AGENTIC HIGH-DIMENSIONAL BAYESIAN OPTIMIZATION WITH HYPOTHESIS- AND EVIDENCE-GUIDED SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

High-dimensional Bayesian optimization (HDBO) seeks sample-efficient optimization when the number of variables is large relative to the evaluation budget. Recent LLM-based and agentic BO methods incorporate task knowledge and adapt search decisions during a run, but have primarily been evaluated on low- and moderate-dimensional problems. We ask whether this paradigm can transfer to the higher-dimensional regime. Our experiments show that these methods do not remain reliable in the high-dimensional regime, where the challenge is not only where to evaluate, but also which modeling assumption and search geometry to use when the objective's useful structure is unknown. We therefore introduce HERA, a Hypothesis- and Evidence-guided Research Agent that uses task context, optimization feedback, and structural diagnostics to revise search hypotheses, select and configure HDBO strategies, and determine their execution length. PRISM, its numerical optimization engine, generates and evaluates candidates sequentially within each search block, updating numerical models after each observation. HERA remains competitive with strong numerical HDBO baselines and outperforms the evaluated LLM-based and agentic methods on four metadata-free synthetic functions. Across eight real-world tasks, HERA achieves the best mean final objective among all evaluated systems on most benchmarks. Further analyses show that structural diagnostics change strategy use, metadata effects vary across tasks, and adaptive search blocks reduce inference cost.

1 INTRODUCTION

Bayesian optimization (BO) is a standard approach to optimizing expensive black-box functions under limited evaluation budgets (Shahriari et al., 2015; Frazier, 2018). Task descriptions and prior experimental experience can inform this search, but their implications are difficult to encode in numerical optimization rules. Recent LLM-based methods augment BO with task descriptions and other informal priors (Liu et al., 2024b; Chen et al., 2026), while agentic BO places an LLM agent in control of an evolving optimization campaign and retains a Bayesian backend for uncertainty-aware search (Brunzema et al., 2026). This division combines the agent's ability to interpret heterogeneous context and revise decisions with the systematic numerical search that makes BO sample efficient.

These systems have been evaluated mainly on low- to moderate-dimensional problems, leaving their applicability to high-dimensional BO (HDBO) underexplored (Yang et al., 2024; Liu et al., 2024b; Schwanke et al., 2026; Chen et al., 2026; Brunzema et al., 2026). In HDBO, sparse observations make surrogate modeling and candidate search difficult, and numerical methods introduce different biases—including coordinate sparsity, low-dimensional embeddings, additivity, and locality—to make search tractable (Hoang et al., 2025; Letham et al., 2020; Kandasamy et al., 2015; Eriksson et al., 2019; Duque et al., 2024). Recent results also show that carefully designed full-dimensional BO can be competitive (Hvarfner et al., 2024; Papenmeier et al., 2025b), so no single strategy is an obvious default. We therefore ask whether agentic BO can transfer to high-dimensional problems, where limited evaluations must guide both objective optimization and the choice of search strategy.

This uncertainty creates two obstacles to a direct transfer of Agentic BO. First, the numerical backend itself is not uniformly reliable: a generic full-dimensional backend can become fragile, while

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.

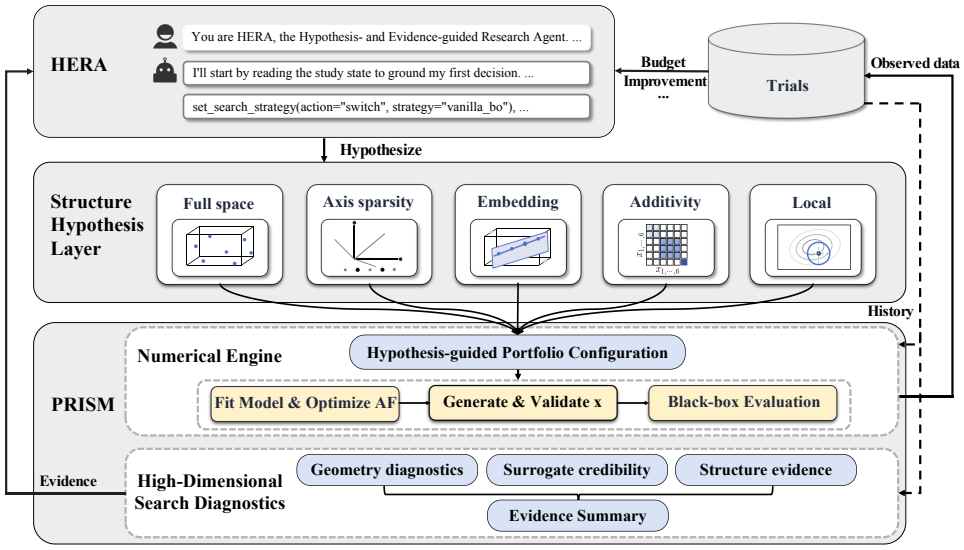

Figure 1: **Agentic HDBO as an iterative research process.** HERA interprets task information, optimization feedback, and structural diagnostics to maintain a revisable working hypothesis. PRISM executes the selected strategy for several evaluations and updates a shared observation history. Outcomes return as evidence for the next round of hypothesis revision.

a fixed specialized backend commits in advance to a bias that may be mismatched to the objective (Papenmeier et al., 2025b; Yu et al., 2026). Second, delegating candidate generation to an LLM does not remove the difficulty; complete high-dimensional proposals can provide poor coverage or become effectively uninformative as coordinate-wise decisions accumulate (Schwanke et al., 2026). These obstacles establish a transfer gap rather than a universal failure of Agentic BO, and raises two related questions: *what should an LLM control in HDBO, and how should that control be exercised while retaining reliable numerical search?* Our answer is to place the agent above candidate generation: it manages revisable search hypotheses and bounded numerical experiments, while specialized methods decide which concrete points to evaluate.

We instantiate this idea in HERA, a Hypothesis- and Evidence-guided Research Agent, and PRISM, a Portfolio of Reconfigurable Inference and Search Methods (Figure 1). HERA interprets task context, optimization progress, and provisional structural diagnostics to maintain a working hypothesis and choose whether to continue, reconfigure, restart, or change a numerical strategy. PRISM validates that intervention, generates and evaluates candidates through the selected HDBO method, and retains all observations in a shared history. The resulting *observe–hypothesize–intervene–evaluate–reflect* loop allows one agent decision to govern several numerical evaluations before reassessment. A working hypothesis is an actionable and revisable account of what may currently make the objective searchable, not a claim that the true structure has been recovered.

Experiments characterize our approach across four metadata-free synthetic functions and eight real-world tasks spanning trajectory planning, continuous control, engineering design, chip placement, and molecular optimization. Across synthetic and real-world benchmarks, HERA matches strong classical high-dimensional BO methods, outperforms the evaluated LLM-based and agentic baselines, and achieves the best mean final objective among all evaluated systems on the majority of real-world tasks. We further examine whether simpler strategy selection is sufficient, and ablate the roles of structural diagnostics, task metadata, decision frequency, and the model family. Our contributions are an HDBO control formulation covering strategy selection, configuration, and execution length; the HERA–PRISM architecture with shared observations and HDBO-specific diagnostics; and an empirical evaluation of its performance, behavior, and inference cost.

2 Background

Given a budget of T evaluations, we minimize an expensive black-box function $f : \mathcal{X} \to \mathbb{R}$ over a bounded domain $\mathcal{X} \subseteq \mathbb{R}^D$. Evaluating x_t yields $y_t = f(x_t) + \epsilon_t$, and the resulting data $\mathcal{D}_t =$

$\{(x_i, y_i)\}_{i=1}^t$ define the incumbent $y_t^\star = \min_{i \le t} y_i$. Standard BO fits a probabilistic surrogate to $\mathcal{D}_t$ and uses an acquisition function to select x_{t+1} (Shahriari et al., 2015; Frazier, 2018).

High-dimensional Bayesian optimization. When D is large relative to T, limited observations make both surrogate fitting and acquisition optimization difficult (Papenmeier et al., 2025b; Duque et al., 2024). HDBO methods impose different structural assumptions or search biases. Variable-selection methods assume that only a subset of variables is relevant and iteratively optimize over a small set of important dimensions (Li et al., 2017; Eriksson & Jankowiak, 2021; Song et al., 2022; Hoang et al., 2025). Embedding-based methods assume that the objective admits a low-dimensional subspace or embedded manifold, and perform optimization in a reduced space before projecting back to the original space (Wang et al., 2016; Nayebi et al., 2019; Letham et al., 2020; Papenmeier et al., 2022). Decomposition-based methods assume that the objective can be decomposed into a sum of low-dimensional functions, thereby exploiting low-order interactions or additive structures (Kandasamy et al., 2015). Trust-region methods instead restrict search to a local region around the current best point (Eriksson et al., 2019). Carefully designed full-dimensional BO can also be competitive when such structural assumptions are absent or hard to verify (Hvarfner et al., 2024; Papenmeier et al., 2025b). These are useful inductive biases for the unknown objective. Section 3 describes how PRISM makes several such biases available to the agent; methodological details and further HDBO references appear in Appendix A.

Large language models for black-box optimization. Foundation models offer several complementary capabilities for black-box optimization: extracting priors from free-form task context, representing optimization behavior with flexible sequence models, and transferring performance prediction across search spaces (Song et al., 2024). One line of work realizes these capabilities within an explicit BO loop by assigning the LLM a particular component. At the modeling stage, LLMs can design or evolve GP kernels (Suwandi et al., 2025), provide representations for probabilistic surrogates (Agarwal et al., 2025; Nguyen et al., 2024; Lin et al., 2026), or supply low-cost predictions that are fused with real evaluations (Chen et al., 2026). At the decision stage, they can select acquisition functions (Ngo et al., 2026), construct or filter candidates (Cissé et al., 2025; Yang et al., 2025), or select between LLM and BO proposals (Chang et al., 2025). LLMs can also translate feedback into observations for preference-based BO (Adachi et al., 2024; Kobalczyk et al., 2026). These methods enrich a prescribed BO pipeline while retaining numerical modeling or acquisition machinery. A second line gives the language model a more direct role in constructing evaluations. OPRO (Yang et al., 2024) conditions on previous solutions and objective values to generate new proposals. LLAMBO (Liu et al., 2024b), a principal baseline in our study, reformulates several BO components as language-model inference: it uses task descriptions for zero-shot warm-starting, samples promising configurations conditioned on the observation history, and predicts candidate performance with an LLM surrogate. LLAMBO therefore extends beyond point proposal, but its intervention interface remains fixed. Across both lines, the location and form of LLM intervention are algorithm-specified.

