forest, predictive posterior standard deviation for the GP, and acquisition-score standard deviation over a probe sample for TPE. Comparisons are paired at the task level, as in the main analysis.

Table 8: Restriction vs. full scoring for three surrogates ($B = 25$, paired at the task level). Both fixed $K = 2000$ and the adaptive-K schedule are shown. The random forest and TPE match full scoring for both modes; GP-UCB shows a small but significant gap that does not widen from fixed to adaptive restriction.

Surrogate	Mode	Regret	Δ vs. Full	p	Cost Red.
Random forest	Restricted	—	—	≥ 0.10	$\sim 20\times$
	Adaptive	—	—	≥ 0.10	$\sim 27\times$
TPE	Full	0.115	—	—	$1\times$
	Restricted	0.121	+0.005	0.65	$13.6\times$
	Adaptive	0.128	+0.012	0.08	$22.8\times$
GP-UCB	Full	0.042	—	—	$1\times$
	Restricted	0.048	+0.005	0.011	$13.7\times$
	Adaptive	0.050	+0.008	0.020	$15.9\times$

For **TPE**, both fixed restriction and adaptive-K match full scoring: restricted scores 0.121 vs. 0.115 for full ($p = 0.65$, $n = 352$ paired observations) at $13.6\times$ lower cost; adaptive-K scores 0.128 ($p = 0.08$) at $22.8\times$ lower cost. For **GP-UCB**, fixed restriction scores 0.048 vs. 0.042 for full ($p = 0.011$, $n = 281$); adaptive-K scores 0.050 ($p = 0.020$) at $15.9\times$ lower cost. The GP is the least favourable of the three surrogates, consistent with the coverage intuition: predictions that are bounded or local (the forest’s tree averages, TPE’s densities) saturate early, whereas a Gaussian process’s smooth global posterior may benefit slightly more from a large candidate set. However, the adaptive-K schedule adds no statistically significant loss over fixed restriction for either surrogate ($p = 0.25$ for TPE, $p = 0.78$ for GP) while further reducing the candidate count.

6 Discussion and Limitations

In practice, the results support a simple default: on grids beyond the full-scoring floor, maximise the acquisition over a few thousand random candidates; on small pools, score everything. The change adds no evaluations and, because accuracy is flat in the candidate budget, requires no per-task tuning. Where a single objective evaluation is very expensive, restriction saves proportionally less, since the acquisition step is then a minor share of the wall-clock. Six limitations qualify these claims.

6.1 Matched, Not Improved, Accuracy

Restriction matches exhaustive scoring; it does not improve on it. The median per-task difference is zero, and the benefit is the reported reduction in acquisition cost. On individual large-pool tasks the outcome varies in both directions; the matched accuracy is a distribution-level statement. A better-calibrated surrogate

is the most plausible route to converting the match into a gain, since it would improve the candidate ranking itself rather than only reduce its variance.

6.2 Evaluation Scope

The primary evidence comes from one standardised benchmark family, HPO-B, with precomputed evaluations on fixed discrete grids; the two live-training studies extend it to real model fits on two learner families (XGBoost and random forest) across 24 OpenML classification datasets, but not to a second tabular suite, to continuous search spaces, or to multi-fidelity settings. The analysis is likewise specific to finite, enumerable pools, where exhaustive scoring is the standard default; in continuous spaces the acquisition is already optimised over a sampled candidate set, and whether the schedule and the small-pool identity transfer to that setting remains open.

6.3 Scope of the Theory

Theorem 1 is a one-step selection bound: it controls the per-iteration gap between the selected configuration and the best then available in the pool, not the cumulative regret of the run. The claim that restriction matches full scoring over entire runs rests on the empirical evidence of the held-out and live-training evaluations; a cumulative-regret analysis remains open and would require stronger surrogate-calibration assumptions than the sub-Gaussian condition we use.

6.4 Comparison with SMAC3

SMAC3's default acquisition maximiser combines random candidate sampling with a local-search refinement, representing an alternative restriction heuristic with a different design point from our pure random restriction [6]. The present experiments compare adaptive-K against full-candidate RF-SMBO, random search, and TPE, but do not include a direct head-to-head comparison with SMAC3's built-in maximiser. Such a comparison is a natural next step; it would isolate whether the local-search component adds value beyond what uniform random restriction provides, and whether the adaptive schedule improves on SMAC3's fixed candidate count.

