SEMANTICBBOBENCH: A BENCHMARK FOR SEMANTIC BLACK-BOX OPTIMIZATION WITH LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Black-box optimization is central to many scientific and engineering disciplines. While numerical methods such as Bayesian optimization (BO) can successfully optimize black-box problems, these methods struggle to incorporate the semantic context that often accompanies real-world experimental problems, including domain knowledge, prior experiments, and expert intuition. Using this information could substantially reduce experimental costs and improve outcomes. Large language models (LLMs) can reason over both semantic and numerical data, offering a pathway for bringing this context into the optimization loop. We present SEMANTICBBOBENCH, a black-box optimization benchmark designed for testing LLM-based methods with semantic context. It provides a uniform framework containing 42 optimization problems spanning diverse scientific and engineering domains, each paired with a curated context at multiple controlled levels of detail and types of information. Our experiments show that semantic context provides its greatest benefit early in the optimization process, enabling substantially stronger initial configurations when few numerical observations are available. We also show that misleading semantic information can persistently bias LLM-based search. These results highlight both the potential and limitations of LLM-based black-box optimization and establish SEMANTICBBOBENCH as a framework for developing methods that use such information effectively.

1 INTRODUCTION

Black-box optimization is widely used in science and engineering when an objective is expensive to evaluate, lacks a tractable analytic form, or is accessible only through experiments or simulations. Despite this broad applicability, commonly used optimization benchmarks are dominated by synthetic test functions and machine-learning hyperparameter-tuning tasks. Although these benchmarks provide controlled and reproducible evaluations, they omit an important feature of many real-world optimization problems: decision variables have meaningful identities, and practitioners possess qualitative knowledge about their effects, interactions, constraints, and trade-offs.

For example, consider optimizing the yield of a crop by choosing its variety, planting density, irrigation, and fertilizer levels. A conventional optimizer may observe a vector of settings and the resulting yield, but a grower also knows that too little water or fertilizer can limit growth, that excessive amounts may be wasteful or harmful, and that the best management strategy depends on the crop variety and growing conditions. Conventional black-box optimization methods generally cannot use this information in its natural-language form. Instead, it must be manually translated into structured priors, kernels, constraints, or engineered features. This requires both domain and optimization expertise, which many experimentalists may not have.

LLM-based Optimization Direct use of an LLM as a proposal mechanism is attractive for both practical and diagnostic reasons. A natural-language interface can condition proposals on task descriptions, variable meanings, qualitative mechanisms, constraints, and previous evaluations, reducing the need to translate each form of knowledge into a task-specific prior, kernel, or acquisition function (Song et al., 2024). Prompt-based methods can operate without task-specific training and have demonstrated the feasibility of iterative LLM-based optimization (Yang et al., 2024; Guo

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.

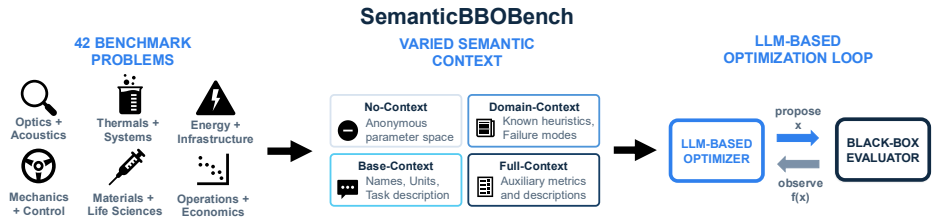


Figure 1: Overview of SEMANTICBBOBENCH benchmarks, context, and optimization loop.

et al., 2023). This capability may be especially useful at the beginning of expensive experimental campaigns, when observations are sparse and pretrained knowledge can warm-start search (Liu et al., 2024). Direct evaluation also provides a clean test of whether an LLM can convert semantic knowledge into better experimental decisions, without attributing improvements to a conventional optimizer embedded in a hybrid pipeline. This distinction is important because prior work finds that LLMs often struggle with purely numerical search yet can benefit substantially from supplied heuristics (Huang et al., 2024). However, these distinctions have received limited evaluation, as existing black-box optimization benchmarks primarily evaluate numerical search on synthetic functions or hyperparameter-tuning tasks. To our knowledge, none systematically control natural-language semantic information to measure how it affects optimization performance.

Contributions We introduce SEMANTICBBOBENCH, a benchmark for evaluating semantic black-box optimization. The benchmark contains 42 optimization problems spanning scientific, engineering, and economic domains, including energy systems, chemical reactions, structural design, control, aerodynamics, biotechnology, and portfolio construction. Each problem provides a standardized search space and black-box evaluator together with curated semantic information describing the domain setting, the meanings of the variables, qualitative mechanisms, constraints, and relevant heuristics. The suite includes a mix of continuous, integer, and categorical search spaces, as well as problems with infeasible regions. Such mixed-type search spaces are common in real-world problems but missing from many benchmarks and sampler implementations. Our key contributions are summarized below:

- We introduce SEMANTICBBOBENCH, a collection of 42 reproducible black-box optimization problems drawn from diverse application settings with varied parameter types. We provide a uniform interface supporting mixed-type search spaces, evaluation failures, auxiliary metrics, and both simulator and data-derived objective functions.
- We pair every problem with structured natural-language semantics and controlled context levels and types, enabling direct ablations of the value of general problem description, extended domain knowledge, and auxiliary metrics.
- We show that semantic context can yield large gains over numerical baselines and that these gains are driven primarily by warm-starting rather than by improved iterative search. We provide our benchmark as a testbed for developing semantic optimizers.

2 RELATED WORK

Black-Box Optimization Benchmarks Established black-box optimization benchmarks primarily measure numerical search. COCO and its BBOB suites provide controlled synthetic functions for continuous optimization (Hansen et al., 2021). Hyperparameter-optimization suites such as HPOBench and YAHPO Gym provide reproducible executable, tabular, and surrogate objectives derived from machine-learning tasks (Eggensperger et al., 2021; Pfisterer et al., 2022). MetaBox similarly provides a standardized platform for evaluating learned black-box optimizers over numerical functions (Ma et al., 2023). These suites enable rigorous comparisons, but are not designed to measure whether an optimizer can use semantic information to improve its search.

A related line of work evaluates language models on scientific reasoning and design. BrainBench tests whether LLMs can predict the outcomes of neuroscience experiments (Luo et al., 2025), while

BoxingGym evaluates agents that sequentially design experiments to identify hidden generative processes (Gandhi et al., 2025). SciDesignBench provides simulator-based inverse design tasks specified through natural-language goals (van Dijk & Vrkic, 2026). These benchmarks contain meaningful semantics, but target outcome prediction, scientific discovery, or inverse design, rather than sequential optimization of a black-box objective.

Knowledge-Informed Bayesian Optimization Bayesian optimization (BO) is widely used to optimize expensive black-box functions by combining a probabilistic surrogate with an acquisition function (Jones et al., 1998; Snoek et al., 2012; Shahriari et al., 2015; Frazier, 2018). Modern methods support continuous, categorical, and constrained search spaces, and some incorporate domain knowledge through priors over the optimum (Souza et al., 2021; Hvarfner et al., 2022), expert-informed kernels (Häse et al., 2021a), or elicited beliefs (Huang et al., 2022). These methods assume that knowledge has already been translated into a particular structured representation, such as a density over the optimum or a kernel encoding similarity. Natural-language context can be less structured, including detailed descriptions, expected relationships, or predicted failure modes. Converting this information to a structured prior is outside the scope of this work. Semantic optimization instead supplies this information directly to the optimizer, avoiding manual prior construction and providing a common interface for evaluating how effectively an optimizer can use it. Future work could explore connecting this information with a prior-informed BO framework.

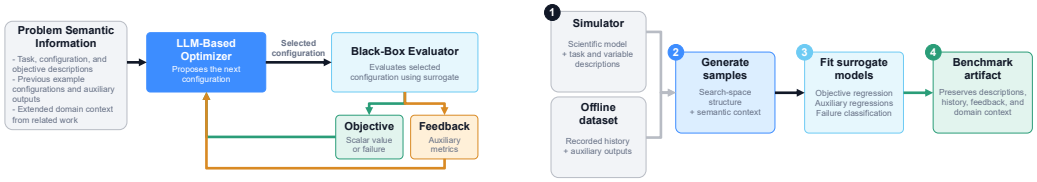
LLMs for Black-Box Optimization Recent work has explored LLMs both as direct optimizers and as components of model-based optimization pipelines. Direct approaches iteratively propose configurations from previous evaluations and can benefit from task descriptions or supplied heuristics, but often underperform specialized optimizers on purely numerical search (Yang et al., 2024; Guo et al., 2023; Huang et al., 2024). LLAMBO integrates LLMs into Bayesian optimization for warm-starting, surrogate modeling, and candidate generation (Liu et al., 2024), while other hybrid methods combine LLM proposals or representations with probabilistic optimization, including LLINBO, LGBO, and GOLLuM (Chang et al., 2025; Yuan et al., 2026; Ranković et al., 2026). Lastly, LILO translates free-form knowledge and feedback into BO preferences (Kobalczyk et al., 2025). Rather than proposing another integration mechanism, we isolate the effect of semantic information itself by evaluating an otherwise fixed LLM optimizer across controlled levels and types of context over diverse black-box problems. This minimizes the additional algorithmic components such that changes in performance can be more directly attributed to the amount and type of semantic information provided.

Overview Existing work provides either rigorous numerical optimization benchmarks without natural-language semantics or semantically meaningful tasks that do not evaluate sequential black-box optimization. To our knowledge, no existing benchmark evaluates sequential black-box optimization across diverse domains while independently controlling the natural-language semantic information available to the optimizer. SEMANTICBBOBENCH addresses this gap by pairing executable black-box problems with controlled information conditions, ranging from no-context search spaces to variable labels, domain descriptions, mechanisms, and qualitative heuristics.

3 METHODOLOGY

Semantic Black-Box Optimization We study sequential black-box optimization in which the optimizer also receives natural-language information about the problem. Each problem combines a mixed-type search space $\mathcal{X}$, spanning continuous, integer, categorical, and Boolean variables, with an objective $f : \mathcal{X} \to \mathbb{R} \cup \{\text{fail}\}$, and a fixed evaluation budget T. Alongside these, a semantic annotation $\mathcal{S}$ describes the task, the variables, the objective, the auxiliary measurements, the underlying domain mechanisms, and historical configurations from related work. At each step, the optimizer draws on $\mathcal{S}$ and the evaluations collected so far to propose a configuration, which we validate against the declared types, bounds, and domains before passing it to the evaluator. An overview of this process is seen in Figure 2a.

Benchmark Domains SEMANTICBBOBENCH comprises 42 optimization problems drawn from scientific, engineering, economic, and computational applications. An overview of the benchmark domains is shown in Table 1. The problems span both simulator-based and data-derived objectives



(a) Overview of semantic black-box optimization. (b) Benchmark creation pipeline.

Figure 2: Overview of the semantic black-box optimization and benchmark creation pipelines.

Domain	Representative problems
Energy & Infrastructure	Photovoltaics, wind-farm layout and wake steering, networks, charging, geothermal fields, anaerobic digestion, buildings
Thermals & Systems	Electrolyzers, thermoelectrics, power cycles, refrigeration, heat exchangers, heat sinks, desalination
Materials & Life Sciences	Reactions, combustion, nanoalloys, molecular design, metabolic engineering, crop management, measurement
Mechanics & Control	Aircraft missions, airfoils, interplanetary transfers, satellite constellations, vehicle powertrains, trusses, controller tuning
Optics & Acoustics	Antennas, matching networks, lenses, optical coatings, loudspeakers
Operations & Economics	Supply chains, pump scheduling, patient flow, pricing, portfolios, housing design

Table 1: We cover a wide range of problems and parameter types in SEMANTICBBOBENCH.

and have between 3 and 29 decision variables. Search spaces contain mixed combinations of continuous, discrete, categorical, and Boolean parameters. Many tasks also exhibit constrained feasible regions where failure cases are present.

Benchmark Creation We construct benchmarks from existing experimental datasets and open-source simulators. Each benchmark follows a common specification containing the optimization direction, parameter types and domains, objective description, auxiliary outputs, and semantic context. For each source system, we formulate a black-box optimization task by defining the tunable variables, objective, constraints, and available auxiliary outputs. The sources provide the underlying dataset, simulator, or scientific model rather than the benchmark formulation itself. We cover a broad range of application domains, parameter types, dimensionalities, and constraint structures. We use Anthropic Claude Cowork to assist with implementing simulator and dataset interfaces and drafting benchmark semantic context, before manually completing the implementations and revising the context (Anthropic, 2026). An overview of the construction process is shown in Figure 2b.