Agentic BBO. Recently, agentic systems organize optimization as a continuing reason–act–observe process and give the model authority beyond a single prescribed component. AgentHPO (Liu et al., 2024a) interprets an HPO task, proposes configurations, and updates its recommendations from trial outcomes; OptiMindTune (Madiraju & Madiraju, 2025) distributes recommendation, evaluation, and decision making across specialized agents. These systems broaden the scope of autonomous optimization, but do not generally retain an explicit uncertainty-aware numerical backend. Hybrid systems couple agent reasoning to such machinery: Centaur (Ferreira et al., 2026) exposes CMA-ES state and proposals to an autoresearch agent, whereas Sara (Brunzema et al., 2026) lets an agent inspect and reconfigure a Bayesian backend during the run. Sara is therefore the most closely related prior system. In HDBO, however, complete candidate generation is a difficult numerical task and the useful search bias is unknown. Therefore, our proposed HERA controls which bias to test and how to configure it, while PRISM generates several consecutive candidates before the agent reassesses the strategy. Table 1 compares the primary control roles.

3 HERA: Hypothesis- and Evidence-guided Research Agent

Extending Agentic BO to high dimensions changes the division of labor between the LLM and the numerical optimizer. As motivated in Section 1, neither direct LLM candidate generation nor com-

Table 1: Primary control roles of Sara with lenz (Brunzema et al., 2026) and our HERA with PRISM.

Aspect	Sara + lenz	HERA + PRISM
Agent role	Experiment designer	Structure strategist
Decision target	Next evaluation point	Structural hypothesis
Candidate generation	LLM or lenz	PRISM only
Control level	Point-level	Strategy-level
Control frequency	Every evaluation	Every search block
Adaptation	Point and BO policy	HDBO strategy
Backend role	Advisory instrument	Numerical executor

mitment to a single numerical backend is reliable when the objective's useful structure is unknown. The central question is therefore not which point the LLM should propose, but which structural hypothesis should guide numerical search and when that hypothesis should be revised. We assign the LLM this strategy-level role: HERA maintains a working hypothesis and selects how it should be operationalized, while PRISM executes the corresponding numerical strategy and generates evaluation points. This gives the LLM strategic authority without asking it to solve high-dimensional numerical optimization directly (Figure 1).

Agentic HDBO. We follow the general metalevel decision-process view of Agentic BO introduced by Sara (Brunzema et al., 2026), but define a different action space for high-dimensional search. Apart from one explicitly separated initialization action, HERA never emits a candidate point. Its admissible actions are

$$\mathcal{A}_{\text{HERA}} = \mathcal{A}_{\text{state}} \cup \mathcal{A}_{\text{init}} \cup \mathcal{A}_{\text{strategy}} \cup \mathcal{A}_{\text{block}} \cup \{\texttt{stop}\}, \tag{1}$$

where the first four sets correspond exactly to the four typed PRISM tools:

$$\begin{aligned} \mathcal{A}_{\text{state}} &= \{\texttt{get_state}(\nu)\}, \\ \mathcal{A}_{\text{init}} &= \{\texttt{propose_initial_solution}(\iota)\}, \\ \mathcal{A}_{\text{strategy}} &= \{\texttt{set_search_strategy}(\rho, b, \lambda, \tau, r) : \rho \in \{\texttt{switch}, \texttt{reconfigure}, \texttt{restart}\}\}, \\ \mathcal{A}_{\text{block}} &= \{\texttt{run_search_block}(L, r) : 1 \le L \le \min(50, T - t)\}. \end{aligned} \tag{2}$$

For a state query, ν specifies the state and structure detail levels, whether to repeat task metadata or the strategy catalog, and the recent window and number of top observations to return. For the one-shot initialization action, $\iota = (v, \phi, \delta, r)$ contains named variable values v in the task's original representation, a normalized fill value ϕ for unnamed coordinates, an optional spread δ, and a rationale r. In a strategy action, b names a PRISM strategy, λ contains its schema-validated configuration, τ optionally identifies the component to restart, and r records the reason for the intervention. Arguments not used by the selected intervention ρ are omitted. Finally, a block action specifies its evaluation horizon L and rationale r; it is the only action that invokes the evaluator. The initialization action merely queues a design, which is charged when the next block evaluates it. The `stop` action is handled by the agent loop rather than PRISM and terminates the campaign without changing numerical state. A decision cycle may therefore contain several zero-budget actions followed by at most one budget-consuming block. Numerical models update after every observation inside that block, whereas HERA reassesses the structural hypothesis only after control returns.

Accordingly, HERA's decision context augments the usual Agentic BO state with the active HDBO strategy, structural evidence, and the realized return of previous blocks. These quantities support decisions under structural uncertainty. The key difference from point-level Agentic BO is thus not merely an action set: HERA controls the structural search process, while PRISM retains authority over complete high-dimensional candidate generation.

HDBO diagnostics and working hypotheses. In high dimensions, the same lack of progress can result from inadequate coverage, an unreliable surrogate, or a mismatch between the active search bias and the objective. PRISM therefore augments routine optimization feedback with three HDBO-specific diagnostic groups: search geometry, summarized by normalized OTSD and observation entropy (Papenmeier et al., 2025a); surrogate credibility, assessed through posterior uncertainty and leave-one-out residuals; and provisional evidence for coordinate sparsity, low-dimensional variation,

and decomposable interactions. These signals are returned as bounded summaries and are aligned with the numerical strategies available to HERA. Appendix G.1 provides their definitions, returned fields, and failure cases.

HERA combines these diagnostics with the active strategy, realized block improvement, and remaining budget to maintain a *working hypothesis*: an explanation of which search bias may currently be useful. The hypothesis links supporting and conflicting evidence to a numerical intervention and a condition for reassessment. Surrogate credibility qualifies the structural signals, while subsequent evaluator outcomes test whether the chosen bias was useful. Evaluator returns remain observed evidence; diagnostics are derived evidence and metadata is prior context. Thus, a working hypothesis guides adaptation without claiming that HERA has recovered the objective's true structure.

PRISM tools. HERA acts through four typed PRISM tools (Table 2). These tools are exposed directly to the language model as structured function calls; HERA does not operate PRISM through a shell or a free-form CLI.

Table 2: PRISM tools available to HERA, with arguments validated by PRISM.

Tool	Returned information or effect
`get_state`	Budget and progress, active strategy and scoreboard, geometry, surrogate credibility, structural evidence, and optional metadata/catalog. No state change.
`propose_initial_solution`	Validate and queue one complete initial design. The next search block evaluates it first.
`set_search_strategy`	Atomically switch, reconfigure, or restart the numerical strategy without an objective evaluation.
`run_search_block`	Return every observed value and best-so-far, block improvement, and remaining budget.

The formal action classes in Equations 1 and 2 map directly to these calls. State queries are read-only and return bounded summaries rather than unrestricted backend objects. Switching selects a different inductive bias; reconfiguration changes the active strategy's semantic hyperparameters; restarting rebuilds a strategy-specific component, such as a trust region, embedding, grouping, or active set. PRISM rejects unavailable strategies, invalid configuration keys, dense candidate vectors, out-of-range block sizes, and actions that violate the one-shot initialization contract.

The numerical strategies available through this interface embody distinct assumptions about the objective. We choose representative methods for several common HDBO hypotheses: Vanilla BO for a full-dimensional default (Hvarfner et al., 2024), LassoBO for axis-aligned sparsity (Hoang et al., 2025), an ALEBO-inspired low-rank strategy for low-dimensional subspace structure (Letham et al., 2020), Add-GP-UCB for additive interactions (Kandasamy et al., 2015), and TuRBO for local regularity (Eriksson et al., 2019). The broader sparse and embedding families also include methods such as SAASBO and BAxUS (Eriksson & Jankowiak, 2021; Papenmeier et al., 2022). These strategies provide numerical mechanisms through which HERA tests and revises working hypotheses as evidence accumulates, rather than fixing one hypothesis for the entire run.

After the shared opening design, HERA may translate documented task semantics into one proposed design. PRISM validates it, places it at the front of the next search block, and charges it to the common evaluation budget. Once any search block has begun, this channel closes and every subsequent candidate is generated by a numerical strategy. This makes the role of semantic initialization explicit rather than conflating it with later Agentic HDBO decisions.

Search blocks. Given an accepted `run_search_block`(L_k, r_k) action, PRISM runs the active backend sequentially for at most L_k evaluations. Any strategy change is validated and applied first, before the block starts. For $t = t_k, \ldots, t_k + L_k - 1$,

$$x_{t+1} = \mathrm{Suggest}_{\mathcal{B}_k, \lambda_k}(\mathcal{D}_t), \tag{3}$$

$$y_{t+1} = f(x_{t+1}) + \epsilon_{t+1}, \qquad \mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}. \tag{4}$$

PRISM fits the numerical model, optimizes its acquisition rule, materializes a complete candidate, and checks its dimensionality, bounds, declared constraints, and duplication status before evaluation. The backend can update after each returned observation. All evaluated points remain in the shared observation history, so a newly selected strategy can reuse previous evidence even when incompatible strategy-specific state must be rebuilt.

The block terminates when L_k evaluations have been completed, the remaining budget is exhausted, or the evaluator fails. PRISM then constructs the next context c_{k+1} from the updated history and strategy state, returning control to HERA. This transition establishes the central boundary of the method: HERA decides what numerical experiment to run and how long to run it, while PRISM retains authority over ordinary high-dimensional candidate generation, model fitting, evaluation, and budget accounting.