6.5 Fixed Design Choices

The study fixes the surrogate (a 64-tree random forest) and the acquisition (UCB with $\beta = 1.96$), and the three adaptive-K constants were frozen on a single development space. A different surrogate, a calibrated-uncertainty variant, or another development space could shift the schedule's operating point; the multi-surrogate comparison shows that the basic restriction result, including the adaptive-K schedule, extends to TPE and GP-UCB (Table 8), but we have not characterised these interactions further.

6.6 Independence Assumption in the Coverage Bound

The coverage bound (Lemma 1) assumes uniform random sampling from the pool, which is valid by construction: the candidate subset is drawn uniformly without replacement. The bound does not require the configurations themselves to be independent or identically distributed (i.i.d.). However, the *practical benefit* of restriction depends on the landscape of the acquisition function: if the near-optimal region of the pool is highly concentrated (e.g., a few isolated configurations far from the rest), the near-optimal fraction p is small and more candidates are needed to achieve the same coverage probability. All HPO-B and live-training grids in our study are factorial or quasi-random designs over the hyperparameter space, and the saturation plateau (Fig. 5) confirms that coverage is ample on these grids. On configuration pools with strong spatial clustering

or adversarial structure, a larger $K_{\min}$ or a stratified sampling scheme may be needed; characterising the interaction between pool geometry and the coverage bound is an open direction.

7 Conclusion

This paper revisited a routine step of random-forest SMBO: scoring every unevaluated configuration with the acquisition function at each iteration. This default originated on small search grids, where it adds little overhead; on the large grids of modern transfer-HPO benchmarks its cost grows linearly with the pool while its benefit saturates after a few hundred candidates. Randomised candidate restriction maximises the acquisition over a small random subset, sized online by ADAPTIVE-K, and reverts to exact full scoring on small pools. Across 14 HPO-B search spaces, 157 held-out tasks and two live-training studies it matches exhaustive scoring while scoring 20–27× fewer candidates per step, and it outperforms random search and TPE by more than 25%. The reason is coverage rather than any reduction in selection bias: a few thousand random candidates contain a near-optimal configuration with high probability, so accuracy saturates long before the pool is exhausted and the candidate budget is not an accuracy-sensitive hyperparameter. The rule is auditable, since its behaviour follows from a stated bound; it needs no per-task tuning, since its three constants are frozen once; and it deploys as a one-line change to any RF-SMBO implementation.

Several directions follow from this work. Better-calibrated surrogates, such as quantile forests [38], jackknife-plus, or conformal intervals, could turn the matched accuracy into a gain within the same bias–coverage framework. Extending the schedule to continuous search spaces, where candidates are sampled rather than enumerated, and to multi-fidelity optimisation, where the surrogate’s reliability varies with fidelity as well as with data, would broaden its applicability. Finally, the coverage analysis, which characterises when scoring more candidates helps, applies to any surrogate whose acquisition is maximised over a finite candidate set, and extending it to other surrogates is a natural next step.

Acknowledgement: None.

Funding Statement: This work is supported by Quanzhou University of Information Engineering, Quanzhou, China, and Universiti Tunku Abdul Rahman, Kampar, Malaysia.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualisation, Irfan Khan, Saqib Alam, Ramesh Kumar Ayyasamy and Miao Zhang; methodology, Irfan Khan and Saqib Alam; software, Irfan Khan; validation, Irfan Khan and Saqib Alam; formal analysis, Irfan Khan; investigation, Irfan Khan; data curation, Irfan Khan; writing—original draft preparation, Irfan Khan; writing—review and editing, Saqib Alam, Ramesh Kumar Ayyasamy and Miao Zhang; visualisation, Irfan Khan; supervision, Ramesh Kumar Ayyasamy and Miao Zhang; project administration, Ramesh Kumar Ayyasamy; funding acquisition, Ramesh Kumar Ayyasamy and Miao Zhang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The HPO-B benchmark [4] and the OpenML datasets [37] used in the live-training studies are publicly available. All code required to reproduce the experiments, figures and tables in this paper (the adaptive- K and baseline runners, the candidate-cost and optimiser’s-curse instrumentation, and the analysis scripts) is publicly released at <https://doi.org/10.5281/zenodo.22886237>.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Full Per-Space Results

Table A1 gives the per-space regret at $B = 25$ for all 14 HPO-B spaces and all methods; it is the disaggregated form of Fig. 2.

