CALIBRATING LLM GUIDANCE FOR ACQUISITION SELECTION IN BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian optimization (BO) is a sample-efficient approach to black-box optimization, but its performance often depends on task-specific design choices, including acquisition functions and trust region methods. Identifying effective configurations often requires human expertise or substantial manual experimentation. Large language models (LLMs) offer a promising approach to automatically recommend these choices using task descriptions and optimization histories. However, it is reportedly difficult for LLMs to track long interaction histories efficiently, which can undermine the reliability of their recommendations for complex tasks. We propose LLM-Prior-Informed Gaussian Upper Confidence Bound (LPG-UCB), a framework that combines LLM guidance with empirical feedback to adaptively select acquisition functions. LPG-UCB encodes LLM recommendations as priors over acquisition function preferences and updates these preferences through Bayesian inference as the BO loop proceeds. This Bayesian update mechanism leverages summary statistics from the BO trajectory to refine and calibrate LLM guidance, reducing reliance on the accuracy of individual LLM recommendations. We model LPG-UCB as a non-stationary rotting-bandit framework, and derive an expected cumulative regret bound, which grows sublinearly under appropriate conditions. Extensive experiments on diverse synthetic benchmarks demonstrate faster convergence than competing LLM-based selection methods. LPG-UCB also achieves competitive performance on a high-dimensional black-box optimization problem.

1 INTRODUCTION

Bayesian optimization (BO) provides a principled framework for scientific optimization, enabling sample-efficient search over expensive black-box functions whose underlying structure is unknown a priori. In spite of its popularity, the performance of BO is often sensitive to the selection of optimization strategies, including acquisition functions (AFs), surrogate models, and global–local search mechanisms (Hoffman et al., 2011; Lu et al., 2023; Van Dieren et al., 2026). Because effective choices depend on problem characteristics, practitioners often configure these strategies through repeated trial and error. This manual tuning introduces additional costs and runs counter to BO's goal of automating expensive black-box optimization (Vasconcelos et al., 2019; 2022; Ngo et al., 2026). This raises a natural question: *Can BO strategy selection be automated as part of the optimization process?*

Large language models (LLMs) offer a promising approach to this problem through their ability to interpret problem context and reason over interaction histories. Recent studies have explored LLMs as optimization agents that adapt optimization strategies, propose query points, or generate AFs (Liu et al., 2024; Aglietti et al., 2024; Ngo et al., 2026). These studies suggest that LLMs can combine contextual information with feedback from previous evaluations to guide BO strategy selection as optimization progresses.

A critical limitation, however, lies in the reliability of LLMs. It has been reported that LLMs can produce inconsistent responses across repeated generations and factual or reasoning errors, including hallucinations (Sahoo et al., 2024; Arana et al., 2026). Such errors are particularly consequential in sequential optimization, where each decision influences subsequent observations and the resulting optimization trajectory (Ranković et al., 2026). This leads to a second question:

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.

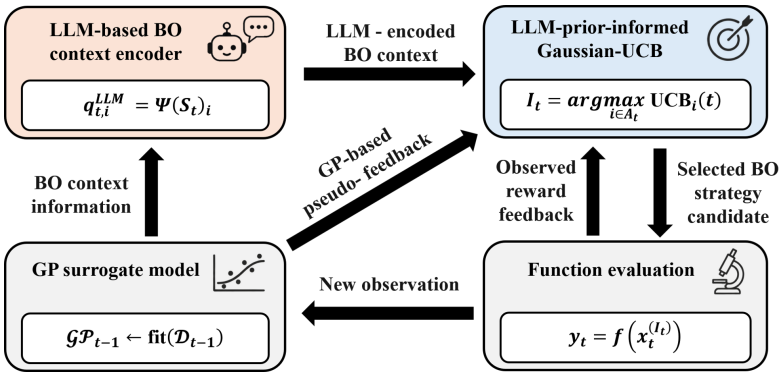


Figure 1: Overall workflow of the proposed LPG-UCB framework.

Can we leverage the ability of LLMs to extract information from complex optimization context while limiting the impact of their errors on strategy selection?

To this end, we propose a Bayesian decision-making framework that encodes LLM recommendations as priors over strategy preferences and updates these priors using observed optimization outcomes. We define the reward for selecting a strategy by comparing the function value at its recommended query point with the best value observed before that evaluation. Combining Bayesian inference with a Gaussian bandit model and an upper confidence bound selection rule, we develop LLM-Prior-Informed Gaussian Upper Confidence Bound (LPG-UCB). The overall workflow of LPG-UCB is illustrated in Figure 1.

The main contributions of this work are summarized as follows:

- We propose LPG-UCB, which leverages Gaussian-UCB to integrate LLM-generated BO strategy preferences as prior guidance with numerical reward feedback for principled strategy selection.
- We formulate LPG-UCB under a non-stationary rotting-bandit model and develop a regret analysis that jointly accounts for statistical uncertainty, non-stationary reward decay, and errors in the LLM-generated prior, establishing conditions for sublinear expected cumulative regret.
- LPG-UCB demonstrates state-of-the-art performance across 31 synthetic benchmark problems and further shows strong performance on a high-dimensional real-world optimization problem.

2 Background

In this section, we will introduce the background of Bayesian optimization and discuss the related literature.

2.1 Bayesian Optimization

BO offers a surrogate-model-based approach to scientific optimization problems where high evaluation costs limit the number of performance evaluations. Rather than searching the design space exhaustively, BO constructs a probabilistic surrogate from previously observed input–output pairs and uses an acquisition function (AF) to guide the selection of the next candidate point (Frazier, 2018).

