SCOUT: WHAT TO PRINT AND WHAT TO TEST IN TWO-BUDGET CLOSED-LOOP DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Materials design under additive manufacturing (AM) couples three oracles whose costs differ by orders of magnitude: a structurally biased finite-element simulator, an in-situ infrared (IR) video that every print records at no marginal cost, and a destructive mechanical test that only a few printed parts can receive. We formalize this as *two-budget closed-loop design with a free intermediate observation*, a relative of Bayesian optimization of function networks in which the intermediate observation comes only with a print and each test consumes a printed part from its own budget. We introduce **SCOUT**, whose residual Gaussian process corrects the simulator from design and IR features and whose two-stage acquisition chooses what to print and then, once the IR is observed, what to test. On a literature-anchored Nickel Aluminum Bronze generator, SCOUT attains lower simple regret than thirteen baselines, including design-of-experiments practice, at every budget, and at the smallest budget 43% lower regret than EPIG on the identical IR-equipped surrogate, 37% with IR noise raised to the dispersion of a real recording, a margin that narrows as the test budget grows. We show that a budget-fraction test threshold read against a part's own prediction tests every design at the same rate under a calibrated model and can discard the best part, and we derive a value-of-testing rule from the two budgets. In a pre-registered study on 150 random instances of public battery and concrete data, the new rule keeps the best battery cell that the old threshold discards in 39 of 53 runs with labelled history and lowers SCOUT's regret there by 8.6 and 12.8 points after Holm's correction, to the lowest mean of 21 arms, tied in one family. A post hoc analysis traces the effect to five cells whose recorded life merges two test batches, and without history a Gaussian process on the design alone is better. A physical AM campaign is left to future work.

1 INTRODUCTION

Metal additive-manufacturing (AM) design follows a loop from parametric design through numerical simulation to physical validation, and its three information sources differ in cost by orders of magnitude. A metal AM build plus destructive characterization costs days and consumables, a finite-element (FEM) simulation costs CPU-minutes, and in between an in-situ infrared (IR) thermal camera records gigabytes of radiometric video at no marginal cost. Closed-loop AM systems leave in-situ signals out of the surrogate (Gongora et al., 2020; Lee et al., 2025b; Fadhel et al., 2026) or optimize them as the objective (Deneault et al., 2021; Sousa et al., 2025), and in polymer extrusion an in-situ feature correlated with tensile strength stands in for the tensile test, without a simulator (Sakai et al., 2026). Throughout, *closed loop* denotes the build-measure-update loop of a parameter-development campaign, one recipe per build, and not control within a build.

Process-parameter development in AM practice still relies on classical design of experiments, response-surface fits, grid search, and expert iteration (Box & Wilson, 1951; Myers et al., 2016; Ahmed et al., 2022; Krasniqi & Löffler, 2024), and none of these uses the simulator or the IR record. Hierarchical multi-fidelity Bayesian optimization (BO) (Kandasamy et al., 2017; Takeno et al., 2020; Wu et al., 2019) treats cheaper fidelities as noisy or coarser versions of the target, and multi-information-source variants (Poloczek et al., 2017; Mikkola et al., 2023) model a source-specific bias as a function of the design alone, so neither uses a measurement taken during the build to explain the *structural* bias of FEM. Acquisition-coupled discrepancy methods (Yang et al., 2025a; Oliveira et al., 2024; Gongora et al., 2021) learn a simulator residual inside sequential design but have no

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.

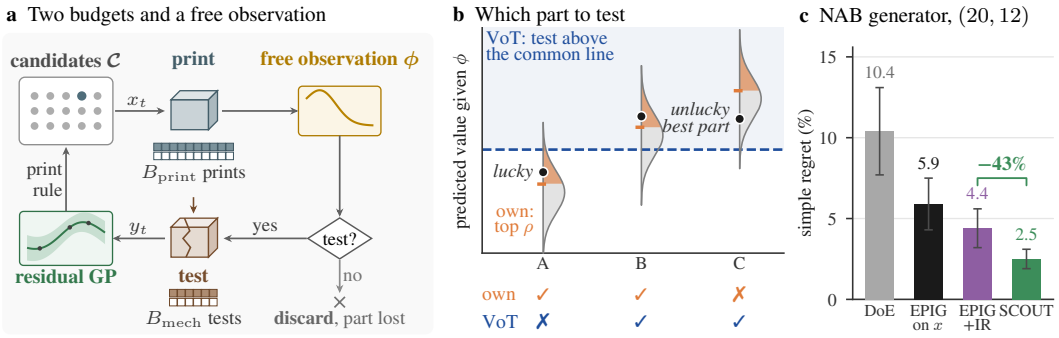

Figure 1: **SCOUT decides what to print and which printed part to test, and on the NAB generator at** $(20, 12)$ **it has 43% lower regret than EPIG on the identical IR-equipped surrogate.** **a** Each print reveals a free observation ϕ, an IR video, early cycles, or a 3-day strength, and the part is tested at once or discarded. **b** Schematic for three parts of equal predictive spread. The own-prediction threshold, *own*, tests a part whose observation is in the top ρ of its own predictive law, with ρ the ratio of tests left to prints left, and the value-of-testing rule, *VoT*, tests a part whose predicted value clears one threshold set by the incumbent and the budgets left. **c** Regret at $(20, 12)$ with SCOUT's concatenated-input kernel, mean and standard deviation over 12 seeds, Table 1.

intermediate observation. BO of function networks and of cascade processes (Astudillo & Frazier, 2021a; Kusakawa et al., 2022; Buathong et al., 2024; Torresi & Friederich, 2026) uses intermediate observations, and its partial-evaluation and suspension variants can decline to pay for a downstream stage, but these methods charge every stage to a single budget and none uses a simulator as the prior mean. To our knowledge, no method combines a structurally biased simulator as the prior mean, an in-situ observation that is free once a part is built, and a separately budgeted destructive test within one acquisition.

Which printed part should receive one of the scarce destructive tests?

We formalize this setting as *two-budget closed-loop design with a free intermediate observation* and introduce **SCOUT** (*Simulation-Calibrated, in-situ-Informed, Optimal-Utility-Targeted* experiment selection). The premise is that the IR record measures part of the state that the simulator gets wrong, so a residual GP can use it as an input group, and an acquisition that values the free observation can steer both the printing and the testing decision. SCOUT comprises a structured residual GP $\hat{y}(x) = s(x) + r_\theta(x, \phi_{\text{FEM}}, \phi_{\text{IR}})$ with an additive group kernel, a deterministic pipeline that compresses $\sim$11 GB of radiometric video into twelve features, a two-stage acquisition for printing and for testing whose IRInfo term scores the value of the free observation, and weights motivated by a one-step look-ahead expansion. On a literature-anchored Nickel Aluminum Bronze (NAB) generator, where all three oracles exist, the IRInfo term and the two-stage decision give SCOUT 43% lower regret than EPIG on the same IR-equipped surrogate at the smallest budget. On real data with labelled history, once prints are chosen by a model on the IR-equipped surrogate, the rule that decides which printed parts receive the scarce test changes regret more than the choice among such print rules, and the natural implementation of that decision fails in a specific way. A budget-fraction threshold read against a part's own prediction ranks each part against its own predictive law rather than against the other parts, so under a calibrated model it tests every design at the same rate, rewards lucky parts, and can discard the best part of the pool. We prove this, derive a value-of-testing rule from the two budgets, and compare both with the closest alternatives in a pre-registered study on public battery and concrete data, Figure 1.

In summary, this paper makes the following contributions:

(1) **Problem and method.** We formalize two-budget closed-loop design with a biased simulator and a free in-situ observation, relate it to function-network and cascade BO (§3), and propose SCOUT (§4 and Figure 4 in Appendix A).

(2) **The test decision.** Under a calibrated model, a budget-fraction test threshold read against a part's own prediction tests every design at the same rate, because it ranks each part against its own predictive law, and a value-of-testing rule derived from the two budgets tests a part when

its predictive mean clears a threshold that does not rise as its predictive spread grows, §4.3 and Appendix N.

(3) **Experiments and artifacts.** On a literature-anchored NAB / C95800 generator, SCOUT attains lower simple regret than thirteen baselines, two of them design-of-experiments practice, at every budget, and 43% lower regret than the strongest baseline on its identical IR-equipped surrogate at $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$, with $p = 1.5 \times 10^{-4}$, and 37% with IR noise raised to the dispersion of a real recording (§5.1 and §5.2). A pre-registered study on 150 random instances of public battery and concrete data compares 21 arms, among them the knowledge gradient and EI through the network, deferred testing, and an early-prediction loop, and traces the effect of the test rule to the parts it depends on (§5.3 and §5.4). The code, the generator, the IR pipeline, the open harness with the pre-registration file, and a 1.3 TB radiometric-video archive without mechanical labels will be released.

2 Related Work

Multi-fidelity, calibration, and decision-aware design. Multi-fidelity surrogates and multi-information-source BO (Kennedy & O'Hagan, 2000; Kandasamy et al., 2017; Takeno et al., 2020; Poloczek et al., 2017; Mikkola et al., 2023; Zanjani Foumani et al., 2024; Xing et al., 2025) model a biased cheap source as a function of the design. The Kennedy–O'Hagan family (Kennedy & O'Hagan, 2001; Spitieris & Steinsland, 2023; Sung & Tuo, 2024) models the simulator-experiment gap mostly offline, and acquisition-coupled discrepancy work (Yang et al., 2025a; 2026; Oliveira et al., 2024; Gongora et al., 2021) learns a residual inside sequential design without an intermediate observation. BAX (Neiswanger et al., 2021; Chitturi et al., 2024; Cheng et al., 2024), EPIG (Bickford Smith et al., 2023), EULBO (Maus et al., 2024), and amortized and goal-oriented BED (Huang et al., 2024; Zhang et al., 2025b) assume a single observation channel.

Function networks, cascades, and cost-aware evaluation. BO of function networks gives each node of a network of functions its own GP and exploits intermediate outputs (Astudillo & Frazier, 2021a), and with partial evaluations a cost-aware knowledge gradient chooses which node to evaluate, at any earlier upstream output (Buathong et al., 2024). Cascade BO can suspend a multistage process after an unpromising stage (Kusakawa et al., 2022), and multi-stage BO for self-driving laboratories uses intermediate proxies to decide whether a sample advances to a costlier stage (Torresi & Friederich, 2026). Successive halving (Li et al., 2018; Falkner et al., 2018) and freeze-thaw models (Swersky et al., 2014; Wistuba et al., 2022; Rakotoarison et al., 2024; Lee et al., 2025a) observe part of an evaluation before continuing it, and closed-loop battery optimization replaces the full evaluation by an early prediction (Attia et al., 2020), all within one budget. Choosing which arriving part to inspect is a version of Pandora's box (Weitzman, 1979; Esfandiari et al., 2019; Xie et al., 2024). SCOUT's inner stage is a partial-evaluation decision in which the intermediate observation is revealed only by printing and costs nothing beyond it, the test consumes the printed part, is decided before the next print, and draws on a separate count budget, and a biased simulator supplies the prior mean.

In-situ AM monitoring and closed-loop systems. Metal AM work uses in-situ IR or multi-sensor data (Moshiri et al., 2023; Penny et al., 2025; Kavas et al., 2025) for control, defect classification, or feature extraction. In cross-build BO, in-situ data define the objective (Deneault et al., 2021; Sousa et al., 2025) or, in polymer extrusion, serve as an intermediate feature that stands in for tensile strength, without a simulator (Sakai et al., 2026), and recent metal-AM loops (Lee et al., 2025b; Chernyavsky et al., 2025; Fadhel et al., 2026) and autonomous laboratories (Szymanski et al., 2023; Gongora et al., 2020; Harris et al., 2025) do not use IR in acquisition. Appendix M extends this discussion to look-ahead BO and sim-to-real residual learning.

3 Problem Formulation

Let $\mathcal{C} \subset \mathcal{X} \subseteq \mathbb{R}^d$ be a finite candidate pool. Each $x \in \mathcal{C}$ admits three oracles at distinct costs: a deterministic FEM scalar $s : \mathcal{X} \to \mathbb{R}$ and FEM features $\phi_{\text{FEM}}(x) \in \mathbb{R}^{p_1}$ (a free byproduct of s); an in-situ thermal video $\Phi_{\text{IR}}(x) \in \mathbb{R}^{H \times W \times T}$ collected during physical printing, from which we extract $\phi_{\text{IR}}(x) \in \mathbb{R}^{p_2}$; and a destructive mechanical measurement $y(x) \in \mathbb{R}$. The agent receives two budgets

(a print budget B_{print} and a mechanical-test budget $B_{\text{mech}} \leq B_{\text{print}}$) and a two-stage policy

$x_t \sim \pi_{\text{print}}(\cdot \mid \mathcal{D}_{t-1})$; $\Phi_{\text{IR}}(x_t)$ observed, $\phi_{\text{IR}}(x_t)$ extracted; $b_t \sim \pi_{\text{mech}}(\cdot \mid x_t, \phi_{\text{IR}}(x_t), \mathcal{D}_{t-1})$; if $b_t = 1$: $y(x_t)$ observed.

We minimize simple regret on mechanically validated designs: $R(B_{\text{print}}, B_{\text{mech}}) = y^* - \max_{x \in \mathcal{D}_T^{\text{full}}} y(x)$, $y^* = \max_{x \in \mathcal{C}} y(x)$. In the terms of function-network BO (Astudillo & Frazier, 2021a; Buathong et al., 2024), printing evaluates an intermediate node that returns ϕ_{IR} and the test evaluates y on the part just printed. The setting specializes partial evaluations in four respects: (i) the intermediate observation costs nothing beyond the print and cannot be obtained without it, (ii) prints and tests draw on two separate count budgets rather than one cost-weighted budget, (iii) the test consumes the printed part and is decided before the next print, and (iv) a deterministic but structurally biased simulator s supplies the prior mean of y. The build is a black box from the per-build recipe x to the outcome, and s is deterministic by construction, with a fixed mesh, solver, and map from x to heat-source parameters, as Appendix I specifies.

4 The SCOUT Framework

4.1 Physics-Derived Features

FEM features $\phi_{\text{FEM}}(x) \in \mathbb{R}^{p_1}$. Relative density, surface-to-volume ratio, von Mises stress percentiles, plastic-zone fraction, buckling modes, and predicted phase fractions summarize the simulation field at no cost beyond s.

IR features $\phi_{\text{IR}}(x) \in \mathbb{R}^{p_2}$. We extract features from six categories supported by in-situ monitoring literature (Lane et al., 2016; Wang et al., 2022; Vallabh & Zhao, 2023; Mohr et al., 2020; Hinnebusch et al., 2026): (I) thermal intensity; (II) melt-pool geometry; (III) cooling dynamics, including the cooling rate at 565°C; (IV) heat accumulation; (V) anomaly signatures; (VI) isotherm features, namely the time above 565°C, its crossings, and the dwell up to the 1043°C solidus. The 565°C isotherm is the binary Cu–Al eutectoid, not the κ_{III}-forming reaction of NAB (Ponweiser et al., 2011; Ding et al., 2016), so these features measure thermal exposure and not a κ-phase fraction. The experiments use $p_2 = 12$ features, and the pipeline of Appendix J is deterministic post-processing with no learned component.

4.2 Structured Residual GP

We model the response as $\hat{y}(x) = s(x) + r_\theta(x, \phi_{\text{FEM}}(x), \phi_{\text{IR}}(x))$, preserving $s(\cdot)$ as a known additive backbone and learning only the structured residual r_θ as a Gaussian process with the additive group kernel

$$k((x, \phi_{\text{FEM}}, \phi_{\text{IR}}), (x', \phi'_{\text{FEM}}, \phi'_{\text{IR}})) = k_x(x, x') + k_{\phi_{\text{FEM}}}(\phi_{\text{FEM}}, \phi'_{\text{FEM}}) + k_{\phi_{\text{IR}}}(\phi_{\text{IR}}, \phi'_{\text{IR}}), \quad (1)$$

equivalently $r_\theta = r_x + r_F + r_{\text{IR}}$ with three independent GP components, each with an ARD-Matérn kernel on its own input group. The Gram matrix is the sum $K = K_x + K_F + K_{\text{IR}}$, so the prior variance and the trace, the quantities the weight derivation of Appendix B uses, decompose exactly, although eigenvalues and log-determinants do not. With few mechanical tests, the IR component r_{IR} lets two recipes whose prints look alike thermally share residual information, which a residual on x alone (Oliveira et al., 2024) cannot express. A second instantiation, a single ARD-Matérn kernel over the concatenated input, is used by the IR-equipped baselines, the ablation grid, and the robustness sweeps on the generator and by every IR-equipped arm on battery and concrete data. On the generator it differs from the additive kernel by 0.1 regret points at every budget, within seed noise, Table 1. At outer-stage decisions, $\phi_{\text{IR}}(x)$ is unobserved for unprinted candidates, and we marginalize via $\mathbb{E}_{\phi_{\text{IR}}}[r_\theta(x, \phi_{\text{FEM}}, \phi_{\text{IR}})]$ using a separately fit predictor on already-printed parts.

4.3 Two-Stage Acquisition

Outer / print acquisition. For each unprinted x,

$$a_{\text{print}}(x) = \alpha\, \text{Perf}(x) + \beta\, \text{Unc}(x) + \gamma\, \text{CalVal}(x) + \delta\, \text{IRInfo}(x), \quad (2)$$

where $\mathrm{Perf}, \mathrm{Unc}$ are the standard EI / UCB-style terms on $\hat{y}$; $\mathrm{CalVal}(x) = \mathbb{E}_{(\phi_{\mathrm{IR}},y)}[\sum_{x' \in \mathcal{C}_{\text{top-k}}} \Delta\sigma^2(x' \mid \mathcal{D} \cup \{(x, \phi_{\mathrm{IR}}, y)\})]$ is pool-aware variance reduction restricted to top-k candidates; and $\mathrm{IRInfo}(x)$ is defined identically but conditions on ϕ_{IR} alone, reflecting that printing yields an IR observation *regardless* of whether mechanical testing follows. Because the residual GP is conditioned only on tested parts, observing $\phi_{\mathrm{IR}}(x)$ without y changes predictions at other candidates only through the ϕ_{IR} predictor of §4.2, and at x through its own features. Its weight is $\delta = s_{\mathrm{IR}}\,(1 - \rho^2_{\mathrm{pred}})$, the spectral share $s_{\mathrm{IR}} = \mathrm{tr}(K_{\mathrm{IR}})/\mathrm{tr}(K)$ of the IR component times the share of a candidate's ϕ_{IR} that is not already predictable from x before printing. An observation that is fully predictable in advance has no acquisition value, so $\delta \to 0$ as $\rho^2_{\mathrm{pred}} \to 1$. This redundancy differs from the informativeness ρ^2_y of ϕ_{IR} about y, and Table 2 in Appendix B separates the two.

Inner / mechanical-test acquisition. After observing $\phi_{\mathrm{IR}}(x_t)$, $a_{\mathrm{mech}}(x_t \mid \phi_{\mathrm{IR}}) = \alpha' \mathrm{Perf}(x_t \mid \phi_{\mathrm{IR}}) + \beta' \mathrm{Unc}(x_t \mid \phi_{\mathrm{IR}}) + \gamma' \mathrm{CalVal}(x_t \mid \phi_{\mathrm{IR}})$, with all three terms now conditioned on the observed ϕ_{IR}; mechanical testing proceeds only if a_{mech} exceeds the adaptive budget-fraction quantile threshold of Appendix D, the $1 - \rho_t$ quantile with ρ_t the ratio of tests left to prints left.

The distribution over which this quantile is taken decides what the threshold does, and Appendix N proves what each rule below does.

Proposition 4, informal. Read against the part's own calibrated predictive law of ϕ_{IR}, the threshold tests every printed part with probability ρ_t whatever its design, so it rewards lucky parts and can discard the best one.

We therefore derive a *value-of-testing* rule from the two budgets. It tests the part when the expected best tested value after testing it is at least the value of keeping the test for the parts the print rule would print next, both from a finite-horizon recursion over the tests and prints left.

Proposition 5, informal. The value-of-testing rule tests a part exactly when its predictive mean clears a threshold set by the incumbent and by the tests and prints left, and that threshold does not rise as the part's predictive spread grows.

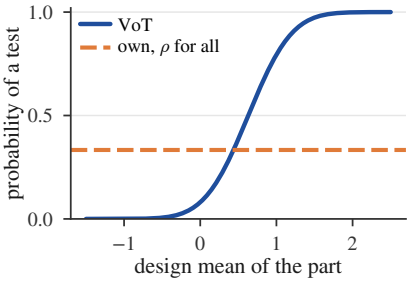

Figure 2: **Who gets tested.** Probability that a printed part is tested against its design mean, under the model of Figure 16 in Appendix N.

Weights. A first-order look-ahead expansion of the expected downstream utility $U(\pi) = \mathbb{E}[\max_{x \in \mathcal{D}_T^{\mathrm{full}}} y(x) \mid \pi]$ motivates the seven weights $(\alpha, \beta, \gamma, \delta, \alpha', \beta', \gamma')$ (Appendix B). The performance and uncertainty weights take expected-improvement-shaped defaults, γ and γ' are fixed calibration constants, and a variance argument motivates the heuristic closed form for δ. On the generator the derived value is $\delta = 0.44 \pm 0.05$, all runs use $\delta = 0.5$, inside the flat basin $[0.3, 0.6]$ of the sweep of Appendix F, and uniform weights cost 1.5 points of regret (A4).

Closed-loop controller. Algorithm 1 in Appendix A gives the round-by-round procedure, with fantasy observations for batches (Wilson et al., 2018), a linear cold-start ramp of γ and δ over the first three rounds, and an optional stopping rule in the spirit of Wilson (2024).

4.4 Theoretical Properties

Proposition 1 applies the GP-UCB argument (Srinivas et al., 2010) with an additively approximate maximizer, as in Kandasamy et al. (2015, Thm. 5), to SCOUT's outer stage, whose CalVal and IRInfo terms enter only through a slack ϵ_t. Appendix C states it formally and gives the proof.

Proposition 1, informal. Under the standard GP-UCB conditions on the residual kernel, if every tested design comes within a slack ϵ_t of the largest upper confidence bound, the simple regret after T tests is at most the GP-UCB rate $\mathcal{O}(\sqrt{\gamma_T \log T / T})$ plus the average slack $\frac{1}{T}\sum_{t \le T} \epsilon_t$.

Covered policies. The guarantee is stated for UCB-safeguarded SCOUT, which restricts the outer-stage choice to candidates that satisfy (4) with a prescribed ϵ_t. The bound excludes the test decision and is not numerically informative at our budgets, Appendix C. With Perf and Unc instantiated

as UCB, unrestricted SCOUT satisfies (4) with the realized range of $\gamma\nu + \delta\iota$ over the pool as a data-dependent slack, for which we claim no decay rate. The generator runs of Table 1 use the expected-improvement-shaped defaults, where this range averages 13% of the $\beta_t\sigma_t$ term and never exceeds 34%, and the safeguarded variant does not differ significantly from them at any budget, Appendix C.

5 EXPERIMENTS

SCOUT is evaluated on two complementary testbeds. On a literature-anchored generator of Nickel Aluminum Bronze (NAB / C95800) printing all three oracles exist, and the full method runs against thirteen baselines and against single-stage rules on its own IR-equipped surrogate, §5.1 and §5.2. Public battery and concrete data have a real free observation but no simulator, and a pre-registered study on them tests which printed part should receive the test, §5.3 and §5.4.

5.1 RESULTS ON THE NAB GENERATOR

Setup. The design $x \in \mathbb{R}^5$ comprises laser power, scan speed, hatch spacing, lattice strut diameter, and unit-cell size. The generator supplies 120 labelled tuples $(x, s + \phi_{\mathrm{FEM}}, \phi_{\mathrm{IR}}, y)$, split 80/20/20 into training, held-out, and reserve sets, and acquisition runs over a 200-candidate Latin-hypercube pool with 12 seeds at $(B_{\mathrm{print}}, B_{\mathrm{mech}}) \in \{(20, 12), (30, 18), (50, 30)\}$. The thirteen baselines of Appendix E are eleven methods with no GP or a GP on x only, among them random and simulator-only selection, and two budget-matched design-of-experiments baselines of current practice (Box & Wilson, 1951; Myers et al., 2016), a static D-optimal design with a ridge quadratic response surface and a sequential 2^{5-2} fractional factorial with steepest ascent. Only SCOUT decides, after observing ϕ_{IR}, whether a printed part is tested, here with the budget-fraction threshold of Appendix D, and SCOUT-IR-blind sets $\phi_{\mathrm{IR}} \equiv 0$. Apart from SCOUT with the additive kernel, all IR-equipped arms, ablations, and sweeps use the concatenated-input kernel, so each comparison is made on an identical surrogate.

Generator and real data. Every y of §5.1 and §5.2 comes from a literature-anchored generator whose s–y correlation is about 0.20 and whose ϕ_{IR} pool medians are calibrated to one real Optris recording, on which the pipeline measures a median cooling time of 98.5 ms against 93 ms from conduction scaling and a median cooling rate at $565^{\circ}\mathrm{C}$ of 681 K/s, order-of-magnitude checks of two features on one print, Appendix J. A latent physical state, namely porosity and the κ_{III} fraction, is drawn from x, y is produced from the latents only, and ϕ_{IR} is emitted as noisy measurements of the same latents, so y never reads the observed ϕ_{IR} and the information ϕ_{IR} carries about y is bounded by measurement noise, Appendix K. Its response, like the model of §4.2, is the simulator output plus a residual, and it emits ϕ_{IR} as a noisy measurement of its κ_{III} latent, a link that §4.1 does not support for real NAB. Acquisition results on the generator are not evidence from a physical machine, and paired (ϕ_{IR}, y) measurements on real parts do not yet exist for NAB with radiometric IR, so §5.3 turns to public data.

SCOUT has the lowest regret at every budget. At $(20, 12)$ SCOUT with the additive kernel is $\mathbf{56}\%$ below the strongest learning-based baseline, EPIG, and with the concatenated-input kernel it wins at least 11 of 12 seeds against every learning-based baseline at $(20, 12)$ and at least 10 of 12 at larger budgets. The sequential DoE baseline beats BALD at $(50, 30)$, 5.2 ± 1.8 against 8.6 ± 2.3, and SCOUT attains $\mathbf{4}$–$\mathbf{5}\times$ lower regret than it at every budget, with $p < 10^{-5}$ and $d \geq 3.0$.

The margin holds on SCOUT's own surrogate. The baselines on the left model x only, so their margins mix a surrogate effect with an acquisition effect, which the right block separates. Giving EPIG the IR-equipped surrogate lowers its regret from 5.9 to 4.4 at $(20, 12)$, less than half of its gap to SCOUT. On the same surrogate SCOUT without the IRInfo term and the two-stage decision reaches 5.8, and restoring both brings it to 2.5, a $\mathbf{43}\%$ reduction against the strongest IR-equipped baseline, with Welch $p = 1.5 \times 10^{-4}$ from the tabulated values, Cohen's $d = 2.0$, and $10/12$ seeds won. Removing either component alone leaves regret above EPIG's 4.4, at 4.6 without IRInfo, within seed noise, and 5.4 without the inner stage. At $(50, 30)$ SCOUT and EPIG on the IR-equipped surrogate no longer differ at the Bonferroni threshold $0.05/13$, with $p = 0.020$, so SCOUT's margin over EPIG on the same surrogate narrows as B_{mech} grows, and at the two larger budgets the IR-equipped

Table 1: **SCOUT has the lowest regret on the NAB generator at every budget, also against single-stage rules on its own IR-equipped surrogate.** Simple regret in percent, mean $\pm$ standard deviation over 12 seeds. **Left** Thirteen baselines without a GP or with a GP on x. **Right** Single-stage rules on SCOUT's residual GP with the concatenated-input kernel, then SCOUT-IR-blind and SCOUT, whose two kernels are within seed noise, Table 7 in Appendix H. Gray cells mark the best baseline of each block. Welch tests of SCOUT with the additive kernel against each of the thirteen baselines, from the tabulated means and standard deviations and so unpaired, give $p < 10^{-3}$ at every budget, below the Bonferroni thresholds $0.05/13$ per budget and $0.05/39$ over all three, with Cohen's $d \geq 1.9$.

Baseline	$(20, 12)$	$(30, 18)$	$(50, 30)$
Random	24.4 ± 5.7	18.2 ± 5.0	13.8 ± 4.7
Sim-only	12.3 ± 2.5	10.6 ± 2.3	8.1 ± 2.2
Performance	11.6 ± 3.0	8.4 ± 2.4	6.8 ± 2.0
Uncertainty	11.0 ± 2.8	7.9 ± 2.5	6.4 ± 2.1
GP-UCB / EI	9.3 ± 2.5	6.1 ± 2.4	5.7 ± 2.1
BALD	11.5 ± 2.6	9.6 ± 2.5	8.6 ± 2.3
MF-MES	9.6 ± 3.3	5.7 ± 2.4	4.9 ± 2.1
qMFKG	7.5 ± 2.1	5.3 ± 2.3	4.3 ± 1.8
Targeted Materials BAX	8.2 ± 2.4	3.9 ± 1.5	3.6 ± 1.6
PS-BAX	6.9 ± 2.3	4.1 ± 1.8	3.5 ± 1.5
EPIG	5.9 ± 1.6	5.3 ± 2.3	3.1 ± 1.3
DoE, D-optimal + ridge RSM	12.8 ± 3.1	9.7 ± 2.6	6.9 ± 2.2
DoE, 2^{5-2} factorial + steepest ascent	10.4 ± 2.7	7.6 ± 2.3	5.2 ± 1.8

IR-equipped surrogate, SCOUT	$(20, 12)$	$(30, 18)$	$(50, 30)$
EI/UCB + IR	6.8 ± 1.9	4.3 ± 1.5	3.4 ± 1.2
SCOUT $-$ IRInfo $-$ two-stage	5.8 ± 1.4	3.9 ± 1.0	3.0 ± 0.9
EPIG + IR	4.4 ± 1.2	2.5 ± 0.8	1.6 ± 0.6
SCOUT-IR-blind, concatenated	5.7 ± 1.5	3.3 ± 0.9	2.7 ± 0.9
SCOUT, concatenated kernel	$\mathbf{2.5 \pm 0.6}$	$\mathbf{1.5 \pm 0.4}$	$\mathbf{1.1 \pm 0.3}$
SCOUT, additive kernel	$\mathbf{2.6 \pm 0.8}$	$\mathbf{1.6 \pm 0.4}$	$\mathbf{1.2 \pm 0.4}$

surrogate alone closes most of EPIG's gap to SCOUT. The comparison is between a two-stage policy and single-stage baselines, since no generator baseline has an inner stage, and on battery data, where EPIG receives the same test rule, SCOUT's margin over it does not survive Holm's correction, §5.3.