Table A1: Per-space results at $B = 25$ (normalised regret, mean over 20 seeds), all 14 HPO-B spaces. Lower is better; **best** in bold. Adaptive-K matches full RF-SMBO across pools; its wins and losses both fall in the large pools, consistent with the pooled result of Table 1. [≦]In search spaces with $d \geq 5$, TPE's model-initialisation phase requires at least $5d$ random evaluations; at the $B = 25$ budget this consumes all evaluations, so TPE reduces to random search. Only the two low-dimensional glmnet spaces ($d = 2, 3$) retain model-guided evaluations within this budget.

Space	Algo	Dim	N	Random	TPE	rf-smbo	Adaptive-K
6794	ranger	8	52,448	0.0349 [≦]	0.0349 [≦]	0.0349	0.0344
5527	svm	6	44,790	0.0711 [≦]	0.0711 [≦]	0.1076	0.0912
6767	xgboost	17	44,258	0.0859 [≦]	0.0859 [≦]	0.0839	0.0624
5636	rpart	6	43,841	0.4635 [≦]	0.4635 [≦]	0.3740	0.3434
6766	glmnet	2	39,155	0.0690	0.0948	0.0694	0.0616
5965	ranger	8	11,228	0.0541 [≦]	0.0541 [≦]	0.0460	0.0464
5859	rpart	6	2,878	0.2883 [≦]	0.2883 [≦]	0.2185	0.1585
5891	svm	6	2,584	0.0384 [≦]	0.0384 [≦]	0.0312	0.0258
5971	xgboost	15	2,265	0.0224 [≦]	0.0224 [≦]	0.0126	0.0169
7609	ranger	7	1,269	0.0235 [≦]	0.0235 [≦]	0.0141	0.0167
7607	ranger	7	647	0.0491 [≦]	0.0491 [≦]	0.0353	0.0361
4796	glmnet	3	300	0.0659	0.0839	0.0523	0.0455
5889	ranger	5	298	0.1143 [≦]	0.1143 [≦]	0.0379	0.0395
5906	xgboost	15	254	0.1152 [≦]	0.1152 [≦]	0.0849	0.0961

Appendix B Complete List of Large-Pool Losses

Table A2 lists every held-out task on which adaptive-K underperforms full-candidate RF-SMBO by more than the tie tolerance, the disaggregated form of the failure analysis (Fig. 7).

Table A2: All held-out tasks where adaptive-K underperforms full-candidate RF-SMBO by more than the 0.005 tie tolerance ($\Delta = r_{\text{smbo}} - r_{\text{adaptK}} < -0.005$), sorted worst-first. Every meaningful loss occurs in a large configuration pool ($N > F = 2500$); none in saturated or small-pool tasks (on which adaptive-K equals full scoring up to tie-breaking, residual $|\Delta| \leq 0.002$). Normalised regret, 20 seeds, $B = 25$.

Space	Task	Split	Dim	N	Adapt-K	rf-smbo
5527	3493	val	6	44,790	0.463	0.267
5636	10101	tes	6	43,841	0.293	0.121
5859	31	tes	6	2,878	0.279	0.130
6767	3493	val	17	44,258	0.195	0.103
5636	146064	tes	6	43,841	0.248	0.172
5636	3917	val	6	43,841	0.194	0.137
5636	10093	val	6	43,841	0.154	0.100
5965	9971	val	8	11,228	0.172	0.138
5859	9957	val	6	2,878	0.043	0.011

(Continued)

Table A2 (continued)