A Gaussian process (GP) is a commonly used surrogate model that defines a probability distribution over functions through a mean function $m(\mathbf{x})$ and a covariance function, or kernel, $k(\mathbf{x}, \mathbf{x}')$, which characterizes the statistical dependence between function values at different input locations. Conditioning the GP on the observed dataset $\mathcal{D}_{t-1} = \{(\mathbf{x}_s, y_s)\}_{s=1}^{t-1}$ yields a posterior predictive mean $\mu_{t-1}^{\mathrm{GP}}(\mathbf{x})$ and standard deviation $\sigma_{t-1}^{\mathrm{GP}}(\mathbf{x})$, providing both a prediction of the objective function and a measure of predictive uncertainty. An AF uses the GP posterior to quantify the utility

of evaluating a new candidate point. For a minimization problem, the next query point is typically selected by maximizing the AF, $\mathbf{x} \in \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}; \mathcal{D}_{t-1})$, where α denotes the AF. Different AFs employ distinct criteria for candidate selection, reflecting varying assumptions and priorities in the optimization process. For example, EI and PI tend to favor exploitation, whereas TS and UCB provide stronger exploration mechanisms (Jones et al., 1998; Srinivas et al., 2010).

2.2 Adaptive BO Strategy Selection

BO comprises several configurable components, including the GP surrogate model, AF, input-space transformation, and search regime, for which the appropriate strategy may vary across optimization problems. Numerous studies have sought to improve BO performance by developing and improving such individual components. This has motivated adaptive approaches that enable strategy selection at each optimization iteration. In this context, a prominent line of research has focused on adaptive AF selection. GP-Hedge (Hoffman et al., 2011) uses an adversarial bandit algorithm to allocate selection probabilities among AFs. No-PASt-BO (Vasconcelos et al., 2019) introduces a memory factor and reward normalization to reduce the influence of distant past rewards, while SETUP-BO (Vasconcelos et al., 2022) automatically adapts portfolio hyperparameters using Thompson sampling and conjugate Bayesian updates. However, these approaches often do not explicitly incorporate optimization-state information into the selection process.

2.3 Non-Stationary Multi-Armed Bandits

Multi-armed bandits (MABs) provide a sequential decision-making framework under uncertainty, in which a decision-maker selects an arm, whose associated action produces a reward. The received reward serves as feedback for subsequent arm selection, with the objective of maximizing cumulative reward. The MAB framework provides a principled approach to adaptive strategy selection while enabling theoretical performance characterization through regret bounds.

Classical stochastic MABs assume stationary arm-reward distributions. In BO strategy selection, however, each new objective evaluation updates the GP posterior and changes the optimization state, causing the reward distribution associated with each strategy to evolve across optimization iterations. GP-Hedge (Hoffman et al., 2011) avoids explicitly modeling this history-dependent selector reward process by adopting an adversarial-style bandit analysis, at the cost of generally more conservative regret bounds. The resulting history-dependent reward process is difficult to characterize within the classical stationary MAB framework, motivating a non-stationary bandit formulation for the theoretical analysis of BO strategy selection.

Non-stationary MABs have been studied under several structural models of reward evolution. A basic distinction is between rested and restless bandits (Whittle, 1988; Seznec et al., 2020). In rested bandits, the reward process of an arm changes only when that arm is selected, whereas in restless bandits, the reward process may evolve regardless of whether the arm is selected. More specific formulations include switching or piecewise-stationary bandits, which assume that reward distributions change at a limited number of change points (Garivier & Moulines, 2011), and variation-budget approaches, which constrain the total amount of reward variation over the decision horizon (Besbes et al., 2014). Rotting and rising bandits impose directional structure on reward evolution, with expected rewards decreasing or increasing according to the evolution of each arm (Levine et al., 2017; Li et al., 2020; Genalti et al., 2024). These structural assumptions enable regret analysis in settings where the stationary MAB assumption no longer holds.

2.4 Large Language model In Bayesian Optimization

Leveraging the ability of LLMs to interpret heterogeneous optimization context, incorporate prior knowledge, and reason over sequential feedback, several recent studies have explored integrating LLMs into the BO loop. LLAMBO (Liu et al., 2024) integrates LLM capabilities into multiple stages of the BO pipeline, including warm-start initialization, surrogate modeling, and candidate generation, allowing language-model outputs to directly influence both the construction of the optimization model and the selection of evaluation points. FunBO (Aglietti et al., 2024) uses an LLM to generate and iteratively refine new acquisition functions as executable code, guided by their performance on training optimization problems. LMABO (Ngo et al., 2026) instead employs an LLM

as an online optimization strategist that interprets the current BO state and adaptively selects an acquisition function at each iteration. LABO (Chen et al., 2026) adopts a multi-fidelity perspective, treating LLM predictions as inexpensive low-fidelity evaluations and using them to guide the selective allocation of costly high-fidelity objective evaluations.

Despite these advances, recent studies have raised concerns about the reliability of directly using LLMs as optimization agents. Several recent studies have reported limitations of current LLMs in scientific optimization, including unreliable decision-making and insufficient uncertainty calibration (Gupta et al., 2025; Ranković et al., 2026). Moreover, LLM-based BO performance can depend strongly on the underlying model, with smaller models exhibiting less stable behavior. (Rychert et al., 2025). Beyond BO, limitations in numerical reasoning and fine-grained quantitative decision-making have also been reported in structured optimization tasks (Song et al., 2026), with such challenges becoming more pronounced for lower-capacity models (Zhang et al., 2026). These findings highlight the risks of directly using unrefined LLM outputs for sequential optimization decisions, as their reliability can vary with both the optimization task and the capability of the underlying model. This motivates integrating LLM-generated information with an explicit numerical decision-making mechanism rather than treating LLM outputs as the sole basis for strategy selection.

3 METHODOLOGY

LPG-UCB incorporates LLM-generated BO strategy preferences as prior information into sequential numerical updates within a closed-loop BO framework. The overall procedure is provided in Algorithm 1 in Appendix A.