We fit offline surrogate models to each benchmark problem to provide a standardized and computationally inexpensive interaction loop without requiring users to install and execute the original simulation packages. Replacing an expensive objective with a fitted model is established practice in surrogate benchmarks for hyperparameter optimization, algorithm configuration, and neural architecture search (Eggensperger et al., 2015; 2018; Zela et al., 2020; Pfisterer et al., 2022), and in scientific-optimization suites that represent experimental systems using learned emulators (Häse et al., 2021b; Felton et al., 2021).

Because optimizers concentrate their evaluations in high-performing regions, we evaluate surrogate quality both globally and where optimization trajectories are likely to search. We combine Sobol samples with optimizer-visited samples, iteratively refine the surrogate in regions where it exhibits large residuals, and evaluate the final model on held-out Sobol samples and optimizer-visited configurations. Across the suite, this procedure uses 9.6 million true evaluations, with a median of 213k per problem. Each problem is fit with a deep gradient-boosted tree ensemble and three neural networks that differ in width, depth, and activation, combined by a convex blend whose weights are fitted on the validation split. The same set of candidate models is also fit for evaluation failure and

	R^2		NRMSE			Optimum	Feasibility classifier	
Statistic	Sobol	Search	All	Top-5%	Top-1%	Regret	Bal. acc.	Brier skill
Mean	0.9951	0.9967	0.0072	0.0028	0.0030	0.0006	0.9829	0.9475
Best decile	> 0.9999	> 0.9999	0.0008	0.0002	0.0002	0.0000	1.0000	1.0000
Median	0.9989	0.9995	0.0055	0.0014	0.0013	0.0000	0.9935	0.9786
Worst decile	0.9887	0.9850	0.0157	0.0072	0.0080	0.0008	0.9604	0.8734

Table 2: Test fit across the 42 surrogates, on a holdout split. R^2 is given on the Sobol holdout and the optimizer-visited search holdout. NRMSE is normalized by the feasible training-target P_1-P_{99} span. Normalized optimum regret is computed by searching the final surrogate under a fixed budget and evaluating its best designs on the original evaluator. The feasibility classifier predicts evaluation failure, scored on its own held-out split for the 30 problems with failures.

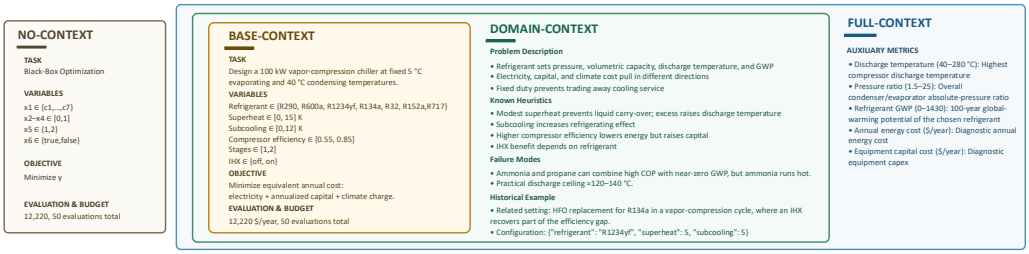


Figure 3: Semantic information example, illustrated using the vapor-compression chiller benchmark. The conditions are nested: *Base-context* assigns semantic meaning to the task, variables, and objective; *Domain-context* additionally provides long-form domain knowledge and relevant historical examples; and *Full-context* further provides auxiliary metric descriptions and observations. *No-context* anonymizes labels and normalizes numerical ranges. The entries above provide abbreviated examples of the complete data.

regressors for each auxiliary output. To further improve surrogate quality, we search the fitted surrogate using an evolutionary strategy and evaluate its highest-ranked designs on the true evaluator. We add the samples with the largest gap to the training dataset and refit in an iterative process. We show the surrogate fits are very close to the ground truth data across a variety of metrics in Table 2 and that optimizer results on surrogate models and ground-truth evaluators are very similar in Appendix D.

Benchmark Semantic Information Each benchmark is paired with semantic context derived from documentation associated with the underlying simulator, dataset, or scientific model, together with relevant source publications when available. We use Claude Cowork to convert these materials into a standardized description containing the task setting, variable meanings and units, qualitative mechanisms, known constraints, and failure modes. We then manually review the generated context against the source material, correcting factual errors, removing unsupported claims, and excluding information that directly reveals benchmark-specific optima. The reviewed information is then divided into the semantic categories used to construct the different context variants.

Context Level Axis We evaluate the language-model optimizer under four context level variants. All variants include the parameter types and optimization direction required to produce a valid configuration. Examples of the information included in each variant are shown in Figure 3. The *no-context* variant uses anonymous parameter identifiers, normalized ranges, and no task description, because exposing the original ranges and units could provide semantic cues. The *base-context* variant adds meaningful parameter names, units, and descriptions of the task, objective, and variables. The *domain-context* variant provides long-form domain knowledge describing relevant mechanisms, trade-offs, constraints, and failure conditions, together with historical configurations. These historical configurations contain a fraction of the overall parameters to inform the optimizer and cannot be used to propose a full configuration. The *full-context* variant further includes descriptions and observations of auxiliary metrics returned by the evaluator. These are secondary quantities computed

```
System: <Defines the model's role and required output>
User: Long-form context: <Further descriptions and domain knowledge>
History: <Examples from related problems or standard configurations>
Problem name and description: <Brief overview of the problem>
Tunable search space: <Parameters, ranges, units, and descriptions>
Objective: <Objective name, direction, range, unit, and description>
Auxiliary metrics: <Auxiliary metric descriptions>
Budget: <Total available evaluations>
Assistant: {"initial_configuration_1": <proposed parameter values>, ...,
    "initial_configuration_5": <proposed parameter values>}
User: Evaluation 1: Configuration: <proposed parameter values>
Objective: <Objective value>, Auxiliary metrics: <Auxiliary metrics>
# Evaluations 2-4 condensed for space
Evaluation 5: Configuration: <proposed parameter values>
Objective: <Objective value>, Auxiliary metrics: <Auxiliary metrics>
Assistant: {"next_configuration": <proposed parameter values>,
  "mean": <predicted objective>, "std": <predicted uncertainty>,
  "failure_probability": <predicted failure probability>}
User: Evaluation 6: Objective: <Objective value>
Auxiliary metrics: <Auxiliary metrics>
```

■ Base-Context ■ Domain-Context ■ Full-Context

Figure 4: Prompt format for semantic black-box optimization. Green text identifies information included in the *base-context* variant. The *domain-context* variant additionally includes the long-form domain description and optimization history, shown in purple. The *full-context* variant further includes auxiliary metric descriptions and observations, shown in blue. The conversation continues in a multi-turn evaluation loop until the available budget is exhausted.

alongside the objective, such as intermediate simulation outputs or constraint margins, that are not optimized directly but can indicate why a configuration performed as it did.

Context Type Axis We are also interested in a model's robustness to different types of contextual information. Regarding memorization, a model may recognize a problem from its wording, parameter names, or source dataset and reproduce a memorized configuration. Corpus filtering does not resolve this, since paraphrased benchmark content survives decontamination and contamination is specific to a benchmark and a model (Sainz et al., 2023; Yang et al., 2023). We therefore build a *counterfactual* variant of each benchmark that preserves the domain and semantic meaning while removing surface identity: semantic context is rewritten using a semantically equivalent phrasing. This type of counterfactual change has been proposed as a method to reduce the effect of memorization in previous works (Kaushik et al., 2019; Gardner et al., 2020; Wu et al., 2024). Next, we test robustness using a deliberately misleading prior, which is an established practice in prior-guided Bayesian optimization (Souza et al., 2021; Hvarfner et al., 2022). We build an *adversarial* variant on the counterfactual base, preserving the search space, variable names, bounds, optimization direction, and evaluator while making the optimization-relevant claims false: heuristics are inverted, mechanisms misattributed, suggested regions relocated, and partial example configurations replaced with unfavorable configurations. Lastly, we include a *mismatched* context variant, which contains plausible context for a completely different real-world problem with the same search space structure. The goal here is to test how unrelated context affects the results. All arms run against the same fitted surrogate at equal budget, so they separate the contribution of the supplied semantic information.

Optimization Process We formulate evaluation as sequential black-box optimization under a fixed evaluation budget and implement the process as a multi-turn LLM conversation, as shown in Figure 4. The optimizer first proposes five initial configurations, which are evaluated individually. It then proceeds iteratively, proposing one configuration per turn based on the semantic context and all observations collected so far. For each proposed configuration, the evaluator returns an objective value or indicates that the evaluation is infeasible. During the iterative phase, the LLM optimizer returns a structured JSON object containing the proposed configuration together with predictions of the objective mean, uncertainty, and failure probability if a failure has been observed.

4 RESULTS

Baselines Our baselines are three popular numerical optimizers with distinct inductive biases (a Gaussian process, a density-ratio model, and an evolution strategy) and two language models (one frontier, one open-weight) that also receive the semantic context.

- **HEBO GP:** A Bayesian optimization method designed for mixed search spaces (Cowen-Rivers et al., 2022), using the Gaussian process surrogate implementation in OptunaHub (Ozaki et al., 2025). We select it as our primary BO baseline for its handling of mixed search spaces and strong performance on classical benchmarks (Turner et al., 2021).
- **TPE:** A Tree-structured Parzen Estimator sampler (Bergstra et al., 2011), using the Optuna implementation (Akiba et al., 2019). It models the search space through density ratios rather than a global surrogate, a different inductive bias from the Gaussian process baseline.
- **CatCMA with Margin:** A mixed-variable evolution strategy extending CMA-ES to integer and categorical spaces (Hamano et al., 2025).
- **GPT-5.6 Luna:** We use GPT-5.6 Luna in medium reasoning mode (OpenAI, 2026), the smaller model in the GPT-5.6 family. This smaller model is affordable for the large numbers of calls in the test suite while remaining a frontier reasoning model.
- **Gemma 4 31B:** An open-weight counterpart (Team et al., 2026) in its non-thinking mode, under the same prompts, context conditions, and budget. Its fixed weights and serving stack make the language model arm reproducible.

To aggregate performance across objective scales, we orient all objectives so that higher is better, then normalize between a typical search-space value and a fixed best-known reference:

$$S = (f_{\text{best}} - f_m)/(f_{\text{ref}} - f_m),$$

where f_{best} is the best objective found by the optimizer within the search budget, f_m is the median objective over a fixed 65,536 evaluation Sobol sample and f_{ref} is the best objective found by a 262,144 evaluation evolution-strategy search on the fitted surrogate. This procedure follows standard practice in black-box optimization benchmarking (Turner et al., 2021; Pfisterer et al., 2022). We clip scores below -1 to prevent extreme outliers having an outsized effect.

Overall Results We show the aggregate performance across the 42 benchmarks for all baselines, with the LLM variants provided full context, in Figure 5. Semantic context provides its largest advantage early. After 10 evaluations, GPT-5.6 Luna with *full-context* reaches a mean normalized score of 0.755, compared with 0.478 for HEBO GP and 0.394 for random search. By 50 evaluations HEBO GP has largely closed this gap, reaching 0.800 against 0.826 for *full-context*. The pattern is consistent with a limitation in the LLM's iterative search rather than in its use of context. The advantage does not require a frontier model: evaluated on the same benchmarks under identical prompts and budget, the open-weight Gemma 4 31B achieves similar performance to GPT-5.6 Luna.

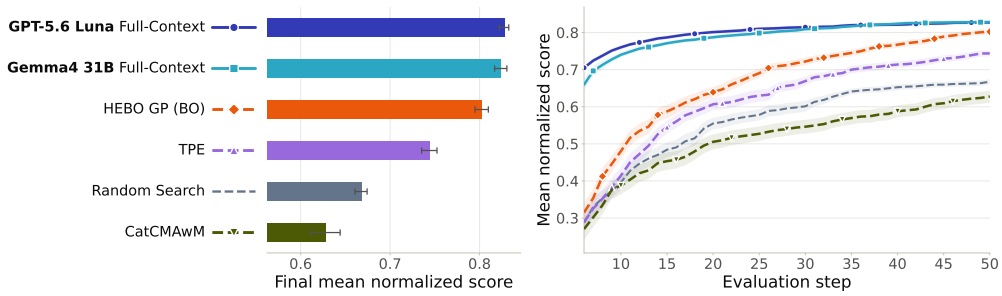


Figure 5: Semantic context provides a strong advantage at small evaluation budgets, while HEBO GP gradually closes the gap over steps. Other baseline methods perform worse than HEBO GP. Bars and curves show the mean and standard error over five random seeds per benchmark.