4 EXPERIMENTS

We first evaluate whether existing methods can transfer to high-dimensional tasks, then compare HERA with simpler controllers over the same numerical backends. Real-world benchmarks and ablations assess the complete system. Our code is available in the supplemental files.

Experimental Setup. Unless stated otherwise, each run has a budget of 100 objective evaluations, including five shared scrambled-Sobol opening points, and uses 10 seeds. All objectives are minimized. We compare HERA with Sobol search; five classical HDBO methods—Vanilla BO (Hvarfner et al., 2024), TuRBO (Eriksson et al., 2019), Add-GP-UCB (Kandasamy et al., 2015), ALEBO (Letham et al., 2020), and LassoBO (Hoang et al., 2025); and LLM-based or agentic systems—LLAMBO (Liu et al., 2024b), Centaur (Ferreira et al., 2026), and Sara (Brunzema et al., 2026). The principal LLM comparison uses DeepSeek-V4.1-Flash. We report incumbent trajectories and final objectives with variability across seeds. A task-level win refers to the mean final objective, not every seed. Real-world LLM systems receive task information that classical HDBO methods do not use, so those comparisons evaluate complete systems. Appendix B documents task definitions, representations, and implementation details.

4.1 SYNTHETIC BENCHMARKS

We first examine whether existing LLM-based and agentic BO methods transfer to high-dimensional problems without task metadata. The suite comprises Hartmann6 (extended to 100-D), Ackley10 (extended to 200-D), Lévy15 (extended to 300-D), and Rosenbrock10 (extended to 300-D), covering a range of optimization challenges. Task identities are anonymized and metadata cards are empty.

As shown in Figure 2, the LLM-based and agentic baselines yield worse mean final objectives than strong numerical HDBO methods across all four tasks (mean $\pm$ std over 10 seeds), indicating that naively extending these methods to high-dimensional settings fails to deliver competitive performance. In contrast, HERA performs on par with the strongest traditional high-dimensional BO baselines: it ranks first on Lévy, second on Ackley, and third on Hartmann and Rosenbrock, matching or outperforming these numerical methods. Appendix B.1 gives the details.

The variation across functions is as informative as the aggregate ranking. No single strategy wins everywhere: full-dimensional or sparse search works best on Hartmann, HERA is best on Lévy, and local trust-region search leads on Rosenbrock. In other words, there is no one high-dimensional search bias that is always preferable. What HERA offers is therefore not a universally better strategy, but a robust one: it stays close to the best numerical methods without relying on the hidden construction of the task or any prescribed structural assumption. This is exactly what our formulation aims for—the agent does not need to recover the true structure, only to keep a useful numerical search strategy as evidence accumulates.

4.2 HEURISTIC AND BANDIT BASELINES

We next ask how much of HERA's performance can be explained by simpler backend selection over the same numerical portfolio. The structure-aware heuristic maps the same high-dimensional diagnostics to backends through fixed rules. UCB (Auer et al., 2002) instead treats each backend

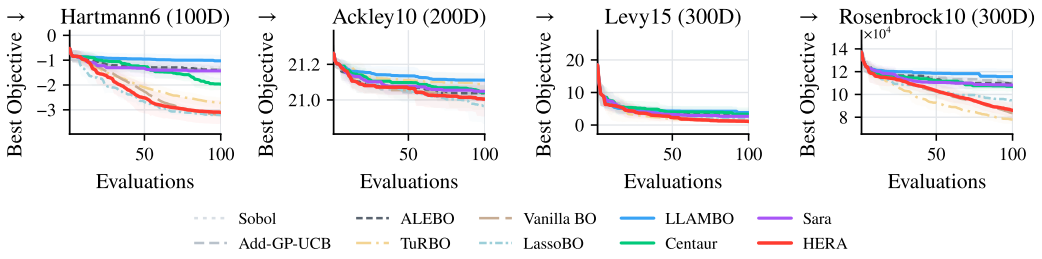

Figure 2: **Metadata-free high-dimensional optimization.** Incumbent trajectories on four synthetic tasks, comparing HERA, classical HDBO methods, and LLM-based methods (LLAMBO (Liu et al., 2024b), Centaur (Ferreira et al., 2026), Sara (Brunzema et al., 2026)) over 10 seeds. The agentic baselines show a transfer gap, while HERA remains competitive.

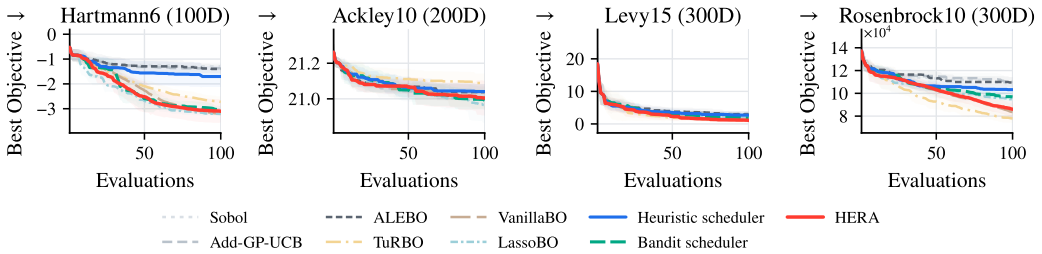

Figure 3: **Comparison with heuristic and bandit baselines.** HERA, a structure-aware heuristic, and UCB use the same five numerical backends on four synthetic tasks. HERA is better than both adaptive baselines on three tasks and ranks second to UCB on Ackley.

as an arm. Both baselines use the same portfolio, initialization, evaluation budget, and fixed five-evaluation blocks on the four synthetic tasks. HERA can additionally configure or restart a backend and choose its execution horizon. Appendix D specifies the policies.

Figure 3 shows that HERA performs better than both controllers on Hartmann, Lévy, and Rosenbrock, while UCB is slightly better on Ackley. The performance of the two baselines also varies by task: UCB is competitive on Hartmann and Ackley, whereas the heuristic is consistently weaker. The scheduler traces clarify why UCB's competitiveness on two tasks does not make it a sufficient replacement for HERA. As Figure 5 in Appendix D shows, the backend allocations of UCB are nearly task-invariant: it frequently switches across the portfolio at a similar rate across the four objectives, regardless of their different structural properties. Its switching is therefore driven primarily by recent normalized improvement rather than by task-specific structural evidence. This is the essential limitation of a reward-only controller: it can react to whichever backend happened to do well last, but it cannot represent, revise, or condition on a structural hypothesis. HERA instead exhibits task-dependent allocation patterns, concentrating on different subsets of strategies across objectives.

4.3 Real-World Benchmarks

We next evaluate the complete HERA system on eight real-world HDBO tasks: Rover (60-D) (Wang et al., 2017), HalfCheetah (102-D) (Wang et al., 2019), MOPTA (124-D) (Eriksson & Jankowiak, 2021), Chip Design (128-D) (Xue et al., 2026), Mazda (148-D) (Kohira et al., 2018), Ant (216-D) (Wang et al., 2020), and Median Molecules 1 and 2 (256-D) (Brown et al., 2019). The suite spans trajectory design, continuous control, engineering, chip placement, and molecular optimization. HERA and the LLM-based or agentic baselines receive the same task information, whereas the classical HDBO methods operate only on evaluated inputs and outputs. Figure 4 reports these two comparator groups separately; the latter comparison is therefore a complete-system evaluation rather than an isolation of language metadata.

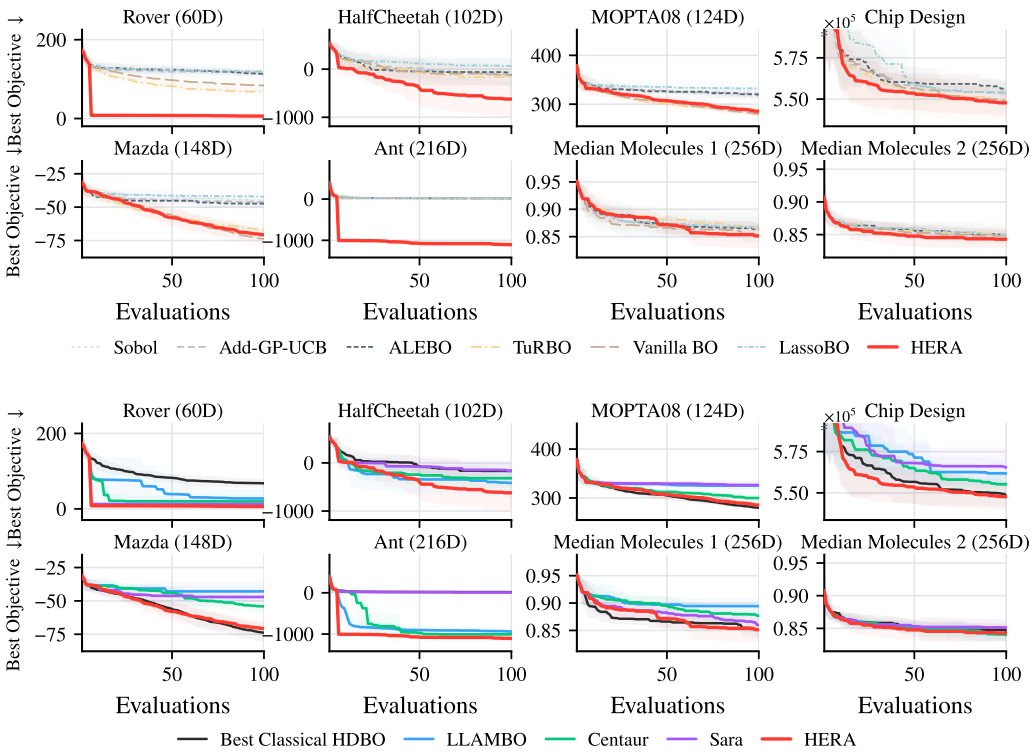

Figure 4: **Real-world tasks.** Comparisons with classical HDBO methods (top) and with LLM-based methods (bottom) over 100 evaluations and 10 seeds. Full results are reported in Appendix E.