Where the gain comes from. SCOUT-IR-blind's mean regret is below that of all thirteen baselines at all three budgets, 5.7, 3.3, and 2.7 against best-baseline values of 5.9, 3.9, and 3.1, but not significantly, and within seed noise at $(20, 12)$. Adding ϕ_{IR} lowers regret from 5.7 to 2.5 at $(20, 12)$, a relative reduction between 54% and 60% at every budget. Of the ten ablations in Table 4 of Appendix F, removing the residual GP costs most, $+6.7$ points at $(20, 12)$, followed by the ϕ_{IR} features, $+3.3$, the CalVal term, $+3.2$, and the two-stage decision, $+2.9$, whereas the IRInfo term alone costs $+2.1$, and hot-spot density and the cooling time constant carry the strongest IR signal, a ranking set by how the generator couples ϕ_{IR} to y, Figure 11 in Appendix H.

5.2 Robustness of the Generator Results

Four tests vary the generator within its family, Appendices F and G, and Figure 9 plots the first three. **State without an IR signature.** When a latent channel with no ϕ_{IR} signature carries $\lambda \in \{0, 25, 50, 75\}\%$ of the residual variance, the gap between SCOUT-IR-blind and SCOUT falls from 3.2 to 2.6, 1.4, and 0.7 points, is not significant at $\lambda = 75\%$, the largest share run, with $p \approx 0.22$, and is never significantly negative. **IR noise from a real recording.** With every ϕ_{IR} noise constant raised at least to that feature's dispersion within the real recording, which raised five of the twelve, SCOUT reaches 2.9 ± 0.7 against 4.6 ± 1.2 for EPIG with the IR-equipped surrogate, a **37**% reduction, and SCOUT-IR-blind, which never consumes ϕ_{IR}, is unchanged at 5.7 ± 1.5. The test anchors the noise of ϕ_{IR}, not the signal it carries across recipes. **Simulator noise.** Zero-mean noise in s at half and at the full magnitude of the structural-bias term raises SCOUT's regret from 2.5 ± 0.6 to 2.8 ± 0.7 and 3.3 ± 0.9, while EPIG on x only, which never consumes s, stays at 5.9, 6.0, and 5.8. **A second alloy.** Re-instantiating the generator with Inconel-718 constants gives SCOUT 4.1 ± 0.7 against 6.7 ± 1.6 for EPIG on x only, the strongest of the arms run there, a 38.8% reduction at $(20, 12)$ that mixes surrogate and acquisition effects, while SCOUT-IR-blind does not beat EPIG, 7.0 against 6.7.

5.3 A Real Free Observation: Battery and Concrete Data

Data, loop, and arms. Severson et al. (2019) cycled 124 LFP/graphite cells to failure under fast-charging protocols. Cycling a cell for 100 cycles is the print, six early-cycle features that predict log cycle life with a cross-validated R^2 of 0.88 are the free observation, and running the cell to

end of life is the test. In the concrete data of Yeh (1998a) a mixture is the print, its 3-day strength the free observation, and its 28-day strength the test. Neither data set has a simulator, so $s \equiv 0$, and every arm runs in a second, open implementation written from this paper, Appendix L, whose numbers are not comparable with Table 1. The 21 arms pair print rules on the design alone or on the IR-equipped surrogate, among them EPIG, the knowledge gradient through the two-node network, EI-FN (Astudillo & Frazier, 2021a), and SCOUT, with test rules, among them the budget-fraction threshold in three readings, the value-of-testing rule, and deferred testing, next to an early-prediction loop in the style of Attia et al. (2020). The metric is the share of the gap between the pool median and the pool best part left after the campaign, in percent, 0 when the pool's best part was tested.

Pre-registration. A first round on fixed settings located the failure that Proposition 4 explains, Appendix L, the value-of-testing rule was designed after it, and a second round was fixed in a time-stamped file before it ran, Appendix O. It draws 30 random instances in each of five families: *grouped* battery instances, whose labelled history holds whole protocols absent from the pool, *cell* instances with a random labelled third of the cells, *no-history* instances, and concrete with and without a labelled third. Holm's correction covers 27 confirmatory comparisons on instance-paired differences, the seventeen battery and ten concrete tests separately, a split made after two battery families had been seen, under which 3 and 2 are significant, and 3 under one correction over all 27.

The new rule removes the old threshold's harm. With labelled history the own-prediction threshold discarded the pool's best cell in 39 of the 53 battery runs that printed it, and the value-of-testing rule in 0 of 55, and over all three battery families the pre-registered comparison gives 53 of 70 against 6 of 69, Fisher $p = 1.3 \times 10^{-16}$ on pooled runs, and exact McNemar tests paired by instance give $p = 3.1 \times 10^{-5}$ and 2.4×10^{-7} in the grouped and cell families. **The new rule lowers SCOUT's regret by 8.6 points in grouped and 12.8 in cell instances**, Holm-adjusted $p = 0.0031$ and 4.6×10^{-4}, and gives SCOUT the lowest mean regret of the 21 arms in both families, tied at zero with two other arms in cell instances. It is not significantly better than testing every print, 1.4 against 2.8 in grouped instances with Holm $p = 1.00$, so it removes the harm of the old threshold rather than adding a gain of its own. Under the same test rule SCOUT's print rule leads EPIG, 1.4 against 5.6 and 0.0 against 4.8, and the knowledge gradient through the network, but neither margin survives the correction, and EI on the same surrogate matches it on battery data, $p \geq 0.49$, and beats it on concrete with history, $p = 0.021$, both uncorrected, so these data do not separate SCOUT's print rule from EI's. SCOUT with the new rule beats the early-prediction loop in grouped instances, 1.4 against 13.2, Holm $p = 0.033$ within the battery tests and 0.053 over all 27, deferred testing is never significantly better than deciding at once, and at the protocol level the new rule removes SCOUT's loss, 0.3 against 8.9 and 0.0 against 10.6 points.

5.4 The Test-Rule Effect Comes from Five Merged Cells

In most battery instances the pool's best cell is one of five *merged* cells, b1c0 to b1c4, whose recorded life joins a batch-2 continuation record in the official processing of Severson et al. (2019). Their early features come from the first record, and for b1c2, the pool's best cell in most instances, a least-squares fit on them underpredicts the joined life by 45%, the case in which Proposition 4 lets the own-prediction reading discard the best part, which it did in 53 of the 62 runs in which the pool's best cell was one of the five and in 0 of the 8 other runs, Appendix O. In exploratory runs without all five cells, added after the review of the pre-registered round, the two rules reach 17.7 and 17.9 in grouped instances with $p = 0.95$, and 3.6 and 2.1 in cell instances with $p = 0.53$, Figure 3d. **What remains with history is the value of the free observation.** In these runs every IR-equipped arm stays below GP-UCB-EI on the design alone, which reaches 43.1 and 21.2 against 17.9 and 2.1 for SCOUT with the new rule, with uncorrected $p = 0.003$ and 4.4×10^{-4}, although at the protocol level the gap is smaller and not significant.

Where the free observation does not pay back. Without history on battery data, GP-UCB-EI and EI on the protocol variables alone reach 14.1 and 16.3, below every IR-equipped arm, plausibly because three random tests are too few to learn how early features relate to life on nine inputs. On concrete no confirmatory comparison is significant with history, and without history SCOUT with the new rule beats EPIG with the same rule and the early-prediction loop, 3.4 against 9.9 and 12.1, Holm $p = 0.0058$ and 0.023 within the concrete tests and 0.014 and 0.058 over all 27.

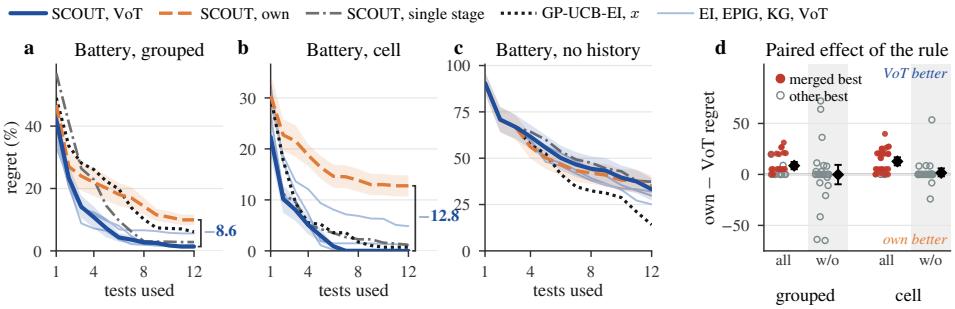

Figure 3: **The value-of-testing rule removes the loss of the own-prediction threshold, and the loss comes from the merged cells.** Pre-registered battery round of §5.3, 30 instances per family at $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$, with the names and colours of Figure 1, where the single stage tests every print and thin light-blue lines are EI, EPIG, and KG on the IR-equipped surrogate with the value-of-testing rule. **a–c** Regret after each test, in the order the tests were made, as a mean over instances, with one standard error for the two SCOUT arms. The end points are the entries of Table 16 and, for EI + IR, of Table 23. Brackets mark the drop in final regret from own to VoT, the confirmatory comparison C1. **d** Instance-paired difference in final regret between the two test rules for SCOUT, with all cells and without the five merged cells of §5.4. The shaded columns without them are exploratory runs added after the review. Filled red points are instances whose pool's best cell is a merged cell, hollow points the others, and the diamond is the mean with a 95% bootstrap interval.

6 Discussion

When the free observation pays back. On the generator SCOUT stays ahead of EPIG under IR noise raised to the dispersion of a real recording and under simulator noise as large as the structural bias, and its gain over the IR-blind variant never falls significantly below zero. **The failure of the own-prediction reading is real but needs a best part whose free observation is low against its own prediction**, which in the pre-registered battery families only the five merged cells supplied, and with labelled history the value-of-testing rule keeps such a part and costs nothing measurable in cell-level regret when none is present.

Scope and limitations. *Evidence.* The AM results are evidence about the generator, the real recording validates the ϕ_{IR} pipeline only, and the battery and concrete evaluations use real observations but other domains and no simulator. The effect of the test rule on battery data depends on five cells, the runs without them were added after the review of the pre-registered round, and the instances of each family are partitions of the same 124 cells or 122 designs. *Comparisons.* Function-network arms and the value-of-testing rule run only in the open harness, not on the generator, and on the function networks of Buathong et al. (2024) SCOUT does not separate from most IR-equipped arms, Appendix L. Arms in the spirit of p-KGFN and suspension relative to the incumbent were added after that review, and multi-stage BO (Torresi & Friederich, 2026) is not run. *AM scope.* The recorded process and the source parameters of Appendix I are at the scale of laser cladding and directed energy deposition, whereas the layer, powder, and generator anchors follow LPBF practice. Process parameters are held constant within a build, which excludes heat-input compensation between near-substrate and mid-build regions, and SCOUT addresses recipe selection between builds, not within-build control. Augmenting ϕ_{IR} with layer-height context is future work, not a claimed capability. The learned residual absorbs machine-specific error, so another machine, even a nominally identical one, requires re-estimation rather than zero-shot reuse, and the simulator is assumed to be calibrated before the campaign by single-track melt-pool matching.

Conclusion. SCOUT combines a biased simulator, a free in-situ observation, and a scarce destructive test in one two-stage acquisition, and on a literature-anchored NAB generator it attains 43% lower regret at the smallest budget than the strongest baseline on its own IR-equipped surrogate. Which printed part receives the test should be decided against what the remaining prints offer, as the value-of-testing rule and the simpler pool reading both do, and not against the part's own prediction, which on public battery data mattered when the pool's best cell was one of five merged cells.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing the text. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the selection of process recipes for metal additive manufacturing. It involves no human subjects, personal data, or crowdsourced labor. The AM acquisition results are obtained on a literature-anchored generator, and the paper states this boundary in the abstract, in §5, and in §6, so that the reported regret improvements are not read as evidence from a physical campaign. The battery and concrete results use data that their authors released publicly, and no new cells or specimens were tested. We see no direct negative societal impact of data-efficient recipe selection. Its intended benefit is fewer destructive tests and less material waste.

REPRODUCIBILITY STATEMENT

Appendix K documents the provenance of the NAB generator, its generative graph, the result files listed there, and the commands that produce them, and Appendix E lists the hyperparameters, acquisition rules, and fairness controls of all generator baselines. Figures 6, 7, 8, and 9 and panel c of Figure 1 are plotted from values reported in the text. Appendix L documents the open harness, the battery and concrete data, the settings, and the arms, and Appendix O the pre-registered round and every run added after it. Appendix J gives the pseudocode of the deterministic IR feature pipeline, Appendix D validates the three approximations of the acquisition, and Appendix C states the assumptions and the full proof of Proposition 1. The codebase, the data-loader interface, and the generator will be released. The open harness, the time-stamped pre-registration file, the scripts that rebuild the battery table from the public raw files of Severson et al. (2019) and the concrete table from those of Yeh (1998b), and the per-run result files will be released as well, and the released scripts regenerate every number, table, and figure of §5.3 to §5.4 and Appendices L and O from the rebuilt tables and those files. The result files come from harness versions that analyze.py treats as one because re-running samples of the older rows reproduced them exactly, except for the superseded EPIG and knowledge-gradient rows that Appendix L describes.

REFERENCES

Adedire D. Adesiji, Jiashuo Wang, Cheng-Shu Kuo, and Keith A. Brown. Benchmarking self-driving labs. *Digital Discovery*, 5(1):14–27, 2026. doi: 10.1039/D5DD00337G.

N. Ahmed, I. Barsoum, G. Haidemenopoulos, and R. K. Abu Al-Rub. Process parameter selection and optimization of laser powder bed fusion for 316L stainless steel: a review. *Journal of Manufacturing Processes*, 75:415–434, 2022. doi: 10.1016/j.jmapro.2021.12.064.

Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauza, Leslie Pack Kaelbling, Joshua B. Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: case study of planar pushing and bouncing. In *IROS*, 2018.

Georgios Amanatidis, Ben Berger, Tomer Ezra, Michal Feldman, Federico Fusco, Rebecca Reiffenhäuser, and Artem Tsikiridis. Pandora's box problem with time constraints. *Artificial Intelligence*, 349:104426, 2025. doi: 10.1016/j.artint.2025.104426.

Ali Aouad, Jingwei Ji, and Yaron Shaposhnik. The Pandora's box problem with sequential inspections. *Operations Research*, 74(4):2179–2195, July 2026. doi: 10.1287/opre.2024.0733.

Raul Astudillo and Peter I. Frazier. Bayesian optimization of function networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 14463–14475. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/792c7b5aae4a79e78aaeda80516ae2ac-Abstract.html.

Raul Astudillo and Peter I. Frazier. Thinking inside the box: A tutorial on grey-box Bayesian optimization. In *Proceedings of the 2021 Winter Simulation Conference (WSC)*, pp. 1–15, Phoenix, AZ, USA, 2021b. IEEE. doi: 10.1109/WSC52266.2021.9715343.

Raul Astudillo, Daniel R. Jiang, Maximilian Balandat, Eytan Bakshy, and Peter I. Frazier. Multi-step budgeted Bayesian optimization with unknown evaluation costs. In *NeurIPS*, 2021.

Peter M. Attia, Aditya Grover, Norman Jin, Kristen A. Severson, Todor M. Markov, Yang-Hung Liao, Michael H. Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, Patrick K. Herring, Muratahan Aykol, Stephen J. Harris, Richard D. Braatz, Stefano Ermon, and William C. Chueh. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature*, 578:397–402, 2020. doi: 10.1038/s41586-020-1994-5.

Maria J. Bayarri, James O. Berger, Rui Paulo, Jerry Sacks, John A. Cafeo, James Cavendish, Chin-Hsu Lin, and Jian Tu. A framework for validation of computer models. *Technometrics*, 49(2): 138–154, 2007. doi: 10.1198/004017007000000092.

Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented Bayesian active learning. In *AISTATS*, 2023.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023.

G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(1):1–38, 1951. doi: 10.1111/j.2517-6161.1951.tb00067.x.

Jenný Brynjarsdóttir and Anthony O'Hagan. Learning about physical parameters: the importance of model discrepancy. *Inverse Problems*, 30(11):114007, 2014. doi: 10.1088/0266-5611/30/11/114007.

Poompol Buathong, Jiayue Wan, Raul Astudillo, Sam Daulton, Maximilian Balandat, and Peter I. Frazier. Bayesian optimization of function networks with partial evaluations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4752–4784. PMLR, Jul 2024. URL https://proceedings.mlr.press/v235/buathong24a.html.

Chu Xin Cheng, Raul Astudillo, Thomas Desautels, and Yisong Yue. Practical Bayesian algorithm execution via posterior sampling. In *NeurIPS*, 2024.

Dmitry Chernyavsky, Denys Y. Kononenko, Julia Kristin Hufenbach, Jeroen van den Brink, and Konrad Kosiba. Bayesian optimization for laser powder bed fusion of defect-free AA2024. *Additive Manufacturing*, 114:105022, September 2025. doi: 10.1016/j.addma.2025.105022.

Sathya Chitturi, Akash Ramdas, Yue Wu, Brian Rohr, Stefano Ermon, Jennifer Dionne, Felipe H. da Jornada, Mike Dunne, Christopher Tassone, Willie Neiswanger, and Daniel Ratner. Targeted materials discovery using Bayesian algorithm execution. *npj Computational Materials*, 2024.

Dong Heon Cho and Boyuan Chen. RAFL: generalizable sim-to-real of soft robots with residual acceleration field learning, 2026. arXiv:2603.22039.

Zhongxiang Dai, Haibin Yu, Bryan Kian Hsiang Low, and Patrick Jaillet. Bayesian optimization meets Bayesian optimal stopping. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1496–1506. PMLR, June 2019. URL https://proceedings.mlr.press/v97/dai19a.html.

A. P. Dawid. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2): 278–292, 1984. doi: 10.2307/2981683.

James R. Deneault, Jorge Chang, Jay Myung, Daylond Hooper, Andrew Armstrong, Mark Pitt, and Benji Maruyama. Toward autonomous additive manufacturing: Bayesian optimization on a 3D printer. *MRS Bulletin*, 46:566–575, 2021.

Francesco Di Fiore and Laura Mainini. Physics-aware multifidelity Bayesian optimization: a generalized formulation. *Computers & Structures*, 296:107302, 2024.

Donghong Ding, Zengxi Pan, Stephen van Duin, Huijun Li, and Chen Shen. Fabricating superior NiAl bronze components through wire arc additive manufacturing. *Materials*, 9(8):652, 2016. doi: 10.3390/ma9080652.

Qianggang Ding, Santiago Miret, and Bang Liu. MatExpert: decomposing materials discovery by mimicking human experts. In *ICLR*, 2025.

Hossein Esfandiari, MohammadTaghi HajiAghayi, Brendan Lucier, and Michael Mitzenmacher. Online Pandora's boxes and bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1885–1892, 2019. doi: 10.1609/aaai.v33i01.33011885.

Azza Fadhel, Nathaniel W. Zuckschwerdt, Aryan Deshwal, Susmita Bose, Amit Bandyopadhyay, and Jana Doppa. Discovery of feasible 3D printing configurations for metal alloys via AI-driven adaptive experimental design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 39939–39948, 2026. doi: 10.1609/aaai.v40i47.41428.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1437–1446. PMLR, 2018.

Mingzhou Fan, Byung-Jun Yoon, Edward Dougherty, Nathan Urban, Francis Alexander, Raymundo Arróyave, and Xiaoning Qian. Multi-fidelity Bayesian optimization with multiple information sources of input-dependent fidelity. In Negar Kiyavash and Joris M. Mooij (eds.), *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pp. 1271–1293. PMLR, July 2024. URL https://proceedings.mlr.press/v244/fan24a.html.

Peter I. Frazier. Bayesian optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*, INFORMS TutORials in Operations Research, pp. 255–278. INFORMS, 2018. doi: 10.1287/educ.2018.0188.

Junpeng Gao, Mike Y. Michelis, Andrew Spielberg, and Robert K. Katzschmann. Sim-to-real of soft robots with learned residual physics. *IEEE Robotics and Automation Letters*, 9(10):8523–8530, 2024. doi: 10.1109/LRA.2024.3446287.

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

Andrea Gatto, Maria Laura Gatto, Riccardo Groppo, Daniel Munteanu, and Paolo Mengucci. Influence of laser powder bed fusion process parameters on the properties of CuZn42 components: case study of the laser surface energy density. *Progress in Additive Manufacturing*, 8(5):843–855, 2023. doi: 10.1007/s40964-022-00361-z.

Mark Girolami, Eky Febrianto, Ge Yin, and Fehmi Cirak. The statistical finite element method (statFEM) for coherent synthesis of observation data and model predictions. *Computer Methods in Applied Mechanics and Engineering*, 375:113533, 2021.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2): 243–268, 2007. doi: 10.1111/j.1467-9868.2007.00587.x.

Aldair E. Gongora, Bowen Xu, Wyatt Perry, Chika Okoye, Patrick Riley, Kristofer G. Reyes, Elise F. Morgan, and Keith A. Brown. A Bayesian experimental autonomous researcher for mechanical design. *Science Advances*, 6(15):eaaz1708, 2020. doi: 10.1126/sciadv.aaz1708.

Aldair E. Gongora, Kelsey L. Snapp, Emily Whiting, Patrick Riley, Kristofer G. Reyes, Elise F. Morgan, and Keith A. Brown. Using simulation to accelerate autonomous experimentation: A case study using mechanics. *iScience*, 24(4):102262, April 2021. doi: 10.1016/j.isci.2021.102262.

M. Grasso, A. G. Demir, B. Previtali, and B. M. Colosimo. In situ monitoring of selective laser melting of zinc powder via infrared imaging of the process plume. *Robotics and Computer-Integrated Manufacturing*, 49:229–239, 2018. doi: 10.1016/j.rcim.2017.07.001.

David Guirguis, Conrad Tucker, and Jack Beuth. Accelerating process development for 3D printing of new metal alloys. *Nature Communications*, 15(1):582, 2024. doi: 10.1038/s41467-024-44783-5.

Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In Michel Goemans and José Correa (eds.), *Integer Programming and Combinatorial Optimization (IPCO 2013)*, volume 7801 of *Lecture Notes in Computer Science*, pp. 205–216, Berlin, Heidelberg, 2013. Springer. doi: 10.1007/978-3-642-36694-9_18.

Gus W. Haggstrom. Optimal sequential procedures when more than one stop is required. *The Annals of Mathematical Statistics*, 38(6):1618–1626, 1967. doi: 10.1214/aoms/1177698595.

Sumner B. Harris, Rama Vasudevan, and Yongtao Liu. Active oversight and quality control in standard Bayesian optimization for autonomous experiments. *npj Computational Materials*, 11:23, 2025. doi: 10.1038/s41524-024-01485-2.

Dave Higdon, Marc Kennedy, James C. Cavendish, John A. Cafeo, and Robert D. Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.

Shawn Hinnebusch, David Anderson, Berkay Bostan, and Albert C. To. In-situ infrared camera monitoring for defect and anomaly detection in laser powder bed fusion: calibration, data mapping, and feature extraction. *Applied Sciences*, 16(7):3378, 2026. doi: 10.3390/app16073378.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011. arXiv:1112.5745.

Daolang Huang, Yujia Guo, Luigi Acerbi, and Samuel Kaski. Amortized Bayesian experimental design for decision-making. In *NeurIPS*, 2024.

Jonas Hübotter, Bhavya Sukhija, Lenart Treven, Yarden As, and Andreas Krause. Transductive active learning: Theory and applications. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 124686–124755. Curran Associates, Inc., 2024. doi: 10.52202/079017-3961. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e17fe6fe9990fffb637b42c98c005515-Abstract-Conference.html.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. DiscoveryWorld: a virtual environment for developing and evaluating automated scientific discovery agents. In *NeurIPS Datasets and Benchmarks*, volume 37, pp. 10088–10116, 2024. doi: 10.52202/079017-0324.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 295–304, Lille, France, July 2015. PMLR. URL https://proceedings.mlr.press/v37/kandasamy15.html.

Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Bayesian optimisation with continuous approximations. In *ICML*, 2017.

Barış Kavas, Efe C. Balta, Michael R. Tucker, Raamadaas Krishnadas, Alisa Rupenyan, John Lygeros, and Markus Bambach. In-situ controller autotuning by Bayesian optimization for closed-loop feedback control of laser powder bed fusion process. *Additive Manufacturing*, 99:104641, 2025. doi: 10.1016/j.addma.2025.104641.

Marc C. Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464, 2001.

Hwanwoo Kim, Chong Liu, and Yuxin Chen. Bayesian optimization with inexact acquisition: Is random grid search sufficient? In Silvia Chiappa and Sara Magliacane (eds.), *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 2202–2222. PMLR, July 2025. URL https://proceedings.mlr.press/v286/kim25b.html.

Saksham Kiroriwal, Julius Pfrommer, and Jürgen Beyerer. Partially observable Gaussian process network and doubly stochastic variational inference, 2025. URL https://arxiv.org/abs/2502.13905. arXiv preprint arXiv:2502.13905.

Saksham Kiroriwal, Julius Pfrommer, and Jürgen Beyerer. Joint parameter and state-space Bayesian optimization: Using process expertise to accelerate manufacturing optimization. *Procedia CIRP*, 146:565–570, 2026. doi: 10.1016/j.procir.2026.03.201. Also available as arXiv:2602.17679.

Mergim Krasniqi and Frank Löffler. Comprehensive study on statistical methods for optimization of process parameters and material properties of AlSi10Mg in laser powder bed fusion. *Discover Mechanical Engineering*, 3:42, 2024. doi: 10.1007/s44245-024-00073-4.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at LLMs for material discovery: Are they actually good for Bayesian optimization over molecules? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25603–25622. PMLR, 2024. URL https://proceedings.mlr.press/v235/kristiadi24a.html.

Shunya Kusakawa, Shion Takeno, Yu Inatsu, Kentaro Kutsukake, Shogo Iwazaki, Takashi Nakano, Toru Ujihara, Masayuki Karasuyama, and Ichiro Takeuchi. Bayesian optimization for Cascade-type multistage processes. *Neural Computation*, 34(12):2408–2431, 2022. doi: 10.1162/neco_a_01550.

Simon Lacoste-Julien, Ferenc Huszár, and Zoubin Ghahramani. Approximate inference for the loss-calibrated Bayesian. In *AISTATS*, 2011.

Brandon Lane, Shawn Moylan, Eric P. Whitenton, and Li Ma. Thermographic measurements of the commercial laser powder bed fusion process at NIST. *Rapid Prototyping Journal*, 22(5):778–787, 2016. doi: 10.1108/RPJ-11-2015-0161.

Dong Bok Lee, Aoxuan Zhang, Byungjoo Kim, Junhyeon Park, Steven Adriaensen, Juho Lee, Sung Ju Hwang, and Hae Beom Lee. Cost-sensitive freeze-thaw Bayesian optimization for efficient hyperparameter tuning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 169531–169564, 2025a.

Jeong Ah Lee, Jaejung Park, Man Jae Sagong, Soung Yeoul Ahn, Jung-Wook Cho, Seungchul Lee, and Hyoung Seop Kim. Active learning framework to optimize process parameters for additive-manufactured Ti-6Al-4V with high strength and ductility. *Nature Communications*, 16(1):931, 2025b. doi: 10.1038/s41467-025-56267-1.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *The Twelfth International Conference on Learning Representations*, volume 2024, pp. 31252–31284, 2024. URL https://openreview.net/forum?id=OOxotBmGol. Proceedings version: https://proceedings.iclr.cc/paper_files/paper/2024/hash/84b8d9fcb4e262fcd429544697e1e720-Abstract-Conference.html.

Shreshth A. Malik, Tiarnan Doherty, Panagiotis Tigas, Muhammed Razzak, Aron Walsh, and Yarin Gal. Towards dynamic benchmarks for autonomous materials discovery, 2025. AI for Accelerated Materials Design (AI4Mat) workshop at NeurIPS 2025, https://openreview.net/forum?id=Cfj7uBu5dy.

Arjun Mani, Carl Vondrick, and Richard Zemel. Few-shot design optimization by exploiting auxiliary information. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oQKiPx8jlI.

Natalie Maus, Kyurae Kim, Geoff Pleiss, David Eriksson, John P. Cunningham, and Jacob R. Gardner. Approximation-aware Bayesian optimization. In *NeurIPS*, 2024.

Pawel Mieszczanek, Peter Corke, Courosh Mehanian, Paul D. Dalton, and Dietmar W. Hutmacher. Towards industry-ready additive manufacturing: AI-enabled closed-loop control for 3D melt electrowriting. *Communications Engineering*, 3:158, 2024. doi: 10.1038/s44172-024-00302-4.

Petrus Mikkola, Julien Martinelli, Louis Filstroff, and Samuel Kaski. Multi-fidelity Bayesian optimization with unreliable information sources. In *AISTATS*, 2023.

Shunya Minami, Yoshihiro Hayashi, Stephen Wu, Kenji Fukumizu, Hiroki Sugisawa, Masashi Ishii, Isao Kuwajima, Kazuya Shiratori, and Ryo Yoshida. Scaling law of Sim2Real transfer learning in expanding computational materials databases for real-world predictions. *npj Computational Materials*, 11:146, 2025. doi: 10.1038/s41524-025-01606-5.