Context Level Axis We show the performance of LLM-based black-box optimization and Bayesian optimization methods as a function of the evaluation budget in Figure 6. The context ablation shows that the LLM's performance depends strongly on semantic information. The *no-context* variant remains well below all context-aware variants, reaching a score of 0.630 after 50 evaluations, below both HEBO GP and random search. This highlights the limited iterative search ability of the model when semantic information is withheld. Including *base-context* leads to a large increase in initial and overall performance relative to *no-context*. Adding long-form domain knowledge and historical configuration data in *domain-context* and auxiliary metrics in *full-context* each further improve performance, showing that adding more information helps the LLM.

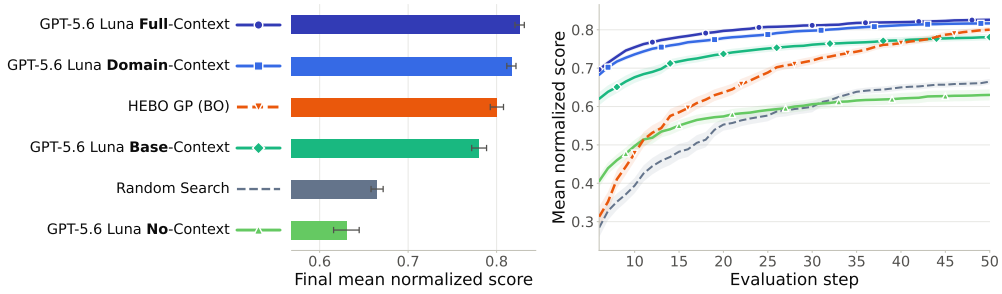


Figure 6: Increasing the level of semantic information provided improves LLM optimization performance, while the LLM performs very poorly when given no context. Bars and curves show the mean and standard error over five random seeds per benchmark.

Context Type Axis The robustness of LLM-based methods to misleading or incorrect semantic information is another important condition to consider when adding priors to the problem space. If the LLM is easily misled by incorrect information and fails to update priors based on the evaluation results, we may see worse performance than BO. We test this by generating adversarial contextual information within the long-form and history context that misleads the model to poor search regions. The goal is to provide plausible but misleading suggestions, testing if the model adapts its search after seeing poor evaluations. We show the results in Figure 7. The *counterfactual* rewrite leads to a slight decrease in performance, but is a fairly limited change overall. The limited degradation under counterfactual rewriting suggests that surface-form recognition is unlikely to be the primary explanation for the observed semantic advantage. The *adversarial* context, by contrast, has a large and persistent effect. The model only partially recovers from its poor starting points even after repeatedly observing evaluations that contradict the supplied guidance. Lastly, the *mismatched* variant shows near-random performance similar to *no-context* and also fails to adapt to the numerical data.

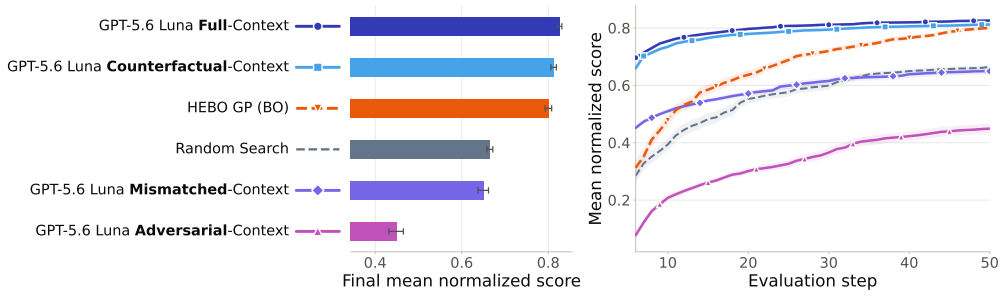


Figure 7: Counterfactual rewriting causes only a modest degradation, suggesting limited impact of memorization, whereas adversarial context substantially reduces performance and mismatched context performs on par with no-context. Bars and curves show the mean and standard error over five random seeds per benchmark.

LLM + HEBO GP Combination The previous results suggest that the two methods have complementary strengths. The LLM uses semantic context to select strong initial configurations, while HEBO GP searches more effectively once it has collected enough observations. We therefore test a

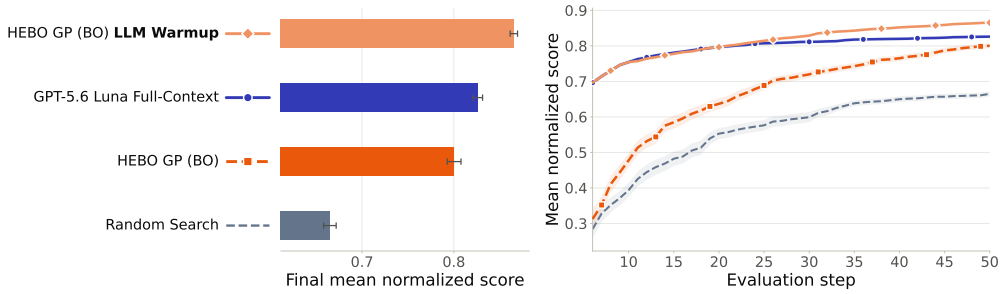


Figure 8: The combination sampler uses GPT-5.6 Luna with *full-context* for the first ten evaluations and then transfers the resulting observations to HEBO GP. The combination method outperforms both individual methods, suggesting a path toward stronger semantic optimizers: preserve the LLM's ability to identify promising regions while improving or handing off the iterative search process. Bars and curves show the mean and standard error over five random seeds per benchmark.

combined sampler, where GPT-5.6 Luna with *full-context* proposes the first ten configurations and HEBO GP is then initialized on these evaluations and proposes the rest. We show the results in Figure 8, where the combination reaches 0.865 after 50 evaluations compared to 0.826 for GPT-5.6 Luna and 0.800 for HEBO GP. This shows that the semantic warm-start remains useful when combined with HEBO GP, and that GPT-5.6 Luna has weaknesses in its iterative search capabilities.

Limitations SEMANTICBBOBENCH provides a controlled evaluation of semantic black-box optimization, but it does not fully reproduce real experimental settings. The semantic context is also manually curated. Its usefulness depends on what information is included and how it is expressed, while real users may provide incomplete, ambiguous, or conflicting knowledge. In addition, pretraining contamination cannot be ruled out, since models may have encountered related domains, datasets, or scientific relationships during training. The benchmark also covers a broad but incomplete set of problems and evaluates only a limited set of optimizers. The results should therefore not be interpreted as a universal comparison between LLMs and numerical optimization methods. The robustness results also show that misleading context can strongly bias the LLM and that current models may fail to recover from incorrect semantic priors.

5 CONCLUSION

We introduce SEMANTICBBOBENCH, a benchmark for evaluating semantic black-box optimization across 42 problems drawn from diverse scientific, engineering, economic, and computational domains. By pairing each problem with controlled levels of semantic context, the benchmark separates an optimizer's ability to exploit semantic knowledge from its ability to learn numerically from previous evaluations. This enables structured evaluation of LLM-based optimization methods.

Our results show that semantic information provides its largest benefit early in optimization, enabling stronger initial configurations when few observations are available. This advantage narrows as evaluations accumulate, while an LLM warm start followed by HEBO GP outperforms either method alone, suggesting that current LLMs are better at identifying promising regions than at iterative numerical refinement. However, adversarial context substantially degrades performance and models may fail to revise misleading priors, while mismatched context provides no benefit over no context. These findings motivate semantic optimization methods that exploit language-based priors early while increasingly relying on numerical evidence as optimization proceeds.

As a whole, these results argue for continued development of LLM-based optimization, guided by the strengths and weaknesses the benchmark exposes. Language models can translate domain knowledge into promising initial regions, hypotheses, and constraints, while Bayesian optimization can provide systematic search and adaptation from observations. SEMANTICBBOBENCH provides a controlled framework for developing and evaluating improved LLM-based methods, including identifying when semantic information helps, when it is unnecessary, and when it becomes harmful.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, designing and providing feedback on research methodology or experiments, proposing or refining hypotheses, implementing methods, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, and interpreting results. We have not used generative AI tools for assisting with translation or formulating mathematical claims. Providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, suggesting experimental parameters, creating or editing software code, creation of artifacts, drafting parts of a research paper, summarizing or analyzing existing literature, discovering research topics or identify gaps, brainstorming, sourcing/searching for information, and identifying relevant literature. We have reviewed all AI-assisted work. The LLM-generated code in this work was reviewed and tested for correctness. In particular to this benchmark, the semantic information and artifacts relevant to this work were reviewed for correctness. AI-tools were used for light editing of this document. 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 benchmark and the associated evaluations are created for the purpose of the advancement of understanding optimizer capabilities. We do not see any ethical concerns related to this.

REPRODUCIBILITY STATEMENT

The code and data for this paper are available at `https://anonymous.4open.science/r/semanticbbobench_iclr2027-C429/README.md`. The release contains the definition of every benchmark problem (search space, objective, semantic context in all context variants, and source evaluator), the surrogate-construction pipeline, the samplers and experiment drivers for every method we compare, and the recorded results of all experiments. A single script, `scripts/reproduce_paper.sh`, regenerates every data-driven figure and table in the paper from these results. Appendix E describes the release in detail, including which surrogate artifacts are distributed and how to evaluate new methods or new problems.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Paul M Anglin and Ramazan Gencay. Semiparametric estimation of a hedonic price function. *Journal of applied econometrics*, 11(6):633–648, 1996.

Anthropic. Claude cowork. `https://www.anthropic.com`, 2026.

Alexander Arsenovic, Julien Hillairet, Jackson Anderson, Henrik Forsten, Vincent Rieß, Michael Eller, Noah Sauber, Robert Weikle, William Barnhart, and Franz Forstmayr. scikit-rf: An open source python package for microwave network creation, analysis, and calibration [speaker's corner]. *IEEE Microwave Magazine*, 23(1):98–105, 2021.

Brent R Asplin, David J Magid, Karin V Rhodes, Leif I Solberg, Nicole Lurie, and Carlos A Camargo Jr. A conceptual model of emergency department crowding. *Annals of emergency medicine*, 42(2):173–180, 2003.

Astropy Collaboration, Thomas P. Robitaille, Erik J. Tollerud, Perry Greenfield, Michael Droettboom, Erik Bray, Tom Aldcroft, Matt Davis, Adam Ginsburg, Adrian M. Price-Whelan, Wolfgang E. Kerzendorf, Alexander Conley, Neil Crighton, Kyle Barbary, Demitri Muna, Henry Ferguson, Frédéric Grollier, Madhura M. Parikh, Prasanth H. Nair, Hans M. Unther, Christoph Deil, Julien Woillez, Simon Conseil, Roban Kramer, James E. H. Turner, Leo Singer, Ryan Fox, Benjamin A. Weaver, Victor Zabalza, Zachary I. Edwards, K. Azalee Bostroem, D. J. Burke,

Andrew R. Casey, Steven M. Crawford, Nadia Dencheva, Justin Ely, Tim Jenness, Kathleen Labrie, Pey Lian Lim, Francesco Pierfederici, Andrew Pontzen, Andy Ptak, Brian Refsdal, Mathieu Servillat, and Ole Streicher. Astropy: A community Python package for astronomy, October 2013.

Richard W Baker. *Membrane technology and applications*. John Wiley & Sons, 2023.

Damien J Batstone, Jurg Keller, Irini Angelidaki, Sergey V Kalyuzhnyi, Spyros G Pavlostathis, Alberto Rozzi, Wendy TM Sanders, Hansruedi Siegrist, and Vasily A Vavilin. The iwa anaerobic digestion model no 1 (adm1). *Water Science and technology*, 45(10):65–73, 2002.

Ian H Bell, Jorrit Wronski, Sylvain Quoilin, and Vincent Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & engineering chemistry research*, 53(6):2498–2508, 2014.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Neural Information Processing Systems*, 2011.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

Steven J Byrnes. Multilayer optical calculations. *arXiv preprint arXiv:1603.02720*, 2016.