Compared with classical HDBO methods, HERA obtains a better mean final objective than the strongest evaluated baseline on six of eight tasks. The largest improvements occur on Rover, HalfCheetah, and Ant. On MOPTA and Mazda, full-dimensional BO remains better, while HERA is competitive. On Median Molecules 1 and 2 and Chip Design, HERA has the better performance. Thus, HERA is broadly competitive across heterogeneous tasks. Compared with the LLM-based and agentic methods, HERA achieves the best mean final objective on seven tasks. The exception is Median Molecules 2, where Centaur and LLAMBO are slightly better. The contrast is especially clear on Rover, HalfCheetah, and Ant: although several agentic methods benefit from task information, HERA attains a better final mean than each of them and also matches or exceeds the strongest classical HDBO baseline (black curve). On Chip Design, HERA also has the best mean across both comparator groups. Notably, other LLM-based and agentic methods perform worse than the strongest classical HDBO baseline on tasks like Chip Design and Median Molecules 1, whereas HERA is still competitive. Appendix E reports the complete final values and seed variability.

Taken together, the results support HERA as a strong complete system for real-world Agentic HDBO. The largest gains appear on Rover, HalfCheetah, and Ant, where semantic context can inform initialization and numerical feedback remains important during search. Smaller gains on the engineering and molecular tasks show that an effective fixed optimizer can remain difficult to improve upon. Because HERA combines task semantics and online strategy adaptation, these experiments establish the value of the complete framework but do not attribute the gains to backend orchestration alone; the following ablations examine these factors separately.

4.4 Ablation Studies

Having established end-to-end performance, we next ask which information and control choices shape HERA's behavior, performance, and cost. We study structural diagnostics, task metadata, decision frequency, and agent model. Detailed trajectories and statistics are deferred to the appendix.

Structural evidence. We first ask whether the HDBO-specific diagnostics provide a meaningful control signal. We remove the sparsity, low-dimensional-subspace, and interaction channels while retaining the other components. Removal alters the strategies adopted and makes backend use substantially more concentrated, most visibly toward Vanilla BO on MOPTA and Mazda (Appendix Figure 6). Final performance, however, remains similar on most tasks, so the experiment supports a behavioral effect rather than an aggregate optimization gain. An offline audit reinforces this interpretation: variable-relevance and low-dimensional-subspace readings become more informative as observations accumulate, whereas the interaction-grouping diagnostic remains unreliable. These diagnostics should therefore be viewed as provisional evidence for deciding which search bias to test, not as recovered ground-truth structure; Appendices F and G provide the full results.

Task metadata. We next test whether semantic task information contributes beyond numerical observations. The strict-removal condition withholds task identity, descriptions, and the corresponding prompt instructions while preserving the optimization budget and interface. Removal degrades performance on most tasks, with the clearest effects on Rover and Ant, but slightly improves performance on a few other tasks (Appendix H). It also almost eliminates the use of the initial-solution tool. Metadata therefore acts as a task-dependent prior rather than a universally beneficial input, and its effect can enter through both initialization and later reasoning. This intervention consequently does not isolate metadata-driven strategy selection from semantic initialization.

Decision frequency. We then ask whether HERA should reconsider its strategy after every evaluation. On HalfCheetah, MOPTA, and Mazda, we compare adaptive blocks with blocks of size one, both with and without context compaction. With context compaction, returning control after every evaluation increases inference by roughly an order of magnitude without a consistent improvement across paired runs. Without context compaction, making decisions after every evaluation helps on some tasks, but the improvement is not consistent across tasks (Figure 12). Adaptive reassessment is therefore the more practical default: it allows the numerical backend to accumulate evidence before deliberation while avoiding a large inference overhead.

Model family. To measure sensitivity to the underlying LLM, we replace DeepSeek-V4.1-Flash with Kimi-K3 and GPT-6-astra on the same three tasks. Kimi-K3 has the best final mean on HalfCheetah, whereas GPT-6-astra has the best means on MOPTA and Mazda (Appendix I). No model dominates across tasks. Agent-model choice is therefore another task-dependent design factor rather than a uniformly improving substitution.

Additional analyses of tool-use trajectories and token consumption are provided in Appendices J and L. They show selective, task-dependent use of the backend portfolio, but remain descriptive and do not establish that more frequent interventions cause better performance. Collectively, the ablations show that HERA's information and control interface materially changes how optimization is conducted. Structural evidence primarily changes strategy use, metadata can change both initialization and subsequent decisions, and per-evaluation deliberation sharply increases inference cost. Model rankings and metadata effects remain task dependent.

5 Conclusion

We presented HERA and PRISM, an agent–engine framework for high-dimensional Bayesian optimization in which an LLM manages revisable structural hypotheses and search strategies while numerical backends generate high-dimensional candidates. On metadata-free synthetic functions, HERA remains competitive with strong HDBO methods and improves over the evaluated LLM-based and agentic baselines; across eight real-world tasks, it performs strongly against other methods. The ablations further show that structural evidence changes strategy use, metadata provides task-dependent priors, and adaptive decision intervals control inference cost. Together, these results support a view of Agentic HDBO in which the LLM manages the evolving search strategy rather than directly solving high-dimensional candidate generation. The current framework is bounded by the diagnostics and numerical strategies exposed by PRISM: HERA can test and revise hypotheses within this interface, but its working hypotheses do not guarantee recovery of the objective's true structure. Future work could expand the available tool and strategy space, calibrate the reliability of structural diagnostics, and transfer strategy experience across related optimization tasks.

AI USE STATEMENT

In this work, we used generative AI tools for translation and language-related assistance, including improving grammar, clarity, concision, and the presentation of author-written text. We have not used generative AI tools for formulating research questions or hypotheses, developing the conceptual or theoretical framework, designing the methodology or experiments, implementing the proposed method, generating or processing data, or analyzing and interpreting the experimental results. We have reviewed all AI-assisted work. All AI-assisted revisions were checked against the authors' intended meaning and reviewed for technical accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The appendices specify task representations, opening designs, evaluation budgets, backend adaptations, scheduler rules, intervention scopes, and sample counts. All objective evaluations, including prior proposals, count toward the same budget. No additional results are inferred from unexecuted experiments. The code is provided in supplementary material.

REFERENCES

Masaki Adachi, Brady Planden, David A. Howey, Michael A. Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. Looping in the human: Collaborative and explainable Bayesian optimization. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238, pp. 505–513, Valencia, Spain, 2024.

Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with LLMs via Bayesian optimization. In *The 13th International Conference on Learning Representations*, Singapore, 2025.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3): 1096–1108, 2019.

Paul Brunzema, Louis Tiao, Nhat Le, Kevin De Angeli, Yao Xuan, and Djordje Gligorijevic. Agentic Bayesian optimization through surrogate-augmented autoresearch. *arXiv:2608.00316*, 2026.

Chih-Yu Chang, Milad Azvar, Chinedum Okwudire, and Raed Al Kontar. LLINBO: Trustworthy LLM-in-the-loop Bayesian optimization. *arXiv:2505.14756*, 2025.

Zhuo Chen, Xinzhe Yuan, Jianshu Zhang, Jinzong Dong, Ruichen Zhou, Yingchun Niu, Tianhang Zhou, Yu Yang Fredrik Liu, Yuqiang Li, Nanyang Ye, and Qinying Gu. LABO: LLM-accelerated Bayesian optimization through broad exploration and selective experimentation. In *43rd International Conference on Machine Learning*, 2026.

Abdoulatif Cissé, Xenophon Evangelopoulos, Vladimir V. Gusev, and Andrew I. Cooper. Language-based Bayesian optimization research assistant. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, pp. 4967–4975, 2025.

Miguel González Duque, Richard Michael, Simon Bartels, Yevgen Zainchkovskyy, Søren Hauberg, and Wouter Boomsma. A survey and benchmark of high-dimensional Bayesian optimization of discrete sequences. In *Advances in Neural Information Processing Systems 37*, Vancouver, BC, Canada, 2024.

D. Eriksson and M. Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, pp. 493–503, Virtual, 2021.

D. Eriksson, M. Pearce, J. R. Gardner, R. D. Turner, and M. Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems 32*, pp. 5497–5508, Vancouver, Canada, 2019.

Fabio Ferreira, Lucca Wobbe, Arjun Krishnakumar, Frank Hutter, and Arber Zela. Can LLMs beat classical hyperparameter optimization algorithms? a study on autoresearch. *arXiv:2603.24647*, 2026.

P. I. Frazier. A tutorial on Bayesian optimization. *CoRR abs/1807.02811*, 2018.

E. Han, I. Arora, and J. Scarlett. High-dimensional Bayesian optimization via tree-structured additive models. In *Proceedings of the 35th Association for the Advancement of Artificial Intelligence*, pp. 7630–7638, Virtual, 2021.

Vu Viet Hoang, Hung The Tran, Sunil Gupta, and Vu Nguyen. High dimensional Bayesian optimization using lasso variable selection. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, pp. 3106–3114, Mai Khao, Thailand, 2025.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20793–20817, Vienna, Austria, 2024.

K. Kandasamy, J. G. Schneider, and B. Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 295–304, Lille, France, 2015.

Katarzyna Kobalczyk, Zhiyuan Jerry Lin, Benjamin Letham, Zhuokai Zhao, Maximilian Balandat, and Eytan Bakshy. LILO: Bayesian optimization with natural language feedback. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Takehisa Kohira, Hiromasa Kemmotsu, Akira Oyama, and Tomoaki Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the 20th ACM Conference on Genetic and Evolutionary Computation*, pp. 183–184, Kyoto, Japan, 2018.

B. Letham, R. Calandra, A. Rai, and E. Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, pp. 1546–1558, Vancouver, Canada, 2020.

C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton. High dimensional Bayesian optimization using dropout. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2096–2102, Melbourne, Australia, 2017.

Zhiyuan Jerry Lin, Benjamin Letham, Samuel Dooley, Maximilian Balandat, and Eytan Bakshy. Embedding by elicitation: Dynamic representations for Bayesian optimization of system prompts. *arXiv:2605.19093*, 2026.