Space	Task	Split	Dim	N	Adapt- K	rf-smbo
5891	3899	tes	6	2,584	0.072	0.052
5965	10101	tes	8	11,228	0.109	0.093
6767	10101	val	17	44,258	0.199	0.184
5527	146064	tes	6	44,790	0.030	0.016
5891	9980	tes	6	2,584	0.091	0.082
5891	6566	tes	6	2,584	0.024	0.016
5965	145836	tes	8	11,228	0.063	0.055
5527	3913	val	6	44,790	0.079	0.072
5636	43	val	6	43,841	0.096	0.088
5965	37	val	8	11,228	0.085	0.077
5891	3492	tes	6	2,584	0.007	0.000

Appendix C Reproducibility

All results are reproducible from the public code release at <https://doi.org/10.5281/zenodo.22886237>. The frozen method is fully specified by five constants, set once on a single development space (HPO-B space 6767) and never retuned: candidate floor $K_{\min} = 1000$, growth exponent $c = 0.75$, full-scoring floor $F = 2500$, random-forest surrogate with 64 trees, and UCB acquisition with $\beta = 1.96$, after a 5-step random warm-start. The fixed- K variant uses $K = 2000$.

The evaluation uses HPO-B v3 (precomputed evaluations) over the 14 non-saturated search spaces, 77 meta-test and 80 meta-validation tasks, and 20 seeds. The live-training studies use 3-fold cross-validated AUC with 10 seeds each: XGBoost over a 22,680-configuration grid on 14 OpenML datasets, and a random forest over a 6048-configuration grid on 10 OpenML datasets.

The code release provides every runner and analysis script used, each with fixed random seeds; all reported comparisons are paired Wilcoxon signed-rank tests at a 0.005 normalised-regret tie tolerance. The main-study components are the adaptive- K and baseline runners, the candidate-cost instrumentation, the optimiser's-curse measurement and its controlled synthetic validation, and the live-training runner.

References

1. Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, et al. Hyperparameter optimization: foundations, algorithms, best practices, and open challenges. *WIREs Data Min Knowl Discov*. 2023;13(2):e1484.
2. Franceschi L, Donini M, Perrone V, Klein A, Archambeau C, Seeger M, et al. Hyperparameter optimization in machine learning. *Found Trends Mach Learn*. 2025;18(6):1054–201.
3. Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. In: *Learning and intelligent optimization (LION)*. Berlin/Heidelberg, Germany: Springer; 2011. p. 507–23.
4. Pineda-Arango S, Jomaa HS, Wistuba M, Grabocka J. HPO-B: a large-scale reproducible benchmark for black-box HPO based on OpenML. *arXiv:2106.06257*. 2021.
5. Eriksson D, Pearce M, Gardner JR, Turner RD, Poloczek M. Scalable global optimization via local Bayesian optimization. In: *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*; 2019 Dec 8–14; Vancouver, BC, Canada.
6. Lindauer M, Eggensperger K, Feurer M, Biedenkapp A, Deng D, Benjamins C, et al. SMAC3: a versatile Bayesian optimization package for hyperparameter optimization. *J Mach Learn Res*. 2022;23(54):1–9.
7. Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N. Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE*. 2016;104(1):148–75.