Antonio C Caputo, Pacifico M Pelagagge, and Paolo Salini. Heat exchanger design based on economic optimisation. *Applied thermal engineering*, 28(10):1151–1159, 2008.

Marcelo Carmo, David L Fritz, Jürgen Mergel, and Detlef Stolten. A comprehensive review on pem water electrolysis. *International journal of hydrogen energy*, 38(12):4901–4934, 2013.

Chih-Yu Chang, Milad Azvar, Chinedum Okwudire, and Raed Al Kontar. Llinbo: Trustworthy llm-in-the-loop bayesian optimization. *arXiv preprint arXiv:2505.14756*, 2025.

Massimo Cimmino, D Thorne, CD Langevin, and MC Sukop. Pygfunction: an open-source toolbox for the evaluation of thermal response factors for geothermal borehole fields. *Proceedings of eSim*, 2018, 2018.

Andrew J Clark and Herbert Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, 6(4):475–490, 1960.

Alexander I Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan Rhys Griffiths, Alexandre Max Maraval, Hao Jianye, Jun Wang, Jan Peters, et al. Hebo: Pushing the limits of sample-efficient hyper-parameter optimisation. *Journal of Artificial Intelligence Research*, 74:1269–1349, 2022.

Ali Ebrahim, Joshua A Lerman, Bernhard O Palsson, and Daniel R Hyduke. Cobrapy: constraints-based reconstruction and analysis for python. *BMC systems biology*, 7(1):74, 2013.

Katharina Eggensperger, Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. Efficient benchmarking of hyperparameter optimizers via surrogates. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Katharina Eggensperger, Marius Lindauer, Holger H Hoos, Frank Hutter, and Kevin Leyton-Brown. Efficient benchmarking of algorithm configurators via model-based surrogates. *Machine Learning*, 107(1):15–41, 2018.

Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. Hpobench: A collection of reproducible multi-fidelity benchmark problems for hpo. *arXiv preprint arXiv:2109.06716*, 2021.

Kobi C. Felton, Jan G. Rittig, and Alexei A. Lapkin. Summit: Benchmarking machine learning methods for reaction optimisation. *Chemistry–Methods*, 1(2):116–122, 2021. doi: https://doi.org/10.1002/cmtd.202000051. URL https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/cmtd.202000051.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Sawyer Fuller, Ben Greiner, Jason Moore, Richard Murray, René van Paassen, and Rory Yorke. The python control systems library (python-control). In *60th IEEE Conference on Decision and Control (CDC)*, pp. 4875–4881. IEEE, 2021.

Kanishk Gandhi, Michael Y Li, Lyle Goodyear, Agam Bhatia, Louise Li, Aditi Bhaskar, Mohammed Zaman, and Noah D Goodman. Boxinggym: Benchmarking progress in automated experimental design and model discovery. *arXiv preprint arXiv:2501.01540*, 2025.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1307–1323, 2020.

David G Goodwin, Raymond L Speth, Harry K Moffat, and Bryan W Weber. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. *Zenodo*, 2018.

Kristen B Gorman, Tony D Williams, and William R Fraser. Ecological sexual dimorphism and environmental variability within a community of antarctic penguins (genus pygoscelis). *PloS one*, 9(3):e90081, 2014.

Pei-Fu Guo, Ying-Hsuan Chen, Yun-Da Tsai, and Shou-De Lin. Towards optimizing with large language models. *arXiv preprint arXiv:2310.05204*, 2023.

Ryoki Hamano, Masahiro Nomura, Shota Saito, Kento Uchida, and Shinichi Shirakawa. Catcma with margin: Stochastic optimization for continuous, integer, and categorical variables. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 737–745, 2025.

Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. Coco: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, 2021.

Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, 8(3), 2021a.

Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

William F Holmgren, Clifford W Hansen, and Mark A Mikofski. pvlib python: A python package for modeling solar energy systems. *Journal of Open Source Software*, 3(29):884, 2018.

Christopher A Hone, Nicholas Holmes, Geoffrey R Akien, Richard A Bourne, and Frans L Muller. Rapid multistep kinetic model generation from transient flow data. *Reaction chemistry & engineering*, 2(2):103–108, 2017.

Beichen Huang, Xingyu Wu, Yu Zhou, Jibin Wu, Liang Feng, Ran Cheng, and Kay Chen Tan. Exploring the true potential: Evaluating the black-box optimization capability of large language models. *arXiv preprint arXiv:2404.06290*, 2024.

Daolang Huang, Louis Filstroff, Petrus Mikkola, Runkai Zheng, and Samuel Kaski. Bayesian optimization augmented with actively elicited expert knowledge. *arXiv preprint arXiv:2208.08742*, 2022.

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πbo: Augmenting acquisition functions with user beliefs for bayesian optimization, 2022. URL `https://arxiv.org/abs/2204.11051`.

Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, Melodie Christensen, Elena Liles, Jason E Hein, and Alán Aspuru-Guzik. Olympus: a benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2(3):035021, 2021b. ISSN 2632-2153. doi: 10.1088/2632-2153/abedc8. URL http://dx.doi.org/10.1088/2632-2153/abedc8.

Prageeth Jayathissa, Mauro Luzzatto, Jeremias Schmidli, Johannes Hofer, Zoltan Nagy, and Arno Schlueter. Optimising building net energy demand with dynamic bipv shading. *Applied Energy*, 202:726–735, 2017.

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

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

Rudolf Kingslake and R Barry Johnson. *Lens design fundamentals*. academic press, 2009.

Katherine A Klise, Regan Murray, and Terra Haxton. An overview of the water network tool for resilience (wntr)., 2018.

Katarzyna Kobalczyk, Zhiyuan Jerry Lin, Benjamin Letham, Zhuokai Zhao, Maximilian Balandat, and Eytan Bakshy. Lilo: Bayesian optimization with natural language feedback. *arXiv preprint arXiv:2510.17671*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Greg Landrum et al. Rdkit documentation. *Release*, 1(1-79):4, 2013.

Wiley J Larson and James Richard Wertz. Space mission analysis and design. Technical report, Torrance, CA (United States); Microcosm, Inc., 1992.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. *arXiv preprint arXiv:2402.03921*, 2024.

Xiaoliang Luo, Akilles Rechardt, Guangzhi Sun, Kevin K Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O Cohen, Valentina Borghesani, Anton Pashkov, et al. Large language models surpass human experts in predicting neuroscience results. *Nature human behaviour*, 9(2): 305–315, 2025.

Zeyuan Ma, Hongshu Guo, Jiacheng Chen, Zhenrui Li, Guojun Peng, Yue-Jiao Gong, Yining Ma, and Zhiguang Cao. Metabox: A benchmark platform for meta-black-box optimization with reinforcement learning. *Advances in Neural Information Processing Systems*, 36:10775–10795, 2023.

National Laboratory of the Rockies. FLORIS: Wake modeling and wind farm controls software, 2026. URL https://github.com/NatLabRockies/floris.

Masahiro Nomura, Masashi Shibata, and Ryoki Hamano. cmaes: A simple yet practical python library for cma-es. *arXiv preprint arXiv:2402.01373*, 2024.

OpenAI. GPT-5.6. https://openai.com/index/previewing-gpt-5-6-sol/, 2026.

Yoshihiko Ozaki, Shuhei Watanabe, and Toshihiko Yanase. Optunahub: A platform for black-box optimization. *arXiv preprint arXiv:2510.02798*, 2025.

Florian Pfisterer, Lennart Schneider, Julia Moosbauer, Martin Binder, and Bernd Bischl. Yahpo gym – an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization, 2022. URL https://arxiv.org/abs/2109.03670.

R. Quinlan. Auto MPG. UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5859H.

Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery. *Nature Machine Intelligence*, pp. 1–12, 2026.

Frank M Redington. Review of the principles of life-office valuations. *Journal of the Institute of Actuaries (1886-1994)*, 78(3):286–340, 1952.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, 2023.

LA Schmit Jr and B Farshi. Some approximation concepts for structural synthesis. *AIAA journal*, 12(5):692–699, 1974.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Peter Sharpe and R John Hansman. Neuralfoil: An airfoil aerodynamics analysis tool using physics-informed machine learning. *arXiv preprint arXiv:2503.16323*, 2025.

Muhammed E Shibu, Peter A Leffelaar, Herman Van Keulen, and Pramod K Aggarwal. Lintul3, a simulation model for nitrogen-limited situations: Application to rice. *European Journal of Agronomy*, 32(4):255–271, 2010.

SimPy Developers. Simpy: Discrete event simulation for python, 2023. URL `https://simpy.readthedocs.io/`. Version 4.1.1.

Richard H Small. Vented-box loudspeaker systems–part 1: Small-signal analysis. *J. Audio Eng. Soc*, 21(5):363–372, 1973.

Jasper Snoek, Hugo Larochelle, and Ryan Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

G Jeffrey Snyder and Eric S Toberer. Complex thermoelectric materials. *Nature materials*, 7(2): 105–114, 2008.

Xingyou Song, Yingtao Tian, Robert Tjarko Lange, Chansoo Lee, Yujin Tang, and Yutian Chen. Position: Leverage foundational models for black-box optimization. *arXiv preprint arXiv:2405.03547*, 2024.

Artur Souza, Luigi Nardi, Leonardo B Oliveira, Kunle Olukotun, Marius Lindauer, and Frank Hutter. Bayesian optimization with a prior for the optimum. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 265–296. Springer, 2021.

Junzi Sun, Jacco M Hoekstra, and Joost Ellerbroek. Openap: An open-source aircraft performance model for air transportation studies and simulations. *Aerospace*, 7(8):104, 2020.

Jonathan Taylor, Trevor Hastie, Gareth James, Robert Tibshirani, and Daniela Witten. ISLP: Data sets and helper functions for an introduction to statistical learning with applications in python, 2026. URL `https://github.com/intro-stat-learning/ISLP`. Python software package.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

P Teertstra, MM Yovanovich, and JR Culham. Analytical forced convection modeling of plate fin heat sinks. *Journal of Electronics Manufacturing*, 10(04):253–261, 2000.

Leon Thurner, Alexander Scheidler, Florian Schäfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, 2018.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments, 2025. URL https://arxiv.org/abs/2407.17032.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pp. 3–26. PMLR, 2021.

David van Dijk and Ivan Vrkic. Scidesignbench: Benchmarking and improving language models for scientific inverse design. *arXiv preprint arXiv:2603.12724*, 2026.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 1819–1862, 2024.

Hidetsugu Yagi and Shintaro Uda. Projector of the sharpest beam of electric waves. *Proceedings of the Imperial Academy*, 2(2):49–52, 1926.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers, 2024. URL https://arxiv.org/abs/2309.03409.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.

I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

Xinzhe Yuan, Zhuo Chen, Jianshu Zhang, Huan Xiong, Nanyang Ye, Yuqiang Li, and Qinying Gu. Unleashing llms in bayesian optimization: Preference-guided framework for scientific discovery. In *The Fourteenth International Conference on Learning Representations*, 2026.

Arber Zela, Julien Siems, Lucas Zimmer, Jovita Lukasik, Margret Keuper, and Frank Hutter. Surrogate nas benchmarks: Going beyond the limited search spaces of tabular nas benchmarks. *arXiv preprint arXiv:2008.09777*, 2020.

A Benchmark Creation

We construct each benchmark from either an existing experimental dataset or an open-source simulator. Regardless of its source, every benchmark is represented using a common specification that records the optimization direction, the names, types, and domains of the decision variables, a description of the objective, any auxiliary outputs returned by an evaluation, and semantic context describing the application domain. This standardized representation allows dataset-backed and simulator-backed problems to be accessed through the same evaluation interface. We use Anthropic Claude Cowork to assist with implementing simulator and dataset interfaces and with drafting the semantic context associated with each benchmark, before manually completing implementations and updating context (Anthropic, 2026).

Common Benchmark Specification Each decision variable is designated as continuous, integer, categorical, or Boolean and is associated with an explicit domain. Continuous and integer variables are assigned lower and upper bounds, categorical variables are assigned a finite set of admissible values, and Boolean variables take values in $\{0, 1\}$. The benchmark specification also indicates whether the objective is minimized or maximized. Auxiliary outputs are retained when available, even when they are not directly optimized, because they provide additional information about the physical, experimental, or computational process represented by the benchmark. We then proceed with surrogate fitting through sampling of the benchmarks.