Siyi Liu, Chen Gao, and Yong Li. Large language model agent for hyper-parameter optimization. *arXiv:2402.01881*, 2024a.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *The 12th International Conference on Learning Representations*, Vienna, Austria, 2024b.

Meher Bhaskar Madiraju and Meher Sai Preetam Madiraju. OptiMindTune: A multi-agent framework for intelligent hyperparameter optimization. *arXiv:2505.19205*, 2025.

A. Nayebi, A. Munteanu, and M. Poloczek. A framework for Bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4752–4761, Long Beach, CA, 2019.

Giang Ngo, Dat Phan Trong, Dang Nguyen, Sunil Gupta, and Svetha Venkatesh. Adaptive acquisition selection for Bayesian optimization with large language models. In *The 14th International Conference on Learning Representations*, 2026.

Tung Nguyen, Qiuyi Zhang, Bangding Yang, Chansoo Lee, Jorg Bornschein, Yingjie Miao, Sagi Perel, Yutian Chen, and Xingyou Song. Language model embeddings can be sufficient for Bayesian optimization. *arXiv:2410.10190*, 2024.

Mohammad Nabi Omidvar, Xiaodong Li, Yi Mei, and Xin Yao. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):378–393, 2014.

Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. In *Advances in Neural Information Processing Systems 35*, pp. 11586–11601, New Orleans, LA, 2022.

Leonard Papenmeier, Nuojin Cheng, Stephen Becker, and Luigi Nardi. Exploring exploration in Bayesian optimization. In *Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence*, pp. 3388–3415, Rio de Janeiro, Brazil, 2025a.

Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Understanding high-dimensional Bayesian optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, Vancouver, Canada, 2025b.

Bojana Ranković and Philippe Schwaller. GOLLuM: Gaussian process optimized LLMs—reframing LLM finetuning through Bayesian optimization. *arXiv:2504.06265*, 2025.

P. Rolland, J. Scarlett, I. Bogunovic, and V. Cevher. High-dimensional Bayesian optimization via additive models with overlapping groups. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, pp. 298–307, Playa Blanca, Spain, 2018.

Andrej Schwanke, Lyubomir Ivanov, David Salinas, Fabio Ferreira, Aaron Klein, Frank Hutter, and Arber Zela. Improving LLM-based global optimization with search space partitioning. In *The 14th International Conference on Learning Representations*, 2026.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. Monte carlo tree search based variable selection for high dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 35*, pp. 28488–28501, New Orleans, LA, 2022.

Xingyou Song, Yingtao Tian, Robert Tjarko Lange, Chansoo Lee, Yujin Tang, and Yutian Chen. Position: Leverage foundational models for black-box optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 46168–46180, 2024.

Richard Cornelius Suwandi, Feng Yin, Juntao Wang, Renjie Li, Tsung-Hui Chang, and Sergios Theodoridis. Adaptive kernel design for Bayesian optimization is a piece of CAKE with LLMs. In *Advances in Neural Information Processing Systems 38*, 2025.

Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. In *Advances in Neural Information Processing Systems 33*, pp. 19511–19522, Virtual, 2020.

Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv:1907.02057*, 2019.

Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Freitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55(1):361–387, 2016.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, pp. 745–754, Canary Islands, Spain, 2017.

Nathan Wycoff, Mickaël Binois, and Stefan M. Wild. Sequential learning of active subspaces. *Journal of Computational and Graphical Statistics*, 30(4):1224–1237, 2021.

Zhitong Xu, Haitao Wang, Jeff M. Phillips, and Shandian Zhe. Standard gaussian process is all you need for high-dimensional Bayesian optimization. In *Proceedings of the 13th International Conference on Learning Representations*, Singapore, 2025.

Ke Xue, Ruo-Tong Chen, Rong-Xi Tan, Xi Lin, Yunqi Shi, Siyuan Xu, Mingxuan Yuan, and Chao Qian. Bboplace-bench: Benchmarking black-box optimization for chip placement. *arXiv:2510.23472*, 2026.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The 12th International Conference on Learning Representations*, Vienna, Austria, 2024.

Zhuo Yang, Daolang Wang, Lingli Ge, Beilun Wang, Tianfan Fu, and Yuqiang Li. Reasoning BO: Enhancing Bayesian optimization with long-context reasoning power of LLMs. *arXiv:2505.12833*, 2025.

Rosen Ting-Ying Yu, Cyril Picard, and Faez Ahmed. Git-bo: High-dimensional Bayesian optimization with tabular foundation models. *arXiv:2505.20685*, 2026.

Juliusz Krysztof Ziomek and Haitham Bou-Ammar. Are random decompositions all we need in high dimensional Bayesian optimisation? In *Proceedings of the 40th International Conference on Machine Learning*, pp. 43347–43368, Honolulu, Hawaii, 2023.

A Further Related Work

Classical High-dimensional Bayesian Optimization. Random embeddings such as REMBO and HeSBO search through lower-dimensional representations (Wang et al., 2016; Nayebi et al., 2019). ALEBO modifies embedding geometry, whereas BAxUS expands nested subspaces as the search progresses (Letham et al., 2020; Papenmeier et al., 2022). Other approaches fit sparse-coordinate models or select active variables (Eriksson & Jankowiak, 2021; Song et al., 2022; Hoang et al., 2025), model additive components with overlapping, tree-structured, or randomized decompositions (Kandasamy et al., 2015; Rolland et al., 2018; Han et al., 2021; Ziomek & Bou-Ammar, 2023), or search with adaptive local trust regions (Eriksson et al., 2019). Full-dimensional BO with dimension-aware priors and careful candidate generation remains a strong alternative (Hvarfner et al., 2024; Xu et al., 2025; Papenmeier et al., 2025b). These lines of work supply different biases.

Language models at different BO interfaces. Methods with an explicit numerical BO loop use language models at several interfaces. CAKE uses LLM operations to generate and refine GP kernels (Suwandi et al., 2025). Embed-then-Regress uses fixed language-model embeddings, GOLLuM jointly adapts an LLM representation with a GP through marginal likelihood, and ReElicit repeatedly constructs a semantic representation for surrogate modeling (Nguyen et al., 2024; Ranković & Schwaller, 2025; Lin et al., 2026). LABO treats LLM predictions as a low-cost information source and fuses them with experimental observations in a multi-fidelity model (Chen et al., 2026). At the decision layer, LMABO selects an acquisition function (Ngo et al., 2026); BOPRO, BORA, and Reasoning BO condition, propose, or rank candidates (Agarwal et al., 2025; Cissé et al., 2025; Yang et al., 2025); LLINBO arbitrates between LLM and BO proposals (Chang et al., 2025); and LILO maps natural-language feedback to preference observations (Kobalczyk et al., 2026). A separate family places proposal generation more directly in the language model: OPRO generates solutions from the optimization trajectory, LLAMBO combines language-model prediction with promising-point generation, and HOLLM augments iterative LLM optimization with search-space partitioning (Yang et al., 2024; Liu et al., 2024b; Schwanke et al., 2026). These approaches differ in how much numerical BO machinery they retain, but each prescribes the interface through which the LLM affects search.

Agentic control and adaptive numerical policies. Agentic optimization spans several levels of control. AgentHPO iteratively proposes hyperparameters from task descriptions and trial outcomes and OptiMindTune divides HPO among cooperating agents (Liu et al., 2024a; Madiraju & Madiraju, 2025). Centaur exposes CMA-ES progress to an autoresearch agent and combines numerical and LLM proposals (Ferreira et al., 2026). Sara gives an agent access to a Bayesian backend, including surrogate inspection, evaluation selection, and run-time policy changes (Brunzema et al., 2026). Its scope of control makes it the closest agentic BO comparison, while the high-dimensional setting motivates our block-level control interface.

B Experimental Settings and Task Definitions

B.1 Tasks and representations

Table 3 lists the optimization suite. Unless an ablation specifies otherwise, each task–method pair uses seeds 0–9, a total budget of 100, and a shared five-point scrambled-Sobol opening design. Numerical optimization uses normalized coordinates, converted to the native representation before evaluation. Reward-maximization tasks are oriented toward minimization. The molecular tasks optimize a continuous latent representation of the GuacaMol median-molecule objectives (Brown et al., 2019); the reported dimension is that representation's dimension, not the number of atoms or a native discrete search-space dimension.

The synthetic embeddings are not sparse-coordinate embeddings: all ambient coordinates contribute through block averages. Shifted tasks apply a fixed coordinatewise shift with wrap-around, $(x + \delta) \bmod 1$, before evaluating the underlying construction.

Anonymized synthetic runs expose opaque identifiers and empty metadata cards. This reduces access to named benchmark formulas, but does not guarantee that an agent cannot infer properties from observations. The optional initial-solution tool remains available.

Table 3: Optimization tasks.

Task	D	Representation or construction
Hartmann-100	100	Six latent inputs formed by a seeded block-average embedding.
Ackley-200	200	Dense orthogonal mixing over all coordinates; shifted coordinates.
Lévy-300	300	Fifteen block-average latent inputs; shifted coordinates.
Rosenbrock-300	300	Full 300-coordinate chained Rosenbrock; shifted coordinates.
Rover	60	Trajectory-design parameters.
HalfCheetah	102	Continuous-control policy parameters.
MOPTA	124	Engineering-design benchmark.
Chip Design	128	Coordinates of 64 macros.
Mazda	148	Engineering-design benchmark.
Ant	216	Continuous-control policy parameters.
Median Molecules 1	256	Continuous molecular latent representation.
Median Molecules 2	256	Continuous molecular latent representation.

B.2 Baselines and Implementation Details

Fixed numerical comparators comprise the five portfolio members and Sobol search. The strongest classical comparator is selected for each task by mean final objective. This is an evaluation reference, not a deployable oracle or an additional numerical method.

LLAMBO, Centaur, and Sara are evaluated under common task wrappers, budgets, and opening designs. The principal real-world comparison uses the same DeepSeek-V4.1-Flash endpoint and archived task-card condition. The main HERA configuration uses DeepSeek-V4.1-Flash at temperature zero. Model ablations use Kimi-K3 at temperature one, required by that endpoint, and GPT-6-astra at temperature zero through a separate relay.

