DO LANGUAGE MODELS PROVIDE USEFUL PRIORS FOR AUTONOMOUS SCIENTIFIC SEARCH? A CALIBRATION STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models (LLMs) are increasingly positioned as potential autonomous scientific agents, a central open question is whether they can reliably generate hypotheses that drive iterative discovery. We study this question in a minimal autonomous search loop where a model proposes candidate solutions, receives scalar objective feedback, and iteratively refines proposals. Using continuous black-box optimization as a controlled proxy for scientific search, we compare random search, Tree-structured Parzen Estimator (TPE), LLM-driven proposal generation, and a hybrid TPE+LLM scheme under equal evaluation budgets. Across five independent seeds on a shifted ellipsoid benchmark, we find that LLM-only search performs better than random sampling but substantially worse than TPE, while hybridization achieves the best mean final performance. However, both LLM and hybrid methods exhibit high variance across seeds, indicating limited reliability. These results suggest that current LLMs do not encode sufficiently strong inductive biases for autonomous discovery and must be coupled with explicit optimization machinery in post-AGI scientific systems.

1 INTRODUCTION

Recent discourse around artificial general intelligence (AGI) increasingly assumes that future systems may function as autonomous scientists, generating hypotheses, designing experiments, and discovering laws with minimal human intervention (Bengio, 2023; Lake et al., 2017). This raises foundational questions about how scientific processes might evolve in a post-AGI world: Will machines generate discoveries for humans to validate, or will humans remain responsible for formulating and constraining the underlying search processes?

A natural subproblem is whether present-day large language models (LLMs) already possess useful inductive biases for iterative discovery. LLMs demonstrate strong performance on symbolic reasoning, planning in text, and code synthesis (Yao et al., 2023; Chen et al., 2021), motivating speculation that they may also guide scientific search. However, scientific discovery frequently requires navigating continuous, high-dimensional spaces using scalar feedback, a setting far removed from the discrete token distributions on which LLMs are trained.

In this work, we empirically test a minimal capability prerequisite for autonomous discovery: can an LLM serve as a reliable proposal generator inside an iterative search loop? We instantiate a simple closed-loop system where a model proposes candidate solutions, receives objective values, and proposes new candidates based on history. Using continuous optimization benchmarks as a proxy for scientific search spaces, we compare LLM-driven search against classical Bayesian optimization and random baselines.

Our goal is not to build a state-of-the-art optimizer, but to calibrate expectations about what language-model priors currently provide in discovery-like settings.

2 Related Work

Bayesian Optimization. Bayesian optimization methods such as TPE model the distribution of promising regions in a search space and have demonstrated strong performance on expensive black-box optimization problems (Bergstra et al., 2011).

LLMs for Tool Use and Planning. Recent work explores LLMs as planners, code generators, and controllers (Yao et al., 2023). However, most evaluations focus on symbolic or discrete domains rather than continuous scalar-feedback optimization.

Automated Scientific Discovery. Systems for symbolic regression, theorem proving, and program synthesis demonstrate that structured inductive biases remain crucial for discovery (Schmidt & Lipson, 2009; Polu & Sutskever, 2020).

3 Minimal Autonomous Search Loop

We consider a loop consisting of:

1. A proposal generator produces candidate vectors x.
2. An objective function evaluates $f(x)$.
3. The history of $(x, f(x))$ pairs is provided back to the proposal generator.

We study four instantiations:

- **Random**: Uniform sampling.
- **TPE**: Optuna's Tree-structured Parzen Estimator.
- **LLM-only**: LLM proposes candidates conditioned on history.
- **Hybrid**: TPE candidate pool augmented with LLM proposals.

This loop represents a minimal abstraction of hypothesis generation with scalar feedback.

4 Objective Function

We use the shifted ellipsoid function:

$$f(x) = \sum_{i=1}^{D} w_i (x_i - c_i)^2, \quad w_i = 10^{6(i-1)/(D-1)}$$

where c is a randomly sampled shift vector and $D = 12$. Search space bounds are $x_i \in [-50, 50]$ integers.

This function exhibits strong anisotropy and is widely used to test optimization algorithms.

5 LLM Proposal Mechanism

At each iteration, the LLM receives the best-so-far history:

$$[x^{(1)} \to f^{(1)}, \ldots, x^{(k)} \to f^{(k)}]$$

The prompt instructs the model to infer which coordinate changes reduce the objective and propose $K = 8$ new candidate vectors, with half exploiting near the best solution and half exploring wider perturbations. The model outputs a JSON list of vectors.

We use `gpt-4.1-mini` with temperature 0.2. If parsing fails or the API errors, the system falls back to random sampling.

