Local Thompson Sampling via Prompting for Bayesian Optimization with LLM Generators

Anonymous Authors

Abstract

Steering generative models to propose candidates that optimize expensive black-box functions is a central problem with applications in scientific discovery. Recent approaches integrate Bayesian optimization (BO) with large language models (LLMs) to guide candidate generation, but often require complex architectures or tight coupling between surrogate models and generators. We propose *Local Thompson Prompting* (LTP), a simple and modular alternative that leverages a Gaussian process (GP) surrogate to perform Thompson sampling over previously evaluated candidates, and uses the resulting samples to directly condition LLM prompts. This yields a lightweight mechanism for uncertainty-aware exploration without modifying the LLM or training additional components. We evaluate LTP on two domains—Semantle (word-guessing) and molecular optimization—and show that it consistently outperforms existing BO-LLM baselines while remaining architecturally minimal. These results suggest that effective BO-style exploration can be achieved through prompt-level interventions alone.

1. Introduction

Many scientific discovery problems require optimizing expensive black-box functions over discrete, structured spaces. Drug molecule design is a canonical example: given a protein target, the goal is to find a molecule with high binding affinity and drug-likeness, where each evaluation may require computationally expensive molecular docking. The challenge is that the search space is combinatorially large, evaluations are costly, and optimal candidates often lie far outside the observed data.

Large language models (LLMs) have emerged as powerful priors over structured discrete spaces such as SMILES strings for molecules. They can generate diverse, syntactically valid candidates and leverage in-context learning to adapt their outputs based on provided examples. This makes them attractive as generators in a Bayesian optimization (BO) loop, where they can be guided to produce desirable candidates. However, existing approaches either forgo principled uncertainty estimation Yang et al. (2023), or optimize an acquisition function in a continuous latent space and decode proposals back to the input space Agarwal et al. (2025), introducing a gap between where the surrogate model operates and where candidates are evaluated.

We ask: `Can we leverage in-context learning to steer the LLM?`

Our approach, Local Thompson Prompting (LTP), uses a Gaussian process (GP) surrogate purely as a scoring function via Thompson sampling. The resulting samples are used to guide the selection of in-context exemplars, which steer the LLM toward promising regions of the search space. An inner refinement loop further enables local exploration. This eliminates the continuous-to-discrete decoding bottleneck while retaining the GP's ability to guide generation through uncertainty.

Our contributions are as follows:

- We propose a latent optimization-free framework for Bayesian optimization with LLMs, where optimization is performed by steering the model via in-context prompts rather than optimizing over inputs or latent representations.
- We introduce Local Thompson Prompting (LTP), which uses Thompson sampling to construct prompts that induce an uncertainty-aware proposal distribution over candidates directly in the input space.
- We demonstrate that this approach achieves strong empirical performance on discrete optimization tasks.

2. Background

Bayesian optimization. BO is a sample-efficient approach for optimizing expensive black-box functions $f : \mathcal{X} \to \mathbb{R}$. A GP surrogate $p(f|\mathcal{D}_t)$ is fit to evaluated observations $\mathcal{D}_t = \{(x_i, f(x_i))\}_{i=1}^t$, and an acquisition function is maximised to select the next candidate. For example, in our MolOpt setting, $\mathcal{X}$ is the space of SMILES strings, and f is a scalarised combination of binding affinity (Vina score) and drug-likeness (QED).

Thompson Sampling. Rather than optimizing an acquisition function, Thompson sampling draws a single function sample $\tilde{f} \sim p_t(f|\mathcal{D}_t)$ and selects candidates that maximise $\tilde{f}$. This provides natural exploration-exploitation balance and is particularly well-suited to batch selection.

Table 1. Comparison of ICL-based, training-free LLM optimization methods

Method	Surrogate	Generator	Latent BO?	Train-free?	Domain
OPRO	None (greedy)	LLM	×	✓	General
LLAMBO	LLM-as-surrogate	LLM	×	✓	Tabular
BoPRO	GP (input space)	LLM	✓	✓	General
LTP	**GP (input space)**	**LLM**	×	✓	**General**

3. Related Work

We focus our comparison on in-context learning (ICL)-based, training-free methods that use LLMs as generators over structured discrete spaces. Refer Table 1 for a comparison

OPRO Yang et al. (2023). OPRO frames optimization as iterative prompting: the LLM is provided with previously evaluated $(x, f(x))$ pairs and proposes new candidates at each step. It requires no surrogate and no fine-tuning, but relies on a greedy top-k selection policy without an uncertainty model. As a result, it tends to stall on harder problems and fails to explore unseen regions of the search space.