C Tool details

Table 4: PRISM tools exposed to HERA and their objective-budget effects.

Tool	Contract	Budget
`get_state`	Inspect a bounded state view. Summary mode reuses compact evidence; full mode refreshes surrogate credibility and structure diagnostics.	None
`propose_initial_solution`	Propose one prior-informed design after the shared opening design.	One when evaluated
`set_search_strategy`	Atomically switch, reconfigure, or restart a strategy. Arguments are checked against the selected strategy's semantic schema.	None
`run_search_block`	Execute the requested number of sequential proposals, validate every candidate, call the evaluator, and update the shared observation history.	Evaluations returned

HERA calls these operations through the in-process `PrismToolbox` function interface. PRISM also provides command-line commands for human inspection and reproducibility, but the agent does not issue shell commands or parse terminal output. The observed dataset is authoritative; diagnostics, scoreboards, and hypotheses are derived views. Switching starts a new backend-specific episode while preserving the complete observation history.

D Synthetic Results and Scheduler Definitions

D.1 Heuristic and bandit policies

The heuristic begins with full-dimensional BO (Vanilla BO) and reassesses after each five-evaluation block. It activates sparse search when active fraction is below 0.1, the low-dimensional subspace backend when dimension ratio is at most 0.02, grouped-GP search when the partition score is at least 0.9, and local search when normalized observation-tour length exceeds 0.05. Multiple triggers are resolved by a seeded random choice; no trigger retains the active backend. The geometry rule reflects the search distribution, not a statistical test of local smoothness.

UCB treats each block as one play. After selecting each untried backend, it maximizes

$$\bar{r}_i + \sqrt{\frac{2 \log N}{n_i}}, \qquad r_k = \frac{y^\star_{t_k} - y^\star_{t_{k+1}}}{s(\mathcal{D}_{t_k})}, \tag{5}$$

where n_i is the backend's play count and N the total number of plays. The response scale s uses 1.4826 times median absolute deviation, with interquartile-range and standard-deviation fallbacks. This makes the bonus comparable across objective scales.

Figure 5 shows the resulting per-seed timelines and backend allocations. The heuristic's partition-score trigger saturates and frequently routes evaluations to Add-GP-UCB, which explains a weakness of this particular rule. UCB is a stronger baseline and its trajectories are close to HERA on several tasks, but its allocations remain largely task-invariant and are driven by recent normalized improvement rather than by structural evidence. HERA instead exhibits task-dependent allocation patterns, committing longer blocks to a smaller subset of backends. These measurements characterize the tested policies rather than learned portfolio selection in general.

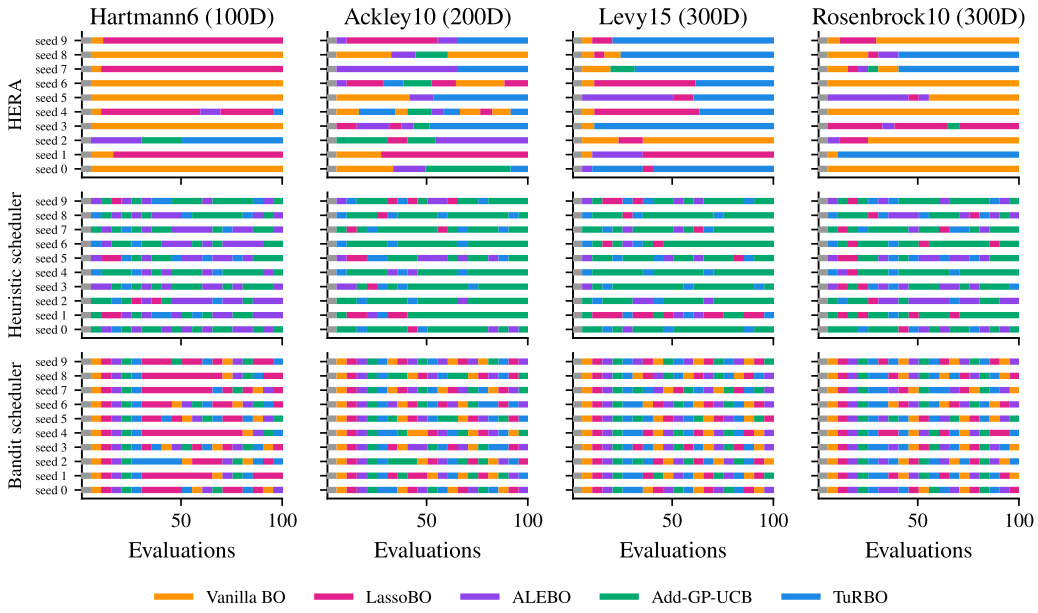

Figure 5: **Scheduler trajectories and allocations.** Per-seed timelines compare HERA, heuristic routing, and UCB. The heuristic's saturated partition-score trigger frequently allocates evaluations to grouped-GP search.

E Detailed Real-World Results

Tables 5–6 report the final objectives. MM1 and MM2 abbreviate Median Molecules 1 and 2, and HC abbreviates HalfCheetah. Full-dimensional BO outperforms HERA on MOPTA and Mazda. On

MM2, both Centaur and LLAMBO have better means than HERA, even though HERA beats every fixed numerical comparator. On Chip Design, HERA has the best mean, but its margin over full-dimensional BO is small relative to seed variation. The small MM1 gap likewise does not support a claim of substantial improvement. Large control-task gains coexist with substantial variability, particularly on HalfCheetah.

Table 5: Real-world final objectives: HERA and strong numerical comparators, mean ± sample SD over ten seeds. Lower is better.

Task	HERA	Full-dimensional BO	TuRBO
Rover	6.14 ± 2.04	84.03 ± 14.23	67.64 ± 12.16
HalfCheetah	-621.40 ± 370.83	-135.37 ± 204.38	-166.72 ± 166.04
MOPTA	285.57 ± 11.58	280.01 ± 3.97	286.82 ± 8.52
Mazda	-70.73 ± 5.42	-73.85 ± 6.88	-67.08 ± 5.30
Ant	-1105.20 ± 85.48	16.85 ± 3.20	19.93 ± 3.48
MM1	0.8514 ± 0.0224	0.8521 ± 0.0152	0.8663 ± 0.0271
MM2	0.8430 ± 0.0137	0.8483 ± 0.0105	0.8507 ± 0.0174
Chip (10^5)	5.477 ± 0.085	5.480 ± 0.061	5.483 ± 0.114

Table 6: Real-world final objectives for Sobol and LLM/agentic comparators, mean ± sample SD over ten seeds.

Task	Sobol	LLAMBO	Centaur	Sara
Rover	111.50 ± 6.19	27.46 ± 24.94	19.64 ± 16.46	12.65 ± 0.32
HC	-145.60 ± 265.58	-415.66 ± 317.55	-318.77 ± 199.42	-166.03 ± 224.58
MOPTA	316.38 ± 5.95	325.25 ± 8.63	299.84 ± 8.50	326.50 ± 8.76
Mazda	-46.30 ± 2.63	-42.86 ± 7.36	-54.17 ± 5.05	-47.07 ± 7.27
Ant	19.63 ± 1.91	-939.50 ± 91.13	-1007.07 ± 31.16	7.39 ± 23.50
MM1	0.8644 ± 0.0130	0.8945 ± 0.0140	0.8770 ± 0.0164	0.8596 ± 0.0238
MM2	0.8469 ± 0.0163	0.8421 ± 0.0112	0.8397 ± 0.0070	0.8513 ± 0.0071
Chip (10^5)	5.550 ± 0.055	5.617 ± 0.083	5.552 ± 0.072	5.654 ± 0.113

F Structural-Evidence Intervention

The intervention removes the variable-sparsity, low-dimensional-subspace, and additivity entries from the explicit evidence package. Task cards, tools, opening designs, and the main prompt remain available. Backend-reported state is retained, so the intervention does not remove every possible source of structural information.

Figure 6 shows how the intervention changes backend use across tasks and seeds. Each row traces one run over the 100-evaluation budget; the upper and lower blocks respectively show the full-evidence and removal conditions. Withholding the three structural channels produces more concentrated strategy use on several tasks, most visibly through longer Vanilla BO episodes on MOPTA and Mazda. The supported conclusion concerns control behavior: backend diversity and switching are not direct measurements of hypothesis correctness. Evidence removal may also change initialization or earlier actions, so a later trajectory difference cannot be assigned to an isolated switch.

G Reliability of Structural Diagnostics

G.1 HDBO diagnostic state

PRISM derives its HDBO diagnostic state from the normalized observation matrix and objective values; no diagnostic spends an objective evaluation. The interface is tailored to three questions that are coupled in HDBO: whether the observed design covers the ambient space, whether a surrogate fitted with $n \ll D$ is credible enough to interpret, and which structural bias is worth testing next. Table 7 summarizes the quantities exposed to HERA.