3.1 LLM-CONTEXTUALIZED AF PRIOR MODELING

In our method, the LLM receives the current BO context and generates arm-specific strategy recommendations that are subsequently parsed into numerical preference scores for each candidate strategy. At each optimization iteration, the prompt is constructed from three components: (i) the current BO state and optimization history, including optimization progress, objective statistics, GP hyperparameters, and recent reward feedback; (ii) metadata describing the optimization-state variables and the available AFs; and (iii) instructions specifying the required output format and parsing rules for extracting numerical preference scores for each arm. The prompt asks the LLM to reason over the current optimization context and assign each candidate AF a preference score in $[0, 1]$. The higher the score, the stronger the LLM recommends the AF. Complete prompt templates are provided in the Appendix E.

Let $\Psi(\cdot)$ denote the LLM response from a given prompt, and let S_t denote the BO context provided at iteration t. The preference score for arm i is then retrieved as $q_{t,i}^{\mathrm{LLM}} = \texttt{Parse}\left(\Psi(S_t), i\right)$, where `Parse` is the output parser that identifies the score associated with the i-th AF from the natural language output. Larger scores indicate greater suitability of arm i for the current BO context.

3.2 GAUSSIAN BAYESIAN UPDATE

LLM-generated preference scores $q_{t,i}^{\mathrm{LLM}}$ provide useful guidance for strategy selection, but relying solely on these scores can lead to unreliable decisions. To account for uncertainty in LLM recommendations, we treat these scores as prior information about each arm's expected reward and update this prior using accumulated reward observations. We adopt a Gaussian model that yields a closed-form Bayesian update, combining LLM guidance with empirical evidence from the optimization process.

We first present the Bayesian update procedure for reward modeling, then define the reward in terms of improvement in the objective function.

3.2.1 GENERAL GAUSSIAN BAYESIAN UPDATE

Consider an unknown mean reward θ with Gaussian prior $\theta \sim \mathcal{N}(\mu_0, \sigma_0^2)$, where μ_0 and σ_0^2 denote the prior mean and variance, respectively. Under a Gaussian observation model, $r^{(n)} \mid \theta \sim \mathcal{N}(\theta, \sigma_r^2)$, where $r^{(n)}$ denotes an observation and σ_r^2 the observation-noise variance.

Given N observations $r^{(1)}, \ldots, r^{(N)}$, Bayes' rule gives $p(\theta \mid r^{(1:N)}) \propto p(r^{(1:N)} \mid \theta)\, p(\theta)$, which yields a Gaussian posterior $\theta \mid r^{(1:N)} \sim \mathcal{N}(\mu_N^{\text{post}}, (\sigma_N^{\text{post}})^2)$. The posterior variance and mean are

$$(\sigma_N^{\text{post}})^2 = \left(\frac{1}{\sigma_0^2} + \frac{N}{\sigma_r^2}\right)^{-1}, \qquad \mu_N^{\text{post}} = (\sigma_N^{\text{post}})^2 \left(\frac{\mu_0}{\sigma_0^2} + \frac{N\bar{r}_N}{\sigma_r^2}\right),$$

respectively, where $\bar{r}_N = \frac{1}{N}\sum_{n=1}^{N} r^{(n)}$.

3.2.2 ARM-SPECIFIC GAUSSIAN BAYESIAN UPDATE

In LPG-UCB, each AF is treated as an arm, where arm i corresponds to the use of candidate AF i for generating the next BO query. For arm i at iteration t, we set the prior mean μ_0 to the LLM-generated preference score $q_{t,i}^{\text{LLM}}$, and denote the corresponding prior variance by $\sigma_{0,i}^2$. Let $A_{t,i}$ denote the number of selector updates received by arm i before iteration t. For $A_{t,i} \geq 1$, the empirical mean of accumulated selector rewards is $\widehat{\theta}_{t,i} = \frac{1}{A_{t,i}} \sum_{u=1}^{A_{t,i}} r_i^{(u)}$, where $r_i^{(u)}$ denotes the u-th selector reward used to update arm i. Let $\mu_{t,i}^{\text{post}}$ and $(\sigma_{t,i}^{\text{post}})^2$ denote the posterior mean and variance, respectively, of the expected selector reward for arm i at iteration t. Applying the Gaussian Bayesian update to arm i gives

$$(\sigma_{t,i}^{\text{post}})^2 = \left(\frac{1}{\sigma_{0,i}^2} + \frac{A_{t,i}}{\sigma_r^2}\right)^{-1}, \qquad \mu_{t,i}^{\text{post}} = (\sigma_{t,i}^{\text{post}})^2 \left(\frac{q_{t,i}^{\text{LLM}}}{\sigma_{0,i}^2} + \frac{A_{t,i}\widehat{\theta}_{t,i}}{\sigma_r^2}\right).$$

The posterior mean can equivalently be written as $\mu_{t,i}^{\text{post}} = w_{t,i} q_{t,i}^{\text{LLM}} + (1 - w_{t,i})\widehat{\theta}_{t,i}$, where $w_{t,i} = a_i/(a_i + A_{t,i})$ and $a_i = \sigma_r^2/\sigma_{0,i}^2$. As selector feedback accumulates, $A_{t,i}$ increases and $w_{t,i}$ decreases, reducing the influence of the LLM prior while increasing the contribution of empirical reward information.

3.3 GAUSSIAN-UCB SELECTION RULE

For each active arm, the UCB value is defined as $\text{UCB}_i(t) = \mu_{t,i}^{\text{post}} + \beta_t \sigma_{t,i}^{\text{post}}$. To reduce the contribution of the uncertainty term toward the later stages of optimization, the exploration coefficient follows the linearly decaying schedule $\beta_t = \beta_0\left(1 - \frac{t}{T}\right)$, where $0 \leq \beta_t \leq \beta_0 =: \beta_{\max}$ and T denotes the optimization budget. The LPG-UCB selector chooses the AF arm with the largest UCB value,

$$I_t \in \arg\max_{i \in \mathcal{A}_t} \text{UCB}_i(t),$$

where I_t denotes the index of the selected arm. $\mathcal{A}_t$ denotes the set of active AF arms at iteration t, excluding arms that fail to produce numerically valid candidate points. The candidate point $x_t^{(I_t)}$ proposed by the selected AF arm is then evaluated using the true objective function.