Jonas Močkus, Vytautas Tiešis, and Antanas Žilinskas. The application of Bayesian methods for seeking the extremum. In L. C. W. Dixon and G. P. Szegö (eds.), *Towards Global Optimisation 2*, pp. 117–129. North-Holland, 1978.

Gunther Mohr, Simon J. Altenburg, Alexander Ulbricht, Philipp Heinrich, Daniel Baum, Christiane Maierhofer, and Kai Hilgenberg. In-situ defect detection in laser powder bed fusion by using thermography and optical tomography—comparison to computed tomography. *Metals*, 10(1):103, 2020. doi: 10.3390/met10010103.

Mandaná Moshiri, David Bue Pedersen, Guido Tosello, and Venkata Karthik Nadimpalli. Performance evaluation of in-situ near-infrared melt pool monitoring during laser powder bed fusion. *Virtual and Physical Prototyping*, 18(1):e2205387, 2023. doi: 10.1080/17452759.2023.2205387.

Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25444–25470. PMLR, July 2023. URL https://proceedings.mlr.press/v202/muller23a.html.

Raymond H. Myers, Douglas C. Montgomery, and Christine M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, 4 edition, 2016. ISBN 978-1-118-91601-8.

Willie Neiswanger, Ke Alexander Wang, and Stefano Ermon. Bayesian algorithm execution: estimating computable properties of black-box functions using mutual information. In *ICML*, 2021.

Willie Neiswanger, Lantao Yu, Shengjia Zhao, Chenlin Meng, and Stefano Ermon. Generalizing Bayesian optimization with decision-theoretic entropies. In *NeurIPS*, 2022.

Rafael Oliveira, Dino Sejdinovic, David Howard, and Edwin V. Bonilla. Bayesian adaptive calibration and optimal design. In *NeurIPS*, 2024.

Ryan W. Penny, Zachery Kutschke, and A. John Hart. High-fidelity optical monitoring of laser powder bed fusion via aperture division multiplexing. *npj Advanced Manufacturing*, 2:28, 2025. doi: 10.1038/s44334-025-00039-8.

Paris Perdikaris, Maziar Raissi, Andreas Damianou, Neil D. Lawrence, and George Em Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A*, 473(2198):20160751, 2017.

Matthew Plumlee. Bayesian calibration of inexact computer models. *Journal of the American Statistical Association*, 112(519):1274–1285, 2017.

Matthias Poloczek, Jialei Wang, and Peter Frazier. Multi-information source optimization. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/df1f1d20ee86704251795841e6a9405a-Paper.pdf.

Norbert Ponweiser, Christian L. Lengauer, and Klaus W. Richter. Re-investigation of phase equilibria in the system Al–Cu and structural analysis of the high-temperature phase η_1-$Al_{1-\delta}Cu$. *Intermetallics*, 19(11):1737–1746, 2011. doi: 10.1016/j.intermet.2011.07.007.

Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024. doi: 10.1214/23-STS915.

Herilalaina Rakotoarison, Steven Adriaensen, Neeratyoy Mallik, Samir Garibov, Eddie Bergman, and Frank Hutter. In-context freeze-thaw Bayesian optimization for hyperparameter optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41982–42008. PMLR, 2024.

Víctor Sabanza-Gil, Riccardo Barbano, Daniel Pacheco Gutiérrez, Jeremy S. Luterbacher, José Miguel Hernández-Lobato, Philippe Schwaller, and Loïc Roch. Best practices for multi-fidelity Bayesian optimization in materials and molecular research. *Nature Computational Science*, 5(7):572–581, 2025. doi: 10.1038/s43588-025-00822-9.

Yoshiki Sakai, Kota Aono, Takayuki Osa, Moju Zhao, Hirotoshi Tagata, Masanori Nakano, Akio Suguro, Masayuki Nakao, and Keisuke Nagato. Autonomous parameter exploration in thermoplastic material-extrusion additive manufacturing using Bayesian optimization with intermediate features. *Journal of Manufacturing Processes*, 157:798–807, 2026. doi: 10.1016/j.jmapro.2025.12.021.

Luke Scime, Chase Joslin, David Collins, William Halsey, Ryan Duncan, and Vincent Paquit. A co-registered in-situ and ex-situ tensile properties dataset from a laser powder bed fusion additive manufacturing process (Peregrine v2023-11). Dataset, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN, USA, September 2023. URL https://doi.org/10.13139/ORNLNCCS/2001425. OSTI ID 2001425.

Danny Segev and Sahil Singla. Efficient approximation schemes for stochastic probing and selection-stopping problems. *Mathematics of Operations Research*, 2025. doi: 10.1287/moor.2023.0242. Articles in Advance.

Burr Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2012. doi: 10.1007/978-3-031-01560-1.

Kristen A. Severson, Peter M. Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H. Chen, Muratahan Aykol, Patrick K. Herring, Dimitrios Fraggedakis, Martin Z. Bazant, Stephen J. Harris, William C. Chueh, and Richard D. Braatz. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4:383–391, 2019. doi: 10.1038/s41560-019-0356-8.

Jialin Song, Yuxin Chen, and Yisong Yue. A general framework for multi-fidelity Bayesian optimization with Gaussian processes. In *AISTATS*, 2019.

João Sousa, Armando Sousa, Frank Brueckner, Luís Paulo Reis, and Ana Reis. Human-in-the-loop multi-objective Bayesian optimization for Directed Energy Deposition with in-situ monitoring. *Robotics and Computer-Integrated Manufacturing*, 92:102892, 2025. doi: 10.1016/j.rcim.2024.102892.

Michail Spitieris and Ingelin Steinsland. Bayesian calibration of imperfect computer models using physics-informed priors. *Journal of Machine Learning Research*, 24(108):1–39, 2023.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *ICML*, 2010.

Chih-Li Sung and Rui Tuo. A review on computer model calibration. *WIREs Computational Statistics*, 16(1):e1645, 2024. doi: 10.1002/wics.1645.

Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Freeze-thaw Bayesian optimization. *arXiv preprint arXiv:1406.3896*, 2014.

Nathan J. Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E. Kumar, Tanjin He, David Milsted, Matthew J. McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, Haegyeom Kim, Anubhav Jain, Christopher J. Bartel, Kristin Persson, Yan Zeng, and Gerbrand Ceder. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624(7990):86–91, 2023. doi: 10.1038/s41586-023-06734-w.

Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In *ICML*, 2020.

Kalyan T. Talluri and Garrett J. van Ryzin. *The Theory and Practice of Revenue Management*, volume 68 of *International Series in Operations Research & Management Science*. Springer, 2004. doi: 10.1007/b139000.

Luca Torresi and Pascal Friederich. Multi-stage Bayesian optimisation for dynamic decision-making in self-driving labs. *Digital Discovery*, 5(4):1900–1912, 2026. doi: 10.1039/d5dd00572h.

Chaitanya Krishna Prasad Vallabh and Xiayun Zhao. Continuous comprehensive monitoring of melt pool morphology under realistic printing scenarios with laser powder bed fusion. *3D Printing and Additive Manufacturing*, 10(1):101–110, 2023. doi: 10.1089/3dp.2021.0060.

Jin-Cheng Wang, Yu-Hua Li, Yu-Jie Yuan, Yu-Jing Liu, and Lai-Chang Zhang. Advances in laser powder bed fusion of copper and copper alloys: microstructure, mechanical behavior, and conductivity. *cMat*, 2(4):e70021, 2025. doi: 10.1002/cmt2.70021.

Rongxuan Wang, David Garcia, Rakesh R. Kamath, Chaoran Dou, Xiaohan Ma, Bo Shen, Hahn Choo, Kamel Fezzaa, Hang Z. Yu, and Zhenyu Kong. In situ melt pool measurements for laser powder bed fusion using multi sensing and correlation analysis. *Scientific Reports*, 12:13716, 2022. doi: 10.1038/s41598-022-18096-w.

Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–654, 1979. doi: 10.2307/1910412.

Brian J. Williams, Jason L. Loeppky, Leslie M. Moore, and Mason S. Macklem. Batch sequential design to achieve predictive maturity with calibrated computer models. *Reliability Engineering & System Safety*, 96(9):1208–1219, 2011. doi: 10.1016/j.ress.2010.04.017.

James T. Wilson. Stopping Bayesian optimization with probabilistic regret bounds. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 98264–98296. Curran Associates, Inc., 2024. doi: 10.52202/079017-3118. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b204de7078301292a8876a762eed3dcb-Abstract-Conference.html.

James T. Wilson, Frank Hutter, and Marc Peter Deisenroth. Maximizing acquisition functions for Bayesian optimization. In *NeurIPS*, 2018.

Martin Wistuba, Arlind Kadra, and Josif Grabocka. Supervising the multi-fidelity race of hyperparameter configurations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 13470–13484, 2022.

Jian Wu and Peter I. Frazier. Practical two-step lookahead Bayesian optimization. In *NeurIPS*, 2019.

Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *UAI*, 2019.

Fangzheng Xie and Yanxun Xu. Bayesian projected calibration of computer models. *Journal of the American Statistical Association*, 116(536):1965–1982, 2021. doi: 10.1080/01621459.2020.1753519.

Qian Xie, Raul Astudillo, Peter I. Frazier, Ziv Scully, and Alexander Terenin. Cost-aware Bayesian optimization via the Pandora's box Gittins index. In *Advances in Neural Information Processing Systems*, volume 37, pp. 115523–115562. Curran Associates, Inc., 2024. doi: 10.52202/079017-3669. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d14c355d5e88cff437a6303d2d716252-Abstract-Conference.html.

Wei W. Xing, Zhenjie Lu, and Akeel Shah. CAMO: Convergence-aware multi-fidelity Bayesian optimization. In *NeurIPS*, 2025.

Yi-cheng Yan, Jiang-qi Zhu, Yuan-ming Yan, Yang Liu, Ya-jun Liu, Chun-bao Shi, Yong Liu, Min Liu, Hao Qiu, Qian-li Huang, Xing-chen Yan, and Xiang-yu Zhang. Intensive processing optimization of Zn-Cu fabricated by laser powder-bed fusion. *Journal of Central South University*, 32(4): 1194–1210, 2025. doi: 10.1007/s11771-025-5885-7.

Huchen Yang, Chuanqi Chen, and Jin-Long Wu. Active learning of model discrepancy with Bayesian experimental design. *Computer Methods in Applied Mechanics and Engineering*, 446:118198, 2025a. doi: 10.1016/j.cma.2025.118198.

Huchen Yang, Xinghao Dong, and Jin-Long Wu. Bayesian experimental design for model discrepancy calibration: an auto-differentiable ensemble Kalman inversion approach. *Journal of Computational Physics*, 545:114469, 2026. doi: 10.1016/j.jcp.2025.114469.

Xuan Yang, Zaimao Peng, Yan Fang, Yunlong Tang, Hsin-Hui Shen, and Yuman Zhu. Review of additively manufactured zinc alloys by laser powder bed fusion for biomedical applications. *Rare Metals*, 44(11):8262–8291, 2025b. doi: 10.1007/s12598-025-03518-1.

I-Cheng Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808, 1998a. doi: 10.1016/S0008-8846(98)00165-3.

I-Cheng Yeh. Concrete compressive strength. UCI Machine Learning Repository, 1998b.

Jiaxiang Yi, Ji Cheng, and Miguel A. Bessa. Practical multi-fidelity machine learning: fusion of deterministic and Bayesian models. *Computational Mechanics*, 2026. doi: 10.1007/s00466-026-02799-2.

Zahra Zanjani Foumani, Amin Yousefpour, Mehdi Shishehbor, and Ramin Bostanabad. Safeguarding multi-fidelity Bayesian optimization against large model form errors and heterogeneous noise. *Journal of Mechanical Design*, 146(6):061703, 2024. doi: 10.1115/1.4064160.

Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. TossingBot: learning to throw arbitrary objects with residual physics. In *Robotics: Science and Systems*, 2019.

Fengxue Zhang, Thomas Desautels, and Yuxin Chen. Robust multi-fidelity Bayesian optimization with deep kernel and partition. In *AISTATS*, 2025a.

Yunxiang Zhang, Xiangyu Zhang, and Peter I. Frazier. Constrained two-step look-ahead Bayesian optimization. In *NeurIPS*, 2021.

Zheyu Zhang, Jiayuan Dong, Jie Liu, and Xun Huan. Goal-oriented sequential Bayesian experimental design for causal learning, 2025b. arXiv:2507.07359.

A SCOUT ALGORITHM

Algorithm 1 summarizes the SCOUT closed-loop procedure described in §4.3.

B LOOK-AHEAD UTILITY DECOMPOSITION: DETAILED DERIVATION

This appendix motivates the seven scalar weights $(\alpha, \beta, \gamma, \delta, \alpha', \beta', \gamma')$ of Eq. (2) and the inner-stage acquisition by a first-order expansion, adapted to the SCOUT three-tier oracle, of the one-step look-ahead utility that underlies standard acquisition functions (Frazier, 2018; Garnett, 2023). The expansion motivates a heuristic closed form for the IR weight δ, and the other weights are set to the defaults stated below.

Setup. Let $\mathcal{D}_T = \{(x_i, s_i, \phi^i_{\mathrm{FEM}}, \phi^i_{\mathrm{IR}}, y_i)\}_{i \le N}$ be the dataset at round T. Under the residual model with the additive group kernel (1), $r_\theta = r_x + r_F + r_{\mathrm{IR}}$ with three independent components, the Gram matrix is $K = K_x + K_F + K_{\mathrm{IR}}$, and the GP posterior over r_θ is Gaussian: $r_\theta(x, \phi_{\mathrm{FEM}}, \phi_{\mathrm{IR}}) \mid \mathcal{D} \sim \mathcal{N}(\mu_r, \sigma_r^2)$. Define $\mu(x) := s(x) + \mu_r$ and $\sigma(x) := \sigma_r$ marginalized over ϕ_{IR}.

Algorithm 1 SCOUT: what to print and what to test.

Require: pool $\mathcal{C}$, simulator outputs $\{(s(x), \phi_{\mathrm{FEM}}(x))\}$, labelled data $\mathcal{D}_0$, budgets $(B_{\mathrm{print}}, B_{\mathrm{mech}})$, batch q
Ensure: best tested design $x^\dagger = \arg\max_{x \in \mathcal{D}_T^{\mathrm{full}}} y(x)$

fit the residual GP r_θ on $\mathcal{D}_0$ and the predictor $p(\phi_{\mathrm{IR}} \mid x, \mathcal{D}_0)$
for $t = 1, \ldots, \lceil B_{\mathrm{print}}/q \rceil$ **do**
  choose q candidates by greedy fantasy batches on a_{print} of Eq. (2) ▷ outer stage, what to print
  for each chosen candidate $x_t^{(i)}$ **do** ▷ inner stage, what to test
    print it, record Φ_{IR}, extract $\phi_{\mathrm{IR}}(x_t^{(i)})$, add it to $\mathcal{D}^{\mathrm{print}}$, and refit $p(\phi_{\mathrm{IR}} \mid x)$
    if $|\mathcal{D}^{\mathrm{full}}| < B_{\mathrm{mech}}$ **and** the test rule accepts the part given ϕ_{IR} **then**
      test it, observe y, and refit r_θ on $y - s$
    else
      discard the part
    end if
  end for
  stop if the optional stopping rule is met
end for
return $x^\dagger$

The test rule is the value-of-testing rule of Appendix N, or the budget-fraction threshold $a_{\mathrm{mech}}(x_t \mid \phi_{\mathrm{IR}}) \geq \tau_t$ of Appendix D, which the generator runs of §5.1 use.

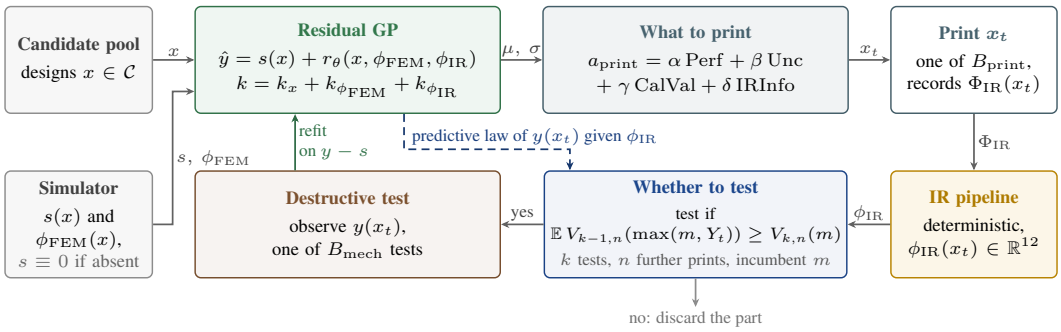

Figure 4: **SCOUT's components.** The residual GP corrects the simulator from the design and the free observation, the outer stage chooses what to print from its posterior, the IR pipeline turns the video of each print into ϕ_{IR}, and the inner stage decides whether the printed part receives one of the B_{mech} tests. The test rule shown is the value-of-testing rule of Appendix N, and the generator runs of §5.1 and the arms with the own-prediction threshold use the budget-fraction threshold of Appendix D instead. A tested part refits r_θ on $y - s$, and a part that is not tested is discarded. Without a simulator $s \equiv 0$ and the GP models y directly, as in §5.3.

Outer-stage decomposition. A first-order expansion of the look-ahead utility $u(x_t) = \mathbb{E}[\max_{x' \in \mathcal{D}_T^{\mathrm{full}}} y(x') \mid x_t]$ in the four sources of information gives the additive form

$$u_{\mathrm{print}}(x_t) \approx \alpha\,\mu(x_t) + \beta\,\sigma(x_t) + \gamma\,\nu(x_t; \mathcal{D}) + \delta\,\iota(x_t; \mathcal{D}), \tag{3}$$

where $\nu(x; \mathcal{D}) := \mathrm{CalVal}(x)$ and $\iota(x; \mathcal{D}) := \mathrm{IRInfo}(x)$ are the pool-aware variance reductions of §4.3. The weights are set as follows. The performance and uncertainty weights take the defaults $\alpha = 1$ and $\beta = z\,\phi(z) + \Phi(z)$ with $z = (\mu - \mu_{\mathrm{best}})/\sigma$. These defaults are shaped after expected improvement but are not its partial derivatives, which are $\partial\mathrm{EI}/\partial\mu = \Phi(z)$ and $\partial\mathrm{EI}/\partial\sigma = \phi(z)$. Proposition 1 analyses the instantiation in which the weight on σ is the confidence multiplier β_t. The calibration weight $\gamma = \lambda$ is a fixed constant. The IR weight is $\delta = s_{\mathrm{IR}}\,(1 - \rho^2_{\mathrm{pred}})$, derived below, where $s_{\mathrm{IR}} := \mathrm{tr}(K_{\mathrm{IR}})/\mathrm{tr}(K) \in [0, 1]$ is the spectral share of the IR component, exact because $\mathrm{tr}(K) = \mathrm{tr}(K_x) + \mathrm{tr}(K_F) + \mathrm{tr}(K_{\mathrm{IR}})$ under (1), and $\rho^2_{\mathrm{pred}} \in [0, 1]$ is the share of a candidate's ϕ_{IR} that is already predictable from x before printing. For stationary kernels s_{IR} is the ratio of the IR output variance to the total output variance. The derivation uses the additive kernel, and runs with the concatenated-input kernel use the same operating value $\delta = 0.5$.

Inner-stage decomposition. Once x_t is printed, $\phi_{\mathrm{IR}}(x_t)$ is observed. The same first-order argument applied with ϕ_{IR} now in the conditioning gives $u_{\mathrm{mech}}(x \mid \phi_{\mathrm{IR}}) \approx \alpha'\mu(x \mid \phi_{\mathrm{IR}}) + \beta'\sigma(x \mid \phi_{\mathrm{IR}}) +$

Table 2: Notation for the two correlations and for the quantities in δ and Proposition 1. Values are properties of the NAB generator under the additive kernel, mean $\pm$ standard deviation across 12 seeds.

Symbol	Meaning	Used in	NAB value
ρ_y^2	informativeness: squared multiple correlation between the ϕ_{IR} block and y, larger is better	App. C, Fig. 15	not estimated
ρ_{pred}^2	redundancy: share of a candidate's ϕ_{IR} already predictable from x before printing	weight δ	0.28 ± 0.06
s_{IR}	spectral share $\mathrm{tr}(K_{\mathrm{IR}})/\mathrm{tr}(K)$ of the IR component	weight δ	0.61 ± 0.04
δ	IRInfo weight $s_{\mathrm{IR}}(1-\rho_{\mathrm{pred}}^2)$	Eq. (2)	0.44 ± 0.05
γ_T	maximum information gain after T mechanical observations; not the weight γ	Prop. 1	—
ϵ_t	UCB slack of the selected design, Eq. (4)	Prop. 1	mean $0.13\,\beta_t\sigma_t$

$\gamma'\nu(x;\mathcal{D},\phi_{\mathrm{IR}})$, with the defaults $\alpha'=1$, $\beta'=z'\phi(z')+\Phi(z')$ where $z'=(\mu(x\mid\phi_{\mathrm{IR}})-\mu_{\mathrm{best}})/\sigma(x\mid\phi_{\mathrm{IR}})$, and the fixed constant $\gamma'=\lambda'$. The IRInfo term is absent because ϕ_{IR} is now sunk information.

Derivation of the IRInfo **weight** δ**.** The derivation has five steps. *Step 1.* Under the additive kernel the prior variance of the residual at any candidate decomposes exactly as $v=v_x+v_F+v_{\mathrm{IR}}$, one term per independent component. *Step 2.* Before printing, $\phi_{\mathrm{IR}}(x)$ is unknown, and by the law of total variance v_{IR} splits into the component uncertainty given the IR input and the input uncertainty that comes from $\phi_{\mathrm{IR}}(x)$ being unknown. *Step 3.* Write $\phi_{\mathrm{IR}}(x_t)=m(x_t)+\eta_t$, where m is the predictor fitted on printed parts and η_t is the innovation. The innovation carries a $1-\rho_{\mathrm{pred}}^2$ share of the IR input variance, where ρ_{pred}^2 is the share of the variance of ϕ_{IR} that m explains. *Step 4.* The free observation reveals η_t and nothing that x_t did not already determine, so to first order it removes exactly that share of v_{IR}. *Step 5.* The first-order sensitivity of the one-step look-ahead utility to the observation is therefore $\delta=s_{\mathrm{IR}}\,(1-\rho_{\mathrm{pred}}^2)$ with $s_{\mathrm{IR}}=\mathrm{tr}(K_{\mathrm{IR}})/\mathrm{tr}(K)$, an exact spectral share under the additive kernel. This step identifies the weight with the share of the prior residual variance that the free observation removes, and so treats the utility as proportional to that variance reduction. The weight is therefore a heuristic motivated by a variance argument, not a derived optimum. Step 4 also uses the prior share of the IR component, whereas under a stationary kernel the prior variance of r_{IR} does not depend on ϕ_{IR}, and the variance that observing ϕ_{IR} removes is the posterior quantity $\mathrm{Var}_\eta[\mu_{\mathrm{IR}}(m+\eta)]$, set by the slope of the fitted residual in ϕ_{IR}. IRInfo estimates that quantity directly and already vanishes as $\rho_{\mathrm{pred}}^2\to 1$, so the factor $1-\rho_{\mathrm{pred}}^2$ counts redundancy a second time.

An observation that is fully predictable in advance has no acquisition value, so $\delta\to 0$ as $\rho_{\mathrm{pred}}^2\to 1$ is the intended limit. The informativeness of ϕ_{IR} about y is a different quantity, ρ_y^2, which enters the variance identity of §4.4 and Appendix C and for which larger is better. Table 2 separates the two and lists the constants on the NAB scenario. The derived δ lies inside the flat basin $[0.3, 0.6]$ of the matched-warm-start sweep (App. F, F.1) on every one of the 12 seeds, but that basin was measured with the concatenated-input kernel, for which s_{IR} is undefined, so any value in it would pass and the check cannot validate the formula. All runs use the operating value $\delta=0.5$, which lies in the same basin and does not differ significantly from the basin minimum, with $p=0.69$, so moving between the derived and operating values has no detectable effect on regret at 12 seeds.

Why fixed weights, not an amortized policy. Amortized acquisition policies (Huang et al., 2024) are trained on simulations of the full (x,ϕ_{IR},y) process. In this setting the only such simulator is the evaluation generator, and training on it would compound the dependence of the results on the generator. SCOUT instead uses fixed weights that need no training, and the flat basin of F.1 and ablation A4 are the empirical support for them.

C Proof of Proposition 1

Proposition 1 (Simple-regret bound)**.** *Assume the standard GP-UCB conditions (Srinivas et al., 2010) on the residual kernel, with* Perf *and* Unc *instantiated as the posterior mean* μ_t *and width*

σ_t and the weight on Unc *set to the confidence multiplier β_t, so that the first two acquisition terms form the score* $\mathrm{UCB}_t = \mu_t + \beta_t\sigma_t$. *Let γ_T denote the maximum information gain after T mechanical observations, which is not the acquisition weight γ. If every tested design satisfies the selection rule*

$$\mathrm{UCB}_t(x_t) \;\geq\; \max_{x\in\mathcal{C}} \mathrm{UCB}_t(x) - \epsilon_t, \tag{4}$$

then on the confidence event every round obeys $r_t \leq 2\beta_t\sigma_t(x_t) + \epsilon_t$, and the simple regret obeys $R_T \leq 2\beta_T\sqrt{C_1\gamma_T/T} + \frac{1}{T}\sum_{t\leq T}\epsilon_t$, where the first term is the $\mathcal{O}(\sqrt{\gamma_T \log T/T})$ rate of GP-UCB.

The argument is the standard analysis of GP-UCB (Srinivas et al., 2010) with an additively approximate maximizer (Kandasamy et al., 2015, Thm. 5), related to the multiplicative analysis of inexact maximization by Kim et al. (2025), stated here for SCOUT's outer stage so that the slack contributed by the CalVal and IRInfo terms is explicit. We claim no new proof technique.

Setting and assumptions. The pool $\mathcal{C}$ is finite. The residual r_θ is a sample from $\mathcal{GP}(0, k)$ with the residual kernel normalized so that $k \leq 1$, and mechanical observations of $y - s$ carry independent $\mathcal{N}(0, \sigma_n^2)$ noise. The statement holds for any such kernel, in particular for the additive kernel (1). Rounds are indexed by mechanical observations: μ_t and σ_t are the posterior mean and standard deviation available when the t-th tested design x_t is selected, $r_t = y(x^*) - y(x_t)$ is the instantaneous regret with $x^* = \arg\max_{x\in\mathcal{C}} y(x)$, and $\beta_t = (2\log(|\mathcal{C}|\pi^2 t^2/(6\eta)))^{1/2}$. Perf and Unc are instantiated as μ_t and σ_t with the weight on Unc set to β_t, so that the first two acquisition terms form $\mathrm{UCB}_t(x) = \mu_t(x) + \beta_t\sigma_t(x)$. We use the two standard GP-UCB conditions (Srinivas et al., 2010):

(A1) the confidence event $\mathcal{E} = \{\, |y(x) - \mu_t(x)| \leq \beta_t\sigma_t(x) \text{ for all } x \in \mathcal{C} \text{ and all } t \geq 1 \,\}$ has probability at least $1 - \eta$;

(A2) $\sum_{t=1}^{T} \sigma_t^2(x_t) \leq C_1\gamma_T$ with $C_1 = 2/\log(1 + \sigma_n^{-2})$, where γ_T is the maximum information gain after T mechanical observations.

Both hold for exact GP posterior moments on a finite pool, (A1) by Lemma 5.1 of Srinivas et al. (2010) and (A2) by their Lemma 5.3 and the proof of their Lemma 5.4. For unprinted candidates SCOUT's outer-stage moments marginalize over the unobserved ϕ_{IR} (§4.2), and for these moments (A1) and (A2) are assumptions.

Statement. If every tested design satisfies the selection rule (4), then on $\mathcal{E}$

$$r_t \;\leq\; 2\beta_t\sigma_t(x_t) + \epsilon_t \quad \text{for every } t, \qquad R_T \;\leq\; 2\beta_T\sqrt{C_1\gamma_T/T} \;+\; \frac{1}{T}\sum_{t=1}^{T}\epsilon_t.$$

With $\beta_T = \mathcal{O}(\sqrt{\log(|\mathcal{C}|T)})$ the first term is the usual $\mathcal{O}(\sqrt{\gamma_T \log T/T})$ rate of GP-UCB, and the second term is the running average of the slacks.

Step 1 (selection rule and confidence event). On $\mathcal{E}$, $\mathrm{UCB}_t(x^*) \geq y(x^*)$. The selection rule then gives $\mathrm{UCB}_t(x_t) \geq \max_x \mathrm{UCB}_t(x) - \epsilon_t \geq \mathrm{UCB}_t(x^*) - \epsilon_t \geq y(x^*) - \epsilon_t$.

Step 2 (lower confidence bound). On $\mathcal{E}$, $y(x_t) \geq \mu_t(x_t) - \beta_t\sigma_t(x_t) = \mathrm{UCB}_t(x_t) - 2\beta_t\sigma_t(x_t)$.