LLAMBO Liu et al. (2024). LLAMBO replaces the GP surrogate with the LLM itself, which both predicts objective values and generates candidates conditioned on a target score. Uncertainty is approximated via Monte Carlo prompt permutation—a heuristic that improves with more observations but remains uncalibrated relative to a GP posterior. LLAMBO excels in early search when the LLM's pretrained priors are informative, but its surrogate quality degrades relative to GPs as data accumulates. It was designed and validated exclusively on hyperparameter tuning over tabular spaces.

BoPRO Agarwal et al. (2025). BoPRO extends OPRO with a GP fitted in a continuous embedding space $\mathcal{Z}$. At each step, the acquisition function is maximized in $\mathcal{Z}$ to obtain a latent proposal, and the k nearest observed candidates in $\mathcal{Z}$ are retrieved via KNN and used as in-context examples. This provides principled, uncertainty-driven exploration, but the KNN retrieval serves as a heuristic workaround for the ill-posed problem of decoding a latent proposal back to the input space $\mathcal{X}$.

4. Method: Local Thompson Prompting

Our method, Local Thompson Prompting (LTP), replaces latent-space optimization with direct GP-guided exemplar selection. The key insight is that the GP posterior can be used to rank evaluated molecules, and these rankings can be communicated to the LLM through in-context exemplars.

Algorithm. Algorithm 1 provides a formal description of LTP.

The inner refinement loop over $r = 1, \ldots, R$ serves a dual exploratory role. At $r = 1$, the exemplar pool is initialized as $\mathcal{P}_1 = \mathcal{D}_t$, anchoring generation in the neighbourhood of the best previously evaluated candidates. For $r > 1$, the pool transitions to $\mathcal{P}r = \bigcup i = 1^{r-1}\mathcal{G}_i$, shifting from global to local exploration by conditioning the LLM on candidates generated in earlier rounds.

At each round, m exemplars $\mathcal{E}_r \subset \mathcal{P}_r$ are selected by ranking under $\tilde{f}$. The LLM then generates n candidates $\mathcal{G}_r = \texttt{LLMGenerate}(\mathcal{E}_r, n)$, which are accumulated into $\mathcal{C}_t$. After R rounds, the top-k candidates $\mathcal{B}_t \subset \mathcal{C}t$ are evaluated using the true oracle f and appended to $\mathcal{D}t + 1$.

The Thompson sample $\tilde{f} \sim p_t(f \mid \mathcal{D}_t)$ is drawn once per outer iteration and held fixed across all inner rounds via the Matheron sampling rule, ensuring consistent ranking throughout the refinement trajectory.

A key architectural difference from BOPRO is that the GP acts as a *ranker* of candidates rather than a *proposal generator*. As a result, GP miscalibration directly affects which molecules are selected as exemplars for the LLM (see Section 6).

Prompt Design. The LLM prompt includes: (1) a task description specifying the objective(s) and their relative importance (if multiple objectives are present), (2) task-specific context when required (e.g., the target protein name, to leverage the LLM's chemical knowledge), and (3) the top-k exemplars. Scores are not shown explicitly—the LLM instead relies on structural patterns in the exemplars rather than numerical interpolation.

5. Experiments

Setup. We use `Mistral-large-2407` (via AWS Bedrock) as the LLM generator. The GP uses a Matern kernel with hyperparameters fit via marginal log-likelihood on the warmstart data and refit at each iteration. We compare against BOPRO and OPRO.

Task 1: Semantle. Semantle is a word-guessing game where the black-box function returns the cosine similarity between a guessed word and the hidden target word. We use `gte-qwen-2-1.5b-instruct` as the encoder model ϕ for generating the embeddings. We evaluate on 10 hidden words with a budget of 400 black-box evaluations for each word. Results are reported as average best similarity score across 5 seeds.

Task 2: MolOpt. MolOpt requires finding drug molecules that optimize a scalarised objective $f(x) = 0.8 \cdot \text{Vina}(x) + 0.2 \cdot \text{QED}(x)$ for a given protein target x, following Agarwal et al. (2025). We use `Molformer` as the encoder model ϕ for generating the embeddings. We evaluate on 10 protein targets from the Dockstring benchmark with a budget of 200 black-box evaluations per protein.