3.4 GP-BASED PSEUDO-FEEDBACK

When only the reward of the selected arm is fed back to the algorithm, reward information for the remaining arms is unavailable. This partial-feedback setting can limit the characterization of each arm's reward distribution, potentially requiring unnecessary exploration of low-potential arms and increasing the risk of premature lock-in to a limited subset of strategies. Motivated by GP-Hedge (Hoffman et al., 2011), which provides surrogate reward feedback to AF arms using GP posterior predictions, we employ GP-based pseudo-feedback only for unselected active arms, while the selected arm receives feedback from the true objective evaluation. For each active arm $i \in \mathcal{A}_t$, the candidate point is obtained as $x_t^{(i)} \in \arg\max_{x \in \mathcal{X}} \alpha_i(x; \mathcal{G}_{t-1})$, with GP-predicted objective value $\widehat{v}_{t,i} = \mu_{t-1}^{\text{GP}}(x_t^{(i)})$. After evaluating the candidate proposed by the selected arm with the true objective function, the objective-level feedback is defined as

$$v_t^{(i)} = \begin{cases} y_t, & i = I_t, \\ \widehat{v}_{t,i}, & i \in \mathcal{A}_t,\ i \neq I_t. \end{cases}$$

This feedback is converted into the selector reward

$$r_i^{(t)} = \text{clip}\left(\frac{f_{\text{best}}^{t-1} - v_t^{(i)}}{\sigma_y^t + \epsilon}, -c, c\right)$$

Here, f_{best}^{t-1} is the best objective value prior to the current evaluation, σ_y^t denotes the empirical standard deviation of the observed objective values, $\epsilon > 0$ is a small numerical constant added for numerical stability, and c is the clipping threshold. Normalization and clipping mitigate premature lock-in to particular AFs caused by disproportionate reward magnitudes. These dense selector updates allow the algorithm to accumulate information about the entire AF portfolio rather than only the AF selected for evaluation.

4 Theoretical Analysis

To characterize the theoretical performance of LPG-UCB under non-stationary reward dynamics, we adopt a rotting-bandit formulation to derive a regret bound and establish conditions for sublinear regret. Classical stochastic MABs typically assume stationary arm-reward distributions. This assumption is generally violated in BO strategy selection, where each new objective evaluation updates the GP posterior and changes the optimization state. Consequently, the expected reward of a given strategy arm can evolve with the optimization history. BO strategy selection is therefore naturally modeled as a non-stationary bandit problem, motivating regret analysis under non-stationary reward dynamics. As defined above, the selector reward measures the arm-specific improvement relative to the current best objective value. As BO progresses and the incumbent improves, achieving further improvement typically becomes more challenging, suggesting a gradual decrease in the expected selector reward. We model this behavior using a rotting-bandit formulation.

Let $\theta_{t,i}$ denote the mean selector reward of arm i at iteration t, and let $i_t^\star$ denote the optimal arm at iteration t with respect to the mean selector reward. We consider the cumulative dynamic regret: $R_T = \sum_{t=1}^{T} \left(\theta_{t,i_t^\star} - \theta_{t,I_t}\right)$.

Under the following assumptions, we derive a regret bound for LPG-UCB and establish conditions under which the expected cumulative regret is sublinear:

1. **Restless reward evolution:** The mean selector reward of each arm depends on the evolving BO state, $\theta_{t,i} = \Theta_i(S_t)$, and may change regardless of whether the arm is selected.
2. **Monotone rotting:** The mean selector reward of each arm is non-increasing over time, $\theta_{t+1,i} \leq \theta_{t,i}$.
3. **Conditionally sub-Gaussian reward noise:** Reward residuals are conditionally zero-mean and sub-Gaussian with variance proxy σ_r^2.
4. **LLM prior informativeness:** The LLM prior is assumed to provide more accurate arm-specific guidance than a uniform random prior in expectation.

The analysis considers the full active-set setting, in which every arm receives either observed or GP-based pseudo-feedback at each round. Under these assumptions, the regret bound of LPG-UCB is given in Theorem 1. The full derivation of the regret bound is provided in Appendix H.

Theorem 1 (Regret bound of LPG-UCB)**.** *Under Assumptions 1–4 and the full active-set setting $A_{t,i} = t - 1$, the expected cumulative regret of LPG-UCB satisfies*

$$\mathbb{E}[R_T] = O\left((\alpha_T + \beta_{\max})\sigma_r\sqrt{T} + \mathbb{E}[R_T^{\text{rot}}] + \gamma C_{\text{rand}} \sum_{i=1}^{K} a_i \log\left(1 + \frac{T}{a_i}\right) + 2cT\delta\right). \quad (1)$$

Here, K denotes the number of arms, $\delta \in (0,1)$ is the confidence-failure probability, and $\alpha_T = \sqrt{2\log\left(\frac{K\pi^2T^2}{3\delta}\right)}$ is the statistical confidence factor.

Corollary 1 (Sublinear regret)**.** *For fixed K, C_{rand}, a_i, γ, and $\beta_{\max}$, choosing $\delta = T^{-2}$ makes the confidence-failure contribution $2cT\delta = O(T^{-1})$, while $\alpha_T = O(\sqrt{\log(KT)})$. Under the bounded*

monotone rotting assumption, the cumulative rotting contribution is asymptotically negligible relative to the decision horizon,

$$\frac{\mathbb{E}[R_T^{\mathrm{rot}}]}{T} \to 0 \qquad \text{as } T \to \infty.$$

Consequently, every term in the regret bound is sublinear in T, implying $\mathbb{E}[R_T] = o(T)$ and establishing sublinear expected cumulative regret for LPG-UCB.