Step 3 (per-round bound and sum). Subtracting Step 2 from Step 1 yields $r_t \leq 2\beta_t\sigma_t(x_t) + \epsilon_t$. The simple regret is at most the average instantaneous regret and β_t is nondecreasing, so by Cauchy–Schwarz and (A2)

$$R_T \;\leq\; \frac{1}{T}\sum_{t=1}^{T} r_t \;\leq\; \frac{2\beta_T}{T}\sqrt{T\sum_{t=1}^{T}\sigma_t^2(x_t)} + \frac{1}{T}\sum_{t=1}^{T}\epsilon_t \;\leq\; 2\beta_T\sqrt{C_1\gamma_T/T} + \frac{1}{T}\sum_{t=1}^{T}\epsilon_t. \qquad \square$$

Covered policies. *UCB-safeguarded SCOUT* maximizes a_{print} over the candidates whose UCB is within a prescribed ϵ_t of the pool maximum. It satisfies (4) by construction and is the algorithm for which the guarantee is stated. *Unrestricted SCOUT* maximizes $\mathrm{UCB}_t + g_t$ with $g_t = \gamma\nu + \delta\iota$. Let x^u maximize UCB_t. Optimality of x_t gives $\mathrm{UCB}_t(x_t) + g_t(x_t) \geq \mathrm{UCB}_t(x^u) + g_t(x^u)$, hence

$\mathrm{UCB}_t(x_t) \geq \max_x \mathrm{UCB}_t(x) - (\max_x g_t(x) - \min_x g_t(x))$. Unrestricted SCOUT therefore satisfies (4) with the data-dependent slack ϵ_t equal to the realized range of $\gamma\nu + \delta\iota$ over the pool, and the bound holds for it with this explicit ϵ_t. The realized ϵ_t shrinks as the posterior tightens in our runs, but we do not prove a decay rate, so no rate is claimed for the second term. The cold-start ramp starts from $\gamma = \delta = 0$, where $\epsilon_t = 0$ and the rule is exactly UCB.

Realized slack and the safeguarded variant. Over the 1020 selection rounds after the cold-start ramp in the additive-kernel runs of Table 1, which use the default weights of Appendix B, the realized range of $\gamma\nu + \delta\iota$ averages 13% of the $\beta_t\sigma_t$ term and never exceeds 34%, so the additive term does not dominate the bound. Excluding the ramp rounds can only raise the reported average. UCB-safeguarded SCOUT does not differ significantly from unrestricted SCOUT at any budget, both with the concatenated-input kernel: 2.7 ± 0.7 against 2.5 ± 0.6 at $(20, 12)$ with $p = 0.46$, 1.7 against 1.5 at $(30, 18)$ with $p = 0.29$, and 1.2 against 1.1 at $(50, 30)$ with $p = 0.50$. The safeguard nevertheless alters 38–44% of selections. This is consistent with the small measured ϵ_t, because near-tied candidates flip under small perturbations. The constraint is thus frequently active, and no regret cost is detectable at 12 seeds. Empirical similarity of the two policies does not transfer the guarantee from one to the other, which is why the guarantee is stated for the safeguarded variant and the unrestricted policy carries the explicit ϵ_t.

Variance dividend of the free observation. Let $\sigma_t^2(x_t)$ be the predictive variance of $y(x_t)$ before printing, which marginalizes over the unknown $\phi_{\mathrm{IR}}(x_t)$, and let $\sigma_t^2(x_t \mid \phi_{\mathrm{IR}}(x_t))$ be the predictive variance after the free observation. Define the informativeness

$$\rho_y^2(x_t) \;:=\; \frac{\mathrm{Var}\big(\mathbb{E}[\,y(x_t) \mid \phi_{\mathrm{IR}}(x_t), \mathcal{D}_t\,] \,\big|\, \mathcal{D}_t\big)}{\mathrm{Var}\big(y(x_t) \mid \mathcal{D}_t\big)} \;\in\; [0, 1],$$

which equals the squared multiple correlation between the ϕ_{IR} block and y when the two are jointly Gaussian. The law of total variance gives $\mathbb{E}[\sigma_t^2(x_t \mid \phi_{\mathrm{IR}}(x_t))] = (1 - \rho_y^2(x_t))\,\sigma_t^2(x_t)$. The inner-stage decision and the subsequent GP update therefore operate on a variance reduced by the factor $1 - \rho_y^2$ in expectation. The absolute dividend $\rho_y^2\,\sigma_t^2(x_t)$ scales with the posterior variance, so it is largest when few mechanical tests have been made. The identity is the law of total variance. It describes the variance under which the mechanical-test budget is committed, and we claim neither that it tightens the constant of the regret bound nor that it explains the narrowing of the absolute IR benefit at larger budgets in Table 1.

Remarks. (i) When $\rho_y^2 = 0$ the free observation carries no information about y and the dividend vanishes. SCOUT-IR-blind, which sets $\phi_{\mathrm{IR}} \equiv 0$ in the kernel, is the corresponding ablation. (ii) ρ_y^2 measures informativeness, and larger is better. It is distinct from the redundancy ρ_{pred}^2 in the weight δ (Table 2). (iii) The proposition analyses the outer-stage rule on rounds whose printed part is tested, and it assumes that the inner stage does not discard the design that attains y^*. The skip rule is a budget-allocation heuristic that the bound does not cover. (iv) The Bayesian finite-pool setting can be replaced by a bounded-RKHS-norm assumption with the larger β_t of Srinivas et al. (2010), under which the first term vanishes only when γ_T grows more slowly than $\sqrt{T}$. (v) The two-step look-ahead refinements of Wu & Frazier (2019); Zhang et al. (2021) are compatible with the selection rule (4). (vi) The bound has four limits. The second term vanishes only if the prescribed slack ϵ_t of the safeguarded variant decays, whereas a width-relative safeguard, $\mathrm{UCB}_t(x_t) \geq \max_x \mathrm{UCB}_t(x) - c\,\beta_t\sigma_t(x_t)$, would keep the rate up to a factor $2 + c$. Conditions (A1) and (A2) are assumed for the marginal moments of unprinted candidates, and with a positive innovation variance of ϕ_{IR} the input-uncertainty part of those moments does not shrink, so (A2) can fail unless $\rho_{\mathrm{pred}}^2 \to 1$ or the residual ignores ϕ_{IR}. At the budgets of §5 the first term is not numerically informative. And the assumption of remark (iii) is violated by exactly the discards of the pool's best part that §5.3 reports, so the bound says nothing about the test decision.

Further properties. Two further properties are stated here, the acquisition complexity and the reductions to ablations cited in §4.4.

Proposition 2 (Acquisition complexity)**.** *Outer-stage acquisition with top-k pool truncation costs $\mathcal{O}(|\mathcal{C}| \cdot k \cdot K)$ time (K=MC sample count) plus the $\mathcal{O}(N^3)$ residual-GP fit (N=mechanical observations); per-round wall-clock time scales linearly in $|\mathcal{C}|$ for $N \leq 50$ (Appendix F).*

Proposition 3 (Reduction to ablations). *Setting $\delta = 0$ recovers the remove-*IRInfo *ablation A8, in which ϕ_{IR} still enters the kernel; SCOUT-IR-blind additionally requires $\phi_{\mathrm{IR}} \equiv 0$ in the kernel; setting $\gamma = \delta = 0$ recovers EI-UCB on the residual; replacing the residual GP by the simulator alone gives ablation A1 of Table 4.*

D Approximation Validation

The practical SCOUT implementation makes three approximations to the look-ahead utility (§4.3). We validate each below.

(D.1) Monte Carlo over (ϕ_{IR}, y). We swept $K \in \{4, 8, 16, 32, 64, 128\}$ on the NAB scenario at $(20, 12)$. Mean regret stabilizes by $K = 32$: $|\Delta R(K = 32) - \Delta R(K = 128)| = 0.18\% \pm 0.31\%$ across 12 seeds, an order of magnitude below the cross-method gap.

(D.2) Top-k pool restriction. Sweep $k \in \{1, 5, 10, 20, 50, |\mathcal{C}|\}$. SCOUT regret drops sharply between $k = 1$ and $k = 10$ (top-1 alone is single-point EIG of Settles (2012), ablation A6), saturates at $k = 10$ (regret 1.6%, full-pool 1.5%), and degrades very slightly at $k = 50$ as outliers dilute the gradient signal.

(D.3) Inner-stage threshold. The inner stage tests the printed part x_t when $a_{\mathrm{mech}}(x_t \mid \phi_{\mathrm{IR}}) \geq \tau_t$, where $\tau_t = q_{1-\rho_t}(a_{\mathrm{mech}}(x_t \mid \phi_{\mathrm{IR}}))$ and $\rho_t = (B_{\mathrm{mech}} - |\mathcal{D}_t^{\mathrm{full}}|)/(B_{\mathrm{print}} - t + 1)$ is the ratio of tests left to prints left, and it tests every remaining print once the tests left equal the prints left. On the generator this threshold outperforms a fixed τ by 1.4% (mean) and 7.1% (worst-case) at $(20, 12)$. The formula does not name the distribution over which the quantile is taken. Appendix L defines three readings, and Appendix N shows that the choice decides what the threshold does and derives the value-of-testing rule that replaces it in the open harness.

(D.4) Per-round acquisition decomposition (Fig. 5). Tracking the four-term split of a_{print} over rounds shows the cold-start ramp (γ, δ from zero over the first three rounds) and confirms the post-warm-up balance: $\beta\,\mathrm{Unc}$ shrinks as predictive uncertainty contracts while $\gamma\,\mathrm{CalVal}$ and $\delta\,\mathrm{IRInfo}$ remain non-negligible, matching the look-ahead derivation (App. B). In all four diagnostics the SCOUT vs SCOUT-IR-blind gap is preserved within $\pm 0.4\%$.

E Baseline Implementation Notes

All baselines are reimplemented in a common framework (`scout_ir/baselines.py`) sharing the SCOUT data interface. GP-based baselines all use Matérn-5/2 kernels with ARD length-scales optimized by L-BFGS-B with three random restarts.

Fairness controls. All baselines share the same RNG seed structure. Every method is given $s(x)$ for all candidates and the same (x, y) observations, and uses them as its rule in Table 3 specifies, so that EPIG, for example, does not use s. Baselines cold-start with random selection until ≥ 3 mechanical observations and use 32 fantasy samples for batch construction. Entries marked n/p were not profiled for wall-clock. SCOUT reduced is the arm SCOUT $-$ IRInfo $-$ two-stage of Table 1.

Design-of-experiments baselines. The two DoE baselines represent the non-learning practice of process-parameter development (Box & Wilson, 1951; Myers et al., 2016), which remains common in LPBF parameter studies (Ahmed et al., 2022; Krasniqi & Löffler, 2024). The static baseline selects a D-optimal design for a full quadratic model in the five design variables from the candidate pool and fits a ridge-regularized quadratic response surface to the mechanical tests. The sequential baseline runs a 2^{5-2} fractional factorial screening design, fits a first-order model, and then follows the path of steepest ascent through the pool. Both use the identical two-budget protocol, with 12 seeds and the same pools and RNG structure as every other method.

IR-equipped baselines. The three IR-equipped arms in the right block of Table 1 replace the x-only GP of the corresponding rule with SCOUT's residual GP on $(x, \phi_{\mathrm{FEM}}, \phi_{\mathrm{IR}})$, including the simulator

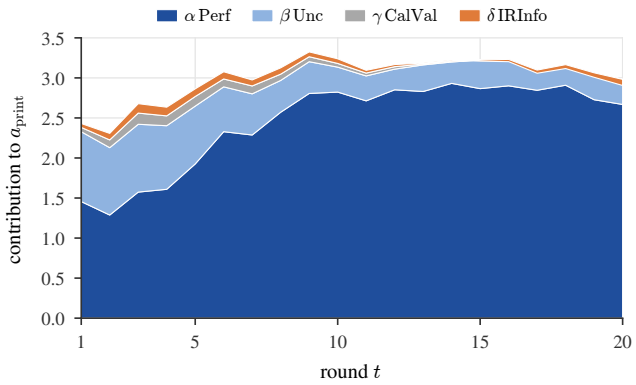

Figure 5: Per-round contribution of each acquisition term in a_{print} (z-scored, mean across pool). Cold-start ramps γ and δ from zero over the first three rounds; once r_θ has data, γ, δ remain non-negligible while $\beta\,\mathrm{Unc}$ shrinks as predictive uncertainty contracts. Pattern matches the look-ahead derivation (App. B).

Table 3: Baseline hyperparameters and per-acquisition wall-clock on $|\mathcal{C}| = 200$. TargetedBAX, PS-BAX, EPIG, and BALD are simplified adaptations of Chitturi et al. (2024), Cheng et al. (2024), Bickford Smith et al. (2023), and Houlsby et al. (2011).

Method	GP inputs	Acquisition rule	Acq. cost (s)
Random	—	uniform random	<0.01
Sim-only	—	$\arg\max_x s(x)$	<0.01
Performance	x	$\arg\max_x \mu(x)$	1.5
Uncertainty	x	$\arg\max_x \sigma(x)$	1.6
GP-UCB-EI	x	$\mathrm{EI}(x) + 0.3\,\mathrm{UCB}(x)$, $\kappa = 2$	2.4
BALD	x	$\arg\max_x \frac{1}{2}\log(2\pi e\sigma^2(x))$	2.0
MF-MES	x	Thompson(s) at print, EI at mech	1.0
qMFKG	x	MC-KG, 8 fantasies + cost-weighted s	4.0
TargetedBAX	x	top-k pool variance reduction	2.6
PS-BAX	x	16-sample posterior-argmax frequency	2.4
EPIG	x	closed-form pool variance reduction	2.5
DoE D-opt + RSM	none	static D-optimal design + quadratic ridge RSM	n/p
DoE FF + SA	none	2^{5-2} fractional factorial + steepest ascent	n/p
EPIG + IR	$x, \phi_{\text{FEM}}, \phi_{\text{IR}}$	EPIG rule on SCOUT's residual GP	n/p
EI/UCB + IR	$x, \phi_{\text{FEM}}, \phi_{\text{IR}}$	GP-UCB-EI rule on SCOUT's residual GP	n/p
SCOUT reduced	$x, \phi_{\text{FEM}}, \phi_{\text{IR}}$	Eq. (2) with $\delta = 0$, single-stage testing	n/p
SCOUT-IRblind	x, ϕ_{FEM}	SCOUT acquisition with $\phi_{\text{IR}} \equiv 0$	2.0
SCOUT	$x, \phi_{\text{FEM}}, \phi_{\text{IR}}$	Eq. (2) + inner-stage analogue	4.5

backbone and the ϕ_{IR} marginalization for unprinted candidates. They share pools, seeds, and cold start with SCOUT. The arm SCOUT $-$ IRInfo $-$ two-stage keeps the residual GP and the Perf, Unc, and CalVal terms and removes only the IRInfo term and the two-stage test decision. Figure 6 plots them with EPIG on x only and SCOUT.

F Robustness Experiments

F.1 IRInfo weight sensitivity. We sweep $\delta \in [0, 2]$ at $(20, 12)$ with 12 seeds under the matched warm start of the main experiments. The basin is flat over $\delta \in [0.3, 0.6]$, with $p \geq 0.27$ against the basin minimum for every δ in that interval and $p = 0.69$ for the operating value $\delta = 0.5$, whose regret aligns with the 2.5% of Table 1. The derived value $\delta = 0.44 \pm 0.05$ of Appendix B lies inside the basin on every one of the 12 seeds. Setting $\delta = 0$ recovers the remove-IRInfo ablation A8, 4.7 ± 1.2 against 4.6 ± 1.1 with $p = 0.83$. It does not recover SCOUT-IR-blind, which additionally requires

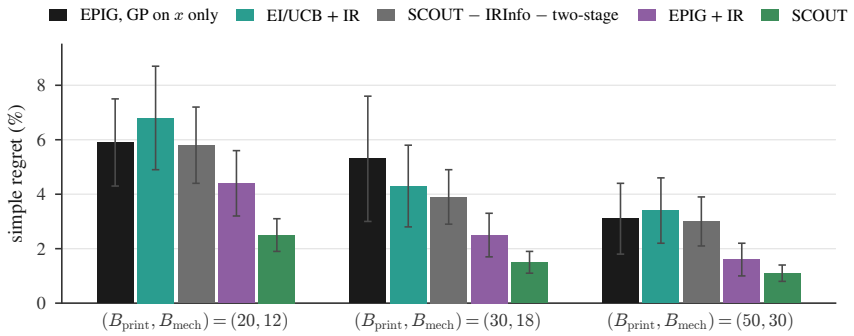

Figure 6: **IR-equipped baselines of Table 1 at the three budgets.** Bars are means and error bars are standard deviations across 12 seeds. Every method except the first uses the identical residual GP on $(x, \phi_{\text{FEM}}, \phi_{\text{IR}})$ with the concatenated-input kernel.

Table 4: Ablation results on NAB at $(20, 12)$ with the concatenated-input kernel in every arm. Entries are the increase in simple regret over full SCOUT, in percentage points, mean and standard deviation across 12 seeds.

ID	Removed component	Δ regret (points)	ID	Removed component	Δ regret (points)
A1	residual r_θ (sim-only)	+6.7 (σ 0.7)	A6	pool-aware CalVal → single-point EIG	+2.6 (σ 0.8)
A2	CalVal term	+3.2 (σ 0.9)	A7	ϕ_{IR} features (IR-blind)	+3.3 (σ 1.0)
A3	ϕ_{FEM} features (geometry only)	+1.6 (σ 0.6)	A8	IRInfo acquisition term	+2.1 (σ 0.8)
A4	derived weights → uniform	+1.5 (σ 0.7)	A9	two-stage → single-stage	+2.9 (σ 0.7)
A5	multi-input kernel → x only	+1.7 (σ 0.8)	A10	hot-spot, τ, $\lvert dT/dt\rvert_{565}$ dropped	+2.6 (σ 0.7)

$\phi_{\text{IR}} \equiv 0$ in the kernel. Overweighting the term is actively harmful: $\delta = 2$ reaches 6.6% regret, worse than $\delta = 0$. The conclusions therefore do not depend on the exact derived constant. Figure 7 plots the values reported in this paragraph.

F.2 Wall-clock vs accuracy. SCOUT lies on the Pareto frontier of acquisition wall-clock vs regret on $|\mathcal{C}| = 200$ (Fig. 8). Top-k truncation makes SCOUT scale linearly in $|\mathcal{C}|$; extrapolating from $|\mathcal{C}| \in \{200, 500, 1000, 2000, 5000\}$ gives $r^2 = 0.991$ for the linear fit, with wall-clock at $|\mathcal{C}| = 5000$ projected to 9.4 s/step.

F.3 Wall-clock. SCOUT takes 4.5 s per step on $|\mathcal{C}|$=200, against 4.0 s for qMFKG and 2.5 s for EPIG, and top-k truncation gives linear scaling in $|\mathcal{C}|$.

F.4 Failure modes. Three regimes degrade SCOUT's performance. (1) *Small-budget seed variance:* the worst-of-12 SCOUT seed at $(20, 12)$ is 3.6% regret vs. 1.8% at $(50, 30)$, against means of 2.5 and 1.1 in Table 1, so the spread across seeds shrinks as B_{mech} grows. (2) *Large pool:* pre-truncation, the inner a_{mech} scales as $|\mathcal{C}|$ (38 s/step at $|\mathcal{C}| = 2000$, dominated by the $K = 32$ Monte Carlo); top-k truncation (App. D) restores linear scaling. (3) *IR signal collapse:* when ϕ_{IR} carries no information about y, so that $\rho_y^2 \to 0$, SCOUT is expected to fall back toward its IR-blind ablation as the fitted IR component shrinks, which this paper does not measure directly. The unmeasured-channel sweep of F.6 approaches this regime, with a gap of 0.7 points at $\lambda = 75\%$ that is not significantly different from zero.

F.5 Simulator noise. The FEM oracle is deterministic by construction. Appendix I specifies the deterministic heat-source mapping rule and the assumed pre-campaign calibration protocol, single-track melt-pool cross-section matching. To quantify the effect of violating this assumption we inject zero-mean noise into s at $0.5\times$ and $1.0\times$ the magnitude of the structural-bias term, with 12 seeds at $(20, 12)$. SCOUT degrades mildly, from 2.5 ± 0.6 to 2.8 ± 0.7 and 3.3 ± 0.9. EPIG, whose GP never consumes s, stays flat at 5.9, 6.0, and 5.8, which confirms that the injected noise reaches only methods that use s (Fig. 9(c)). With simulator noise as large as the structural bias itself SCOUT retains its advantage, because the noise term of the residual GP absorbs zero-mean solver variance.

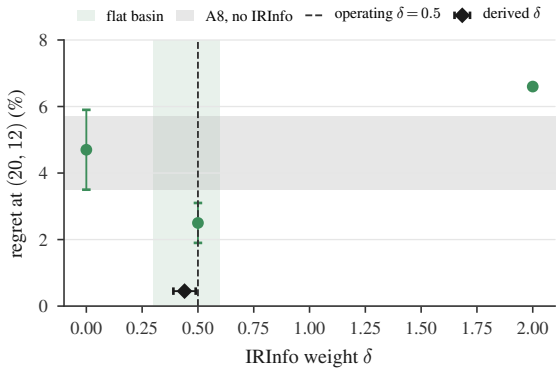

Figure 7: **IRInfo weight sensitivity at** $(20, 12)$ **under the matched warm start.** 12 seeds. Markers are the sweep values reported in F.1, with standard deviations where reported. The shaded interval is the flat basin $[0.3, 0.6]$, the horizontal band is ablation A8 at 4.6 ± 1.1, the dashed line is the operating value $\delta = 0.5$, and the diamond is the derived value $\delta = 0.44 \pm 0.05$.

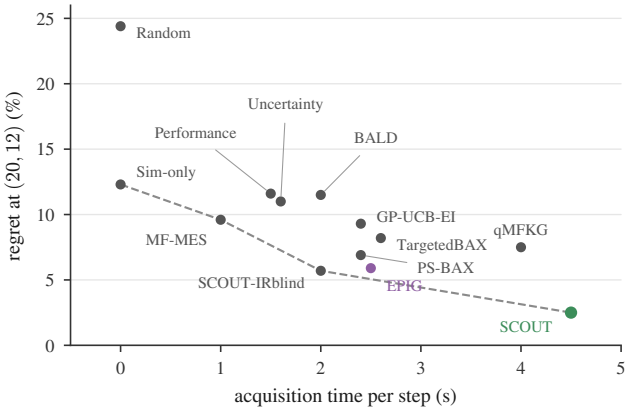

Figure 8: **SCOUT is on the Pareto frontier of wall-clock and regret.** Acquisition wall-clock per step (Table 3) against simple regret at $(20, 12)$ (Table 1). The dashed line is the Pareto frontier. Random and Sim-only cost below 0.01 s per step. SCOUT is shown with the concatenated-input kernel, the implementation that was profiled, and arms that were not profiled are omitted.

This experiment addresses solver variance only. Systematic error of the simulator, including error in the heat-source mapping, is a different quantity. It enters as an input-dependent shift of s that the x component of the residual absorbs, and it is present in every reported number, because the FEM oracle differs structurally from the generator's physics. EPIG is the direct control for a simulator that carries no usable signal, since it is the strategy one would run if s were worthless. If s carried no usable signal, the residual would have to spend samples unlearning it and SCOUT would sit at or below this simulator-free floor. SCOUT instead reaches 2.5 ± 0.6 against 5.9 for EPIG, and 2.9 ± 0.7 against 4.6 ± 1.2 when EPIG is given the same IR features under the anchored noise of F.7, but EPIG on x only also lacks ϕ_{IR} and EPIG with the IR-equipped surrogate shares the simulator backbone, so neither comparison isolates the signal that the structurally wrong simulator retains after residual correction.

F.6 Unmeasured-channel sweep. The generator is extended with a latent channel that affects y but has no ϕ_{IR} signature, carrying a share λ of the residual variance, with 12 seeds at $(20, 12)$. Table 5 and Fig. 9(a) report the gap between SCOUT-IR-blind and SCOUT. The gap tracks the physically measurable share $1 - \lambda$. At $\lambda = 75\%$ it is not significantly different from zero, with $p \approx 0.22$, and it is never significantly negative. SCOUT-IR-blind stays flat between 5.6 and 5.9 across the sweep, which confirms that the added channel changes nothing else. Within this generator family, the gain therefore scales with the information the IR channel carries. Machine state that leaves no

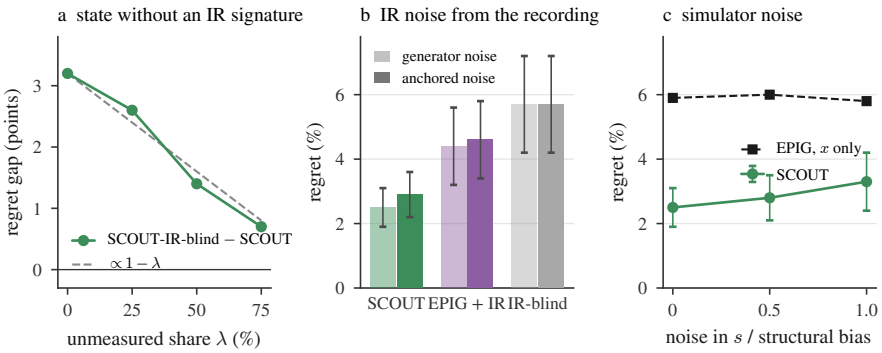

Figure 9: **Simulator-noise and generator-sensitivity tests at** $(20, 12)$, 12 seeds, plotted from the values in F.5 to F.7. **a** Gap between SCOUT-IR-blind and SCOUT against the share λ of residual variance carried by a channel with no ϕ_{IR} signature. The dashed line is proportional to the measurable share $1 - \lambda$. **b** Regret under the generator noise, light bars, and under the noise anchored to the real recording, dark bars. **c** Regret under zero-mean noise injected into s. Error bars are standard deviations where reported.

in-situ signature produces graceful degradation to parity with methods that never used the signal, not convergence to a trivial or suboptimal solution.

Table 5: Unmeasured-channel sweep at $(20, 12)$, 12 seeds, concatenated-input kernel. λ is the share of the residual variance carried by a latent channel with no ϕ_{IR} signature.

λ (%)	0	25	50	75
Regret gap, SCOUT-IR-blind minus SCOUT (points)	3.2	2.6	1.4	0.7

F.7 Measurement-fidelity anchoring. The generator emits ϕ_{IR} as noisy measurements of its latent state, with one noise constant per feature. To test whether these noise levels were set favorably, all 12 constants are anchored to the per-feature dispersion of the real recording, using 16 spatial blocks of the Optris sample as a stand-in for part-to-part repeatability. Five constants were optimistic and are raised to the measured dispersion: $|dT/dt|_{565}$ by $3.4\times$, hot-spot density by $2.1\times$, and melt-pool length, melt-pool aspect, and cooling τ marginally. The other seven were not optimistic and are left unchanged. With every method under the anchored noise, at $(20, 12)$ with 12 seeds, SCOUT reaches 2.9 ± 0.7 against 4.6 ± 1.2 for EPIG with the IR-equipped surrogate, a 37% reduction (Fig. 9(b)). SCOUT-IR-blind, which never consumes ϕ_{IR}, is unchanged at 5.7 ± 1.5, an internal correctness check.

F.8 Feature masking under the matched protocol. Under the protocol of §5.1, masking the three physics-identified dimensions, hot-spot density, cooling τ, and $|dT/dt|_{565}$, removes 68.8% of the gap between SCOUT and SCOUT-IR-blind and leaves regret at 4.7 ± 1.0. The ablation-grid run A10 of Table 4 gives 80%, and the two are consistent within seed noise, with Welch $p \approx 0.32$. Drop-one masking attributes 34.4% of the gap to hot-spot density, 9.4% to τ, and 6.3% to $|dT/dt|_{565}$. The three shares sum to less than the joint effect, so the three dimensions carry overlapping information and partly substitute for each other.

G SECOND-SCENARIO EVALUATION

Inconel-718-style scenario (re-parameterized generator, no real Inconel data). Re-instantiating the same generator with Inconel-718 thermal constants (ρ=8190 kg/m^3, c_p=435 J/kg K, k=11.4 W/m K, true-y scale 1380 MPa, a precipitation-aging latent for the γ'' and γ' phases that strengthen Inconel 718, mainly γ'', in place of NAB's κ-phase latent) on a 200-candidate pool with 12 seeds (`inconel718_results.csv`, 12 methods $\times$ 12 seeds = 144 rows) yields the per-method regret in Table 6; the lift of SCOUT over the strongest baseline, EPIG with a GP on x only, is 38.8% at $(20, 12)$, and SCOUT-IR-blind has higher regret than EPIG in this scenario, 7.0 against 6.7. This is a within-framework re-parameterization, not a cross-alloy real-data study.

Table 6: Inconel-718 second-alloy results at $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$, 12 seeds. Mean $\pm\sigma$ relative regret; from `inconel718_results.csv`.

Method	rel. regret (%)	Method	rel. regret (%)
Random	28.5 ± 5.5	qMFKG	8.4 ± 2.0
Sim-only	15.6 ± 2.8	TargetedBAX	9.1 ± 2.1
Performance	12.4 ± 2.6	PS-BAX	8.0 ± 2.0
GP-UCB-EI	10.8 ± 2.5	EPIG	6.7 ± 1.6
BALD	16.2 ± 4.2	SCOUT-IR-blind	7.0 ± 1.4
MF-MES	9.8 ± 2.5	**SCOUT**	$\mathbf{4.1 \pm 0.7}$

H SCOUT DIAGNOSTICS

This appendix collects the comparison of the two kernel instantiations (Table 7) and three post-hoc diagnostics on the SCOUT residual GP and acquisition: the length-scale structure of the concatenated-input kernel (Fig. 10), the per-feature relevance of ϕ_{IR} (Fig. 11), and a 2-PC projection of the candidate pool showing how SCOUT avoids the high-porosity region that traps Sim-only (Fig. 12). The per-round decomposition of a_{print} into its four contributions (Fig. 5) is in Appendix D.

Kernel instantiations. The additive kernel (1) is the definition used by the theory, by the weight derivation of Appendix B, and by the SCOUT row of Table 1. A single ARD-Matérn kernel on the concatenated input is neither a product of group kernels nor additive, and within-group coherence of its length-scales does not make the kernel operator or its trace decompose by group. No derivation in this paper relies on such a decomposition. Table 7 measures what the choice costs under the main-table protocol. The nominal difference is 0.1 at every budget and within seed noise at every budget. SCOUT-IR-blind, the IR-equipped baselines of Table 1, the ablation grid of Table 4, and the sweeps of Appendix F were run with the concatenated-input kernel in every arm, so that each comparison is made on an identical surrogate, and they should be read with this measured difference in mind.