Algorithm 1 Local Thompson Prompting (LTP)

Require: Initial dataset $\mathcal{D}_0$, evaluation budget T, batch size k, refinement rounds R, exemplars per round m, candidates per round n

for $t = 0, 1, \ldots, T-1$ **do**
  $\tilde{f} \sim p_t(f \mid \mathcal{D}_t)$
  $\mathcal{C}_t \leftarrow \emptyset$
  for $r = 1, \ldots, R$ **do**
    if $r = 1$ **then**
      $\mathcal{P}_r \leftarrow \mathcal{D}_t$
    else
      $\mathcal{P}_r \leftarrow \bigcup_{i=1}^{r-1} \mathcal{G}_i$
    end if
    Select m exemplars $\mathcal{E}_r \subset \mathcal{P}_r$
    $\mathcal{G}_r \leftarrow \mathrm{LLM}(\mathcal{E}_r, n)$
    $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \mathcal{G}_r$
  end for
  Select top-k candidates $\mathcal{B}_t \subset \mathcal{C}_t$
  Evaluate $\{f(x)\}_{x \in \mathcal{B}_t}$; $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x, f(x))\}_{x \in \mathcal{B}_t}$
  Update GP posterior $p_{t+1}(f \mid \mathcal{D}_{t+1})$
end for
return $\arg\max_{(x,y) \in \mathcal{D}_T} y$

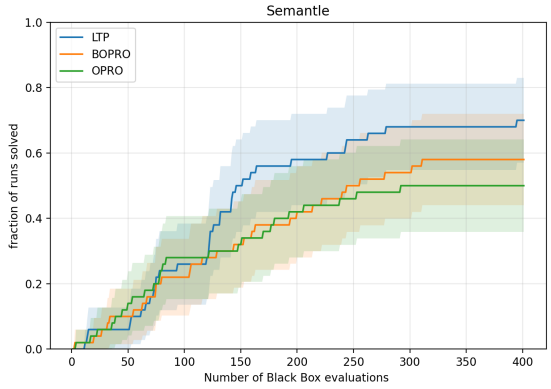

Figure 1. Our method LTP outperforms both BOPRO and OPRO across the full evaluation budget of 50 runs.

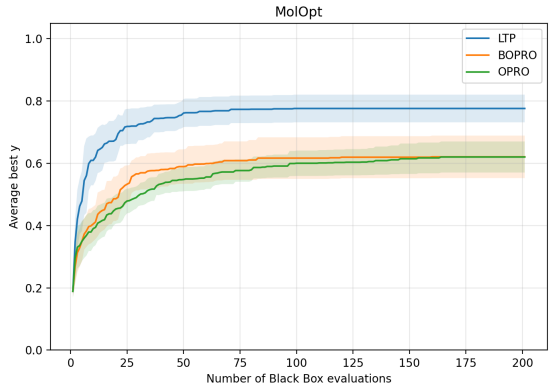

Figure 2. LTP substantially outperforms both BOPRO and OPRO.

Results. On Semantle (Figure 1), our method is able to solve $\approx 70\%$ of runs on average compared to BOPRO's $\approx 58\%$ and OPRO's $\approx 50\%$.

On MolOpt (Figure 2), LTP achieves a scalarised objective value of ≈ 0.78 at 200 evaluations compared to BOPRO's ≈ 0.63 and OPRO's ≈ 0.62. The consistent gap throughout the run suggests our GP-guided exemplar selection provides a stronger signal than BOPRO's latent-space approach for this task.

6. Discussion and Future Work

We have shown that adaptive generation with LLMs can be achieved using GP-guided exemplar selection, yielding a simpler architecture that remains competitive. At the same time, our analysis

indicates that removing the latent-space buffer increases sensitivity to the quality of the GP, making the following directions important.

Constrained BO. We can integrate a classifier to enforce validity constraints and avoid selecting invalid candidates for black-box evaluation (e.g., in the MolOpt task).

Representation learning. Instead of relying on pre-trained embedding models, we can learn candidate representations (e.g., using deep kernel GPs).

Alternative generative models. Our framework is not specific to LLMs; diffusion models or flow-based models could replace the LLM as the generator.

7. Conclusion

We propose LTP, a simple method for performing Bayesian optimization with LLM-based generators, and demonstrate strong performance on both Semantle and MolOpt. Our analysis highlights the role of in-context examples as anchor points for steering generation, suggesting that prompt construction is a key lever for controlling the exploration–exploitation tradeoff. These findings motivate future work on constraint-aware prompting, more principled exploration strategies, and improved representations for guiding generation.

References

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 Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=aVfDrl7xDV.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Miguel García-Ortegón, Gregor N. C. Simm, Austin J. Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. Dockstring: Easy molecular docking yields better benchmarks for ligand design, 2021.

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context?, 2024.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=OOxotBmGol.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models. *arXiv*, 2024.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties, 2022. URL https://arxiv.org/abs/2106.09553.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*, 2023. URL https://arxiv.org/pdf/2309.03409.