Candidate	Family	Configuration
`hgb_deep`	Gradient-boosted trees	900 boosting iterations, learning rate 0.03, 191 leaves, L_2 regularization 10^{-4}, no early stopping
`mlp_deep`	MLP, ReLU	widths $(640, 320, 160)$, up to 800 iterations
`mlp_xl`	MLP, ReLU	widths $(1280, 640, 320)$, up to 1000 iterations
`mlp_smooth`	MLP, tanh	widths $(1024, 512, 256)$, up to 900 iterations

Table 3: Candidate surrogate models fitted for every benchmark. The published objective surrogate is a convex blend of all four, with weights fitted on the validation split.

Simulator-backed Benchmarks For simulator-backed tasks, we generate configurations using Sobol sampling. We evaluate the original simulator independently at every generated configuration. For each evaluation, we record the submitted configuration, the objective value, all available auxiliary outputs, and whether the simulator completed successfully. Exceptions, invalid simulator states, missing outputs, and non-finite objective values are recorded as failed evaluations. Most simulator-backed benchmarks draw 65,536 Sobol training points, with per-problem overrides between 128 and 262,144 for small or high-dimensional tasks, together with 4,096 Sobol points each for validation and test. Optimizer-visited rows are added on top of these: one primary search of 32,768 true evaluations and three further independently seeded searches of 32,768 evaluations each join the training set, while separate searches of 16,384 evaluations each form the searched portion of the validation and test splits. Every search excludes all previously materialized configurations, so the three splits remain disjoint in both regimes.

The validation configurations are obtained from a nonoverlapping continuation of the sampling used for training. Consequently, the validation set is generated by the same sampling procedure but contains no configurations used to fit the surrogate. This separation is maintained throughout model selection and error reporting.

Dataset-backed Benchmarks For dataset-backed tasks, we import the configurations and recorded outcomes from the source dataset and convert them to the same schema used for simulator-backed tasks. Parameter types and domains are inferred from the dataset documentation and available metadata, and the recorded outcome is converted to the benchmark's stated optimization direction. When the source data contain auxiliary measurements, these are retained alongside the objective. For dataset-backed tasks, a seeded random shuffle holds out a validation set of at most 128 rows and at most 20% of the data. It is used to select the surrogate configuration, and the final surrogate is then refit on all rows. We find that the surrogates fit very close to the ground-truth data.

Objective-surrogate Fitting For surrogate fitting, categorical variables are represented using one-hot encoding, Boolean variables are represented as binary indicators, and numerical variables are retained on their original scales. A small number of derived columns are appended: sine and cosine components for parameters declared in radians, a base-ten logarithm for strictly positive parameters whose bounds span at least two decades, and, for categorical parameters whose levels are numeric, the level's rank and value. The same fitted representation and category ordering are used for the training, validation, and test sets. Evaluations whose objective is missing, non-finite, or associated with a simulator failure are excluded when fitting the objective surrogate. These configurations are retained when learning the feasibility model, since failures may occupy structured regions of the parameter space and therefore constitute useful information about the underlying task.

We fit four candidate models per benchmark, summarized in Table 3: one deep histogram-based gradient-boosted tree ensemble and three multilayer perceptrons that differ in width, depth, and activation. One network uses a smooth activation so that the blend is not composed exclusively of piecewise-linear members. The multilayer perceptrons standardize both features and targets and use Adam with $\alpha = 3 \times 10^{-4}$, an initial learning rate of 10^{-3}, internal early stopping on a 10% split once at least 100 rows are available, and a batch size of 1024 beyond 8192 rows. The gradient-boosted ensemble consumes raw feature values and collapses each one-hot block back into a single native categorical split.

The published objective surrogate is the convex blend of all four candidates rather than the single best one. Blend weights are obtained by sequential least-squares programming, minimizing the

mean squared error of the blended prediction on the validation split subject to non-negativity and a unit sum, and are refreshed after every refinement round. Individual candidates are still scored, both as reported diagnostics and to choose the target transform, using

$$s = \tfrac{1}{2}\,\mathrm{NRMSE} + \tfrac{1}{2}\,\mathrm{NRMSE}_{\text{top-5\%}},$$

where both terms are normalized by the P_1–P_{99} span of the feasible training targets and the second is computed over the true best 5% of held-out designs. A per-benchmark probe fits one gradient-boosted candidate under the identity and $\log(1 + x)$ target transforms and adopts the logarithmic transform only when it improves this score by more than 2%.

Surrogate refinement runs for up to ten rounds of 8192 true evaluations, with a minimum of five rounds. Each round searches the current surrogate, evaluates its proposals on the true evaluator, and adopts those that reveal failures, genuine improvements, large residuals, or uncovered regions. Refinement stops once the largest error among the surrogate's own optimizer proposals falls below 1% of the training P_1–P_{99} span; the batch that triggers the stop is retained as an acceptance holdout rather than added to training.

Feasibility Modeling For benchmarks containing failed or invalid evaluations, we separately fit a soft-voting ensemble of the same four candidate models, trained as classifiers over all rows, to predict whether an evaluation will fail. Let $\hat{p}_{\text{fail}}(\mathbf{x})$ denote the predicted probability that configuration $\mathbf{x}$ produces an invalid evaluation. A proposed configuration is rejected when

$$\hat{p}_{\text{fail}}(\mathbf{x}) > 0.5$$

This boundary is fixed rather than calibrated to a target sensitivity, since a threshold fitted to the validation failures is an extreme order statistic and moves with a single unlucky point. For configurations that pass this feasibility check, the selected objective is returned. A rejected configuration is reported as infeasible through the constraint interface and returned as a failure.

Because global error can obscure inaccuracies in the region most relevant to optimization, we also compute RMSE within the best 5% and best 1% of validation configurations according to their true objective values. For minimization tasks, these subsets contain configurations below the corresponding objective quantiles; for maximization tasks, they contain configurations above the corresponding quantiles. These metrics measure how accurately the surrogate represents both the full sampled domain and the near-optimal regions that an optimizer is expected to explore.

A.1 Domains Covered

To provide a clearer overview of the benchmark suite, we organize the 42 problems into six broad application domains: energy and infrastructure, thermal and energy-conversion systems, materials and life sciences, mechanics and control, optics and acoustics, and operations and economics. This organization highlights the breadth of the benchmark while making it easier to compare the dimensionality, parameter composition, and optimization objectives represented within each area. Because several tasks span multiple disciplines, each problem is assigned to the domain that best reflects its primary application. Tables 4–9 report the resulting domain-level groupings and summarize the search-space structure and objective of each benchmark problem. The benchmark formulations and optimization objectives are original to this suite; the citations accompanying each problem identify its underlying dataset, simulation framework, or canonical scientific model. The goal is to cover a wide variety of domains, parameter types, and problem scales.

B Example Prompt

Figure 9 shows a representative excerpt from the semantic black-box optimization interaction. The context conditions are nested: *base-context* provides the problem, search-space, and objective information; *domain-context* additionally provides domain knowledge and prior configurations; and *full-context* further provides auxiliary evaluation information. For compactness, initial configurations and evaluations 2–4 are omitted.

Name	Dim.	Types	Description
Building Energy EUI (Jayathissa et al., 2017)	28	8R, 5Z, 7C, 8B	Co-design a commercial-office envelope, glazing, HVAC, ventilation, shading, and controls to minimize annual lifecycle cost per unit floor area.
Distribution DG Placement (Thurner et al., 2018)	9	6R, 3Z	Place and size three distributed generators on the IEEE 33-bus feeder to minimize annual loss-energy and capacity cost while respecting voltage limits.
EV Charging Hub Profit (SimPy Developers, 2023)	11	5R, 2Z, 2C, 2B	Co-design chargers, prices, solar, storage, and operating policies to maximize simulated daily profit for a public EV charging hub.
Geothermal Borehole Field Capacity (Cimmino et al., 2018)	9	3R, 2Z, 3C, 1B	Design a borehole field to minimize annualized installation and pumping cost while sustainably serving a fixed ten-year thermal load.
Rooftop PV Energy Yield (Holmgren et al., 2018)	7	2R, 2Z, 3C	Design a residential rooftop PV system to maximize annual net value after self-consumption, export revenue, and annualized system costs.
Windfarm Layout AEP (National Laboratory of the Rockies, 2026)	24	24R	Offset twelve turbines from a fixed scaffold to maximize windrose-weighted farm power while satisfying a minimum-spacing constraint.
Windfarm Wake Steering (National Laboratory of the Rockies, 2026)	9	9R	Choose yaw angles for a 3×3 turbine grid to maximize total power by trading local yaw loss against downstream wake deflection.
Anaerobic Digester Biogas Value (Batstone et al., 2002)	13	8R, 4C, 1B	Design and operate a farm and food-waste co-digestion plant simulated with the IWA ADM1 model to maximize net annual value subject to biological stability limits.

Table 4: Energy and infrastructure benchmark problems. The tasks cover building systems, electrical distribution networks, EV charging infrastructure, geothermal and photovoltaic resource design, waste-to-energy conversion, and wind-farm layout and operation. The table reports the dimensionality, parameter composition, and optimization objective associated with each problem. The Types column lists the number of variables of each kind: R (real-valued/continuous), Z (integer), C (categorical), and B (Boolean).

Name	Dim.	Types	Description
PEM Electrolyzer Efficiency (Carmo et al., 2013)	10	9R, 1C	Design a fixed-area PEM electrolyzer to minimize levelized hydrogen cost while balancing efficiency, production, capital, and delivery pressure.
TEG Module Power (Snyder & Toberer, 2008)	7	5R, 1Z, 1C	Design a thermoelectric module to maximize annual net value by balancing electricity output against module and heat-exchanger capital costs.
ORC Thermal Efficiency (Bell et al., 2014)	5	4R, 1C	Design an organic Rankine cycle to maximize equivalent annual net value from a fixed-temperature waste-heat source.
Vapor Compression Chiller COP (Bell et al., 2014)	6	3R, 1Z, 1C, 1B	Design a 100 kW vapor-compression chiller to minimize equivalent annual electricity, equipment, and refrigerant climate cost.
Shell Tube HX Cost (Caputo et al., 2008)	6	4R, 2C	Size and configure a shell-and-tube heat exchanger to minimize capital plus discounted pumping cost for a fixed heat duty.
Heatsink Thermal Resistance (Teertstra et al., 2000)	8	4R, 1Z, 1C, 2B	Design a plate-fin heat sink to minimize annualized material, manufacturing, and fan-energy cost while maintaining its temperature limit.
SWRO Desalination Energy (Baker, 2023)	6	2R, 2Z, 2C	Design a single-pass seawater reverse-osmosis train to minimize levelized water cost under water-quality, production, and recovery constraints.

Table 5: Thermal and energy-conversion systems benchmark problems. The tasks span electrochemical hydrogen production, thermoelectric conversion, power and refrigeration cycles, heat-exchanger and heat-sink sizing, and membrane desalination. Each problem trades thermodynamic performance against capital and operating cost under equipment and service constraints. Parameter-type abbreviations follow Table 4.

Name	Dim.	Types	Description
Snar Flow Space Time Yield (Hone et al., 2017)	4	4R	Choose residence time, reactant equivalents, concentration, and temperature to maximize space-time yield of a continuous-flow S_NAr reaction.
Lean Premixed Low NOx (Goodwin et al., 2018)	6	6R	Choose hydrogen/methane flame conditions to minimize combined NO and CO emissions while satisfying flame-speed, temperature, and CO constraints.
CuAu Nanoalloy Ordering (Hjorth Larsen et al., 2017)	13	13B	Assign six Au atoms to a fixed 13-site Cu–Au icosahedron to minimize relaxed mixing energy at fixed composition.
Concrete Compressive Strength (Yeh, 1998)	8	7R, 1Z	Minimize the embodied carbon of a concrete mix while achieving at least 35 MPa predicted compressive strength.
Drug Likeness QED Design (Bickerton et al., 2012; Landrum et al., 2013)	7	7C	Assemble a two-ring scaffold by choosing rings, a linker, and substituents to maximize RDKit's quantitative estimate of drug-likeness.
Ecoli Succinate Strain Design (Ebrahim et al., 2013)	18	3R, 15B	Select central-metabolism knockouts and cultivation conditions to maximize growth-coupled succinate secretion under flux-balance analysis.
Spring Wheat Cultivar Management (Shibu et al., 2010)	28	9R, 8Z, 3C, 8B	Choose wheat cultivar and management practices to maximize net crop return after nitrogen and irrigation costs at a fixed site and weather year.
Palmer Penguin Body Mass (Gorman et al., 2014)	6	6B	Select field measurements that accurately predict penguin body mass while penalizing capture time, handling burden, and laboratory work.