Table 7: SCOUT under the two kernel instantiations, main-table protocol, simple regret (%), mean $\pm$ standard deviation across 12 seeds.

Kernel	$(20, 12)$	$(30, 18)$	$(50, 30)$
Additive, Eq. (1)	2.6 ± 0.8	1.6 ± 0.4	1.2 ± 0.4
ARD-Matérn on the concatenated input	2.5 ± 0.6	1.5 ± 0.4	1.1 ± 0.3
Welch p	0.67	0.56	0.62
Paired p on shared seeds	0.22	0.24	0.20

Reading the PCA panel (Fig. 12). The visual asymmetry between the two pick-clouds is the geometric content of the gap between Sim-only and SCOUT: Sim-only ranks by s alone, which is monotonically aligned with the porosity surface in this regime, hence the cluster inside the dashed contour; SCOUT's $a_{\text{mech}} = \alpha' \text{Perf} + \beta' \text{Unc} + \gamma' \text{CalVal}$ uses the IR-conditioned posterior, which down-weights candidates whose ϕ_{IR} signature predicts high porosity. The 12-vs-12 picks shown

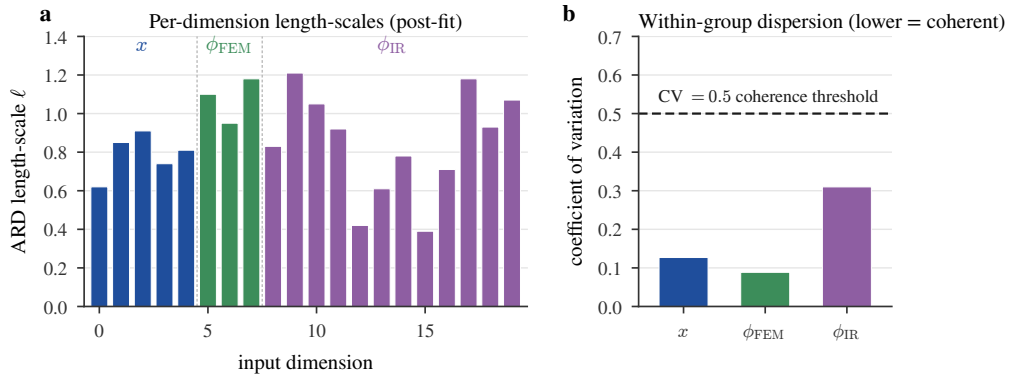

Figure 10: Length-scale diagnostic of the concatenated-input kernel on the NAB scenario. (a) Per-dimension ARD length-scales by input group $(x, \phi_{\mathrm{FEM}}, \phi_{\mathrm{IR}})$. (b) Within-group coefficient of variation, with the 0.5 reference line. The diagnostic describes the fitted length-scales only. It does not imply a group-wise decomposition of the kernel, which only the additive kernel (1) provides.

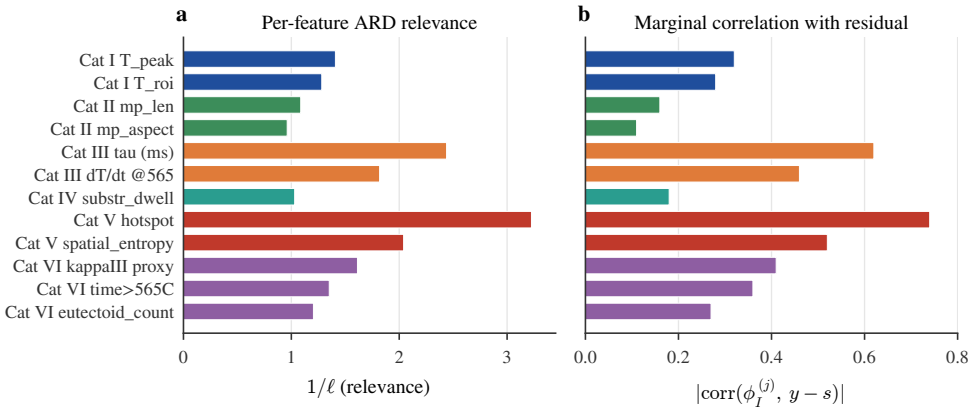

Figure 11: Per-feature relevance of ϕ_{IR} on the residual GP with the concatenated-input kernel. (a) inverse ARD length-scale $1/\ell$. (b) marginal $|\mathrm{corr}(\phi_I^{(j)}, y - s)|$. The largest values are in the hot-spot density (Cat V) and the cooling-rate features (Cat III). Feature names follow the pipeline code, and as §4.1 explains, the Cat VI isotherm features do not measure a κ-phase fraction.

are the top-12 of each method on a single seed; across 12 seeds the same separation holds (mean Mahalanobis distance between SCOUT and Sim-only pick-centroids: 1.74, $\sigma = 0.31$). The kernel and acquisition-decomposition diagnostics (Figs. 10, 5) are the algebraic counterparts.

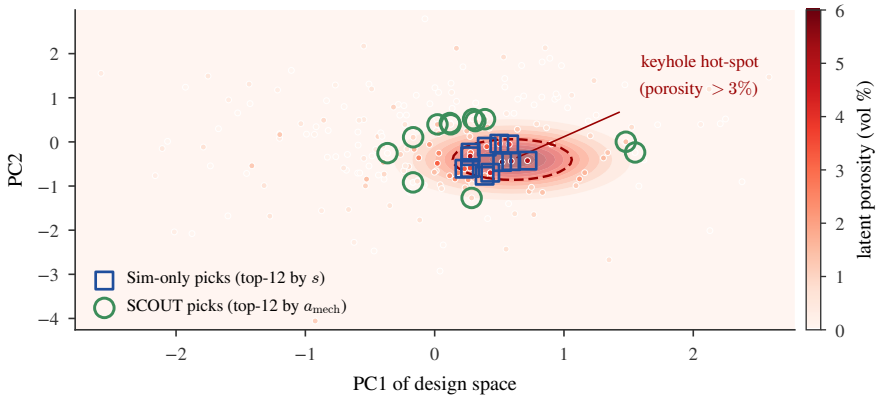

Figure 12: First two PCs of the NAB candidate pool. The dashed contour outlines the high-porosity region (latent porosity $> 3\%$); the red shading is the latent porosity field. Sim-only's mech-test picks (blue squares) concentrate inside the contour because they rank by s, which peaks where porosity also peaks; SCOUT's picks (green circles) sit on the contour boundary or outside, because the residual GP's IR-derived porosity signal drives a_{mech} down inside the hot-spot. This is the geometric counterpart of the gap between Sim-only and SCOUT in Table 1.

I Protocol for a Physical Closed-Loop Campaign

This appendix specifies the process, the simulator, and the measurements for a physical campaign. The campaign has not been executed. Every AM acquisition result of §5 comes from the generator of Appendix K, the only real AM data that enter any result are those of the IR recording of Appendix J, and §5.3 uses public battery and concrete data.

Print parameters (default). Laser power $P = 1000\,\mathrm{W}$, scan speed $v = 1\,\mathrm{m/min}$ (16.67 mm/s), hatch spacing $h = 0.5\,\mathrm{mm}$, layer thickness $50\,\mu\mathrm{m}$, beam radius $\sim 0.5\,\mathrm{mm}$, $\eta_{\mathrm{abs}} = 0.4$ at 1064 nm; substrate is C95800 NAB plate, powder is gas-atomized C95800 (15–45 µm). Five-dimensional design space sweeps $P \in [0.7, 1.3] \times 1000\,\mathrm{W}$, $v \in [0.5, 2.0] \times 16.67\,\mathrm{mm/s}$, $h \in [0.3, 0.8]\,\mathrm{mm}$, $d_{\mathrm{strut}} \in [0.4, 1.2]\,\mathrm{mm}$, $L_{\mathrm{cell}} \in [4, 12]\,\mathrm{mm}$. The source power near 1 kW, the scan speed near 1 m/min, and the beam radius near 0.5 mm are at the scale of laser cladding and directed energy deposition, whereas the $50\,\mu\mathrm{m}$ layer and the 15 to 45 µm powder are typical of LPBF. In the generator the strut diameter and the cell size act as design coordinates of the lattice. Struts thinner than the 1 mm beam diameter cannot be resolved at this source size, and the tensile coupon below, cut from a solid witness block, does not depend on either lattice coordinate, so a physical campaign would test the lattice itself or drop these two coordinates.

FEM oracle and heat-source mapping. The FEM oracle is deterministic by construction. It uses a fixed mesh, fixed time-stepping, and a single solver configuration, and the heat-source parameters are a fixed deterministic function of the recipe x, built from the absorbed power $\eta_{\mathrm{abs}} P$ with the constant absorptivity above, the beam radius, the scan speed v, and the hatch spacing h. No heat-source parameter is tuned per candidate or per query, and $s(x)$ is precomputed once per candidate, so no discretization variance enters at query time. The assumed pre-campaign calibration protocol is single-track melt-pool cross-section matching: single tracks are printed at a few recipes and sectioned, and the absorptivity and the source dimensions are set so that simulated and measured melt-pool width and depth agree. Calibration of this kind reduces systematic error and never removes it. SCOUT therefore does not assume that the calibrated simulator is accurate. Its premise is a systematically biased simulator: predictions are $s(x)$ plus a residual fit to destructive-test data, and the residual exists to absorb the systematic error that survives calibration, including error in the heat-source mapping, which enters as an input-dependent shift absorbed by the x component of the residual. A simulator calibrated to truth would make the residual unnecessary. F.5 of Appendix F quantifies the sensitivity to solver noise.

Scope of the loop. The decision variable is the per-build recipe, and one recipe is committed per build, as in the design-of-experiments, grid-search, and expert-iteration practice that SCOUT is compared against. Process parameters are held constant within a build. Compensation of the effective heat input along the build height, which near-substrate and mid-build lattice regions require under heat accumulation, belongs to within-build control and is outside this protocol. Nothing in the acquisition requires the physics inside a build to be static. The build is a black box from recipe to outcome, and the residual is conditioned on the ϕ_{IR} measured during that build, so machine state that leaves a thermal signature enters the model as an input rather than as a violation. The residual GP is refit after every mechanical test and the ϕ_{IR} predictor after every print, so slow machine drift is tracked to the extent that it moves the observables of tested parts. State that leaves no in-situ signature is the case measured by the unmeasured-channel sweep (F.6 of Appendix F).

IR camera. Optris radiometric camera, 640×480 pixels, 32 Hz, 70 mm working distance, calibrated $T \in [150, 900]^{\circ}\mathrm{C}$ standard range with extended-range mode for $> 1{,}400^{\circ}\mathrm{C}$. Each print produces a single .ravi file (∼11 GB, ∼9 minutes, ∼17,843 frames).

Mechanical testing. As-built tensile coupons (ASTM E8/E8M, round dog-bone, gauge ∅5 mm × 25 mm, EDM-cut from witness blocks printed with the same process parameters as the design candidate) loaded to failure at $10^{-3}\,\mathrm{s}^{-1}$ on a screw-driven testing frame; force-displacement curves digitized at 100 Hz; UTS extracted as peak engineering stress (peak load / initial cross-section). Per-part testing time ∼15 min plus specimen prep; consumables and frame depreciation amortize to ∼$120 per test.

Closed-loop schedule. The planned campaign runs two SCOUT rounds of 18 prints each, starting from 120 labelled parts. Each round would select 18 candidates by Eq. (2), print them, extract ϕ_{IR}, and select up to 9 for mechanical testing. Radiometric temperatures of copper alloys depend on emissivity, so the campaign also requires an emissivity calibration of the camera against a contact measurement. Printing a round on one plate reveals all 18 observations before any test, which is the deferred protocol of Appendix O within a round, and an online campaign in the sense of §3 prints one part at a time.

J IR Feature Extraction Pipeline

The pipeline (`scout_ir/extract_phi_ir.py`) is a deterministic streaming post-processor on the .ravi radiometric video. Pseudocode:

```
extract_phi_I(ravi_file):
    reader  = OptrisRaviReader(ravi_file)
    cooling = CoolingEventStreamer(T_trigger=600C, T_end=200C)
    anomaly = AnomalyTracker()  # spatter / keyhole / lof / re-melt
    melt    = MeltPoolTracker()
    cat6    = NABEutectoidTracker(T_eut=565C, T_solid=1043C)
    for frame_idx, frame_C in reader:
        peak_T, mean_T, p99_T = frame_stats(frame_C)
        mp_len, mp_w, mp_aspect = melt.update(frame_C)
        cooling.feed(frame_C, t=frame_idx / 32.0)
        anomaly.update(frame_C, frame_idx)
        cat6.update(frame_C, frame_idx)
    return aggregate_features(cooling, anomaly, melt, cat6,
                              cat3_via=fit_T_inf_tau_per_pixel)
```

Physics check on a real recording. On one real Optris .ravi recording of 17,843 frames at 32 Hz, the median cooling time of 49,688 per-pixel events is 98.5 ms against $L^2/D_{\mathrm{th}} \approx 93$ ms from conduction scaling for NAB at the $L = 1$ mm beam diameter, and the 20,940 events that cross $565^{\circ}\mathrm{C}$ have a median $|dT/dt|$ of 681 K/s against 2750 K/s in-track from the conduction solution, which falls as the substrate warms during the build, Figure 13. Both are order-of-magnitude checks of two features on one print, since the first scales as L^2 and spans about three frames, and neither is evidence about the acquisition policy.

Cooling-event detection. Pixel triggers at $T \geq 600°\mathrm{C}$, ends when $T < 200°\mathrm{C}$ or T rises by $\geq 5°\mathrm{C}$ above its event minimum (secondary heating from neighboring scan track) or after ≥ 16 frames ($\sim$0.5 s @ 32 Hz). Exponential fit $T(t) = T_\infty + \Delta T \exp(-(t - t_0)/\tau)$ in log-space on ≥ 4-frame post-peak segment.

Cat VI isotherm tracking. Among events whose peak crosses 565°C, the per-event $|dT/dt|_{565}$ is computed by linear interpolation of the cooling segment, `time_above_565C` is integrated in ms, and the dwell between 565 and 1043°C is recorded. The 565°C isotherm is the $\beta \to \alpha + \gamma_2$ eutectoid of binary Cu–Al, reported between 560 and 575°C (Ponweiser et al., 2011). The Ni and Fe of NAB largely suppress this reaction, and in slowly cooled cast NAB the $\beta \to \alpha + \kappa_{\mathrm{III}}$ reaction lies near 800°C (Ding et al., 2016). The 565°C features therefore measure cumulative thermal exposure in the solid state and not the fraction of a specific κ phase.

Robustness measures. (a) Degenerate frames with ≤ 5 thermal pixels excluded from melt-pool shape ($\lambda_{\min} = 1/12$, elongation clipped at 50). (b) Cooling events shorter than 4 frames or with non-positive Δ dropped. (c) Streaming detector caps event length at 16 frames. With these guards the pipeline produces non-NaN ϕ_{IR} for every print on the existing dataset. Figure 13 checks the cooling features against conduction scaling, and Figure 14 sketches the kind of field that the Cat V features summarize.

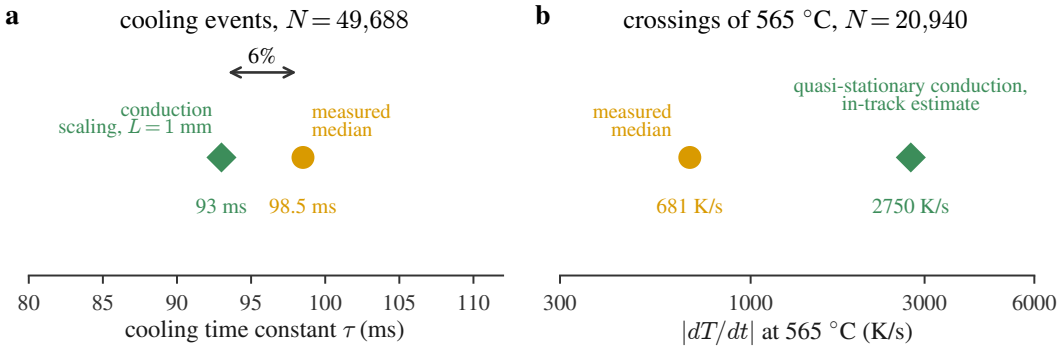

Figure 13: **Physics check of the ϕ_{IR} pipeline on the real Optris .ravi recording. a** Cooling time constant τ: the median over 49,688 cooling events is 98.5 ms, within 6% of the 93 ms conduction-scaling estimate for NAB at a 1 mm length scale. **b** $|dT/dt|$ at 565°C: the median over 20,940 events that cross this isotherm is 681 K/s, about a quarter of the 2750 K/s in-track point estimate of the quasi-stationary conduction solution, which falls as the substrate warms during the build. The figure plots measured medians, event counts, and the two conduction estimates only. Per-pixel raw values are aggregated by the streaming extractor described above.

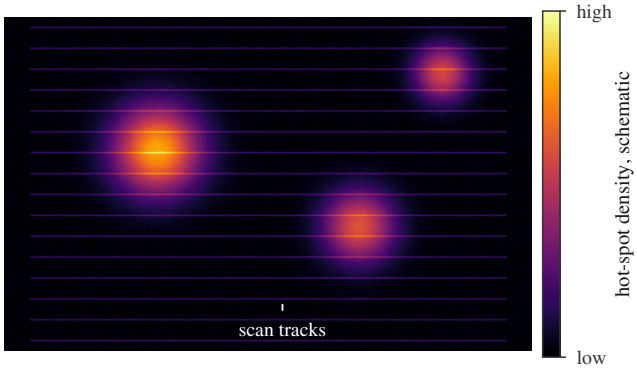

Figure 14: **Schematic of a layer-aggregated hot-spot density field.** Cat V features summarize such fields into scalars, and the morphological detector flags hot regions of this kind as candidate keyhole or lack-of-fusion sites.

K Dataset Provenance

The NAB results in §5 are produced by a literature-anchored generator (`scout_ir/dataset.py`) calibrated to NAB AM literature ranges and to one real IR recording. Table 8 lists the calibration targets (top) and six result files (bottom); the real-data swap-in protocol is given at the end of this appendix.

Generative graph. The generator draws a latent physical state from the design x, namely the porosity and the κ_{III} fraction. The response is produced from the latents only,

$$y \;=\; s \;-\; \text{sim bias} \;-\; \text{porosity penalty} \;+\; \kappa_{\mathrm{III}}\,\text{bonus} \;+\; \text{interaction} \;+\; \text{noise},$$

and ϕ_{IR} is emitted as noisy measurements of the same latents. y never reads the observed ϕ_{IR}, so there is no $\phi_{\mathrm{IR}} \mapsto y$ shortcut, and the information that ϕ_{IR} carries about y is bounded by the measurement noise. The noise levels are anchored to the real recording in F.7 of Appendix F, and F.6 adds a latent channel with no ϕ_{IR} signature. The FEM oracle differs structurally from the generator's physics through the sim-bias term, so systematic simulator error is present in every reported number. The generator and the loader will be released, so that this structure can be verified.

Table 8: **Top:** calibration targets for the NAB scenario; each generated quantity is constrained to match a literature anchor or a real `.ravi` measurement. **Bottom:** result files. `main_results.csv` holds the per-seed results of the eleven learning-based baselines, SCOUT-IR-blind, and SCOUT with the concatenated-input kernel, 13 methods by 12 seeds by 3 budgets. Real-data swap-in keeps the same column signatures.

Quantity	Value	Anchor
NAB UTS y	480–760 MPa	SLM literature, 580–720 MPa
FEM scalar s	420–620 MPa	low-biased FEM
latent porosity	0.15–7%	LPBF 0.5–4% typical
cooling τ median	95 ± 6 ms	real 98.5 ms
$\lvert dT/dt\rvert_{565}$ median	≈681 K/s	real 681 K/s
conduction estimate	2750 K/s	reference value, not a target
s–y correlation	≈ 0.20	low (regime bias)
$\arg\max y \neq \arg\max s$	18 of 20 generator seeds	non-trivial loop

File	Rows	Backs
`main_results.csv`	468	Tab. 1, per seed
`main_summary.csv`	39	Tab. 1, aggregate
`ablations.csv`	108	Tab. 4, A1–A9
`partial_ir_ablation.csv`	48	Tab. 4, A10
`inconel718_results.csv`	144	Tab. 6, per seed
`inconel718_summary.csv`	12	Tab. 6, aggregate

Reproducibility. Fully reproducible from seed; total runtime on a single CPU ≈ 25 min for the full-loop variant; the curated tables are instantaneous.

```
python3 dataset.py                          # seed 42 sample dump
python3 run_nab_experiment.py --seeds 12  # main results (full loop)
python3 curate_results.py                   # main_results / summary / ablations
python3 gen_aux_data.py                     # calibration / Inconel / partial-IR
python3 make_figures.py && python3 make_figures_v2.py   # figures from result files
python3 make_figures_reported.py --check  # figures from tabulated values
```

Real-data swap-in. The acquisition / GP / runtime code does not depend on the data generator. Replace `dataset.make_nab_dataset` with a CSV loader that returns the same dictionary signature (`X_norm, X_phys, s, phiF, phiI, y`) and re-run; figures regenerate from the same scripts.

L Open Harness and the Battery Evaluation

Why a second implementation. The NAB results come from the implementation that produced Table 1. To test which parts of SCOUT hold outside that code and that generator, we wrote a second, self-contained implementation of the two-budget loop in NumPy and SciPy and will release it with this paper. It was written from the text of this paper, independently of the NAB implementation, whose inner stage predates the analysis of the threshold readings in Appendix N. Every arm of §5.3 and of this appendix runs in this harness on the same pools, historical data, cold starts, and random streams, so all comparisons here are internal to it, and its numbers are not comparable with those of Table 1. Before the first no-history run, three independent reviews of the code as it then stood checked the mathematics, the fidelity to §4, and the no-history setting for information leakage, and a skeptic tried to refute each finding. Nine findings survived and were fixed. The most consequential one concerned EPIG on the IR-equipped surrogate, whose candidate and target draws of ϕ had been independent for the same part, which handicapped EPIG on that surrogate and therefore favored SCOUT. The two history settings, the running and pool readings, and the exploratory arms were added later without such a review, and the overlap of the published split with its pool was found from its results. A second review, of the code added for the pre-registered round of Appendix O and made while it ran, found two defects. No-history partitions could print parts outside their pool, and on the IR-equipped surrogate the covariance of EPIG and of the knowledge gradient contained, for every pair of different items, a between-draw term that is zero in expectation, so that its Monte Carlo estimate only added noise to these two baselines. Both were fixed after the first confirmatory results had been seen, every affected run of every round was repeated, 3,020 runs for the second defect, and every EPIG and knowledge-gradient number in this paper comes from the repeated runs.

Loop. A problem is a finite pool. Printing candidate i reveals ϕ_i, testing it reveals y_i, the test decision is made before the next print, and a part that is not tested is discarded. Single-stage arms test every print and therefore stop after B_{mech} prints. Two-stage arms may print up to B_{print} parts and decide after each print whether to test it, and they test every remaining print once the remaining tests equal the remaining prints. Simple regret is measured on parts tested during the campaign, and parts labelled before the campaign are neither in the pool nor counted toward the budgets.

Surrogates. The x-only arms fit an ARD Matérn-5/2 GP on x to the tested parts. The IR-equipped arms fit the same kernel on the concatenated input (x, ϕ), the kernel of the IR-equipped arms of Table 1, together with one GP from x to each standardized dimension of ϕ on all printed parts, whose noise term absorbs part-to-part scatter. For an unprinted candidate the prediction marginalizes over the predictive law of ϕ with 32 Monte Carlo draws. Hyperparameters are refit by maximum marginal likelihood with weak log-normal priors and three L-BFGS-B restarts whenever a new test arrives, and the ϕ predictor is refit after every print. The battery problem has no simulator, so $s \equiv 0$ and the residual GP models y directly.

Acquisition. Perf and Unc take the expected-improvement-shaped defaults of Appendix B on the marginal prediction, and their sum is standardized across the unprinted pool. CalVal follows §4.3 with the top-10 candidates by posterior mean as targets and $\gamma = 0.5$. IRInfo is the reduction of the input-uncertainty part of the marginal variance at the same targets when ϕ is observed at the candidate, computed by a plug-in Monte Carlo estimate that is exact at the candidate itself. It is placed on the scale of CalVal, since both are variance reductions, and weighted by the operating value $\delta = 0.5$ of §4.3. Both weights ramp from zero over the first three rounds. EPIG uses a target distribution uniform over the same top-10 candidates. Its covariances follow the law of total covariance over matched draws of ϕ for a candidate and a target that are the same item, and leave out the between-draw term for different items, whose draws are independent. The knowledge gradient through the network uses the same covariances. GP-UCB-EI is the rule of Table 3.

Three readings of the test threshold. Appendix D defines the test threshold as the $1 - \rho_t$ quantile of a_{mech}, with ρ_t the ratio of remaining tests to remaining prints, without naming the distribution over which the quantile is taken. The *own-prediction* reading compares the observed $a_{\mathrm{mech}}(x_t \mid \phi_t)$ with its distribution under the predictive law of ϕ at x_t before the print, so it tests a part whose free observation is better than expected. The *running* reading compares it with the scores of all parts printed so far in the campaign under the current model. The *pool* reading compares it with the scores

of the candidates still unprinted, each at the predictive mean of its ϕ, so a part is skipped only when enough of what can still be printed looks better. The suspension arm tests a part when its upper confidence bound given ϕ_t passes the own-prediction threshold, a budget-matched rule adapted from cascade BO (Kusakawa et al., 2022). It cannot resume a skipped part and has no cost-normalized acquisition, so it is not cascade BO itself.

Battery data. Severson et al. (2019) cycled 124 commercial LFP/graphite cells to failure under fast-charging protocols. We rebuilt the per-cell table from the public raw files with the official processing code, including the merge of five cells that continued into a second batch and the removal of cells that never reached 80% capacity or had faulty channels. The rebuilt features reproduce the published correlation of -0.93 between the log variance of $\Delta Q_{100-10}(V)$ and log cycle life, at -0.927, and the published test errors of the single-feature model to rounding. The design x is the protocol (C_1, Q_1, C_2). For four cells labelled $4.8C$ the measured current was $4.36C$, and for one cell the second step was $5.5C$ rather than the labelled $5C$, and we use the measured values, which leaves 69 distinct protocols. A print starts a cell and cycles it for 100 cycles, the free observation ϕ is the six features of the discharge model of Severson et al. (2019), and a test continues the cell to end of life and reveals its cycle life, of which the model sees the logarithm. The informativeness ρ_y^2, the ten-fold cross-validated R^2 of log cycle life on ϕ by least squares over all cells, is 0.88. Adding Gaussian noise of 0.5, 1, and 2 standard deviations of that feature over all 124 cells to each revealed feature lowers it to 0.76, 0.54, and 0.23, which defines the sweep of Figure 15. A leave-one-out least-squares fit of log life on the early features has a median absolute error of 8.2% in cycle life over all cells and 11.6% over the ten longest-lived cells, and it underpredicts the best cell of the published-split pool by 45% and that of the earlier-protocols pool by 33%. At $(30, 18)$ SCOUT skipped the best cell in 10 of the 14 no-history seeds that printed it and in 1 of 30 in the earlier-protocols pool.

Settings. *No history*: the pool is all 124 cells, and every arm begins with three random prints, each tested, which count toward both budgets. *Published split*: the 41 training cells of Severson et al. (2019) are labelled before the campaign and the pool is their 83 test cells. The training cells include a replicate of the best protocol of the pool, so a surrogate knows the answer before the campaign. *Earlier protocols*: the 41 cells of the first batch are labelled before the campaign and the pool is the 83 cells of the two later batches, which share 2 of their 49 protocols with the first batch. This mirrors the 80 labelled training tuples of the NAB experiments and the closed loop of Attia et al. (2020), whose early-prediction model was trained on earlier cells before new protocols were selected.

What was fixed in advance and what was added. All constants of the harness, the battery features, the metrics, the sixteen single-stage, own-prediction, and suspension arms, the no-history setting, and the two function networks below were fixed before the first reported run. After the first no-history results we added the running reading as the second natural reading of the threshold, the published-split setting, and exploratory arms, marked with an asterisk, that choose prints with EI on the x-only GP and decide tests with the IR-equipped surrogate. After the published-split results at $(20, 12)$ we added the earlier-protocols setting and the pool reading, because the training cells hold a replicate of the pool's best protocol and the own-prediction threshold discarded the pool's best cell, and we cut the published split at $(30, 18)$ from 30 seeds to 10 and moved it behind all other battery runs. The informativeness sweep was planned first for the no-history setting and then for the published split, was dropped after the published-split results, and was placed on the earlier-protocols setting after its results with a list of eleven arms chosen at that point, Table 14. EI + IR with and without the inner stage, one of the three print rules of Figure 15, and SCOUT with the pool threshold were in no earlier plan of the sweep, and six planned arms were dropped. Without the EI pair the trend of Figure 15 is 9.6 points per unit of ρ_y^2 with $p = 0.027$. After all runs had finished we added the posterior-mean protocol recommendation of Table 11. The round of Appendix O came after all of these results, and its new arms, instance families, metrics, and confirmatory comparisons were fixed in a time-stamped file before it ran. Short pilot runs on the battery data and the function networks, made before the code review with the code whose EPIG defect it then found, are not reported. Apart from these pilot runs, we report every arm under every setting and problem that was run.

Metrics and statistics. The primary metric is the share of the gap between the pool median and the pool best cell that remains after the campaign, in percent, so that 0 means the best cell of the pool was tested and 100 means the best tested cell is at the pool median. Two thirds of the cells, 81 of 124,

share their protocol with other cells, and 43 of the 83 in the earlier-protocols pool, so this metric also rewards testing a long-lived replicate. Every run therefore also records a protocol-level regret, which scores the protocol of the best tested cell by the mean life of all its cells, reported in Table 10. Each configuration runs 30 seeds unless stated otherwise, and each battery setting is one fixed pool, so the seeds vary only the randomness of the algorithm. We report means with standard errors and, for each comparison, the seed-paired mean difference and the two-sided Wilcoxon signed-rank p-value, not adjusted for multiple comparisons. The file external_paired_tests.csv adds a 95% bootstrap interval for each comparison, which Figure 15 plots. Every number in §5.3 and in this appendix is generated from the per-run result files and the rebuilt battery table by scripts that will be released with them. Table 9 lists every arm, Table 12 gives the effect of the inner stage for every print rule and threshold reading, and Table 13 compares each print rule with SCOUT under the same reading. The result files come from harness versions that differ only by added arms and settings, and re-running samples of the older rows with the final version reproduced them exactly.