5 EXPERIMENTS

5.1 SYNTHETIC BENCHMARK PROBLEM

5.1.1 EXPERIMENTAL SETTINGS

The experimental setup for the synthetic BO benchmarks largely follows the experiments of LMABO (Ngo et al., 2026). We conducted numerical experiments on 31 synthetic benchmark problems using 10 independent random seeds, comparing LPG-UCB against the baselines listed below. The GP surrogate model employs a radial basis function (RBF) kernel, and Gemma 3 12B is used as the LLM for all LLM-based BO methods. For a benchmark with input dimension D, $2D + 1$ initial points are sampled. The optimization budget is set to 50 iterations for benchmarks with $D < 10$ and 100 iterations for benchmarks with $D \geq 10$. Optimization performance is evaluated using the relative area under the simple-regret curve (RAUC), mean RAUC rank, variability measures, and Wilcoxon signed-rank statistics for cross-benchmark comparison; detailed definitions are provided in Appendix B.2.

5.1.2 BASELINE METHODS

We consider the following four groups of baselines.

- **Single static AFs.** 12 AFs included in the AF portfolio of LPG-UCB : PosSTD, PosMean, PI, LogPI, EI, LogEI, UCB, TS, KG, PES, MES, and JES.
- **Classical adaptive AF portfolio strategies.** For comparison with classical adaptive AF portfolio strategies, GP-Hedge (Hoffman et al., 2011), No-PASt-BO (Vasconcelos et al., 2019), and SETUP-BO (Vasconcelos et al., 2022) were included.
- **LLM-based BO methods.** To compare LPG-UCB with recent LLM-based BO methods, LLAMBO (Liu et al., 2024), LMABO (Ngo et al., 2026), and LABO (Chen et al., 2026) serve as baselines.
- **Random-prior LPG-UCB.** To isolate the contribution of LLM-generated prior information, we replace the LLM preference scores with independent random values sampled from $\mathrm{Unif}(0, 1)$ while retaining the remaining LPG-UCB settings.

5.1.3 EXPERIMENTAL RESULTS ANALYSIS

The overall results are summarized in Table 1. LPG-UCB achieves the strongest aggregate performance and the most consistent performance across benchmarks among the evaluated methods. Among the LLM-based baselines, LLAMBO shows relatively weak performance under our Gemma 3 12B setting, consistent with recent reproducibility results suggesting sensitivity to the underlying LLM (Rychert et al., 2025). For LMABO, the elaborate prompt improves aggregate performance over the original prompt (Ngo et al., 2026). LPG-UCB consistently outperforms its random-prior variant, indicating that the LLM-derived prior provides informative arm-specific guidance beyond random prior assignment. This advantage is also observed across different LLM backbones in the LLM ablation study, suggesting that the contribution of the LLM prior is not limited to a particular model; detailed results are provided in Appendix F.

5.2 HIGH-DIMENSIONAL REAL-WORLD BENCHMARK PROBLEMS

High-dimensional real-world optimization has been a major challenge in BO. Various approaches, including enhanced vanilla BO (Hvarfner et al., 2024), Spherical Linear-Kernel BO (Doumont et al.,

Table 1: Summary statistics across synthetic function benchmarks.

Method	MeanRAUC (IQR)	P-value (RAUC)	MeanRank (Min-Max)	P-value (Rank)	CV of RAUC
Single static AF					
EI	2.25 (1.17-1.75)	1.73e-06	7.87 (1-19)	1.67e-06	1.11
LogEI	2.25 (1.15-1.65)	9.31e-10	7.71 (2-23)	1.13e-06	1.12
PI	2.60 (1.16-1.75)	9.31e-10	8.23 (2-22)	1.15e-06	1.32
LogPI	2.47 (1.18-1.99)	8.30e-06	8.52 (1-23)	8.15e-06	1.18
PosMean	2.46 (1.04-2.35)	1.23e-05	8.45 (1-23)	1.22e-05	1.16
PosSTD	12.74 (2.03-4.57)	9.31e-10	21.94 (6-24)	8.96e-07	2.25
UCB	4.87 (1.60-3.20)	9.31e-10	18.32 (7-24)	1.12e-06	1.59
TS	5.40 (1.44-2.21)	9.31e-10	15.03 (5-21)	1.14e-06	2.02
KG	3.89 (1.51-2.42)	9.31e-10	14.74 (2-20)	1.13e-06	1.65
PES	9.76 (1.58-3.62)	9.31e-10	18.97 (2-23)	9.95e-07	2.2
MES	3.79 (1.37-2.43)	9.31e-10	13.55 (2-21)	1.16e-06	1.55
JES	2.04 (1.13-1.84)	8.30e-06	11.13 (1-22)	8.02e-06	1.05
Classical adaptive AF portfolio strategy					
GP-Hedge	2.48 (1.35-2.13)	9.31e-10	11.13 (4-20)	1.14e-06	0.93
GP-Hedge (Curated)	2.72 (1.28-2.24)	1.73e-06	11.81 (1-23)	1.70e-06	1.14
No-PASt-BO	2.49 (1.34-2.83)	9.31e-10	11.35 (2-22)	1.16e-06	0.84
No-PASt-BO (Curated)	2.79 (1.26-2.79)	2.56e-06	10.94 (1-19)	2.50e-06	1.07
SETUP-BO	2.53 (1.34-2.87)	9.31e-10	11.90 (3-22)	1.16e-06	0.83
SETUP-BO (Curated)	3.27 (1.26-2.88)	9.31e-10	12.35 (2-23)	1.16e-06	1.38
LLM-based AF selection strategy					
LMABO (Elaborate)	1.82 (1.34-1.75)	1.73e-06	11.74 (1-20)	1.71e-06	0.58
LMABO (Original)	2.61 (1.15-2.06)	3.79e-06	10.32 (1-20)	3.71e-06	1.22
LLAMBO	11.53 (2.55-8.64)	9.31e-10	23.39 (20-24)	5.80e-07	1.49
LABO	5.68 (1.39-2.72)	5.61e-06	15.68 (1-24)	5.22e-06	2.19
LPG-UCB (Random Prior)	2.57 (1.11-2.01)	1.73e-06	8.84 (1-21)	1.71e-06	1.27
LPG-UCB (Our Method)	1.51 (1.13-1.42)	2.56e-06	6.10 (1-18)	2.42e-06	0.46