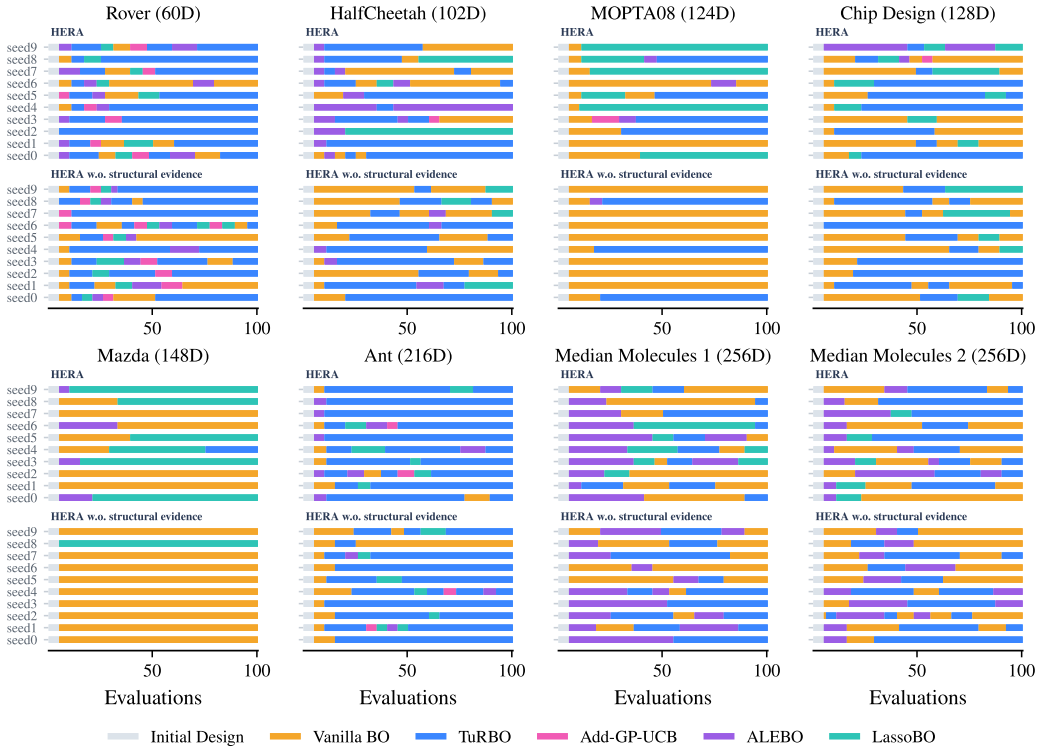

Figure 6: **Structural evidence changes strategy use.** Each row is one seed and each colored segment is a backend episode over the 100 evaluations. The upper block uses full evidence and the lower block withholds the three explicit structural evidence channels.

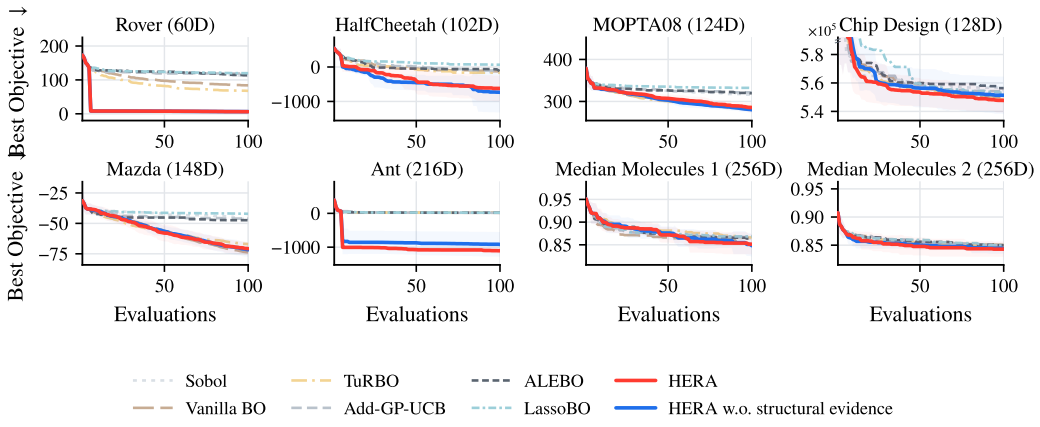

Figure 7: **Final performance with and without explicit structural diagnostics.** Mean best-so-far trajectories on the eight real-world tasks over ten seeds. The curves are difficult to distinguish on most tasks, consistent with the absence of a clear aggregate performance effect.

Table 7: **HDBO-specific diagnostic state exposed by `get_state`.** All quantities are derived signals for controlling search, not observations of the unknown objective structure.

Channel	Principal quantities
Exploration geometry	Normalized OTSD, recent OTSD gain, observation entropy, and their trajectories.
Surrogate credibility	Posterior standard-deviation summaries over Sobol probes; held-out RMSE, standardized residuals, $1\sigma/2\sigma$ coverage, and n/D.
Variable sparsity	LassoBO active-set size and fraction, selected coordinates, and a thresholded provisional verdict.
Low-dimensional subspaces	Surrogate-gradient energy dimension and its ratio to $\min(D, M)$ over $M = 64$ probe points.
Additive structure	Interaction-graph fragmentation, number and size of candidate groups, linked pairs, and strongest normalized mixed differences.

For geometry, let L_t denote the cheapest-insertion tour length through the first t normalized observations. PRISM reports

$$\widetilde{L}_t = \frac{L_t}{2\sqrt{5D}\,(1.5t)^{1-1/D}}, \tag{6}$$

as in (Papenmeier et al., 2025a). Observation entropy is the cumulative k-nearest-neighbor estimator used by the same exploration diagnostic. The two signals are complementary but purely geometric: a coordinate that does not affect the objective can still contribute to both distances and entropy.

Surrogate credibility is evaluated before structural readings are interpreted. PRISM fits the diagnostic GP with normalized inputs and standardized outcomes, summarizes its posterior standard deviation on 256 scrambled-Sobol probes, and performs leave-one-out-style validation. All observations are held out once when $n \leq 12$; otherwise at most 12 deterministically spaced folds are used. For a held-out observation, the standardized residual is

$$r_i = \frac{y_i - \widehat{\mu}_{-i}(x_i)}{\max\{\widehat{\sigma}_{-i}(x_i), 10^{-12}\}}. \tag{7}$$

The returned mean and standard deviation of r_i, empirical 1σ and 2σ coverage, and RMSE normalized by the observed response standard deviation expose bias, miscalibration, and poor generalization separately. The state also attaches n/D to every structural block so that a weak fit in the undersampled regime is not misread as evidence for or against a particular structure.

Figure 8 summarizes the three structural readings and the corresponding offline recovery metrics used in the audit.

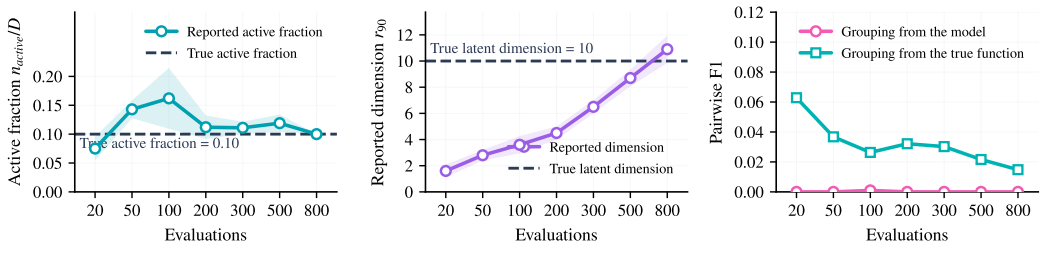

Figure 8: **Offline audit of structural diagnostics.** The panels report the sparsity reading, surrogate-gradient energy dimension, and interaction-group recovery metrics used to assess the diagnostics across observation budgets.

G.2 Sparsity and low-dimensional variation

Support recovery uses Jaccard overlap,

$$J(\widehat{S}, S) = \frac{|\widehat{S} \cap S|}{|\widehat{S} \cup S|}. \tag{8}$$

Given an estimated active set $\widehat{S}$ and, in the offline audit, a known support S, this score measures coordinate-support recovery. The active fraction is the size of the estimated support divided by the ambient dimension; it is a compact sparsity reading for strategy selection, not a guarantee that the selected coordinates are correct.

The low-dimensional subspace diagnostic follows gradient-based active-subspace analysis (Wycoff et al., 2021) and uses gradients of the surrogate posterior mean at probe locations to measure whether response variation is concentrated in a small number of directions. For $C = M^{-1} \sum_{j=1}^{M} g_j g_j^\top$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D$, its energy dimension is

$$d_{0.9} = \min \left\{ r : \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{D} \lambda_i} \geq 0.9 \right\}. \tag{9}$$

The principal configuration uses $M = 64$ probes. This statistic is an effective energy dimension of the surrogate gradient field and therefore a diagnostic for low-dimensional subspace structure, rather than an exact algebraic dimension or a recovery of the underlying subspace basis. Because it is computed from a fitted surrogate and a finite probe set, it should be interpreted together with the surrogate-credibility diagnostics.

G.3 Interaction grouping and fragmentation

The Differential Grouping-inspired diagnostic (Omidvar et al., 2014) computes mixed differences of the surrogate mean under coordinate perturbations, normalizes their magnitude, aggregates over anchors, and builds connected components from detected links. For component sizes $|G_1|, \ldots, |G_K|$, its reported score is

$$S_{\text{frag}} = \frac{D^2 - \sum_{j=1}^{K} |G_j|^2}{D(D-1)}. \tag{10}$$

This is the fraction of coordinate pairs placed in different components. It equals one for all singleton components and zero for one component. A high value can therefore reflect missed interactions, not additive structure.

H Metadata

The main metadata ablation uses a single *strict-removal* condition. It withholds task identity, semantic descriptions, reference-design hints, and the corresponding prompt instructions while preserving the optimization budget and interface. This intervention is broader than simply deleting one semantic card and is the condition reported in the main text.

As shown in Figure 9, strict removal generally worsens performance on several tasks, with the largest effects concentrated in particular tasks. The gains and reversals support task dependence rather than a uniform semantic advantage.

I Agent Model Configurations

We replace DeepSeek-V4.1-Flash with Kimi-K3 and GPT-6-astra on three tasks, with ten completed seeds per configuration and the same interface and evaluation budget. Kimi's required temperature is one rather than zero, and GPT uses a separate relay. This evaluates deployed configurations rather than an isolated scalar notion of model capability.

Kimi has the best mean on HalfCheetah, whereas GPT has the best means on MOPTA and Mazda. Median ordering differs: DeepSeek's HalfCheetah median is better than Kimi's. Substantial seed variation prevents a uniform capability ranking. Model changes also shift backend usage, but behavior alone does not explain the final-objective changes.