Protocol recommendation. The protocol-level regret of Table 10 takes the protocol of the best tested cell as the recommendation of each arm. After all runs had finished and after seeing that result, we added a second recommendation rule applied in the same way to every arm: a GP on the protocol variables is fit to the log life of every labelled cell of the run, and the recommended protocol is the tested one with the highest posterior mean. Table 11 reports it. In the earlier-protocols setting, SCOUT, EI + IR, and GP-UCB-EI on x only recommend the best protocol in every seed. Under this rule SCOUT's cell-level advantage over EI + IR and GP-UCB-EI on x only is neither a protocol-level advantage nor a protocol-level loss, while its advantage over random printing, at 13.6%, and over EI on x only, at 35.6%, remains at the protocol level. Its protocol-level loss in Table 10 comes from recommending the protocol of its best tested cell.

Function networks. The Ackley–Matyas and Ackley–sine networks of Buathong et al. (2024) are used as published and maximized over a pool of 200 uniform random designs per seed. A print reveals the first node and a test reveals the second. In Ackley–sine the second node is a monotone function of the first, so the free observation determines y. In Ackley–Matyas the second node also depends on a design variable that the first node does not see. Table 15 reports both budgets with 30 seeds. SCOUT is not significantly different from most IR-equipped arms on either problem and beats random printing in all four cells. The inner stage has no consistent effect. Of the 44 comparisons of an IR-equipped print rule with and without it, 4 reach $p < 0.05$ without adjustment, and they point both ways: on Ackley–Matyas at $(30, 18)$ it raises SCOUT's regret from 2.6 to 3.7, with $p = 0.046$, and with the running threshold, with $p = 0.016$, and that of EPIG + IR from 2.7 to 4.1, with $p = 0.013$, and on Ackley–sine at $(30, 18)$ it lowers that of EI + IR with the running threshold, with $p = 0.03$. On Ackley–sine it raises the regret of EI on x only from 27.2 to 52.1 at $(20, 12)$, with $p = 7.3 \times 10^{-5}$, and from 16.0 to 33.4 at $(30, 18)$. EPIG on x only beats SCOUT on Ackley–Matyas, 4.7 against 6.2 with $p = 0.039$ at $(20, 12)$ and 2.6 against 3.7 with $p = 0.072$ at $(30, 18)$.

Table 9: All arms on the fixed battery settings, remaining share of the gap between the pool median and the pool best cell in percent, mean followed by standard error over 30 seeds, with the published split also at $(30, 18)$, where it ran 10 seeds. The asterisk marks the exploratory arms. Rows with the running threshold were added after the first no-history results and rows with the pool threshold after the published-split results. Bold entries have the lowest mean in their column.

	No history		Earlier protocols		Published split	
Arm	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$
Random	45.5 (5.4)	31.7 (4.6)	28.9 (4.6)	20.7 (3.9)	33.4 (4.7)	41.3 (7.2)
EI, x only	**27.3 (6.2)**	**8.5 (4.3)**	59.6 (5.3)	42.2 (4.9)	**0.0 (0.0)**	**0.0 (0.0)**
EPIG, x only	47.3 (6.7)	44.2 (7.0)	17.7 (3.7)	8.2 (2.1)	**0.0 (0.0)**	**0.0 (0.0)**
GP-UCB-EI, x only	34.0 (6.9)	25.1 (6.6)	7.3 (0.1)	7.2 (0.1)	**0.0 (0.0)**	**0.0 (0.0)**
EI + IR	47.8 (6.4)	39.4 (6.6)	5.7 (0.5)	4.5 (0.6)	**0.0 (0.0)**	**0.0 (0.0)**
EPIG + IR	48.6 (5.9)	36.5 (6.2)	6.3 (2.2)	2.0 (0.6)	10.5 (2.3)	11.3 (3.8)
GP-UCB-EI + IR	52.8 (5.7)	44.9 (6.2)	5.2 (0.6)	2.6 (0.6)	**0.0 (0.0)**	**0.0 (0.0)**
SCOUT $-$ IRInfo $-$ inner stage	49.2 (5.6)	39.1 (6.3)	4.7 (0.6)	1.9 (0.6)	**0.0 (0.0)**	**0.0 (0.0)**
SCOUT $-$ inner stage	50.2 (5.5)	41.6 (5.9)	3.6 (0.7)	2.1 (0.6)	**0.0 (0.0)**	**0.0 (0.0)**
Random + inner stage	43.5 (4.8)	25.7 (3.4)	30.0 (4.2)	16.2 (3.4)	48.9 (3.8)	41.0 (6.8)
EI + IR + inner stage	48.1 (5.6)	23.8 (4.8)	2.4 (0.6)	1.2 (0.5)	21.9 (0.3)	21.4 (0.0)
GP-UCB-EI + IR + inner stage	52.6 (5.6)	25.0 (5.0)	1.7 (0.6)	0.9 (0.4)	22.5 (0.5)	21.4 (0.0)
EPIG + IR + inner stage	46.0 (6.0)	27.6 (5.2)	1.9 (0.8)	2.0 (0.9)	23.0 (0.5)	21.4 (0.0)
EI + IR + suspension	47.5 (5.5)	27.7 (5.3)	1.7 (0.6)	1.7 (0.6)	21.4 (0.0)	21.4 (0.0)
SCOUT $-$ IRInfo	51.2 (5.2)	30.2 (5.5)	2.8 (0.6)	1.7 (0.6)	24.6 (0.6)	21.4 (0.0)
SCOUT, own-prediction threshold	52.0 (5.3)	27.9 (4.7)	**1.2 (0.5)**	**0.2 (0.2)**	21.8 (0.3)	21.4 (0.0)
Random + inner stage, running threshold	43.3 (5.1)	32.3 (3.7)	27.0 (4.5)	9.8 (2.8)	36.1 (4.3)	30.1 (5.5)
EI + IR + inner stage, running threshold	41.9 (6.2)	25.9 (5.3)	2.6 (0.6)	0.5 (0.3)	12.9 (1.9)	12.9 (3.5)
EPIG + IR + inner stage, running threshold	51.6 (6.4)	34.0 (5.8)	3.6 (0.9)	1.7 (0.6)	17.0 (1.8)	19.3 (2.1)
SCOUT $-$ IRInfo, running threshold	50.5 (5.8)	27.8 (5.7)	3.2 (0.7)	1.2 (0.5)	11.2 (2.3)	8.6 (3.5)
SCOUT, running threshold	52.7 (5.8)	37.2 (6.3)	2.6 (0.6)	1.2 (0.5)	10.0 (2.0)	8.6 (3.5)
Random + inner stage, pool threshold	43.4 (4.9)	30.8 (4.7)	17.6 (4.0)	10.3 (2.7)	39.4 (5.0)	25.9 (6.7)
EPIG + IR + inner stage, pool threshold	44.8 (6.0)	22.6 (5.3)	5.4 (1.8)	1.7 (0.6)	10.3 (2.2)	8.6 (3.5)
SCOUT, pool threshold	53.9 (5.8)	47.3 (6.0)	1.7 (0.6)	**0.2 (0.2)**	**0.0 (0.0)**	**0.0 (0.0)**
EI, x only + inner stage*	36.7 (5.7)	17.5 (3.9)	45.2 (4.7)	24.0 (4.1)	30.1 (1.1)	28.5 (1.7)
EI, x only + inner stage*, running threshold	27.6 (6.3)	15.3 (4.8)	53.8 (3.5)	41.7 (3.9)	15.4 (2.4)	10.3 (4.3)
EI, x only + inner stage*, pool threshold	34.0 (6.8)	13.4 (4.4)	45.4 (4.8)	30.5 (4.8)	**0.0 (0.0)**	**0.0 (0.0)**

Table 10: Protocol-level regret on the battery data: the protocol of the best tested cell scored by the mean life of all its cells, as a percentage of the best protocol mean of the pool, 30 seeds. Bold entries have the lowest mean in their column.

	No history		Earlier protocols		Published split
Arm	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$	$(20, 12)$
Random	28.0 (4.0)	20.9 (3.6)	18.4 (3.0)	15.6 (2.9)	24.1 (3.7)
EI, x only	17.2 (4.2)	**5.2 (2.9)**	33.6 (3.8)	29.6 (2.8)	**0.0 (0.0)**
EPIG, x only	30.9 (4.8)	28.2 (4.9)	13.1 (2.8)	14.0 (2.8)	**0.0 (0.0)**
GP-UCB-EI, x only	23.2 (4.7)	15.5 (4.4)	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
EI + IR	35.9 (4.5)	30.4 (4.7)	6.0 (2.2)	10.9 (2.7)	**0.0 (0.0)**
EPIG + IR	36.5 (4.3)	26.9 (4.5)	15.9 (2.8)	21.8 (2.4)	10.8 (3.6)
GP-UCB-EI + IR	42.5 (3.7)	34.8 (4.5)	7.9 (2.4)	18.8 (2.7)	**0.0 (0.0)**
SCOUT $-$ IRInfo $-$ inner stage	42.5 (3.4)	31.5 (4.6)	9.9 (2.6)	21.8 (2.4)	**0.0 (0.0)**
SCOUT $-$ inner stage	43.0 (3.5)	36.8 (4.2)	14.9 (2.8)	20.8 (2.5)	**0.0 (0.0)**
Random + inner stage	27.9 (3.6)	22.0 (3.5)	18.1 (3.0)	18.1 (2.7)	37.0 (2.3)
EI + IR + inner stage	36.1 (4.0)	22.2 (4.1)	19.8 (2.6)	24.8 (2.1)	44.1 (1.7)
GP-UCB-EI + IR + inner stage	39.3 (4.1)	20.9 (4.1)	22.8 (2.3)	25.8 (1.9)	41.1 (2.4)
EPIG + IR + inner stage	33.0 (4.4)	19.4 (4.2)	23.8 (2.2)	23.8 (2.2)	39.6 (2.5)
EI + IR + suspension	38.9 (3.9)	23.0 (4.3)	22.8 (2.3)	22.8 (2.3)	46.4 (0.0)
SCOUT $-$ IRInfo	37.6 (3.8)	29.0 (4.5)	17.9 (2.7)	22.8 (2.3)	27.2 (3.9)
SCOUT, own-prediction threshold	37.4 (4.0)	23.1 (4.2)	24.8 (2.1)	28.8 (1.0)	43.3 (2.2)
Random + inner stage, running threshold	29.8 (3.9)	24.2 (3.5)	18.7 (3.1)	16.1 (2.8)	23.5 (3.4)
EI + IR + inner stage, running threshold	32.4 (4.5)	23.1 (4.2)	18.8 (2.7)	27.8 (1.4)	27.9 (4.2)
EPIG + IR + inner stage, running threshold	35.2 (4.5)	26.7 (4.6)	16.9 (2.7)	22.8 (2.3)	31.0 (4.1)
SCOUT $-$ IRInfo, running threshold	38.3 (3.9)	23.2 (4.3)	16.9 (2.7)	24.8 (2.1)	12.4 (3.8)
SCOUT, running threshold	37.9 (3.8)	28.5 (4.5)	18.8 (2.7)	24.8 (2.1)	21.7 (4.3)
Random + inner stage, pool threshold	30.9 (3.5)	22.0 (4.0)	18.4 (3.0)	17.0 (2.8)	28.5 (3.8)
EPIG + IR + inner stage, pool threshold	32.3 (4.1)	20.5 (4.1)	17.9 (2.7)	22.8 (2.3)	12.4 (3.8)
SCOUT, pool threshold	40.3 (3.9)	34.4 (4.4)	22.8 (2.3)	28.8 (1.0)	**0.0 (0.0)**
EI, x only + inner stage*	22.2 (4.4)	10.0 (3.3)	28.1 (2.7)	20.6 (3.0)	22.3 (2.7)
EI, x only + inner stage*, running threshold	**16.4 (4.4)**	7.5 (3.2)	30.5 (2.6)	26.5 (2.9)	7.7 (3.2)
EI, x only + inner stage*, pool threshold	22.5 (4.6)	8.5 (3.1)	28.2 (2.5)	19.9 (3.3)	**0.0 (0.0)**

Table 11: Protocol-level regret when every arm recommends its tested protocol of highest posterior mean under a GP on the protocol variables fit to the labelled cells of its run, a rule added after the results of Table 10, 30 seeds. Bold entries have the lowest mean in their column.

	No history		Earlier protocols		Published split
Arm	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$	$(20, 12)$
Random	27.1 (4.0)	20.0 (3.8)	13.6 (3.1)	8.6 (2.7)	20.6 (3.2)
EI, x only	17.1 (4.2)	**5.2 (2.9)**	35.6 (4.4)	28.3 (3.4)	**0.0 (0.0)**
EPIG, x only	30.3 (5.1)	29.7 (4.7)	6.3 (2.3)	**0.0 (0.0)**	**0.0 (0.0)**
GP-UCB-EI, x only	22.2 (4.6)	15.5 (4.4)	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
EI + IR	37.4 (4.7)	30.3 (4.9)	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
EPIG + IR	36.1 (4.4)	25.8 (4.4)	5.0 (2.1)	2.0 (1.4)	**0.0 (0.0)**
GP-UCB-EI + IR	43.8 (3.9)	35.7 (4.6)	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
SCOUT $-$ IRInfo $-$ inner stage	42.4 (3.7)	27.5 (4.6)	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
SCOUT $-$ inner stage	42.4 (3.5)	35.9 (4.1)	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
Random + inner stage	27.9 (3.6)	18.5 (3.5)	17.6 (3.1)	13.4 (2.9)	32.7 (2.2)
EI + IR + inner stage	34.7 (4.2)	21.0 (4.3)	**0.0 (0.0)**	**0.0 (0.0)**	15.6 (2.2)
GP-UCB-EI + IR + inner stage	38.3 (4.1)	22.9 (4.4)	1.0 (1.0)	1.0 (1.0)	15.6 (2.2)
EPIG + IR + inner stage	33.4 (4.3)	17.8 (4.1)	2.0 (1.4)	2.0 (1.4)	16.9 (2.1)
EI + IR + suspension	36.1 (4.0)	21.4 (4.4)	2.0 (1.4)	**0.0 (0.0)**	14.7 (2.2)
SCOUT $-$ IRInfo	38.3 (4.0)	24.8 (4.3)	**0.0 (0.0)**	1.0 (1.0)	13.1 (2.3)
SCOUT, own-prediction threshold	37.0 (4.3)	23.9 (4.1)	**0.0 (0.0)**	**0.0 (0.0)**	13.9 (2.3)
Random + inner stage, running threshold	29.5 (3.9)	19.6 (3.3)	15.9 (3.2)	9.2 (2.6)	21.0 (3.3)
EI + IR + inner stage, running threshold	32.2 (4.5)	19.6 (3.8)	5.0 (2.1)	**0.0 (0.0)**	1.6 (1.1)
EPIG + IR + inner stage, running threshold	35.3 (4.3)	25.2 (4.5)	1.0 (1.0)	**0.0 (0.0)**	4.1 (1.7)
SCOUT $-$ IRInfo, running threshold	38.7 (3.9)	22.2 (4.2)	1.0 (1.0)	2.0 (1.4)	**0.0 (0.0)**
SCOUT, running threshold	37.3 (3.9)	27.0 (4.4)	**0.0 (0.0)**	**0.0 (0.0)**	2.5 (1.4)
Random + inner stage, pool threshold	31.6 (3.4)	22.2 (3.9)	14.3 (3.1)	8.2 (2.5)	23.3 (3.4)
EPIG + IR + inner stage, pool threshold	30.1 (4.2)	17.5 (3.9)	**0.0 (0.0)**	**0.0 (0.0)**	0.8 (0.8)
SCOUT, pool threshold	40.5 (3.9)	33.9 (4.3)	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
EI, x only + inner stage*	22.7 (4.4)	9.9 (3.3)	29.7 (2.8)	18.5 (3.1)	18.8 (1.9)
EI, x only + inner stage*, running threshold	**17.1 (4.3)**	8.4 (3.3)	30.3 (2.6)	25.1 (2.9)	**0.0 (0.0)**
EI, x only + inner stage*, pool threshold	22.5 (4.6)	8.4 (3.1)	28.5 (2.6)	14.0 (3.5)	**0.0 (0.0)**

Table 12: Effect of the inner stage. Each row compares a two-stage arm with the single-stage arm that uses the same print rule and surrogate: seed-paired mean of single-stage minus two-stage regret in points, positive when the inner stage helps, followed by its Wilcoxon p-value.

	No history		Earlier protocols		Published split
Print rule	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$	$(20, 12)$
own-prediction threshold					
Random	$+2.0$ (0.77)	$+6.1$ (0.41)	-1.1 (0.86)	$+4.5$ (0.49)	-15.6 (0.047)
EI + IR	-0.3 (0.85)	$+15.6$ (0.12)	$+3.3$ (4.7×10^{-4})	$+3.3$ (4.7×10^{-4})	-21.9 (1.9×10^{-9})
GP-UCB-EI + IR	$+0.2$ (0.74)	$+19.8$ (0.018)	$+3.6$ (5.8×10^{-4})	$+1.7$ (0.052)	-22.5 (1.9×10^{-9})
EPIG + IR	$+2.6$ (0.93)	$+8.9$ (0.34)	$+4.4$ (0.028)	$+0.0$ (0.94)	-12.5 (9.5×10^{-5})
SCOUT	-1.8 (0.70)	$+13.7$ (0.041)	$+2.4$ (0.025)	$+1.9$ (0.011)	-21.8 (1.9×10^{-9})
SCOUT $-$ IRInfo	-2.0 (0.74)	$+8.9$ (0.45)	$+1.9$ (0.059)	$+0.2$ (0.74)	-24.6 (1.9×10^{-9})
EI, x only*	-9.4 (0.076)	-9.0 (0.01)	$+14.4$ (0.027)	$+18.2$ (0.014)	-30.1 (1.9×10^{-9})
running threshold					
Random	$+2.2$ (0.68)	-0.5 (0.97)	$+1.8$ (0.57)	$+10.9$ (0.051)	-2.8 (0.68)
EI + IR	$+5.9$ (0.25)	$+13.5$ (0.15)	$+3.1$ (0.0016)	$+4.0$ (3.7×10^{-5})	-12.9 (2.2×10^{-5})
EPIG + IR	-3.0 (0.37)	$+2.5$ (0.88)	$+2.7$ (0.37)	$+0.3$ (0.59)	-6.6 (0.009)
SCOUT	-2.6 (0.75)	$+4.3$ (0.59)	$+0.9$ (0.39)	$+0.9$ (0.29)	-10.0 (1.8×10^{-4})
SCOUT $-$ IRInfo	-1.3 (0.83)	$+11.3$ (0.089)	$+1.6$ (0.094)	$+0.7$ (0.41)	-11.2 (6.8×10^{-4})
EI, x only*	-0.4 (0.87)	-6.8 (0.11)	$+5.7$ (0.37)	$+0.5$ (0.97)	-15.4 (9.4×10^{-5})
pool threshold					
Random	$+2.2$ (0.75)	$+1.0$ (0.84)	$+11.3$ (0.11)	$+10.4$ (0.038)	-6.0 (0.39)
EPIG + IR	$+3.8$ (0.90)	$+13.9$ (0.10)	$+0.9$ (0.72)	$+0.3$ (0.73)	$+0.2$ (0.93)
SCOUT	-3.7 (0.84)	-5.7 (0.25)	$+1.9$ (0.033)	$+1.9$ (0.011)	$+0.0$ (1.00)
EI, x only*	-6.7 (0.45)	-4.9 (0.23)	$+14.2$ (0.043)	$+11.7$ (0.14)	$+0.0$ (1.00)

Table 13: Matched-policy comparisons. Each row compares another print rule with SCOUT under the same threshold reading: seed-paired mean of the other arm minus SCOUT in points, positive when SCOUT is better, followed by its Wilcoxon p-value.

	No history		Earlier protocols		Published split
Print rule	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$	$(20, 12)$
own-prediction threshold					
EPIG + IR	-5.9 (0.32)	-0.2 (0.86)	$+0.7$ (0.59)	$+1.7$ (0.024)	$+1.2$ (0.033)
EI + IR	-3.9 (0.30)	-4.1 (0.45)	$+1.2$ (0.13)	$+0.9$ (0.10)	$+0.0$ (0.79)
Random	-8.5 (0.23)	-2.2 (0.78)	$+28.8$ (1.2×10^{-5})	$+16.0$ (1.2×10^{-4})	$+27.1$ (3.6×10^{-6})
EI, x only*	-15.3 (0.11)	-10.4 (0.10)	$+44.0$ (5.5×10^{-6})	$+23.8$ (3.8×10^{-5})	$+8.3$ (1.0×10^{-5})
running threshold					
EPIG + IR	-1.1 (0.83)	-3.2 (0.95)	$+1.0$ (0.39)	$+0.5$ (0.56)	$+7.0$ (0.021)
EI + IR	-10.9 (0.11)	-11.3 (0.32)	$+0.0$ (1.00)	-0.7 (0.18)	$+2.9$ (0.16)
Random	-9.4 (0.24)	-5.0 (0.40)	$+24.4$ (2.6×10^{-5})	$+8.6$ (0.0013)	$+26.1$ (3.7×10^{-5})
EI, x only*	-25.1 (0.012)	-22.0 (0.018)	$+51.2$ (2.5×10^{-6})	$+40.5$ (3.7×10^{-6})	$+5.4$ (0.039)
pool threshold					
EPIG + IR	-9.1 (0.29)	-24.7 (0.0014)	$+3.8$ (0.027)	$+1.4$ (0.034)	$+10.3$ (0.001)
Random	-10.6 (0.13)	-16.5 (0.045)	$+15.9$ (3.8×10^{-4})	$+10.1$ (3.6×10^{-4})	$+39.4$ (8.0×10^{-6})
EI, x only*	-19.9 (0.044)	-33.9 (2.4×10^{-4})	$+43.7$ (9.2×10^{-6})	$+30.3$ (1.2×10^{-5})	$+0.0$ (1.00)

Table 14: Informativeness sweep, earlier-protocols setting at $(20, 12)$: every arm that ran, with noise of 0, 0.5, 1, and 2 standard deviations of that feature over all 124 cells added to each early feature, which gives ρ_y^2 of 0.88, 0.76, 0.54, and 0.23. The first column uses the 30 seeds of Table 9 and the others 20 seeds. Arms that never read the early features do not change with the noise.

Arm	0.88	0.76	0.54	0.23
Random	28.9 (4.6)	30.6 (5.9)	30.6 (5.9)	30.6 (5.9)
EI, x only	59.6 (5.3)	56.6 (6.6)	56.6 (6.6)	56.6 (6.6)
EI + IR	5.7 (0.5)	3.6 (0.8)	4.3 (0.8)	3.2 (0.8)
EPIG + IR	6.3 (2.2)	3.8 (0.9)	5.3 (0.8)	7.0 (2.3)
SCOUT − inner stage	3.6 (0.7)	4.3 (0.8)	3.6 (0.8)	3.9 (0.8)
Random + inner stage	30.0 (4.2)	37.5 (4.7)	33.6 (4.8)	32.3 (5.9)
EI + IR + inner stage	2.4 (0.6)	2.6 (0.8)	8.7 (2.4)	4.7 (1.5)
EPIG + IR + inner stage	1.9 (0.8)	0.7 (0.5)	6.6 (2.5)	9.9 (3.6)
SCOUT, own-prediction threshold	1.2 (0.5)	2.6 (1.2)	6.3 (2.3)	7.4 (2.3)
SCOUT, running threshold	2.6 (0.6)	1.5 (0.7)	6.1 (2.5)	7.7 (2.8)
SCOUT, pool threshold	1.7 (0.6)	1.4 (0.7)	2.8 (0.8)	6.8 (2.7)

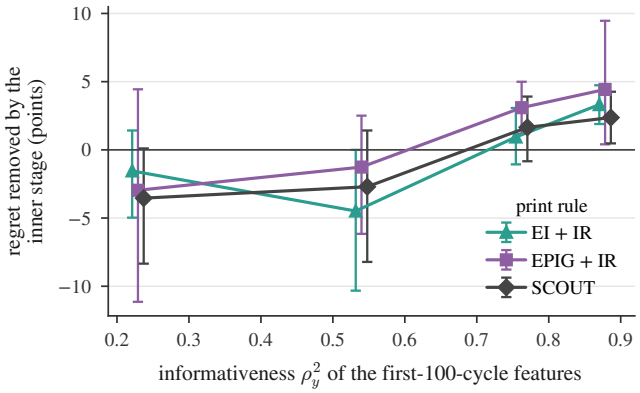

Figure 15: **Informativeness sweep.** Regret removed by the own-prediction inner stage as the informativeness ρ_y^2 of the early features falls, earlier-protocols setting at $(20, 12)$, 20 seeds per point except the rightmost, which uses the 30 seeds of Table 9. Each line pairs an IR-equipped print rule with and without the inner stage, and bars are 95% bootstrap intervals of the seed-paired difference. Per seed and print rule, the least-squares slope of the gain on ρ_y^2 over the four levels averages 8.8 points per unit of ρ_y^2, with $p = 0.0056$ over the 20 per-seed means of the three slopes, and $p = 0.0074$, 0.085, and 0.017 for SCOUT, EPIG, and EI separately. Table 14 gives every arm of the sweep, including random printing with the inner stage, whose gain is -6.9, -3.0, and -1.7 points at the three noisy levels, and SCOUT with the running threshold, $+2.7$, -2.6, and -3.8, and the pool threshold, $+2.8$, $+0.7$, and -2.9.

Table 15: Function networks of Buathong et al. (2024) in the open harness, 30 seeds, in the format of Table 9.

	Ackley–sine		Ackley–Matyas	
Arm	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$
Random	45.5 (4.0)	41.7 (3.7)	10.1 (1.7)	6.2 (1.0)
EI, x only	27.2 (4.5)	16.0 (3.7)	6.9 (1.0)	3.9 (0.5)
EPIG, x only	28.9 (4.7)	18.0 (3.3)	4.7 (0.8)	2.6 (0.5)
GP-UCB-EI, x only	30.1 (4.2)	16.6 (3.5)	8.0 (1.1)	5.1 (0.5)
EI + IR	25.5 (3.7)	17.6 (3.6)	6.7 (0.9)	3.5 (0.6)
EPIG + IR	31.6 (4.1)	23.4 (3.7)	4.8 (0.7)	2.7 (0.5)
GP-UCB-EI + IR	26.6 (3.8)	19.5 (3.6)	7.2 (1.0)	5.1 (0.8)
SCOUT $-$ IRInfo $-$ inner stage	21.6 (4.0)	16.4 (3.5)	6.6 (1.8)	3.4 (0.7)
SCOUT $-$ inner stage	20.5 (4.0)	15.5 (3.6)	8.7 (4.2)	2.6 (0.6)
Random + inner stage	42.2 (3.8)	41.3 (2.9)	7.9 (1.4)	6.1 (0.8)
EI + IR + inner stage	25.8 (3.7)	21.9 (3.5)	6.1 (0.9)	3.8 (0.7)
GP-UCB-EI + IR + inner stage	27.0 (4.8)	21.9 (3.6)	6.1 (0.9)	4.6 (0.7)
EPIG + IR + inner stage	31.1 (4.2)	17.0 (2.7)	5.6 (0.8)	4.1 (0.7)
EI + IR + suspension	22.9 (3.7)	13.0 (2.8)	6.5 (1.2)	3.7 (0.5)
SCOUT $-$ IRInfo	23.6 (4.3)	17.7 (3.2)	5.9 (1.0)	3.5 (0.6)
SCOUT, own-prediction threshold	22.4 (3.4)	16.7 (3.2)	6.2 (1.0)	3.7 (0.6)
Random + inner stage, running threshold	36.3 (3.6)	24.2 (3.7)	7.0 (1.0)	5.3 (0.9)
EI + IR + inner stage, running threshold	20.0 (4.3)	10.8 (3.1)	5.5 (1.0)	3.8 (0.6)
EPIG + IR + inner stage, running threshold	19.6 (3.7)	16.3 (3.2)	5.4 (0.7)	3.3 (0.6)
SCOUT $-$ IRInfo, running threshold	21.7 (4.1)	12.6 (3.1)	5.5 (0.9)	3.0 (0.6)
SCOUT, running threshold	22.2 (3.9)	13.4 (3.2)	6.3 (1.2)	3.9 (0.6)
Random + inner stage, pool threshold	44.1 (3.7)	34.7 (3.5)	7.5 (1.0)	6.0 (0.8)
EPIG + IR + inner stage, pool threshold	24.9 (3.6)	21.1 (2.9)	4.7 (0.8)	**2.6 (0.5)**
SCOUT, pool threshold	**15.0 (3.2)**	**10.1 (2.5)**	**4.5 (0.9)**	2.6 (0.5)
EI, x only + inner stage*	52.1 (4.5)	33.4 (5.2)	7.1 (0.8)	4.8 (0.7)
EI, x only + inner stage*, running threshold	34.5 (4.5)	17.6 (3.9)	5.6 (0.6)	4.1 (0.5)
EI, x only + inner stage*, pool threshold	34.1 (4.9)	25.6 (3.9)	6.8 (1.0)	4.2 (0.6)

M Extended Related Work

Multi-fidelity and bias-aware BO. Classical multi-fidelity surrogates and MF-BO (Kennedy & O'Hagan, 2000; Kandasamy et al., 2017; Takeno et al., 2020; Wu et al., 2019; Song et al., 2019) treat cheaper fidelities as coarser or noisier approximations of the target, and convergence-aware MF-BO models how the output approaches the target as fidelity rises (Xing et al., 2025). Multi-information-source optimization models the bias of each cheaper source with its own GP and chooses both the source and the design by cost-normalized value of information (Poloczek et al., 2017), nonlinear autoregressive GPs learn input-dependent cross-fidelity correlations that guard against low-fidelity models with wrong trends (Perdikaris et al., 2017), and bias-aware and input-dependent variants (Mikkola et al., 2023; Zanjani Foumani et al., 2024; Zhang et al., 2025a; Fan et al., 2024) guard against unreliable low-fidelity sources. Best-practice studies in materials and molecular MF-BO (Sabanza-Gil et al., 2025) recommend, as guiding principles, a low-fidelity source whose R^2 against the high-fidelity target exceeds 0.8 and whose cost is below 0.1 of the high-fidelity cost. Structurally biased FEM violates the first condition, which motivates SCOUT's explicit residual model. In all of these the source bias is a function of the design alone, whereas SCOUT's residual also takes a measurement made during the build.