2026), and TuRBO (Eriksson et al., 2019), have been proposed to improve BO performance in high-dimensional search spaces. We extend LPG-UCB beyond AF selection to sequential BO strategy selection for high-dimensional optimization, where each arm represents a combination of an AF and a search regime. At each iteration, LPG-UCB jointly selects the AF and whether it is optimized over the global domain or within the local trust region defined by TuRBO. The overall procedure is summarized in Algorithm 2, with details provided in Appendix C.

5.2.1 Experimental settings

We evaluate TuRBO-Enhanced LPG-UCB on high-dimensional real-world optimization problem, with the results summarized in Table 2 and detailed experimental settings provided in Appendix D. All experimental settings follow those used for the synthetic-function benchmarks, except that the number of BO iterations is fixed at 300. The strategy arm set consists of four AFs: LogPI, LogEI, UCB, and TS. Each AF is instantiated under two search regimes, where it is optimized either over the global domain or within the TuRBO trust region, resulting in a total of eight strategy arms.

5.2.2 Experimental Results Analysis

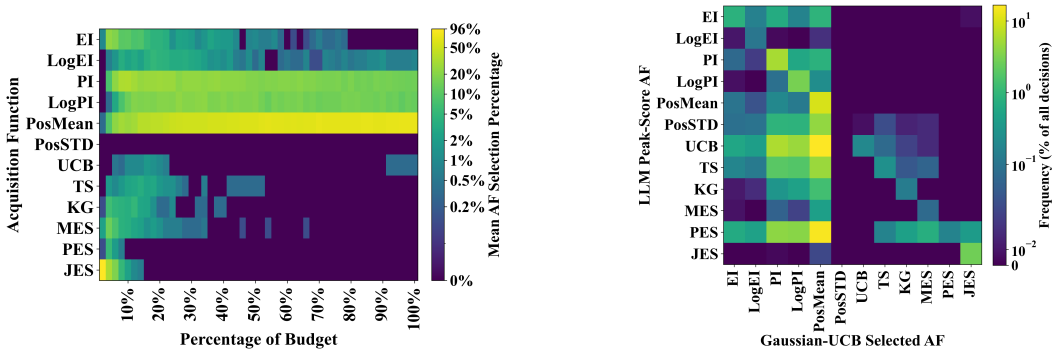
On the Rover benchmark(Wang et al., 2018), the TuRBO-enhanced LPG-UCB outperforms all evaluated fixed global and TuRBO strategies, suggesting that jointly adapting the search regime and AF can improve performance when their relative effectiveness changes during optimization. This result highlights the potential of extending LPG-UCB to joint adaptation of AF and global–local search strategies in high-dimensional optimization.

6 LPG-UCB Strategy-selection Dynamics Analysis

To examine the strategy-selection behavior of LPG-UCB, we analyze the frequency of AF selections throughout the optimization process. For each benchmark, the optimization budget is divided into 2% intervals, the AF selection fractions are averaged across 10 independent runs, and the resulting percentages are averaged across benchmarks with equal benchmark weighting. Figure 2a

Table 2: Raw incumbent AUC (mean ± std) on the high-dimensional Rover benchmark.

Method	Rover
LPG-UCB (TuRBO enhanced)	-269.19 (±333.86)
LogEI	66.98 (±206.34)
LogPI	-13.83 (±230.17)
TS	1055.12 (±232.40)
UCB	534.35 (±188.54)
TuRBO-LogEI	-130.77 (±340.03)
TuRBO-LogPI	-238.73 (±180.10)
TuRBO-TS	333.84 (±445.80)
TuRBO-UCB	-164.77 (±366.96)



(a) AF selection frequency over the optimization budget.

(b) LLM-selector decision frequency.

Figure 2: (a) Mean AF selection frequency as a function of the normalized optimization budget. (b) Joint frequency of the AFs receiving the highest LLM prior scores and the AFs ultimately selected by the Gaussian-UCB selector. Logarithmic color scaling is used to retain visibility of infrequent selection patterns.

shows the resulting AF selection frequency over the normalized optimization budget. LPG-UCB shifts from a broader AF selection pattern in the early stages toward emphasis on a narrower set of AFs as optimization progresses, consistent with a gradual move toward more exploitative behavior. Figure 2(b) further examines the relationship between the AF preference of LLM and the final Gaussian-UCB decision. The diagonal entries indicate that the Gaussian-UCB selector often agrees with the highest-scoring LLM preference, while the off-diagonal entries show that the selector also frequently overrides the LLM preference. To assess whether these overrides correspond to useful optimization decisions, we compute the fraction of improvement events associated with override decisions. Across all runs, such overrides account for **61.69%** of all improvement events. Detailed stage-wise statistics and the corresponding visualization are provided in Table 7 and Figure 3 in Appendix G. These results indicate that the Gaussian-UCB selector does not merely reproduce the LLM preference, but calibrates LLM guidance using accumulated numerical feedback to improve strategy selection.