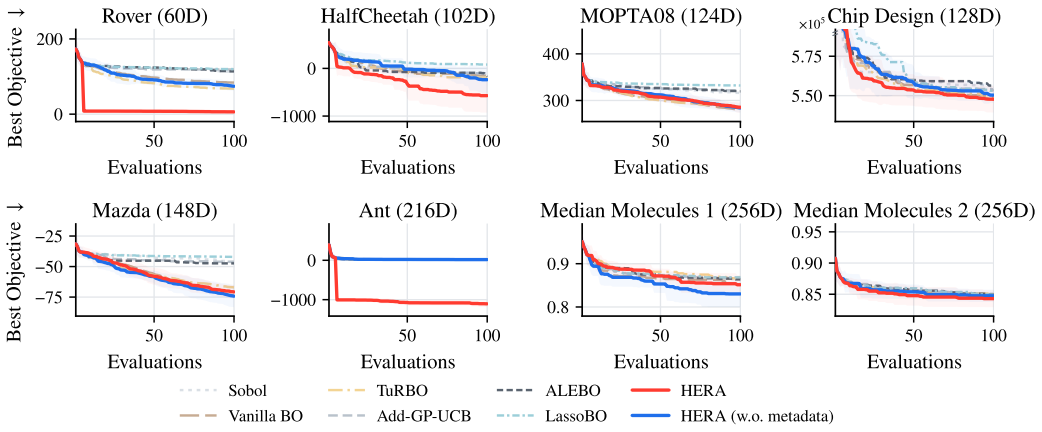

Figure 9: **Metadata affects initialization and search.** Full-context and strict-removal curve. Effects are heterogeneous and do not isolate metadata-driven routing.

Table 8: Agent models: mean final objective ± sample SD over ten seeds. The best mean among models differs by task.

Model	HalfCheetah	MOPTA	Mazda
DeepSeek-V4.1-Flash	-621.4 ± 370.8	285.57 ± 11.58	-70.73 ± 5.42
Kimi-K3	-706.0 ± 485.1	285.20 ± 9.27	-74.51 ± 5.54
GPT-6-astra	-323.0 ± 474.2	281.72 ± 8.78	-75.36 ± 7.23

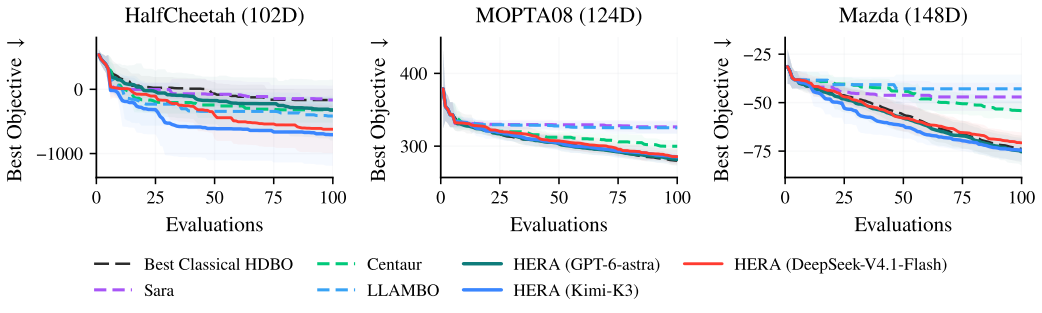

Figure 10: **Model sensitivity.** Ten-seed trajectories for three agent configurations.

J TOOL-USE TRAJECTORIES

The tool-use trajectories show a common initialization pattern followed by task- and seed-dependent control: some tasks trigger frequent strategy interventions, whereas others sustain a stable backend over extended search episodes (Figure 11). This pattern is consistent with adaptive outer-loop control, but the trajectories alone do not establish hypothesis correctness or a causal benefit from more frequent tool use.

Tool calls, assistant responses, and provider requests are distinct accounting levels, so their timelines should be read as descriptions of intervention timing rather than evidence that a hypothesis was correct or that more frequent tool use caused better outcomes.

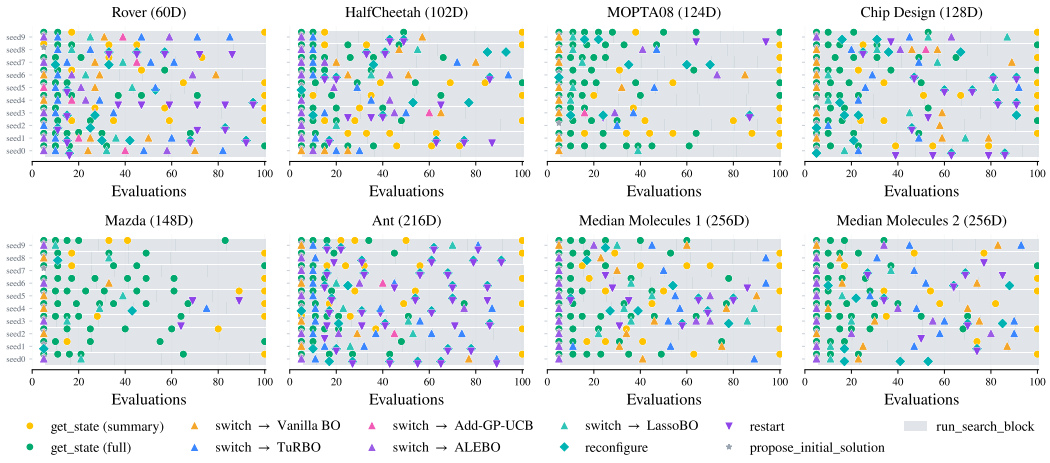

Figure 11: **Tool use over the evaluation budget.** One row per campaign, grouped by task. Markers identify queries and strategy interventions; bands show evaluations consumed by blocks. A common opening protocol is followed by task- and seed-dependent behavior.

K BLOCK SIZE AND CONTEXT COMPACTION

The experiment compares adaptive decision blocks with and without context compaction, together with block-size-one execution under compaction. An additional uncompacted block-size-one arm provides an exploratory matched-subset comparison. Compaction keeps recent tool results detailed, summarizes older results, and preserves assistant reasoning; block size one returns after each evaluation.

Figure 12 shows that context compaction substantially reduces inference cost while preserving broadly comparable optimization behavior. More frequent decisions can improve individual task trajectories, but their additional cost and the limited exploratory comparison do not support a universally best decision interval. The result is therefore a task-dependent trade-off between control frequency and inference efficiency.

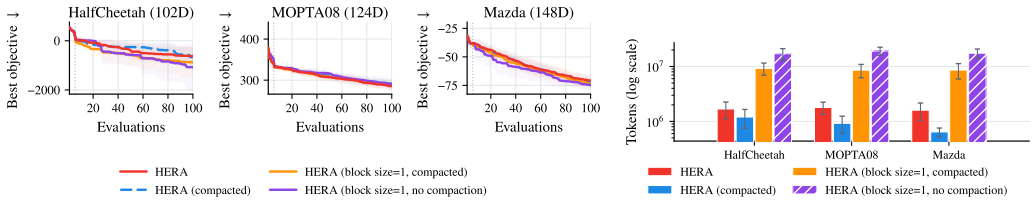

Figure 12: **Performance and inference cost under different decision frequencies.** Adaptive blocks and block-size-one variants are compared with and without context compaction.

L TOKEN ACCOUNTING

The cost study covers 320 complete campaigns: four LLM methods, eight tasks, and ten seeds, each with 100 evaluations. Total tokens are prompt plus completion tokens; prompt tokens are split into provider-reported cache hits and misses. HERA has the highest total-token usage despite fewer requests. Its repeated context gives a prompt-cache hit rate of 94.0%, versus 16.4% for LLAMBO, 27.3% for Centaur, and 36.2% for Sara. Rounded component sums can differ slightly from rounded totals. Figure 13 shows the per-task token totals together with the cached-input, uncached-input, and output components, illustrating that total-token efficiency, request frequency, and billed cost are distinct quantities.

Monetary cost requires prices in consistent token units:

$$C = p_{\text{hit}} T_{\text{hit}} + p_{\text{miss}} T_{\text{miss}} + p_{\text{out}} T_{\text{out}}. \tag{11}$$

Cache hits are still tokens and may incur cost. Neither total tokens nor uncached tokens alone establish which method is cheapest. Counts also exclude numerical computation: fixed HDBO uses no LLM tokens but incurs model-fitting and acquisition-search costs.

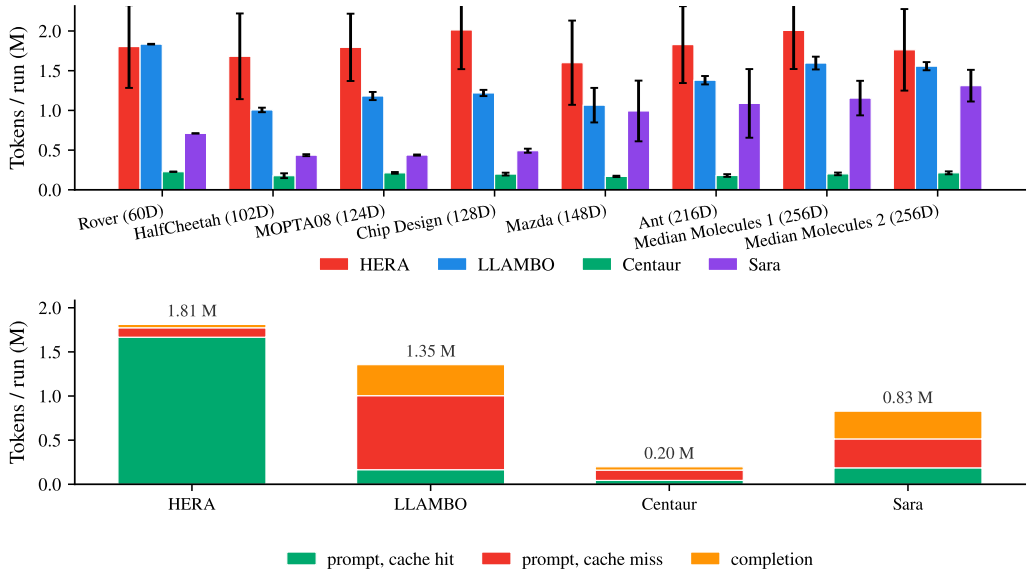

Figure 13: **Tokens and cache decomposition.** Per-task totals and cached-input, uncached-input, and output components. Total-token efficiency, request frequency, and billed cost are distinct quantities.