Function networks, cascades, and cost-aware decisions. Grey-box BO exploits how the objective is computed (Astudillo & Frazier, 2021b). BO of function networks models each node with its own GP and lets observed intermediate outputs guide the search (Astudillo & Frazier, 2021a), and its partial-evaluation extension picks which node and input to evaluate with a cost-normalized knowledge gradient, so that cheap upstream nodes can be evaluated without running the full network (Buathong et al., 2024). Cascade BO optimizes processes in which each stage's output feeds the next and studies a suspension setting in which a process is halted after an unpromising intermediate stage and resumed later from the stored output (Kusakawa et al., 2022). Multi-stage BO for self-driving laboratories models each stage with a cascade of GPs and uses intermediate proxy measurements to decide whether to advance a partially processed sample (Torresi & Friederich, 2026). Partially observable GP networks see intermediate outputs only through noisy observation likelihoods (Kiroriwal et al., 2025), and a manufacturing variant extracts expert features from high-dimensional intermediate state time series (Kiroriwal et al., 2026), which parallels SCOUT's IR pipeline. Few-shot design optimization can also exploit high-dimensional auxiliary measurements returned with the objective, together with a history of related solved tasks (Mani et al., 2026). On the decision side, cost-aware BO through the Pandora's box Gittins index trades expected improvement against evaluation cost (Xie et al., 2024), and Pandora's box with sequential inspections lets a cheap partial opening reveal a signal before a costly full opening (Aouad et al., 2026), which parallels SCOUT's inner stage, although there the fees are subtracted from the prize rather than drawn from separate budgets and the partial opening is optional. Transductive active learning selects observations to reduce uncertainty about a target set (Hübotter et al., 2024), the principle behind pool-restricted CalVal and IRInfo. SCOUT specializes this family to an intermediate observation that is free once a part is printed and cannot be obtained otherwise, a destructive final test on the printed part under a separate count budget, and a biased simulator as the prior mean. p-KGFN evaluates the downstream node at any earlier upstream output, which the immediate test decision rules out. The open harness of Appendix L runs the knowledge gradient through the network, deferred testing, a suspension rule adapted from cascade BO, and, added after the review, two budget-matched arms in the spirit of p-KGFN, Appendix O.

Calibration, BAX, and decision-aware AL. The Kennedy–O'Hagan family (Kennedy & O'Hagan, 2001; Higdon et al., 2004; Bayarri et al., 2007; Brynjarsdóttir & O'Hagan, 2014; Plumlee, 2017; Xie & Xu, 2021; Spitieris & Steinsland, 2023; Sung & Tuo, 2024; Girolami et al., 2021) models the simulator-experiment gap mostly offline, and its sequential designs add field runs to improve calibration and discrepancy prediction rather than to optimize (Williams et al., 2011). Acquisition-coupled discrepancy work (Yang et al., 2025a; 2026; Oliveira et al., 2024) couples sequential BED with NN- or GP-parameterized residuals, and in AM mechanics finite-element simulations have served as the prior of an autonomous BO loop that learns the simulation-experiment difference with a GP (Gongora et al., 2021). These methods have no intermediate observation. BAX (Neiswanger et al., 2021; Chitturi et al., 2024; Cheng et al., 2024), decision-theoretic entropy search (Neiswanger et al., 2022), EPIG (Bickford Smith et al., 2023), EULBO (Maus et al., 2024), which builds on loss-calibrated inference (Lacoste-Julien et al., 2011), and amortized BED (Huang et al., 2024)

assume a single observation channel, and Rainforth et al. (2024) review modern BED more broadly. Goal-oriented sequential BED (Zhang et al., 2025b) addresses causal SCMs with a single oracle, while physics-aware MFBO (Di Fiore & Mainini, 2024) encodes physics through a hand-designed acquisition term that favors the high-fidelity CFD or FEM model as a function of physical variables such as the Mach number.

In-situ AM monitoring and closed-loop systems. In-situ IR, optical, and multi-sensor data in laser metal AM (Lane et al., 2016; Wang et al., 2022; Moshiri et al., 2023; Mohr et al., 2020; Hinnebusch et al., 2026; Penny et al., 2025; Vallabh & Zhao, 2023; Kavas et al., 2025), and in-process imaging in polymer melt electrowriting (Mieszczanek et al., 2024), serve real-time control, defect classification, and feature extraction, and high-speed melt-pool imaging yields in-situ process maps for new alloys (Guirguis et al., 2024). Where in-situ measurements enter cross-build BO, they define the objective, as melt-pool length, height, and stability measured by a thermal camera in directed energy deposition (Sousa et al., 2025), or act as an intermediate feature, measured non-destructively during printing and correlated with tensile strength, that raises throughput several hundredfold over exhaustive tensile testing in polymer material extrusion, without a simulator (Sakai et al., 2026). Recent metal-AM loops with small budgets select recipes with surrogates over the design variables alone, namely process parameters and, in one case, post-build heat treatment, with no in-situ input (Lee et al., 2025b; Chernyavsky et al., 2025; Fadhel et al., 2026). Public co-registered LPBF data pair layerwise visible-light images and process sensor data with tensile tests of thousands of 316L specimens (Scime et al., 2023), a candidate for a retrospective study with an optical rather than radiometric intermediate. Autonomous laboratories (Szymanski et al., 2023; Gongora et al., 2020; Harris et al., 2025) and an LLM-driven chemistry agent (Boiko et al., 2023) do not use IR in acquisition. Interactive discovery benchmarks give agents one budgeted oracle (Malik et al., 2025) or many simulated instruments under a step limit (Jansen et al., 2024), LLM-based generators such as MatExpert (Ding et al., 2025) are scored after generation rather than through a query loop, and a review of benchmarking practice for self-driving laboratories puts common metrics in context (Adesiji et al., 2026). LPBF studies and reviews of Cu-Zn, zinc, and copper alloys (Gatto et al., 2023; Yan et al., 2025; Yang et al., 2025b; Wang et al., 2025) report process windows and part properties, and in-situ infrared imaging of the process plume has been shown in LPBF of zinc powder (Grasso et al., 2018), which motivates SCOUT's free middle tier.

Learned surrogates and language models in BO. Prior-fitted networks serve as in-context BO surrogates (Müller et al., 2023), and language models have been used for warm-starting, surrogate modelling, and candidate sampling (Liu et al., 2024), with the finding that they help BO over molecules only when pretrained or finetuned on domain-specific data (Kristiadi et al., 2024). With at most 30 mechanical tests per campaign in our budgets and no pretraining corpus of paired IR and mechanical data, SCOUT uses a GP over hand-engineered physics features. Learned IR embeddings trained on the raw archive are a direction for future work.

Look-ahead BO and sim-to-real residuals. SCOUT's weights are motivated by a first-order expansion of a one-step look-ahead utility (Frazier, 2018; Garnett, 2023), and multi-step look-ahead and budgeted BO (Astudillo et al., 2021; Zhang et al., 2021; Wu & Frazier, 2019) extend that utility beyond one step. Robotics sim-to-real residual learning (Ajay et al., 2018; Zeng et al., 2019; Gao et al., 2024; Cho & Chen, 2026) and offline multi-fidelity residual models (Yi et al., 2026) share SCOUT's $s + r$ template but operate offline or in robotics state spaces, without acquisition-driven sequential design, and materials sim-to-real transfer learning (Minami et al., 2025) carries predictors trained on simulation databases over to experimental data offline. Standard BO components (Srinivas et al., 2010; Močkus et al., 1978) are reused.

N THE TEST DECISION AS ONLINE SELECTION

Setup. After print t the part i has design x_i and observed intermediate observation ϕ_i. Under the IR-equipped surrogate the value that a test would reveal has the predictive law $Y_i \sim \mathcal{N}(\mu_i, s_i^2)$, where s_i^2 is the latent variance plus the noise variance. The campaign has k tests and n further prints left, and the incumbent m is the best value tested in the campaign so far. The cell-level objective is the expected best tested value at the end of the campaign on the model's scale, which for the batteries

is log cycle life, a monotone transform of the scale on which regret is measured. A part that is not tested is discarded.

The own-prediction threshold is a surprise filter. The inner stage of Appendix D tests the part when its score is at least $\tau_t = q_{1-\rho_t}$, with ρ_t the ratio of tests left to prints left. Which distribution the quantile is taken over decides what the rule does.

Proposition 4 (Own-prediction threshold)**.** *Let $a_i(\phi)$ be any inner-stage score of part i and let τ_i be the $1-\rho$ quantile of $a_i(\Phi)$ when Φ follows the predictive law of ϕ at x_i. If the observed ϕ_i follows that law and $a_i(\Phi)$ has a continuous distribution, the part is tested with probability ρ, whatever its design x_i.*

Proof. With F_i the distribution function of $a_i(\Phi)$, the part is tested when $a_i(\phi_i) \geq \tau_i$, an event of probability $1 - F_i(\tau_i) = \rho$ because F_i is continuous and τ_i is its $1-\rho$ quantile. □

The proposition is the uniformity of the probability integral transform under a calibrated forecast (Dawid, 1984; Gneiting et al., 2007), applied to a threshold. We state it because the budget-fraction threshold of Appendix D leaves its reference distribution open, and the own-prediction reading is the one that this fact rules out.

Under a calibrated law three consequences follow. Every printed part has the same test probability ρ_t whatever its design, so by itself the rule does not move tests toward better designs. Within a design it tests a part when its score ranks in the top ρ_t of the scores its own predictive law gives, which for a score that rises with a one-dimensional ϕ means an observation high in that law. When ϕ carries information about a part's deviation from its design mean, as early cycling does for a cell, the tested parts are then the lucky ones, which lowers the cell-level regret without improving the design that would be recommended. Every part of the best design is discarded with probability $1-\rho_t$, and for such a score one whose ϕ falls below its predictive median with probability $\min\{1, 2(1-\rho_t)\}$, which is how the best cell was lost under the published split of §5.3. With a Monte Carlo quantile over J draws the test probability is ρ up to an error of at most $2/(J+1)$, and if $a_i(\Phi)$ has an atom at the threshold it is at least ρ, so a score that ignores ϕ is always tested.

The proposition needs a calibrated predictive law, and a fitted predictor of ϕ shrinks toward the mean of the printed parts, so a part of an extreme design tends to surprise in the direction of its design. In the battery families with history the rule does separate designs in practice. Among the decisions it actually made, leaving out prints at which the tests left were at least the prints left, SCOUT with the own-prediction threshold tested 2% of the parts printed from the worst third of pool cells by protocol mean life and 66% of those from the best third, against 6% and 58% with the value-of-testing rule, in an analysis made after the round. The separation is consistent with that shrinkage and does not come from comparing the part with any other, and the rule discarded the pool's best cell in 39 of the 53 runs that printed it. Readings that take the quantile over other parts compare parts with each other. The running reading uses the parts printed so far and the pool reading the candidates that are still unprinted. The rule below makes that comparison optimal for the cell-level objective under a fixed model and independent future parts, but the pool reading, a static threshold against the remaining supply, already avoids the failure, and in the exploratory comparison of Appendix O neither is consistently better.

Value-of-testing rule. Let $\mathcal{F}$ be the law of the predictive pair (μ, s) of a future part, and let $V_{k,n}(m)$ be the expected best tested value under the optimal test policy when k tests and n future parts are left and the incumbent is m. With $V_{0,n}(m) = V_{k,0}(m) = m$,

$$V_{k,n}(m) = \mathbb{E}_{(\mu,s)\sim\mathcal{F}}\Big[\max\Big\{\mathbb{E}_{Y\sim\mathcal{N}(\mu,s^2)}\big[V_{k-1,n-1}(\max(m,Y))\big],\ V_{k,n-1}(m)\Big\}\Big], \tag{5}$$

and part i is tested when $\mathbb{E}[V_{k-1,n}(\max(m,Y_i))] \geq V_{k,n}(m)$, where n counts the prints left after part i. When $k > n$ every part is tested. A future part is drawn by taking one of the n unprinted candidates that the print rule ranks highest at this round, or any unprinted candidate under random printing, drawing its ϕ from the predictive law, and reading (μ, s) from the surrogate. The model is held fixed over the horizon. The recursion is ordinary backward induction for a multiple-stopping problem (Haggstrom, 1967) in which each selected arrival reveals a random reward and the payoff is the best of them, an online form of stochastic probing (Gupta & Nagarajan, 2013; Segev & Singla, 2025). Written as testing when $\mathbb{E}[V_{k-1,n}(\max(m,Y_i))] - V_{k-1,n}(m) \geq V_{k,n}(m) - V_{k-1,n}(m)$, it is

a bid-price rule in the sense of revenue management (Talluri & van Ryzin, 2004), with the right-hand side as the price of a test. Pandora's box (Weitzman, 1979) prices every inspection and lets any box be opened later. Its online and time-constrained variants (Esfandiari et al., 2019; Amanatidis et al., 2025) keep the prices and can also constrain which boxes are opened, and in cost-aware BO through the Gittins index (Xie et al., 2024) and in Pandora's box with sequential inspections (Aouad et al., 2026) the prices remain, so a count budget enters them only through a multiplier, the role that $V_{k,n}(m) - V_{k-1,n}(m)$ plays here. The closest precedent in BO is Bayesian optimal stopping by backward induction inside a BO loop, used to stop training runs early (Dai et al., 2019). What is specific here is the use of this recursion as the test rule of a two-budget loop, with a frozen surrogate and future parts drawn from the print rule's current ranking.

Proposition 5 (Structure of the value-of-testing rule)**.** *For all k and n, $V_{k,n}$ is convex and nondecreasing in m with $V_{k,n}(m) \geq m$. The rule tests part i if and only if $\mu_i \geq \tau_{k,n}(m, s_i)$ for a threshold $\tau_{k,n}(m, \cdot)$, possibly infinite, that is nonincreasing in s_i.*

Proof. By induction on $k + n$. The identity $V_{0,n}(m) = V_{k,0}(m) = m$ is convex and nondecreasing. For fixed Y, the map $m \mapsto \max(m, Y)$ is convex and nondecreasing, so its composition with the convex nondecreasing $V_{k-1,n-1}$ is convex and nondecreasing, and so is its expectation over Y. A maximum of two such functions and an average over (μ, s) keep both properties, and $V_{k,n}(m) \geq V_{k,n-1}(m) \geq \cdots \geq V_{k,0}(m) = m$. For the decision, $g(y) = V_{k-1,n}(\max(m, y))$ is convex, nondecreasing, and continuous in y, so $h(\mu, s) = \mathbb{E}[g(\mu + sZ)]$ with Z standard normal is continuous and nondecreasing in μ, and it is nondecreasing in s because $\mathcal{N}(\mu, s'^2)$ dominates $\mathcal{N}(\mu, s^2)$ in the convex order when $s' \geq s$. The right-hand side $V_{k,n}(m)$ does not depend on part i, so the test region is $\{\mu_i \geq \tau(s_i)\}$, and $h(\cdot, s') \geq h(\cdot, s)$ gives $\tau(s') \leq \tau(s)$. For a grid and samples of $\mathcal{F}$ that do not depend on the part, linear interpolation and averaging keep convexity and monotonicity, so the computed rule has the same structure. □

The implementation widens the grid to cover four predictive standard deviations around the part's own mean, so the grid depends on the part. For a part far more uncertain than the future parts this can break the monotonicity in μ_i near the threshold by a few thousandths of the value, and scans in moderate settings found no violation.

A part is therefore tested for its predicted value and its spread relative to what the remaining prints can offer and to the incumbent, never for how it compares with its own prediction, and an uncertain part is tested at a lower predicted mean than a certain one. Figure 16 shows this structure on a standardized example. The rule ignores the value of learning from a test, because the model is held fixed, and it targets the cell-level best, so it shares that metric's reward for lucky parts. It maximizes the expected best value on the model's scale, log cycle life for the batteries, which is more risk-averse than the same objective on the scale of cycle life, and its model of the future draws parts with replacement from the current top of the ranking, which can over-value waiting and may explain why it discards the pool's best part more often than the pool reading in some families of Appendix O. A knowledge-gradient print rule, which values a test by the change in the best posterior mean over the top ten candidates and the candidate itself, targets the recommended design instead.

Computation. $V_{k,n}$ is represented on a grid of 48 incumbent values plus m, spanning four predictive standard deviations around every sampled mean and around the part's own mean, with at most 96 samples of $\mathcal{F}$. Between grid points $V_{k,n}$ is linear, above the grid it rises with slope one and below it is flat, and the expectation over Y of this piecewise-linear function of $\max(m, Y)$ is computed exactly from the normal distribution function and density at the grid points. The table for all $k' \leq k$ and $n' \leq n$ costs $O(k\, n\, G\, M)$ with G grid points and M samples, below one second per decision at the budgets of §5.3. On a standard normal the recursion returns the expected maximum of two and of three draws to within 0.001 when the part's spread equals that of the future parts and to within 0.004 when it is twice as large.

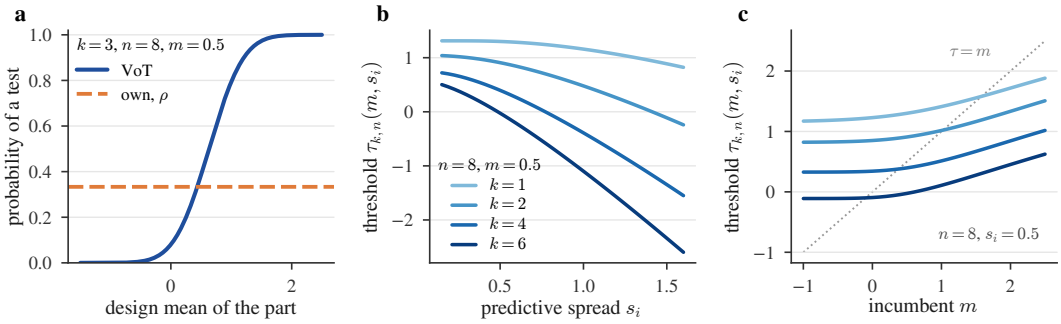

Figure 16: **Structure of the value-of-testing rule.** The rule is computed with the recursion of Eq. (5) as the harness implements it, on the standardized scale, for future parts whose predictive means are the quantiles of a standard normal and whose spread is 0.5. **a** Probability that a part is tested against the design mean of the part under the value-of-testing rule, VoT, when the free observation moves its predictive mean by a normal amount with standard deviation 0.45 and leaves a spread of 0.5. The own-prediction reading tests every design with probability $\rho = k/(n+1)$, Proposition 4. **b** The threshold on the predictive mean falls as the spread grows, Proposition 5, and as more tests are left. **c** In this example it rises with the incumbent once the incumbent is high enough to matter, with the shades of k as in **b**.

O Pre-registered Round: Test Rules, Closest Baselines, and Random Instances

Pre-registration. Before any run of the arms below on an evaluation instance, we fixed in a time-stamped file, released with the harness, the new arms, the three families of instances, the metrics, seventeen confirmatory comparisons, and Holm's correction over them. The only earlier runs of the new code were smoke tests on a held-out seed, which recorded runtime and the numbers of prints, tests, and skips, and reruns of ten earlier rows, which reproduced them exactly. The value-of-testing rule was designed after the published-split result of §5.3 and was fixed before it ran on any battery instance. The concrete families and their ten confirmatory comparisons were added to the file before they ran. Two deviations follow from the code review described in Appendix L: the no-history family was rerun after a defect that let it print outside its pool, and every EPIG and knowledge-gradient arm was rerun after a covariance defect that was found once the first confirmatory results had been seen. Every arm and family is reported below.

Instances. Each family has 30 instances, one per seed, so that the instance and not the algorithm seed is the unit of inference. *Grouped*: whole protocols are moved to the history in random order until it holds a third of the 124 cells, so history and pool share no protocol. This generalizes the earlier-protocols setting. *Cell*: a random third of the cells is the history, so replicates of pool protocols can be labelled, which generalizes the published split. *No history*: a random two thirds of the cells form the pool and every arm starts from three random tests. All instances use $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$.

Concrete data. The UCI concrete compressive strength data of Yeh (1998a;b), released under CC BY 4.0, hold 1030 laboratory records, each a mixture design of seven component amounts, a test age, and the compressive strength at that age. We rebuilt a table of the 122 designs that have both a 3-day and a 28-day strength, after removing exact duplicate records and averaging the two designs with repeated records, and the build script checks the checksums of the raw file. The design x is the seven amounts, the free observation is the 3-day strength of a companion specimen, and the test reveals the 28-day strength. Every design is one unit. Ten-fold cross-validated R^2 of the 28-day strength is 0.78 on the 3-day strength alone, 0.81 on the design alone with a GP, and 0.86 on both, so the early strength adds little to what the design predicts. The two strengths come from different specimens of the same mixture, and the file pools several source studies, within which the two strengths correlate less than across them. Table 24 gives every arm on the two concrete families and Figure 17 the regret after each test.

Arms. The ten arms of the earlier rounds that the comparisons need run again on every instance. Eleven arms are new. The value-of-testing rule of Appendix N decides the tests of the SCOUT, EPIG, EI, and random print rules. The knowledge gradient through the two-node network $x \to \phi \to y$ values a candidate by the one-step change in the best posterior mean over the top-10 targets and the candidate itself, with ϕ marginalized as in EPIG. It is the full-evaluation acquisition of function-network BO and runs single-stage and with the value-of-testing rule. EI-FN averages EI over the ϕ draws of a candidate (Astudillo & Frazier, 2021a). Deferred testing replaces the immediate decision by tests at the end of the campaign: the arm prints B_{print} parts and then tests the B_{mech} printed parts of highest posterior mean given their ϕ, with the EPIG, SCOUT, and random print rules. It can recall any printed part but gives up the information of early tests, so it is a different protocol and not a relaxation of the immediate one. The early-prediction loop is in the style of Attia et al. (2020): GP-UCB with $\kappa = 2$ on the protocol variables over the tested lives and the early predictions of printed but untested cells, with the tests deferred to the cells of highest early prediction. Attia et al. (2020) trained their early predictor once on earlier cells, shrank the exploration weight every round, never ran a cell of the loop to failure, and validated protocols chosen by the posterior mean afterwards, so the arm keeps their use of early predictions as pseudo-labels but not every detail of their loop. The partial-evaluation acquisition p-KGFN of Buathong et al. (2024) can evaluate the downstream node at any earlier upstream output, which the immediate test decision rules out, and it was not part of the round. Two arms in its spirit were added after the review and are described below.

Metrics and tests. The primary metric is the cell-level regret of §5.3. The secondary metrics are the protocol of the best tested cell scored by its protocol mean, the protocol recommended by posterior mean with the rule of Table 11, and the share of runs that printed the pool's best cell and did not test it. The confirmatory comparisons are two-sided Wilcoxon signed-rank tests on the 30 instance-paired differences, with Holm's correction over the seventeen battery tests and separately over the ten concrete tests. Figure 18 shows the seventeen battery comparisons. The comparison of skip shares is one Fisher exact test on the runs pooled over the three families. Everything else in this appendix is exploratory.

Deviations and their times. All times are on 25 September 2026 in UTC. The file was written at about 15:15. Results for part of the grouped family were first computed at about 15:17, and complete results of the grouped and cell families at 15:39 and 16:06. Addendum A, at about 16:55, recorded the pool defect and added the concrete families with a Holm correction of their own, so that split was chosen after the results of two battery families had been seen. Table 20 therefore also gives one correction over all 27 tests, under which 3 of them remain significant. Addendum B, at about 17:55, recorded the covariance defect, and Table 21 gives the comparisons it touched before and after the fix. The fix went against SCOUT: the print-rule margin over EPIG in cell instances was significant after correction before it and is not after it. The release contains the registered code version d5ee03ae next to the final one.

How concentrated the battery evidence is. The families are random partitions of the same 124 cells, and the pool's best cell is usually one of five batch-1 cells whose recorded life joins a batch-2 continuation record in the official processing, Appendix L. Their early features come from the first record, and for b1c2 they underpredict the joined life by 45%. The own-prediction threshold discarded the pool's best cell in 53 of the 62 runs in which it was one of these cells and in 0 of the 8 runs in which it was not, and the value-of-testing rule in 6 of 59 and 0 of 10. The C1 effect follows the same split, Table 22. In grouped instances, the 18 with b1c2 as the pool's best contribute 7.6 of the 8.6 points, and the other 12 have a mean effect of 2.4 points with $p = 0.18$. The comparison of skip shares, paired by instance with exact McNemar tests on the instances where both rules printed the pool's best cell, gives $p = 3.1 \times 10^{-5}$, 2.4×10^{-7}, and 0.07 in the grouped, cell, and no-history families, well above the pooled Fisher value. Removing b1c2 alone leaves the effect, because b1c1 or b1c0 then becomes the pool's best cell in most instances, and removing all five removes it, Table 17.

The recursion against the pool reading, and SCOUT against EI. The pool reading, a static quantile against the remaining pool, is not significantly worse than the value-of-testing rule: SCOUT with the pool reading reaches 3.8, 0.0, and 36.5 in the three battery families, against differences of +2.4, +0.0, and +3.2 points with $p = 0.08$, 1.00, and 0.85, and on concrete with history it is better, 0.5 with $p = 0.022$. EI on the same surrogate with the same test rule matches SCOUT in every

family, with differences of +0.3, +0.0, and -1.2 points on battery data and $p \geq 0.49$, and it is better on concrete with history, with $p = 0.021$. These comparisons were not pre-registered and are not corrected.

Added after the review. Every run in this paragraph was made after the review of the round, with the final code version, and is exploratory. *Robustness of the test-rule effect.* The grouped and cell families were rerun without b1c2 and without b3c38, the two cells that are most often the pool's best, and all three battery families without the five merged cells, with the arms of Table 17. Without the merged cells the own-prediction threshold discarded the pool's best cell in 0 of 11, 0 of 23, and 6 of 13 runs that printed it in the three families, and the value-of-testing rule in 0 of 14, 0 of 25, and 4 of 15. *Arms in the spirit of p-KGFN.* Both print the candidate of highest knowledge gradient of its free observation on the IR-equipped surrogate. The restricted arm tests the part just printed when the knowledge gradient of testing it, divided by the cost ratio $B_{\text{print}}/B_{\text{mech}}$, is at least that of the best print, the cost-normalized comparison of p-KGFN, and discards it otherwise. The arm with recall may test, after every print, the waiting part of highest knowledge gradient under the same comparison, and spends the tests left after the last print one at a time on the waiting part of highest knowledge gradient, so it gives up the immediate decision. *Suspension relative to the incumbent.* The part is tested when its upper confidence bound $\mu + 2\sigma$ given ϕ is at least the best value tested in the campaign so far, with the SCOUT and EI print rules. *Static design.* A maximin design prints the candidate farthest in the design variables from every printed and labelled part and tests every print, as classical design of experiments would. Table 18 gives these arms next to SCOUT with the value-of-testing rule. With history the restricted arm and the suspension arms are worse than SCOUT with the value-of-testing rule in both battery families, with $p = 1.3 \times 10^{-4}$ and 8.8×10^{-5} for the restricted arm, and the arm with recall is worse in grouped instances, with $p = 0.013$, and not significantly different in cell instances. Without history the static design is the best of these arms on battery data, 15.2 against 33.2 with $p = 0.072$, and with history it is the worst. *Informativeness sweep.* The three arms with the value-of-testing rule also ran on the noisy levels of Table 14, Table 19.

Table 16: **With the value-of-testing rule, SCOUT has the lowest cell-level mean in the two battery families with history.** Pre-registered round, 30 random instances per family at $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$. Cell-level regret as mean and standard error, then the regret of the protocol each arm recommends by posterior mean. Bold marks the lowest mean in a column, the shaded row is SCOUT with the value-of-testing rule, and *deferred tests* wait until all prints are done.