8. Jin Y, Wang H, Chugh T, Guo D, Miettinen K. Data-driven evolutionary optimization: an overview and case studies. *IEEE Trans Evol Comput.* 2019;23(3):442–58.
9. He X, Zhao K, Chu X. AutoML: a survey of the state-of-the-art. *Knowl Based Syst.* 2021;212:106622.
10. Kotthoff L, Thornton C, Hoos HH, Hutter F, Leyton-Brown K. Auto-WEKA 2.0: automatic model selection and hyperparameter optimization in WEKA. *J Mach Learn Res.* 2017;18(25):1–5.
11. Feurer M, Eggenberger K, Falkner S, Lindauer M, Hutter F. Auto-Sklearn 2.0: hands-free AutoML via meta-learning. *J Mach Learn Res.* 2022;23(261):1–61.
12. Baratchi M, Wang C, Limmer S, van Rijn JN, Hoos H, Bäck T, et al. Automated machine learning: past, present and future. *Artif Intell Rev.* 2024;57(5):122.
13. Khan I, Zhang X, Rehman M, Ali R. A literature survey and empirical study of meta-learning for classifier selection. *IEEE Access.* 2020;8:10262–81. doi:10.1109/access.2020.2964726.
14. Khan I, Zhang X, Ayyasamy RK, Alhashmi SM, Rahim A. Enhancing classification algorithm recommendation in automated machine learning: a meta-learning approach using multivariate sparse group lasso. *Comput Model Eng Sci.* 2025;142(2):1611–36.
15. Garnett R. Bayesian optimization. Cambridge, UK: Cambridge University Press; 2023.
16. Srinivas N, Krause A, Kakade SM, Seeger MW. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Trans Inf Theory.* 2012;58(5):3250–65. doi:10.1109/tit.2011.2182033.
17. Kandasamy K, Vysyaraju KR, Neiswanger W, Paria B, Collins CR, Schneider J, et al. Tuning hyperparameters without grad students: scalable and robust Bayesian optimisation with Dragonfly. *J Mach Learn Res.* 2020;21(81):1–27.
18. Cowen-Rivers AI, Lyu W, Tutunov R, Wang Z, Grosnit A, Griffiths RR, et al. HEBO: pushing the limits of sample-efficient hyper-parameter optimisation. *J Artif Intell Res.* 2022;74:1269–349.
19. Hvarfner C, Hellsten EO, Nardi L. Vanilla Bayesian optimization performs great in high dimension. *arXiv:2402.02229.* 2024.
20. Rodemann J, Augustin T. Imprecise Bayesian optimization. *Knowl Based Syst.* 2024;300:112186. doi:10.1016/j.knosys.2024.112186.
21. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *J Mach Learn Res.* 2012;13:281–305.
22. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. *Adv Neural Inf Process Syst.* 2012;25:1–9.
23. Hoffman M, Brochu E, de Freitas N. Portfolio allocation for Bayesian optimization. In: *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI); 2011 Jul 14–17; Barcelona, Spain.* p. 327–36.
24. Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A. Hyperband: a novel bandit-based approach to hyperparameter optimization. *J Mach Learn Res.* 2018;18(185):1–52.
25. Falkner S, Klein A, Hutter F. BOHB: robust and efficient hyperparameter optimization at scale. *Proc Mach Learn Res.* 2018;80:1437–46.
26. Mallik N, Bergman E, Hvarfner C, Stoll D, Janowski M, Lindauer M, et al. PriorBand: practical hyperparameter optimization in the age of deep learning. *Adv Neural Inf Process Syst.* 2023;36:7377–91.
27. Zimmer L, Lindauer M, Hutter F. Auto-PyTorch tabular: multi-fidelity metalearning for efficient and robust AutoDL. *IEEE Trans Pattern Anal Mach Intell.* 2021;43(9):3079–90.
28. Feurer M, Springenberg JT, Hutter F. Initializing Bayesian hyperparameter optimization via meta-learning. *Proc AAAI Conf Artif Intell.* 2015;29(1):1128–35.
29. Vanschoren J. Meta-learning. In: *Automated machine learning: methods, systems, challenges.* Berlin/Heidelberg, Germany: Springer; 2019. p. 35–61.
30. Wistuba M, Schilling N, Schmidt-Thieme L. Scalable Gaussian process-based transfer surrogates for hyperparameter optimization. *Mach Learn.* 2018;107(1):43–78. doi:10.1007/s10994-017-5684-y.
31. Smith JE, Winkler RL. The Optimizer's curse: skepticism and postdecision surprise in decision analysis. *Manage Sci.* 2006;52(3):311–22.
32. Breiman L. Random forests. *Mach Learn.* 2001;45(1):5–32.

33. Boucheron S, Lugosi G, Massart P. Concentration inequalities: a nonasymptotic theory of independence. Oxford, UK: Oxford University Press; 2013.
34. Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. *Adv Neural Inf Process Syst.* 2011;24:1–10.
35. Demšar J. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res.* 2006;7:1–30.
36. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2016 Aug 13–17; San Francisco, CA, USA. p. 785–94.
37. Vanschoren J, van Rijn JN, Bischl B, Torgo L. OpenML: networked science in machine learning. *ACM SIGKDD Explor Newsl.* 2014;15(2):49–60.
38. Meinshausen N. Quantile regression forests. *J Mach Learn Res.* 2006;7:983–99. doi:10.32614/cran.package.quantregforest.