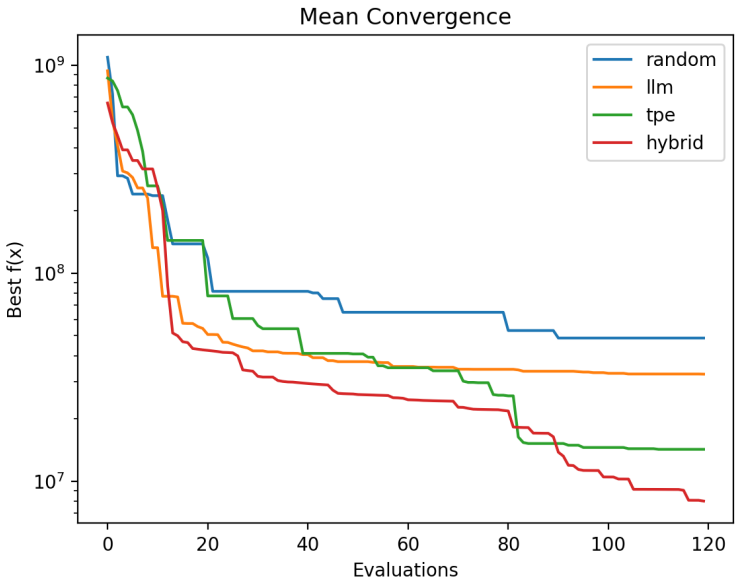

Figure 1: Mean best-so-far objective across 5 seeds (log scale).

6 Hybrid Method

At each iteration:

1. Sample 64 candidate vectors using TPE.
2. Query LLM for additional candidates.
3. Evaluate all candidates.
4. Select best candidate and feed its value back to TPE.

This produces a mixture proposal distribution combining density-based sampling and LLM heuristics.

7 Experimental Setup

- Dimension: $D = 12$
- Budget: 120 evaluations
- Seeds: 5
- Identical evaluation budgets across methods
- Independent random shift vector per seed

We record best-so-far objective value at each evaluation.

8 Results

Figure 1 shows mean convergence behavior.

- LLM-only search improves over random sampling in 60% of seeds.
- TPE outperforms LLM-only in 80% of seeds.
- Hybrid outperforms TPE in 60% of seeds.

Figure 2 shows mean $\pm$ standard deviation across seeds.

Table 1: Final best objective across 5 seeds (mean ± std).

Method	Mean Final $f(x)$	Std
Random	4.87×10^7	1.44×10^7
LLM	3.27×10^7	1.67×10^7
TPE	1.42×10^7	5.22×10^6
Hybrid	$\mathbf{8.02 \times 10^6}$	7.20×10^6

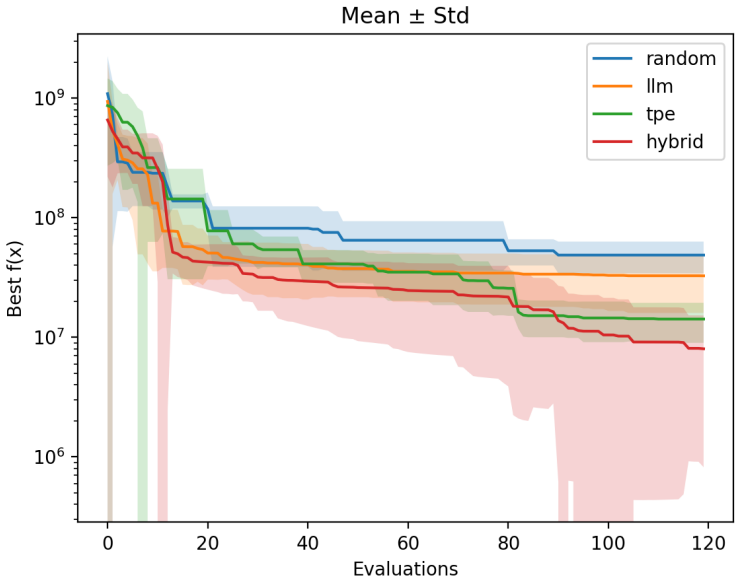

Figure 2: Mean ± standard deviation across seeds. LLM-only and Hybrid exhibit substantially higher variance than TPE.

9 Discussion

The observed ordering—Hybrid < TPE < LLM < Random—indicates that while LLMs can extract weak regularities from past evaluations, they do not internalize the geometric structure required for reliable continuous optimization. Classical Bayesian optimization remains substantially more sample-efficient.

From a post-AGI perspective, these findings caution against assuming that scaling language models alone will yield autonomous scientific agents.

Hybridization partially mitigates this limitation by combining algorithmic search with heuristic language priors, but the resulting system remains high-variance, suggesting that principled integration strategies are necessary.

10 Conclusion

We present a calibration study of LLMs as proposal generators in a minimal autonomous discovery loop. LLM-only search improves over random sampling but remains substantially weaker than classical Bayesian optimization. Hybrid TPE+LLM achieves the best mean performance but remains noisy. These results suggest that current LLMs lack sufficient inductive bias for autonomous scientific discovery and must be embedded within algorithmic search frameworks. Our findings provide grounded evidence for limits of reasoning in present models and inform the design of resilient post-AGI scientific systems.

REFERENCES

Yoshua Bengio. Deep learning for ai. *Communications of the ACM*, 64(7):58–65, 2023.

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

Mark Chen et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Brenden Lake, Tomer Ullman, Joshua Tenenbaum, and Samuel Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 2017.

Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 2009.

Shunyu Yao, Jeffrey Zhao, et al. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2023.