Table 6: Materials and life sciences benchmark problems. The tasks include flow-reaction optimization, combustion, nanoalloy ordering, concrete formulation, molecular design, metabolic engineering, crop management, and biological measurement selection. Together they represent continuous, categorical, and combinatorial design spaces governed by chemical, physical, and biological trade-offs. Parameter-type abbreviations follow Table 4.

Name	Dim.	Types	Description
Aircraft Mission Fuel Burn (Sun et al., 2020)	7	4R, 1Z, 1C, 1B	Choose an airliner and flight profile that minimize direct operating cost for a fixed payload and range, using OpenAP fuel and block-time predictions.
Airfoil Lift To Drag (Sharpe & Hansman, 2025)	18	18R	Optimize a CST/Kulfan airfoil shape and angle of attack to maximize lift-to-drag ratio at a fixed cruise Reynolds number.
Earth Mars Transfer Deltav (Astropy Collaboration et al., 2013)	5	4R, 1B	Choose launch date, flight time, orbit altitudes, and transfer direction to minimize patched-conic Earth-to-Mars mission delta-v.
LEO Constellation Revisit (Larson & Wertz, 1992)	13	5R, 6Z, 1C, 1B	Size, phase, and place a low-Earth-orbit satellite constellation to minimize annualized system cost while meeting population-weighted coverage and revisit-gap requirements.
Auto MPG Fuel Economy (Quinlan, 1993)	4	3R, 1Z	Design a historically plausible 1982 passenger-car engine and curb mass to minimize five-year lifecycle cost subject to performance constraints.
Truss 10bar Weight (Schmit Jr & Farshi, 1974)	11	10R, 1C	Choose material and member areas for the classic 10-bar cantilever truss to minimize weight under stress and deflection constraints.
Pendulum Swingup Control (Towers et al., 2025)	5	5R	Tune five gains of a hybrid energy-shaping and PD controller to maximize mean closed-loop return for an under-actuated pendulum.
PID Controller Tuning (Fuller et al., 2021)	5	5R	Tune a two-degree-of-freedom PID controller to minimize set-point tracking error subject to a maximum-sensitivity robustness constraint.

Table 7: Mechanics and control benchmark problems. The tasks span aircraft mission design, aerodynamics, interplanetary transfers, satellite constellation sizing, vehicle powertrain design, structural optimization, and feedback controller tuning. These problems contain continuous and mixed-variable search spaces with physical, performance, and robustness constraints. Parameter-type abbreviations follow Table 4.

Name	Dim.	Types	Description
Yagi Uda Antenna Gain (Yagi & Uda, 1926)	10	10R	Choose element lengths, spacings, and wire radius for a five-element Yagi–Uda antenna to maximize realized forward gain under matching and front-to-back constraints.
RF Matching Network (Arsenovic et al., 2021)	10	10R	Choose impedances and electrical lengths for a five-section transmission-line network to maximize worst-case return loss over a wide frequency band.
Triplet Lens Spot Size (Kingslake & Johnson, 2009)	16	12R, 3C, 1B	Choose curvatures, thicknesses, air spaces, stop position, and glass types for a Cooke triplet objective to minimize polychromatic RMS spot radius at fixed focal length and f-number.
Broadband AR Coating (Byrnes, 2016)	10	5R, 5C	Choose the materials and thicknesses of a five-layer dielectric stack to minimize solar-spectrum-weighted reflection from cover glass.
Bass Reflex Box Bass Extension (Small, 1973)	11	8R, 1Z, 1C, 1B	Co-design a vented loudspeaker driver and enclosure to minimize cost while meeting bass-extension, sensitivity, excursion, vent-noise, and packaging requirements.

Table 8: Optics and acoustics benchmark problems. The tasks cover antenna design, radio-frequency matching networks, lens design, thin-film optical coatings, and loudspeaker enclosure design. These problems couple wave-propagation physics to broadband performance objectives evaluated over frequency or wavelength. Parameter-type abbreviations follow Table 4.

Name	Dim.	Types	Description
Supply Chain Inventory Control (Clark & Scarf, 1960)	26	7R, 7Z, 7C, 5B	Co-design inventory, replenishment, transport, sourcing, forecasting, and emergency policies to minimize two-echelon operating cost subject to a fill-rate floor.
Water Network Pump Scheduling (Klise et al., 2018)	12	12B	Schedule two pumps over six daily time blocks to minimize time-of-use electricity cost while maintaining hydraulic service and tank storage.
ED Patient Flow LOS (Asplin et al., 2003)	29	5R, 12Z, 4C, 8B	Configure a budget-constrained emergency department to minimize patient-flow harm from length of stay, abandonment, diversion, and boarding.
Child Carseat Sales (Taylor et al., 2026)	3	1R, 1Z, 1C	Choose price, advertising, and shelf placement for child car seats to maximize risk-adjusted annual profit across heterogeneous stores.
LDI Bond Portfolio (Redington, 1952)	10	4R, 1Z, 4C, 1B	Construct a self-funding liability-driven bond portfolio that balances net carry against surplus losses under rate and credit-spread shocks.
Windsor Housing Price (Anglin & Gencay, 1996)	10	1R, 4Z, 5B	Design a historically plausible Windsor infill house to maximize predicted sale price net of land, structural, and amenity development costs.

Table 9: Operations and economics benchmark problems. The tasks include supply-chain policy design, water-network operation, emergency-department planning, pricing and marketing decisions, financial portfolio construction, and housing design. These problems emphasize economic trade-offs, data-driven objectives, and discrete or mixed-variable decision spaces. Parameter-type abbreviations follow Table 4.

```
System: You are a black-box optimization assistant. Before any evaluations, return
    initial_configuration_1,...,initial_configuration_5. Before any observed failure, return
    next_configuration, mean, std; after a failure, additionally return failure_probability.
User: Possibly relevant context: Empirical aircraft models simulate climb, cruise, optional
    step climbs, and descent. Direct operating cost combines fuel and aircraft time cost.
    Narrowbodies may avoid excess weight and hourly cost on this medium-range mission; cruise
     speed trades fuel against block time, while altitude and step climbs may improve
    efficiency but can become infeasible. Candidates fail if fuel capacity or MTOW is
    exceeded, the requested profile cannot be sustained, fuel iteration does not converge, or
     realized distance misses tolerance.
Potentially relevant prior configurations:
- Right-sized narrowbody prior: aircraft=a320
- Oversized widebody example: aircraft=a332, cruise_mach=0.8, cruise_alt_ft=35000
Problem description: Assign an aircraft to a 16,000 kg, 4,500 km mission and choose its flight
     profile to minimize direct operating cost.
Tunable search space:
- aircraft: {a320,a321,b738,b38m,a332,a333,a359,b772,b789,b77w}
- cruise_mach: [0.72,0.86]
- cruise_alt_ft: [28000,41000] ft
- climb_mach: [0.58,0.8]
- step_climb: {true,false}
- n_steps: [0,4]
- step_size_ft: [1000,4000] ft
Objective: minimize, name: direct_operating_cost_usd, Predicted range: [15000,100000] USD
Additional evaluation information: block_time_min, takeoff_mass_kg, trip_fuel_kg,
    aircraft_time_cost_usd
Budget: Total evaluations available (including 5 initial evaluations): 50
Provide 5 initial configurations that explore promising regions.
Assistant: {"initial_configuration_1":{"aircraft":"a320","cruise_mach":0.78,"cruise_alt_ft
    ":36000,"climb_mach":0.72,"step_climb":true,"n_steps":2,"step_size_ft":2000},
# Initial configurations 2-4 omitted
"initial_configuration_5":{"aircraft":"b38m","cruise_mach":0.76,"cruise_alt_ft":39000,"
    climb_mach":0.70,"step_climb":false,"n_steps":0,"step_size_ft":2000}}
User: Initial evaluations:
Evaluation 1: Configuration: {"aircraft":"a320","cruise_mach":0.78,"cruise_alt_ft":36000,"
    climb_mach":0.72,"step_climb":true,"n_steps":2,"step_size_ft":2000}
Objective: 35610 USD
block_time_min: 331.1; takeoff_mass_kg: 74460; trip_fuel_kg: 15100; aircraft_time_cost_usd:
    23180
# Evaluations 2-4 omitted
Evaluation 5: Configuration: {"aircraft":"b38m","cruise_mach":0.76,"cruise_alt_ft":39000,"
    climb_mach":0.70,"step_climb":false,"n_steps":0,"step_size_ft":2000}
Objective: 35250 USD
block_time_min: 337.7; takeoff_mass_kg: 73630; trip_fuel_kg: 12030; aircraft_time_cost_usd:
    25310
Output the next configuration.
Assistant: {"next_configuration":{"aircraft":"b38m","cruise_mach":0.78,"cruise_alt_ft":38000,"
    climb_mach":0.72,"step_climb":true,"n_steps":2,"step_size_ft":2000},"mean":34850,"std
    ":900}
User: Evaluation 6: Objective: 34580 USD
block_time_min: 331; takeoff_mass_kg: 73440; trip_fuel_kg: 11840; aircraft_time_cost_usd:
    24830
Output the next configuration.
```

■ Base-Context ■ Domain-Context ■ Full-Context

Figure 9: Representative prompt excerpt for the nested semantic-context conditions. Green denotes *base-context*; *domain-context* additionally includes domain knowledge and prior configurations, shown in purple; and *full-context* further includes auxiliary evaluation information, shown in blue. The model first proposes five initial configurations and, after receiving their evaluations, proceeds sequentially one configuration per turn. Initial configurations and evaluations 2-4, as well as punctuation details are omitted for space.

C BASELINES

We compare against random search together with five further methods. Three of them span distinct inductive biases and have no access to the problem's semantic description: a Gaussian process surrogate, a density-ratio model, and a mixed-variable evolution strategy. The remaining two are language models, one frontier and one open-weight, that reason jointly over the numerical observations and the semantic context.

Every method is driven through a common Optuna ask-and-tell harness, on the same benchmark suite, under the same evaluation budget, and on the same seeded problem instances; a method's seed fixes both its initial design and its internal randomness. Infeasible evaluations are reported to all methods, as a finite penalty at the worst end of the objective's predicted range for the HEBO baseline. Each numerical baseline spends an initial quota of its budget on a random or quasi-random design before its model takes over, whereas the language-model arms receive no random startup trials and instead propose their own initial batch from the semantic context. When an ablation withholds the semantic context from a language-model arm, we retain the model-proposed initialization rather than substituting a random design, so that the comparison isolates the effect of context rather than that of the initial design.

- **Random search:** Uniform i.i.d. sampling over the search space using Optuna's RandomSampler (Akiba et al., 2019), seeded per instance, with continuous and integer parameters drawn uniformly over their bounds and categorical parameters uniformly over their levels.
- **HEBO GP:** A Bayesian optimization method designed for mixed search spaces (Cowen-Rivers et al., 2022), combining input and output warping with a multi-objective ensemble of acquisition functions optimized by an evolutionary inner loop. We adopt it as our primary BO baseline for its handling of mixed search spaces and its strong performance on classical benchmarks, where it won the NeurIPS 2020 black-box optimization challenge (Turner et al., 2021). We use the OptunaHub registry sampler (Ozaki et al., 2025), which calls the authors' reference implementation as its backend, configured with the exact Gaussian process surrogate so that the arm is a GP baseline in the conventional sense. The first five evaluations are drawn by HEBO's random design phase. While HEBO could use a separate feasibility surrogate model, we find that using a penalty term at the edge of the feasible predicted objective range for infeasible values empirically outperforms a separate feasibility model. Therefore, results are shown with this penalty implementation.
- **TPE:** A Tree-structured Parzen Estimator sampler (Bergstra et al., 2011), using the Optuna implementation (Akiba et al., 2019) at its default settings with a per-instance seed. It models the search space through the ratio of two densities fit to the good and poor performing observations rather than through a global surrogate of the objective, a different inductive bias compared to the Gaussian process baseline, and it treats categorical parameters natively rather than through an embedding. The first ten evaluations are drawn uniformly at random, Optuna's default startup quota, after which every proposal maximizes the density ratio.
- **CatCMA with Margin:** A mixed-variable evolution strategy extending CMA-ES with a margin correction that prevents integer parameters from collapsing onto a single lattice value, and with a categorical distribution updated jointly with the continuous Gaussian (Hamano et al., 2025). Being surrogate-free, it separates an advantage over model-based search from an advantage over sample-efficient search in general. We use the implementation in the `cmaes` library released by the method's authors (Nomura et al., 2024), accessed through its OptunaHub registry wrapper. Because an evolution strategy ranks a population rather than modeling objective values, a generation whose members are tied at the infeasibility penalty carries no ranking signal. We therefore additionally supply this arm with the feasibility indicator, drawing feasibility-seeking points and seeding the initial mean at the first feasible observation.
- **GPT-5.6 Luna:** We use GPT-5.6 Luna in medium reasoning mode (OpenAI, 2026), accessed through the OpenAI API. The benchmark requires many sequential model calls across evaluation steps, context conditions, and seeds, which makes inference cost a central constraint on the study's scale. Its smaller size also mitigates, though it does not eliminate, the concern that performance is driven by memorization of specific problems, while it re-

mains a frontier model with strong reasoning capabilities. The arm receives no random startup trials: the model is shown the search space and the semantic context and proposes the opening batch itself, so its initial design is model-written rather than random, and each subsequent proposal is conditioned on the accumulated record of evaluated points and their outcomes.