7 Conclusions

In this study, we introduced LPG-UCB, a framework that combines LLM-based interpretation of the optimization state as prior information with a MAB-based numerical regulator updated through reward feedback for adaptive BO strategy selection. We formulated the strategy-selection process as a non-stationary rotting-bandit model and established conditions for sublinear expected cumulative regret. Across 31 synthetic benchmark problems, LPG-UCB outperformed state-of-the-art LLM-based BO baselines in aggregate performance, while also demonstrating competitive performance on high-dimensional real-world optimization problem.

AI USE STATEMENT

Generative AI tools were used to aid and polish writing, support retrieval and discovery of related work, provide feedback during research ideation and execution, assist with code development and debugging, and support the development and verification of mathematical arguments and proofs. All AI-assisted content, including technical claims, references, code, and mathematical derivations, was reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Virginia Aglietti, Ira Ktena, Jessica Schrouff, Eleni Sgouritsa, Francisco JR Ruiz, Alan Malek, Alexis Bellot, and Silvia Chiappa. Funbo: Discovering acquisition functions for bayesian optimization with funsearch. *arXiv preprint arXiv:2406.04824*, 2024.

Jasper Meynard Arana, Kristine Ann M. Carandang, Ethan Robert Casin, Christian Alis, and Christopher Monterola. When models hesitate: Answer instability as a label-free uncertainty signal for LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 816–826. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-srw.72.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew G. Wilson, and Eytan Bakshy. BoTorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21524–21538, 2020.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, volume 27, pp. 199–207, 2014.

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. *arXiv preprint arXiv:2605.22054*, 2026. Accepted to ICML 2026.

Colin Doumont, Donney Fan, Natalie Maus, Jacob R. Gardner, Henry Moss, and Geoff Pleiss. We still don't understand high-dimensional bayesian optimization. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, 2026.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Algorithmic Learning Theory*, volume 6925 of *Lecture Notes in Computer Science*, pp. 174–188. Springer, 2011. doi: 10.1007/978-3-642-24412-4_16.

Gianmarco Genalti, Marco Mussi, Nicola Gatti, Marcello Restelli, Matteo Castiglioni, and Alberto Maria Metelli. Graph-triggered rising bandits. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15351–15380. PMLR, 2024.

Rushil Gupta, Jason Hartford, and Bang Liu. LLMs for Bayesian optimization in scientific domains: Are we there yet? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15482–15510. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.838.

Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, 2021. doi: 10.1080/10556788.2020.1808977.

Matthew W. Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for bayesian optimization. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 327–336, 2011.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20793–20817. PMLR, 2024.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. doi: 10.1023/A:1008306431147.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yang Li, Jiawei Jiang, Jinyang Gao, Yingxia Shao, Ce Zhang, and Bin Cui. Efficient automatic CASH via rising bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4763–4771, 2020. doi: 10.1609/aaai.v34i04.5910.

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.

Qin Lu, Konstantinos D. Polyzos, Bingcong Li, and Georgios B. Giannakis. Surrogate modeling for bayesian optimization beyond a single gaussian process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11283–11296, 2023. doi: 10.1109/TPAMI.2023.3264741.

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

Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery. *Nature Machine Intelligence*, 2026. doi: 10.1038/s42256-026-01283-z.

Adam Rychert, Gasper Spagnolo, and Evgenii Posashkov. Reproducibility study of large language model bayesian optimization. *arXiv preprint arXiv:2511.18891*, 2025.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11709–11724. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.685.

Julien Seznec, Pierre Menard, Alessandro Lazaric, and Michal Valko. A single algorithm for both restless and rested rotting bandits. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3784–3794. PMLR, 2020.

Mingxuan Song, Yusen Huo, Bohan Zhou, Shenglin Yin, Zhen Xiao, Jieyi Long, Zhilin Zhang, and Chuan Yu. DARA: Few-shot budget allocation in online advertising via in-context decision making with RL-finetuned LLMs. In *Proceedings of the ACM Web Conference 2026*, pp. 40–50. ACM, 2026. doi: 10.1145/3774904.3792129.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 1015–1022, 2010.

Eliott Van Dieren, Tommaso Vanzan, and Fabio Nobile. LAGO: A local-global optimization framework combining trust region methods and bayesian optimization. *arXiv preprint arXiv:2603.02970*, 2026.

Thiago de P. Vasconcelos, Daniel Augusto R. M. A. de Souza, César L. C. Mattos, and João P. P. Gomes. No-PASt-BO: Normalized portfolio allocation strategy for bayesian optimization. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 561–568, 2019. doi: 10.1109/ICTAI.2019.00084.

Thiago de P. Vasconcelos, Daniel Augusto R. M. A. de Souza, Gustavo C. de M. Virgolino, César L. C. Mattos, and João P. P. Gomes. Self-tuning portfolio-based bayesian optimization. *Expert Systems with Applications*, 188:115847, 2022. doi: 10.1016/j.eswa.2021.115847.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 745–754. PMLR, 2018.

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25(A):287–298, 1988. doi: 10.2307/3214163.

Min Zhang, Yuzhe Lu, Yun Zhou, Panpan Xu, Lin Lee Cheong, Chang-Tien Lu, and Haozhu Wang. Collaborative llm numerical reasoning with local data protection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34772–34780, 2026. doi: 10.1609/aaai.v40i41.40779.

A Details of LPG-UCB

A.1 Pseudocode

The pseudocode of the proposed algorithm is provided in Algorithm 1.

Algorithm 1 LLM-Prior Gaussian-UCB Acquisition-Function Selector

Require: $\mathcal{D}_0$, AF portfolio $\{\alpha_i\}_{i=1}^K$, LLM prior generator Ψ, $\{\sigma_{0,i}^2\}_{i=1}^K$, σ_r^2, clipping threshold c, exploration schedule β_t.