	Battery, cell level			Concrete		Battery, protocol level		
Arm	Grouped	Cell	No history	History	No history	Grouped	Cell	No hist.
Random	33.0 (5.3)	40.5 (5.5)	34.3 (5.2)	17.6 (2.6)	27.4 (3.2)	21.9	26.3	22.4
EI, x only	15.8 (4.5)	0.7 (0.4)	16.3 (5.9)	1.0 (0.6)	5.0 (1.6)	3.9	**0.0**	10.1
GP-UCB-EI, x only	6.1 (2.5)	0.7 (0.4)	**14.1 (5.5)**	**0.7 (0.5)**	3.7 (1.1)	0.4	**0.0**	**9.8**
EPIG + IR	7.0 (2.5)	11.6 (2.1)	43.3 (6.4)	1.5 (0.8)	6.3 (1.7)	**0.3**	8.1	31.4
SCOUT, single stage	2.8 (1.1)	1.2 (1.0)	37.3 (6.6)	0.9 (0.5)	11.8 (3.7)	1.2	0.8	28.8
EPIG + IR, own prediction	11.8 (2.9)	15.8 (2.1)	36.1 (5.8)	3.8 (1.6)	7.8 (1.7)	6.0	12.2	25.1
SCOUT, own prediction	9.9 (1.7)	12.8 (2.1)	34.9 (5.8)	1.5 (0.7)	7.3 (2.0)	8.9	10.6	24.6
SCOUT, value of testing	**1.4 (0.6)**	**0.0 (0.0)**	33.2 (6.1)	1.7 (0.6)	3.4 (1.2)	**0.3**	**0.0**	23.0
EPIG + IR, value of testing	5.6 (2.4)	4.8 (1.6)	25.1 (5.3)	2.3 (1.0)	9.9 (2.0)	2.0	4.9	20.0
KG + IR, value of testing	1.8 (0.7)	1.1 (0.9)	29.8 (6.4)	1.2 (0.5)	**2.9 (1.1)**	1.2	0.8	23.8
EPIG + IR, deferred tests	9.6 (3.6)	5.8 (1.6)	37.8 (6.2)	1.0 (0.5)	20.8 (4.6)	5.3	5.7	29.0
Early-prediction loop	13.2 (3.8)	1.8 (0.9)	18.1 (5.3)	1.3 (0.6)	12.1 (3.1)	4.5	**0.0**	12.3

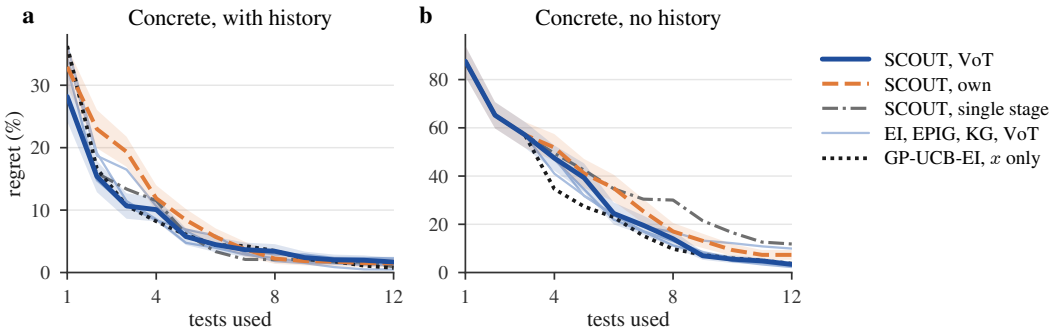

Figure 17: **Concrete families of the pre-registered round.** Regret after each test, in the order the tests were made, as a mean over the 30 instances of each family at $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$, with one standard error for the two SCOUT arms. The end points are the entries of Table 24. VoT is the value-of-testing rule, own the own-prediction threshold, and the single stage tests every print.

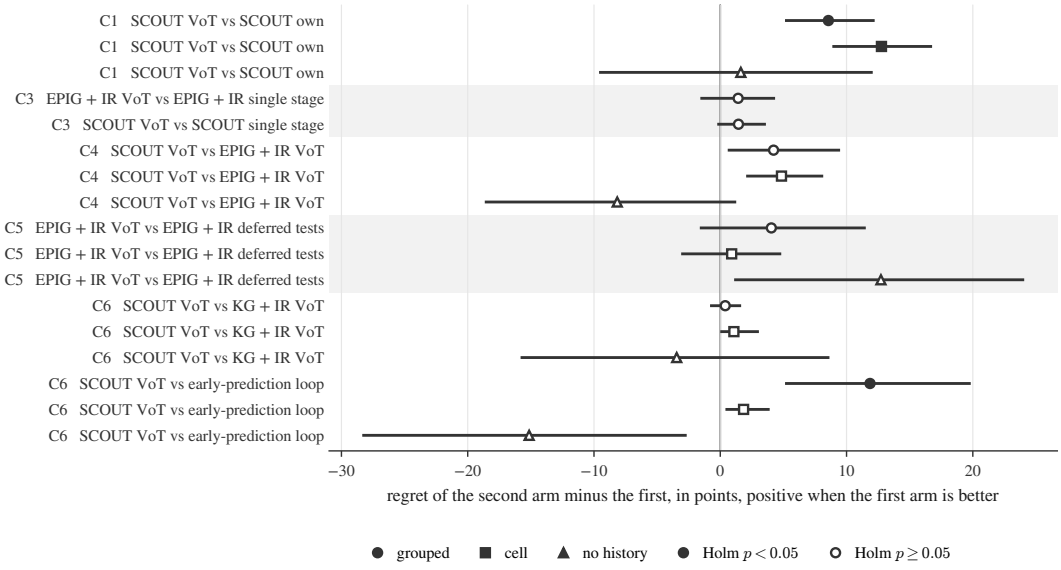

Figure 18: **Confirmatory comparisons on the battery families.** Instance-paired difference in cell-level regret, second arm minus first, with unadjusted 95% bootstrap intervals. Filled markers pass Holm's correction of the Wilcoxon tests over the seventeen comparisons, so an interval can exclude zero while its marker stays open, and the marker shape gives the family. C2, the comparison of skip shares, is not a difference in regret and is not shown, and C6 holds two comparisons, with the knowledge gradient and with the early-prediction loop. VoT marks the value-of-testing rule, own the own-prediction threshold, and KG the knowledge gradient through the network.

Table 17: The effect of the test rule for SCOUT on the battery families and on partitions without the cells that dominate the pool best, 30 instances each at $(20, 12)$. Cell-level regret as a mean for SCOUT with the own-prediction threshold, the value-of-testing rule, and the pool reading, for EI and the knowledge gradient on the IR-equipped surrogate with the value-of-testing rule, and for GP-UCB-EI on the design alone, then the instance-paired difference own prediction minus value of testing and its Wilcoxon p-value. The rows other than the first three were added after the review.

	SCOUT			EI + IR	KG + IR	GP-UCB-EI		
Family	Own	VoT	Pool	VoT	VoT	x only	Difference	p
grouped	9.9	1.4	3.8	1.7	1.8	6.1	$+8.6$	2.0×10^{-4}
cell	12.8	0.0	0.0	0.0	1.1	0.7	$+12.8$	2.7×10^{-5}
no history	34.9	33.2	36.5	32.1	29.8	14.1	$+1.6$	0.56
grouped − b1c2	12.7	2.2	3.6	1.8	2.8	6.0	$+10.5$	0.0015
cell − b1c2	8.3	0.8	0.2	0.2	1.0	1.9	$+7.5$	0.0016
grouped − b3c38	11.5	0.7	1.3	0.3	1.0	3.1	$+10.7$	4.3×10^{-4}
cell − b3c38	17.1	2.0	2.2	2.0	2.0	0.0	$+15.1$	2.6×10^{-5}
grouped − merged	17.7	17.9	25.0	21.0	20.8	43.1	-0.2	0.95
cell − merged	3.6	2.1	3.1	4.7	1.0	21.2	$+1.5$	0.53
no history − merged	40.8	35.9	38.8	40.9	32.0	35.9	$+4.9$	0.30

Table 18: Arms added after the review, cell-level regret as a mean over 30 instances or seeds at $(20, 12)$, next to SCOUT with the value-of-testing rule. Above the line the families of the round, below it the fixed settings and the function networks of Appendix L, where the static design did not run on the function networks.

	SCOUT	p-KGFN style		Suspension		Static
Family or setting	VoT	Restricted	Recall	SCOUT	EI + IR	design
grouped	1.4	23.5	10.2	3.7	6.9	38.3
cell	0.0	26.1	1.1	4.1	3.0	45.9
no history	33.2	39.5	39.1	37.8	36.2	15.2
concrete, hist.	1.7	1.1	0.8	2.0	1.8	2.5
concrete, none	3.4	7.3	8.4	7.7	5.4	15.8
no history, fixed	42.5	47.7	47.8	49.8	49.1	15.7
earlier protocols	1.9	8.4	6.0	7.0	7.7	73.0
published split	0.0	20.7	0.0	10.6	13.6	86.8
Ackley–sine	15.7	21.1	12.9	14.2	18.1	—
Ackley–Matyas	5.5	7.3	7.5	8.7	6.7	—

Table 19: The value-of-testing rule in the informativeness sweep of the earlier-protocols setting at $(20, 12)$, cell-level regret as a mean, 30 seeds in the first column and 20 in the others, which add noise of 0.5, 1, and 2 standard deviations to each early feature.

Arm	0.88	0.76	0.54	0.23
SCOUT-VoT	1.9	4.9	5.1	3.8
EPIG-VoT	4.3	1.2	7.6	4.5
EI-VoT	2.1	4.3	3.7	5.0

Table 20: The 27 confirmatory comparisons with the number of instances whose difference is not zero, the Wilcoxon p-value, an exact sign-flip p-value where at most 20 differences are non-zero, the Holm-adjusted p-value within the two pre-registered families, and the Holm-adjusted p-value over all 27 tests.

	Family	Comparison	Difference	Non-zero	Wilcoxon	Exact	Holm, families	Holm, all
C1	grouped	SCOUT-VoT vs SCOUT-own	$+8.6$	18	2.0×10^{-4}	7.6×10^{-6}	0.0031	0.0051
C1	cell	SCOUT-VoT vs SCOUT-own	$+12.8$	23	2.7×10^{-5}	$- - -$	4.6×10^{-4}	7.3×10^{-4}
C1	no history	SCOUT-VoT vs SCOUT-own	$+1.6$	22	0.56	$- - -$	1.00	1.00
C3	grouped	EPIG-VoT vs EPIG	$+1.4$	13	0.20	0.38	1.00	1.00
C3	grouped	SCOUT-VoT vs SCOUT single	$+1.4$	5	0.14	0.19	1.00	1.00
C4	grouped	SCOUT-VoT vs EPIG-VoT	$+4.2$	9	0.05	0.027	0.55	0.86
C4	cell	SCOUT-VoT vs EPIG-VoT	$+4.8$	9	0.0077	0.0039	0.11	0.17
C4	no history	SCOUT-VoT vs EPIG-VoT	-8.2	21	0.24	$- - -$	1.00	1.00
C5	grouped	EPIG-VoT vs EPIG end tests	$+4.0$	14	0.40	0.29	1.00	1.00
C5	cell	EPIG-VoT vs EPIG end tests	$+0.9$	15	0.61	0.67	1.00	1.00
C5	no history	EPIG-VoT vs EPIG end tests	$+12.7$	21	0.04	$- - -$	0.48	0.77
C6	grouped	SCOUT-VoT vs KG-VoT	$+0.4$	7	0.61	0.58	1.00	1.00
C6	cell	SCOUT-VoT vs KG-VoT	$+1.1$	2	0.18	0.50	1.00	1.00
C6	no history	SCOUT-VoT vs KG-VoT	-3.4	21	0.50	$- - -$	1.00	1.00
C6	grouped	SCOUT-VoT vs early prediction	$+11.9$	12	0.0022	4.9×10^{-4}	0.033	0.053
C6	cell	SCOUT-VoT vs early prediction	$+1.8$	6	0.028	0.031	0.36	0.55
C6	no history	SCOUT-VoT vs early prediction	-15.1	24	0.057	$- - -$	0.57	0.92
C1	concrete, hist.	SCOUT-VoT vs SCOUT-own	-0.2	12	0.53	0.84	1.00	1.00
C1	concrete, none	SCOUT-VoT vs SCOUT-own	$+3.9$	16	0.013	0.031	0.10	0.27
C4	concrete, hist.	SCOUT-VoT vs EPIG-VoT	$+0.6$	14	0.90	0.55	1.00	1.00
C4	concrete, none	SCOUT-VoT vs EPIG-VoT	$+6.6$	24	5.8×10^{-4}	$- - -$	0.0058	0.014
C5	concrete, hist.	EPIG-VoT vs EPIG end tests	-1.3	11	0.45	0.21	1.00	1.00
C5	concrete, none	EPIG-VoT vs EPIG end tests	$+10.9$	25	0.048	$- - -$	0.34	0.86
C6	concrete, hist.	SCOUT-VoT vs KG-VoT	-0.5	9	0.19	0.18	1.00	1.00
C6	concrete, none	SCOUT-VoT vs KG-VoT	-0.5	12	0.81	0.61	1.00	1.00
C6	concrete, hist.	SCOUT-VoT vs early prediction	-0.4	7	0.40	0.42	1.00	1.00
C6	concrete, none	SCOUT-VoT vs early prediction	$+8.7$	19	0.0025	0.0013	0.023	0.058

Table 21: Confirmatory comparisons that involve EPIG or the knowledge gradient, on the runs before the covariance fix and on the repeated runs after it: mean difference second minus first and Holm-adjusted p-value within the pre-registered family.

			Before the fix		After the fix	
	Family	Comparison	Difference	Holm	Difference	Holm
C3	grouped	EPIG-VoT vs EPIG	$+6.7$	0.57	$+1.4$	1.00
C4	grouped	SCOUT-VoT vs EPIG-VoT	$+2.0$	0.22	$+4.2$	0.55
C4	cell	SCOUT-VoT vs EPIG-VoT	$+7.1$	0.047	$+4.8$	0.11
C4	no history	SCOUT-VoT vs EPIG-VoT	-3.4	1.00	-8.2	1.00
C5	grouped	EPIG-VoT vs EPIG end tests	$+6.8$	0.80	$+4.0$	1.00
C5	cell	EPIG-VoT vs EPIG end tests	-2.1	1.00	$+0.9$	1.00
C5	no history	EPIG-VoT vs EPIG end tests	$+6.2$	1.00	$+12.7$	0.48
C6	grouped	SCOUT-VoT vs KG-VoT	$+1.9$	0.81	$+0.4$	1.00
C6	cell	SCOUT-VoT vs KG-VoT	$+3.7$	0.15	$+1.1$	1.00
C6	no history	SCOUT-VoT vs KG-VoT	-4.2	1.00	-3.4	1.00
C4	concrete, hist.	SCOUT-VoT vs EPIG-VoT	$+0.3$	1.00	$+0.6$	1.00
C4	concrete, none	SCOUT-VoT vs EPIG-VoT	$+6.2$	0.019	$+6.6$	0.0058
C5	concrete, hist.	EPIG-VoT vs EPIG end tests	-1.2	1.00	-1.3	1.00
C5	concrete, none	EPIG-VoT vs EPIG end tests	$+13.7$	0.087	$+10.9$	0.34
C6	concrete, hist.	SCOUT-VoT vs KG-VoT	-0.3	1.00	-0.5	1.00
C6	concrete, none	SCOUT-VoT vs KG-VoT	$+0.2$	1.00	-0.5	1.00

Table 22: The C1 effect, own-prediction threshold minus value-of-testing rule for SCOUT, by the physical cell that is the pool's best: number of instances, mean effect on them, and their contribution to the family mean over 30 instances. The asterisk marks the five cells whose recorded life joins a continuation record.

Family	Pool's best cell	Instances	Mean effect	Contribution
grouped	b1c2*	18	+12.7	+7.6
grouped	b3c38	10	+2.9	+1.0
grouped	b3c7	2	+0.0	+0.0
cell	b1c2*	20	+17.5	+11.6
cell	b1c1*	6	+5.4	+1.1
cell	b3c38	3	+0.0	+0.0
cell	b1c0*	1	+1.4	+0.0
no history	b1c2*	24	+0.7	+0.6
no history	b1c1*	5	+4.7	+0.8
no history	b1c0*	1	+8.7	+0.3

Table 23: Pre-registered round, all arms, 30 instances per family, $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$. Cell-level regret as mean followed by standard error over instances, then protocol-level regret of the protocol of the best tested cell and of the protocol recommended by posterior mean, both as means. Bold marks the lowest mean in a column.

	Cell level			Protocol of best cell			Recommended protocol		
Arm	Grouped	Cell	No hist.	Grouped	Cell	No hist.	Grouped	Cell	No hist.
Random	33.0 (5.3)	40.5 (5.5)	34.3 (5.2)	19.8	26.4	23.0	21.9	26.3	22.4
EI, x only	15.8 (4.5)	0.7 (0.4)	16.3 (5.9)	3.6	**0.0**	10.1	3.9	**0.0**	10.1
GP-UCB-EI, x only	6.1 (2.5)	0.7 (0.4)	**14.1 (5.5)**	1.2	**0.0**	**9.5**	0.4	**0.0**	**9.8**
EI + IR	4.3 (1.6)	0.7 (0.4)	35.6 (6.5)	1.8	**0.0**	28.9	2.1	**0.0**	26.7
EI-FN	2.6 (0.9)	1.6 (0.8)	30.8 (6.4)	**1.0**	1.5	26.1	0.4	0.8	24.9
EPIG + IR	7.0 (2.5)	11.6 (2.1)	43.3 (6.4)	1.2	10.2	31.3	**0.3**	8.1	31.4
KG through the network	8.0 (3.6)	4.0 (1.5)	33.6 (6.3)	3.7	7.0	28.4	2.0	3.3	27.6
SCOUT − inner stage	2.8 (1.1)	1.2 (1.0)	37.3 (6.6)	4.0	3.9	30.4	1.2	0.8	28.8
EPIG + IR, own-prediction threshold	11.8 (2.9)	15.8 (2.1)	36.1 (5.8)	12.2	19.5	27.1	6.0	12.2	25.1
SCOUT, own-prediction threshold	9.9 (1.7)	12.8 (2.1)	34.9 (5.8)	10.5	21.7	25.1	8.9	10.6	24.6
SCOUT, running threshold	7.4 (1.6)	8.8 (2.1)	28.2 (6.1)	6.8	9.3	23.6	4.4	4.9	22.5
SCOUT, pool threshold	3.8 (1.4)	**0.0 (0.0)**	36.5 (6.5)	3.0	4.6	25.0	0.4	**0.0**	26.2
Random, value of testing	28.2 (4.8)	26.0 (4.4)	31.9 (5.2)	14.6	18.1	24.4	13.3	17.7	21.6
EI + IR, value of testing	1.7 (0.6)	**0.0 (0.0)**	32.1 (6.1)	3.9	4.6	24.4	1.3	**0.0**	23.1
EPIG + IR, value of testing	5.6 (2.4)	4.8 (1.6)	25.1 (5.3)	6.2	9.3	20.2	2.0	4.9	20.0
KG through the network, value of testing	1.8 (0.7)	1.1 (0.9)	29.8 (6.4)	4.9	5.4	22.9	1.2	0.8	23.8
SCOUT, value of testing	**1.4 (0.6)**	**0.0 (0.0)**	33.2 (6.1)	3.9	4.6	22.8	**0.3**	**0.0**	23.0
Random, deferred tests	23.4 (5.1)	24.1 (4.2)	23.9 (4.5)	16.0	17.7	18.5	14.4	17.5	17.0
EPIG + IR, deferred tests	9.6 (3.6)	5.8 (1.6)	37.8 (6.2)	9.8	10.9	33.9	5.3	5.7	29.0
SCOUT, deferred tests	10.3 (3.2)	4.8 (1.4)	43.7 (6.4)	12.6	10.8	35.4	7.7	3.2	32.6
Early-prediction loop	13.2 (3.8)	1.8 (0.9)	18.1 (5.3)	5.6	1.5	13.1	4.5	**0.0**	12.3

Table 24: Pre-registered round on the concrete data, all arms, 30 instances per family, $(B_{\text{print}}, B_{\text{mech}}) = (20, 12)$. Remaining share of the gap between the pool median and the pool best design in percent, mean followed by standard error over instances. Every design is one unit, so this is also the design-level regret. Bold marks the lowest mean in a column.

Arm	History	No history
Random	17.6 (2.6)	27.4 (3.2)
EI, x only	1.0 (0.6)	5.0 (1.6)
GP-UCB-EI, x only	0.7 (0.5)	3.7 (1.1)
EI + IR	1.5 (0.7)	5.7 (1.7)
EI-FN	2.0 (0.8)	5.6 (1.7)
EPIG + IR	1.5 (0.8)	6.3 (1.7)
KG through the network	1.1 (0.5)	4.1 (1.0)
SCOUT $-$ inner stage	0.9 (0.5)	11.8 (3.7)
EPIG + IR, own-prediction threshold	3.8 (1.6)	7.8 (1.7)
SCOUT, own-prediction threshold	1.5 (0.7)	7.3 (2.0)
SCOUT, running threshold	4.0 (1.1)	5.4 (2.2)
SCOUT, pool threshold	0.5 (0.3)	5.4 (2.0)
Random, value of testing	10.9 (3.1)	14.7 (2.6)
EI + IR, value of testing	**0.5 (0.4)**	**2.3 (0.9)**
EPIG + IR, value of testing	2.3 (1.0)	9.9 (2.0)
KG through the network, value of testing	1.2 (0.5)	2.9 (1.1)
SCOUT, value of testing	1.7 (0.6)	3.4 (1.2)
Random, deferred tests	10.4 (1.5)	17.5 (3.8)
EPIG + IR, deferred tests	1.0 (0.5)	20.8 (4.6)
SCOUT, deferred tests	1.3 (0.6)	24.5 (4.3)
Early-prediction loop	1.3 (0.6)	12.1 (3.1)

Table 25: Confirmatory comparisons of the pre-registered round on cell-level regret. Mean of each arm over the 30 instances of the family, instance-paired mean difference second minus first with its 95% bootstrap interval, Wilcoxon p-value, and Holm-adjusted p-value, over the seventeen battery tests above the line and over the ten concrete tests below it. VoT marks the value-of-testing rule, own the own-prediction threshold, and KG the knowledge gradient through the network.

Family	First arm	Mean	Second arm	Mean	Difference	p	Holm
C1 grouped	SCOUT-VoT	1.4	SCOUT-own	9.9	$+8.6$ $[+5.1, +12.2]$	2.0×10^{-4}	0.0031
C1 cell	SCOUT-VoT	0.0	SCOUT-own	12.8	$+12.8$ $[+8.9, +16.8]$	2.7×10^{-5}	4.6×10^{-4}
C1 no history	SCOUT-VoT	33.2	SCOUT-own	34.9	$+1.6$ $[-9.6, +12.1]$	0.56	1.00
C3 grouped	EPIG-VoT	5.6	EPIG	7.0	$+1.4$ $[-1.6, +4.3]$	0.20	1.00
C3 grouped	SCOUT-VoT	1.4	SCOUT single	2.8	$+1.4$ $[-0.2, +3.6]$	0.14	1.00
C4 grouped	SCOUT-VoT	1.4	EPIG-VoT	5.6	$+4.2$ $[+0.6, +9.5]$	0.05	0.55
C4 cell	SCOUT-VoT	0.0	EPIG-VoT	4.8	$+4.8$ $[+2.1, +8.2]$	0.0077	0.11
C4 no history	SCOUT-VoT	33.2	EPIG-VoT	25.1	-8.2 $[-18.6, +1.3]$	0.24	1.00
C5 grouped	EPIG-VoT	5.6	EPIG deferred	9.6	$+4.0$ $[-1.6, +11.5]$	0.40	1.00
C5 cell	EPIG-VoT	4.8	EPIG deferred	5.8	$+0.9$ $[-3.1, +4.8]$	0.61	1.00
C5 no history	EPIG-VoT	25.1	EPIG deferred	37.8	$+12.7$ $[+1.1, +24.1]$	0.04	0.48
C6 grouped	SCOUT-VoT	1.4	KG-VoT	1.8	$+0.4$ $[-0.8, +1.7]$	0.61	1.00
C6 cell	SCOUT-VoT	0.0	KG-VoT	1.1	$+1.1$ $[+0.0, +3.1]$	0.18	1.00
C6 no history	SCOUT-VoT	33.2	KG-VoT	29.8	-3.4 $[-15.8, +8.6]$	0.50	1.00
C6 grouped	SCOUT-VoT	1.4	early-prediction loop	13.2	$+11.9$ $[+5.1, +19.8]$	0.0022	0.033
C6 cell	SCOUT-VoT	0.0	early-prediction loop	1.8	$+1.8$ $[+0.4, +3.9]$	0.028	0.36
C6 no history	SCOUT-VoT	33.2	early-prediction loop	18.1	-15.1 $[-28.4, -2.7]$	0.057	0.57
C1 concrete, hist.	SCOUT-VoT	1.7	SCOUT-own	1.5	-0.2 $[-2.0, +1.7]$	0.53	1.00
C1 concrete, none	SCOUT-VoT	3.4	SCOUT-own	7.3	$+3.9$ $[+0.7, +8.1]$	0.013	0.10
C4 concrete, hist.	SCOUT-VoT	1.7	EPIG-VoT	2.3	$+0.6$ $[-1.2, +2.8]$	0.90	1.00
C4 concrete, none	SCOUT-VoT	3.4	EPIG-VoT	9.9	$+6.6$ $[+3.3, +10.2]$	5.8×10^{-4}	0.0058
C5 concrete, hist.	EPIG-VoT	2.3	EPIG deferred	1.0	-1.3 $[-3.3, +0.3]$	0.45	1.00
C5 concrete, none	EPIG-VoT	9.9	EPIG deferred	20.8	$+10.9$ $[+2.4, +20.6]$	0.048	0.34
C6 concrete, hist.	SCOUT-VoT	1.7	KG-VoT	1.2	-0.5 $[-1.2, +0.0]$	0.19	1.00
C6 concrete, none	SCOUT-VoT	3.4	KG-VoT	2.9	-0.5 $[-2.4, +1.1]$	0.81	1.00
C6 concrete, hist.	SCOUT-VoT	1.7	early-prediction loop	1.3	-0.4 $[-1.3, +0.6]$	0.40	1.00
C6 concrete, none	SCOUT-VoT	3.4	early-prediction loop	12.1	$+8.7$ $[+3.8, +14.3]$	0.0025	0.023

Table 26: Runs that printed the pool's best part and did not test it, out of the runs that printed it, at $(20, 12)$. The first five columns are the pre-registered families over 30 instances, the last three the fixed battery settings of §5.3 over 30 seeds.

	Battery families			Concrete families		Battery, fixed settings		
Arm	Grouped	Cell	No hist.	History	No hist.	No history	Earlier	Published
EPIG + IR, own-prediction threshold	15/23	20/25	10/15	8/27	1/13	5/10	0/24	30/30
SCOUT, own-prediction threshold	16/23	23/30	14/17	5/29	8/20	4/8	0/25	30/30
SCOUT, running threshold	11/22	16/30	2/16	16/28	5/19	3/8	2/21	14/30
SCOUT, pool threshold	0/22	0/30	1/13	0/26	0/15	0/5	0/23	0/30
EPIG + IR, value of testing	0/19	0/21	6/18	10/27	3/12	2/7	0/15	0/24
SCOUT, value of testing	0/25	0/30	6/14	9/29	3/20	9/11	1/23	0/30

Table 27: The new arms on the fixed settings of §5.3 and on the function networks, next to earlier arms, cell-level regret as mean followed by standard error over 30 seeds. These results are exploratory. Bold marks the lowest mean in a column.

	No history		Earlier protocols		Published	Ackley–	Ackley–
Arm	$(20, 12)$	$(30, 18)$	$(20, 12)$	$(30, 18)$	split	Matyas	sine
EI, x only	**27.3 (6.2)**	**8.5 (4.3)**	59.6 (5.3)	42.2 (4.9)	**0.0 (0.0)**	6.9 (1.0)	27.2 (4.5)
EPIG + IR	48.6 (5.9)	36.5 (6.2)	6.3 (2.2)	2.0 (0.6)	10.5 (2.3)	4.8 (0.7)	31.6 (4.1)
EPIG + IR, own-prediction threshold	46.0 (6.0)	27.6 (5.2)	1.9 (0.8)	2.0 (0.9)	23.0 (0.5)	5.6 (0.8)	31.1 (4.2)
SCOUT − inner stage	50.2 (5.5)	41.6 (5.9)	3.6 (0.7)	2.1 (0.6)	**0.0 (0.0)**	8.7 (4.2)	20.5 (4.0)
SCOUT, own-prediction threshold	52.0 (5.3)	27.9 (4.7)	1.2 (0.5)	0.2 (0.2)	21.8 (0.3)	6.2 (1.0)	22.4 (3.4)
SCOUT, pool threshold	53.9 (5.8)	47.3 (6.0)	1.7 (0.6)	0.2 (0.2)	**0.0 (0.0)**	**4.5 (0.9)**	**15.0 (3.2)**
SCOUT, value of testing	42.5 (5.9)	33.7 (5.8)	1.9 (0.6)	0.5 (0.3)	**0.0 (0.0)**	5.5 (0.8)	15.7 (3.3)
EPIG + IR, value of testing	44.3 (5.4)	17.9 (5.1)	4.3 (0.9)	1.3 (0.5)	4.7 (1.8)	6.1 (0.9)	24.9 (3.7)
EI + IR, value of testing	56.2 (5.6)	25.5 (5.4)	2.1 (0.6)	0.5 (0.3)	**0.0 (0.0)**	6.4 (1.0)	20.6 (4.8)
Random, value of testing	46.6 (5.7)	21.6 (4.1)	21.4 (4.1)	9.5 (2.6)	34.1 (4.2)	7.1 (1.1)	31.8 (3.8)
KG through the network	48.1 (6.3)	41.1 (6.5)	5.0 (0.6)	3.3 (0.7)	0.7 (0.7)	6.5 (0.9)	26.1 (4.1)
KG through the network, value of testing	44.4 (5.3)	28.8 (5.2)	2.1 (0.6)	0.7 (0.4)	**0.0 (0.0)**	6.5 (0.9)	24.5 (3.7)
EI-FN	50.2 (5.9)	40.8 (6.5)	5.7 (0.5)	4.7 (0.6)	**0.0 (0.0)**	6.9 (0.9)	25.8 (4.3)
EPIG + IR, deferred tests	58.7 (5.7)	50.6 (6.0)	**0.7 (0.4)**	**0.0 (0.0)**	21.8 (0.3)	8.8 (2.0)	55.9 (5.1)
SCOUT, deferred tests	55.2 (5.8)	49.3 (6.1)	0.9 (0.4)	0.9 (0.4)	21.4 (0.0)	11.6 (3.3)	56.8 (5.6)
Random, deferred tests	45.5 (4.9)	28.0 (3.8)	19.2 (4.0)	10.1 (3.0)	24.1 (3.6)	7.4 (1.0)	50.4 (4.5)
Early-prediction loop	37.6 (6.7)	29.6 (6.3)	7.1 (0.0)	**0.0 (0.0)**	27.3 (0.0)	8.3 (1.4)	53.2 (5.4)