- **Gemma 4 31B:** An open-weight counterpart (Team et al., 2026) in its non-thinking mode, evaluated under the same prompts, context conditions, initialization scheme, and budget as the frontier arm. We use default sampling parameters specified by the Gemma authors. We use the no-thinking mode because we found the thinking mode failed to reliably produce correct JSON outputs after multiple turns.

For LLMs, the optimization process is represented as a multi-turn conversation. The initial prompt specifies the available evaluation budget, optimization direction, and search space, together with the information included in the selected context condition. At each step, the model proposes a configuration. These configurations are validated and evaluated individually. Thereafter, each model response proposes one additional configuration. Its objective value or failure outcome is appended to the conversation, and the model uses the updated history to select the next configuration. This loop continues until the fixed evaluation budget is exhausted.

After initialization, each model response is a structured object containing the proposed configuration and predictions of its objective mean, uncertainty, and probability of failure if a failure has been observed. Benchmark performance is determined only by the objective values of the evaluated configurations. Proposed configurations are checked against the declared parameter types and domains before evaluation. Numerical values are coerced to their required representations where possible. Malformed, incomplete, or out-of-domain proposals only occur a few times in the completed experiments, but are retried until a valid proposal is reached.

We use medium reasoning mode and default OpenAI parameters for inference. Running the full benchmark suite required around 100,000 API requests to GPT-5.6 Luna, for a total of around 1 billion input tokens and 40 million output tokens. Due to the multi-turn nature of these problems, most tokens are able to be used as cached input, reducing overall costs. We ran Gemma 4 31B on an NVIDIA H200 GPU using vLLM (Kwon et al., 2023), taking around 24 hours to run the full-context suite on this model.

D Further Results

Per Benchmark Results The aggregate curves average over benchmarks and can hide whether a method wins broadly or on a few problems only. Figure 10 therefore reports the final score on every benchmark separately, ordered by the score of GPT-5.6 Luna with *full-context*. We show a range of problem difficulties, ranging from problems most optimizers solve, to ones where most optimizers are largely unsuccessful.

Ranks Mean normalized score is sensitive to benchmarks with a wide score range, so we also report scale-free results. At each evaluation budget we rank the six samplers within every run and average the ranks over benchmarks, shown in Figure 11. The ranking reproduces the picture from the aggregate curves without depending on the size of the gaps. The two *full-context* variants hold the top two ranks at every budget. HEBO GP is third throughout, but at 6 evaluations it is barely separated from random search and TPE (4.03 against 4.11 and 4.15) and by 50 evaluations it has pulled clear of them and closed on the LLM variants (2.87 against 2.52 and 2.72) without overtaking them.

Per-Domain Results To check that the advantage is not carried by one application area, we group the benchmarks into the six domains of Table 1 and plot the trajectories within each, in Figure 12. The early advantage of semantic context appears in all six domains rather than in a subset.

We further compare GPT-5.6 Luna and HEBO GP benchmark by benchmark rather than in aggregate. We compare the seven Luna context variants against HEBO GP on all 42 benchmarks, at 10 and 50 evaluations. Four of them vary how much context the model is given, from *full-context* down through *domain-context* and *base-context* to *no-context*. The other three keep the full context but

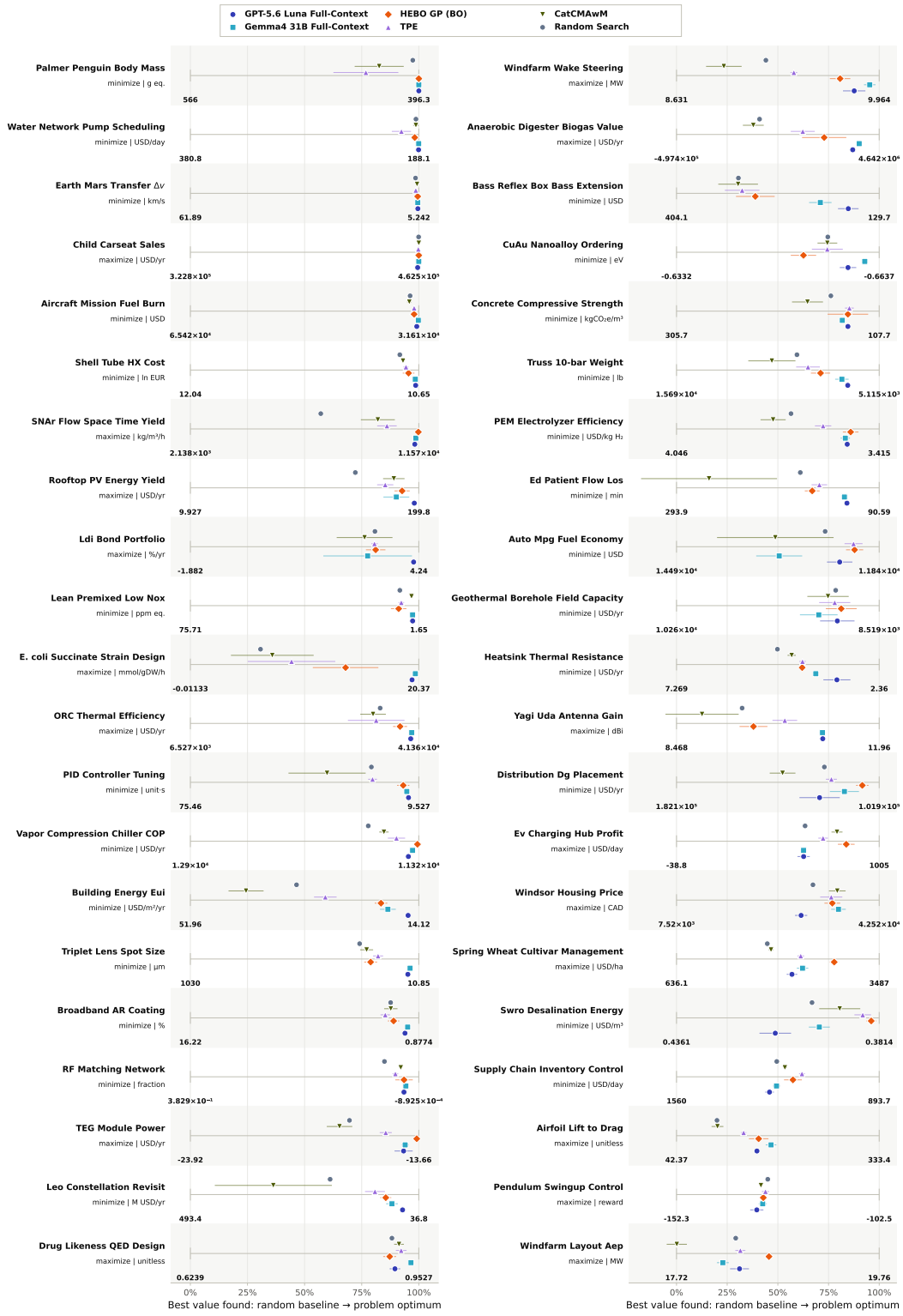


Figure 10: Scores after 50 evaluations on each of the 42 benchmarks, best at the top of the left column and continuing into the right. We see a range of problem difficulties and that different methods perform better on different tasks. Markers are the mean over five seeds and bars are one standard error.

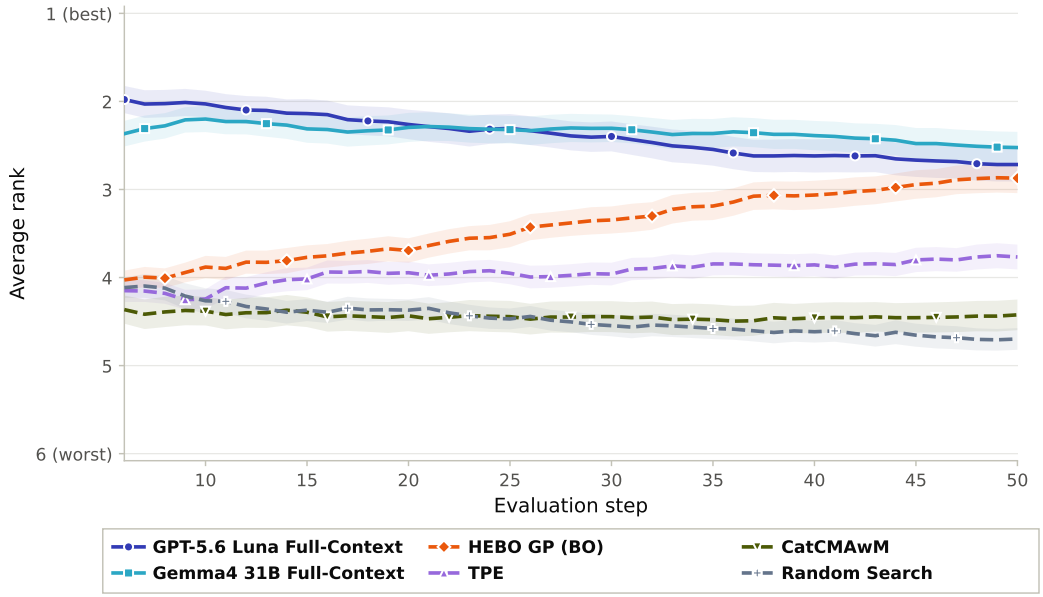


Figure 11: Average rank as a function of evaluation budget, rank 1 being best. Within each run the six samplers are ranked at every budget and the ranks are then averaged over the 42 benchmarks; bands are one standard error over benchmarks.

	10 evaluations		50 evaluations	
Context	Mean Δ [95% CI]	$\#(\Delta{>}0)$	Mean Δ [95% CI]	$\#(\Delta{>}0)$
Full	$+0.277$ $[+0.186, +0.380]$	$38/42$	$+0.026$ $[-0.022, +0.074]$	$24/42$
Domain	$+0.258$ $[+0.169, +0.356]$	$34/42$	$+0.016$ $[-0.031, +0.063]$	$22/42$
Base	$+0.198$ $[+0.115, +0.296]$	$35/42$	-0.020 $[-0.062, +0.027]$	$15/42$
None	$+0.019$ $[-0.076, +0.119]$	$21/42$	-0.170 $[-0.231, -0.109]$	$5/42$
Counterfactual	$+0.257$ $[+0.163, +0.363]$	$36/42$	$+0.012$ $[-0.038, +0.063]$	$23/42$
Mismatched	$+0.032$ $[-0.100, +0.162]$	$26/42$	-0.151 $[-0.250, -0.066]$	$12/42$
Adversarial	-0.265 $[-0.416, -0.114]$	$12/42$	-0.352 $[-0.474, -0.240]$	$4/42$

Table 10: Per-benchmark differences relative to HEBO GP. The upper block varies how much context the model receives; the lower block keeps the full context and rewrites it. Each of the 42 benchmarks contributes one value, obtained by averaging its five matched seeds before differencing. Intervals are percentile bootstrap over those 42 values, and the final column for each budget counts the benchmarks with a positive difference.

rewrite it: the *counterfactual* rewrite removes surface identifying detail while leaving the problem intact, the *mismatched* rewrite replaces the context with the context from a new problem with the same search space format, and the *adversarial* rewrite replaces it with misleading guidance.

Every method sees the same benchmarks under the same five seeds, so the comparisons are paired. For each configuration we report the mean difference in normalized score, a percentile bootstrap 95% confidence interval over the 42 benchmark-level differences, and the number of benchmarks on which the difference is positive. Positive values favor Luna.

The aggregate in Figure 8 could be carried by a few benchmarks, so we also compare the combination sampler to each of its two components benchmark by benchmark, on the same 42 benchmarks under the same five seeds. It improves on both, and on most benchmarks rather than a handful: it beats HEBO GP on 33 of 42 and GPT-5.6 Luna on 28 of 42.

Early-budget performance At 10 evaluations every configuration that receives real context is clearly ahead of HEBO GP. The advantage grows with the amount of context supplied. The rewrites

Compared against	Mean Δ [95% CI]	$\#(\Delta{>}0)$
HEBO GP	$+0.065$ $[+0.028, +0.104]$	$33/42$
GPT-5.6 Luna *full-context*	$+0.039$ $[+0.015, +0.066]$	$28/42$

Table 11: The combination sampler compared benchmark by benchmark against each of its two components, after 50 evaluations. Positive values favor the combination. Each of the 42 benchmarks contributes one value, obtained by averaging its five matched seeds before differencing; intervals are percentile bootstrap over those 42 values, and the last column counts the benchmarks with a positive difference.

separate the content of that context from its surface form. *Counterfactual* keeps almost all of the advantage, suggesting the gain comes from the semantics of the problem rather than from recognizing parameter names or descriptions.

Adversarial falls well below HEBO GP, showing the misleading context severely impacts the optimizer. *Mismatched* is the one undetermined case: its interval spans zero, so at this budget a context belonging to a different problem is neither help nor harm.

Performance at the full evaluation budget By 50 evaluations most of that separation is gone. *Full-context* remains the strongest Luna configuration and is still ahead on 24 of 42 benchmarks, but its interval now spans zero, so we do not claim an advantage over HEBO GP at the full budget. *Domain-context*, *base-context* and *counterfactual* are likewise indistinguishable from it. The degraded conditions stay behind, and *mismatched* has turned clearly negative. The ordering of the context conditions persists at the full budget. Semantic context helps most when evaluations are scarce, and HEBO GP closes the aggregate gap as they accumulate.

Context Composition *Domain-context* adds two kinds of information to *base-context* at once: long-form domain knowledge with optimization heuristics, and partial historical example configurations. To provide further breakdown, we test two intermediate variants, *long-form-context* and *history-context*, each adding one of the two, and show all context levels in Figure 13. After 50 evaluations *base-context* reaches 0.781, *history-context* 0.794, *long-form-context* 0.816 and *domain-context* 0.817. Nearly all of the *domain-context* improvement therefore comes from the long-form domain knowledge: removing the historical configurations costs 0.001, while removing the long-form knowledge costs 0.022 and leaves the model at the level of HEBO GP (0.800). The ordering is the same at 10 evaluations (0.676, 0.696, 0.733 and 0.736). The model appears to use the described mechanism of the problem rather than the tabulated examples.

Benchmark Surrogates vs. Ground Truth Evaluator Each benchmark scores configurations with a fitted surrogate rather than the original simulator, which is what makes a 42-problem suite affordable to run across context conditions and seeds. Table 2 reports how accurate these surrogates are, but accuracy alone does not establish that the substitution preserves a *comparison* between optimizers: an optimizer concentrates its evaluations where the surrogate ranks highest, which is exactly where a small systematic error would matter most. We therefore re-ran the benchmark against the original simulators.

The two legs differ only in the source of a configuration's objective value and feasibility flag, and share the search space, semantic context, budget of 50 evaluations and five random seeds. We evaluate five arms: GPT-5.6 Luna with *full-context*, HEBO GP, TPE, CatCMA with Margin, and random search on 21 benchmarks. These span all six domains of Table 1 and 4 to 29 decision variables. Both legs are normalized with the surrogate leg's f_m and f_{ref}, so that a difference between them reflects a difference in objective value rather than in normalization.

The two legs agree closely, as shown in Figure 14. They identify the same best arm on 18 of 21 benchmarks after both 10 and 50 evaluations, with mean Spearman rank correlations of 0.92 and 0.86. The lowest correlation after 50 evaluations occurs on a benchmark where all arms finish within about 0.01 of one another, so its ordering reflects noise rather than the choice of evaluator. Mean normalized scores differ between legs by at most 0.025 for any arm after 10 evaluations and 0.014 after 50. After 10 evaluations, for example, GPT-5.6 Luna with *full-context* scores 0.746 on

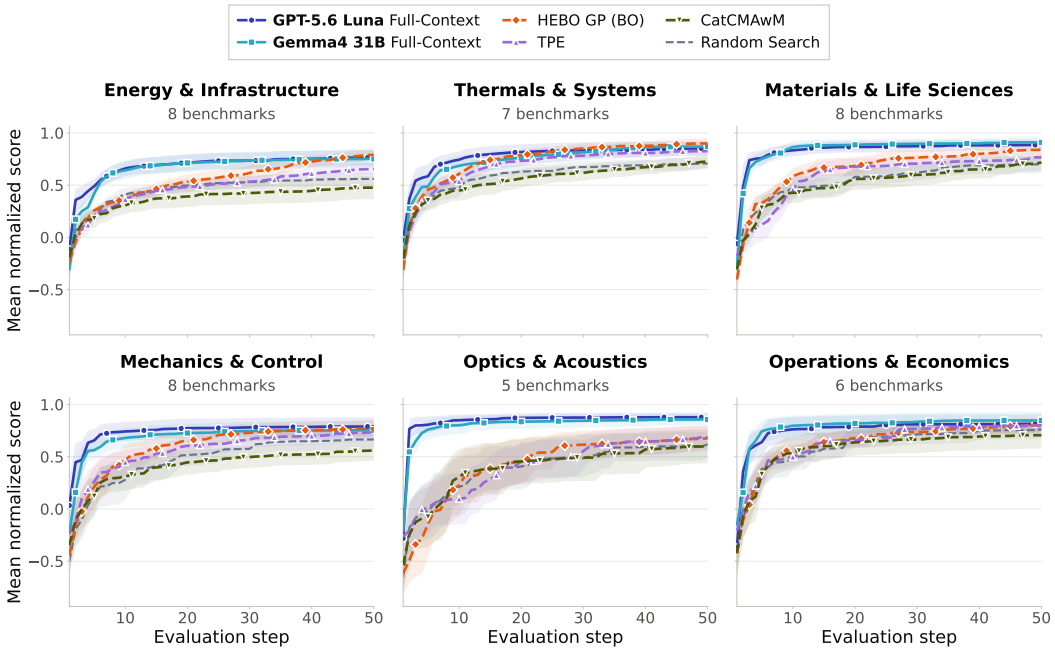


Figure 12: Mean normalized score as a function of evaluation budget within each application domain. Bands are one standard error over the benchmarks of the domain; the benchmark count per domain is given in each panel title.

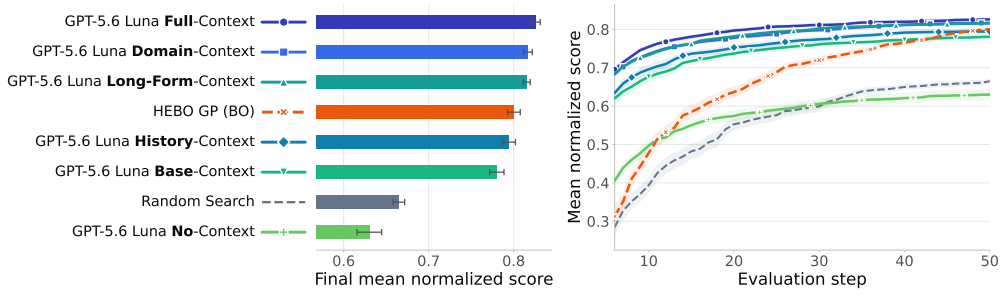
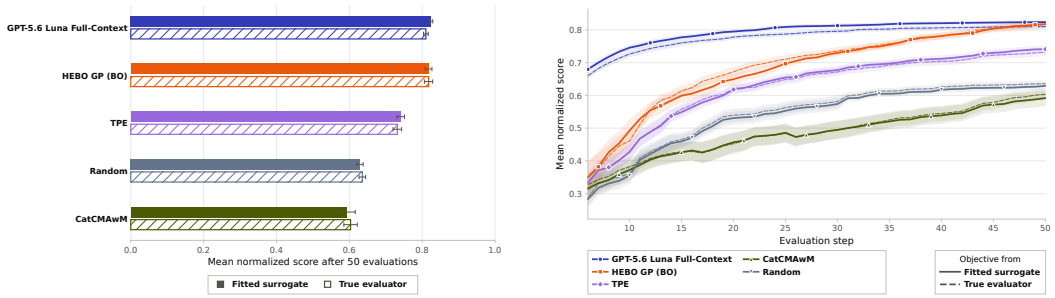


Figure 13: All context levels, including the two variants that add only one half of *domain-context*: *history-context* adds the historical example configurations to *base-context*, *long-form-context* adds the long-form domain knowledge. The long-form knowledge carries most of the *domain-context* gain and the historical configurations add little on top of it. Bars and curves show the mean and standard error over five random seeds per benchmark.



(a) Performance after 50 evaluations.

(b) Performance as a function of evaluation budget.

Figure 14: Optimizer performance is preserved when the fitted surrogate is replaced by the original simulator: the arms keep their order and their spacing, and the paired curves stay together over the whole budget. Solid bars and lines are the fitted surrogate, hatched bars and dashed lines the original simulator. Results are averaged over 21 benchmarks. Bars and curves show the mean and standard error over five random seeds per benchmark.

the surrogate and 0.726 on the simulator. The early advantage of semantic context and its later narrowing therefore appear at the same magnitude in both legs. The paired curves also remain together as the budget grows, whereas a surrogate that degraded in the regions optimizers search would cause them to diverge. Evaluation failure rates differ by at most 0.010 for any arm.

E Code and Artifact Release

An implementation of SemanticBBOBench, the optimization runners, and the analysis utilities is available at `https://anonymous.4open.science/r/semanticbbobench_iclr2027-C429/README.md`. The benchmark is organized as an OptunaHub registry (Ozaki et al., 2025) and exposes each problem through a common Optuna-compatible interface, so any Optuna sampler can be evaluated on it without modification.

Benchmark definitions The release contains the specification of all 42 benchmark problems. For each problem, these files describe the search space, optimization direction, objective, auxiliary outputs, and semantic context under every context level and context type, together with the source evaluator and its supporting data. The interface maps continuous, integer, categorical, and Boolean variables to Optuna distributions and exposes the semantic information in a standardized format.

Surrogate artifacts Each problem is evaluated through a fitted objective surrogate and, when applicable, a failure classifier and predictors for auxiliary outputs, so the benchmark runs without the original simulators. Because of their combined size, the anonymized release includes these artifacts for ten problems spanning the domains and parameter types of the suite, which run directly after cloning the repository. For every problem it includes the surrogate metadata, from which the surrogate-fit summary in Table 2 is regenerated, and the size and checksum of every artifact file, against which any copy of the remaining surrogates is verified before use. The release also includes the benchmark-construction pipeline used to sample the source problems, fit the candidate surrogate families, select models on held-out data, and fit the failure and auxiliary models.

Optimization and analysis code The release includes the samplers and context ablations used in the experiments and the experiment driver for GPT-5.6 Luna, Gemma 4, and the random search, TPE, CatCMAwM, and HEBO GP baselines under a shared evaluation protocol. Run manifests record the benchmark, method, model configuration, evaluation budget, and random seed. The analysis scripts reconstruct the aggregate trajectories, paired comparisons, robustness analyses, and tables from the recorded results, which the release includes for every method, problem, and seed as per-evaluation objective values and best-value progressions; full trial histories and LLM transcripts are omitted.

Reproducing the results The script `scripts/reproduce_paper.sh` regenerates every data-driven figure and table from the included results on a CPU. A second script, `scripts/test.sh`, runs the baselines and the context-aware LLM sampler on any installed problem through the same runner, normalization, and plotting code, with presets that reproduce the inference settings of GPT-5.6 Luna and Gemma 4.