Initialize GP with initial observations $\mathcal{D}_0$
Initialize $A_{1,i} = 0$ and $\widehat{\theta}_{1,i} = 0$ for all $i \in [K]$.
for $t = 1, \ldots, T$ **do**
  Fit GP $\mathcal{G}_{t-1} \leftarrow \mathrm{fit}(\mathcal{D}_{t-1})$.
  Generate LLM priors $q_{t,i}^{LLM} \leftarrow \Psi(S_t)_i$ for all $i \in [K]$.
  Initialize $\mathcal{A}_t \leftarrow \emptyset$.
  for $i = 1, \ldots, K$ **do**
    $x_t^{(i)} \in \arg\max_x \alpha_i(x; \mathcal{G}_{t-1})$.
    if $x_t^{(i)}$ and $\mu_{t-1}^{GP}(x_t^{(i)})$ are valid **then**
      $\mathcal{A}_t \leftarrow \mathcal{A}_t \cup \{i\}$; $\widehat{v}_{t,i} \leftarrow \mu_{t-1}^{GP}(x_t^{(i)})$.
    end if
  end for
  for $i \in \mathcal{A}_t$ **do**
    $(\sigma_{t,i}^{post})^2 \leftarrow \left(\sigma_{0,i}^{-2} + A_{t,i}\sigma_r^{-2}\right)^{-1}, \quad w_{t,i} \leftarrow \dfrac{\sigma_r^2}{\sigma_r^2 + A_{t,i}\sigma_{0,i}^2}.$
    $\mu_{t,i}^{post} \leftarrow w_{t,i} q_{t,i}^{LLM} + (1 - w_{t,i})\widehat{\theta}_{t,i}$,
    $UCB_i(t) \leftarrow \mu_{t,i}^{post} + \beta_t \sigma_{t,i}^{post}$.
  end for
  Select $I_t \in \arg\max_{i \in \mathcal{A}_t} UCB_i(t)$; evaluate $y_t \leftarrow f(x_t^{(I_t)})$.
  $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(x_t^{(I_t)}, y_t)\}$.
  for $i \in \mathcal{A}_t$ **do**
    $v_t^{(i)} \leftarrow \begin{cases} y_t, & i = I_t, \\ \widehat{v}_{t,i}, & i \neq I_t, \end{cases} \quad r_i^{(t)} \leftarrow \mathrm{clip}\left(\dfrac{f_{best}^{t-1} - v_t^{(i)}}{\sigma_y^t + \epsilon}, -c, c\right), \quad A_{t+1,i} \leftarrow A_{t,i} + 1.$
    if $A_{t,i} = 0$ **then**
      $\widehat{\theta}_{t+1,i} \leftarrow r_i^{(t)}$.
    else
      $\widehat{\theta}_{t+1,i} \leftarrow \dfrac{A_{t,i}\widehat{\theta}_{t,i} + r_i^{(t)}}{A_{t+1,i}}$.
    end if
  end for
  for $i \notin \mathcal{A}_t$ **do**
    $(A_{t+1,i}, \widehat{\theta}_{t+1,i}) \leftarrow (A_{t,i}, \widehat{\theta}_{t,i})$.
  end for
end for

B Details of Synthetic Benchmark Function Experiments

B.1 Synthetic Benchmark Function list

The evaluation is performed on 31 synthetic benchmark problems, comprising 14 functions from the COCO-BBOB test suite (Hansen et al., 2021) and 17 functions implemented in the BoTorch library (Balandat et al., 2020). The benchmark set covers a range of objective landscapes, including unimodal and multimodal functions with different dimensionalities and conditioning properties. The complete list of benchmark functions and their input dimensions is provided in Table 3 in the appendix.

Table 3: Synthetic benchmark functions used in the experiments.

COCO-BBOB		BoTorch	
Benchmark	**Dim.**	**Benchmark**	**Dim.**
Discus	5	Branin	2
Bent Cigar	5	Beale	2
Büche–Rastrigin	5	Bukin	2
Attractive Sector	5	Cosine8	8
Step Ellipsoid	5	Dixon–Price	15
Sharp Ridge	5	Drop-Wave	2
Different Powers	5	Eggholder	2
Weierstrass	5	Griewank	9
Schaffers Ill-Conditioned	5	Holder Table	2
Composite Griewank–Rosenbrock	10	Michalewicz	10
Gallagher 21	5	Shekel	4
Gallagher 101	5	Six-Hump Camel	2
Katsuura	5	Styblinski–Tang	21
Lunacek bi-Rastrigin	5	Hartmann6	6
		Rosenbrock	4
		Ackley	5
		Levy	6

B.2 Evaluation metric

Optimization performance is evaluated primarily using simple regret. For a minimization problem, the simple regret at iteration t is defined as

$$r_t^{\text{simple}} = f_{\text{best}}^t - f^\star, \tag{2}$$

where f_{best}^t denotes the best objective value observed up to iteration t, including the initial observations, and $f^\star$ is the known global optimum. The area under the simple regret curve (AUC) is computed as the discrete cumulative simple regret:

$$\text{AUC} = \sum_{t=1}^{T} r_t^{\text{simple}}, \tag{3}$$

where T denotes the evaluation budget. Lower AUC values indicate better optimization performance, reflecting faster convergence toward the global optimum within the given evaluation budget.For each method and benchmark, AUC values are averaged over 10 independent repetitions using a fixed set of 10 random seeds. To enable comparison across benchmarks with different regret scales, we normalize the mean AUC on each benchmark by the best-performing method to obtain the relative AUC (RAUC).

$$\text{RAUC}_{b,m} = \frac{\overline{\text{AUC}}_{b,m}}{\min_{m' \in \mathcal{M}_b} \overline{\text{AUC}}_{b,m'}}, \tag{4}$$

where $\overline{\text{AUC}}_{b,m}$ denotes the mean AUC of method m on benchmark b, and $\mathcal{M}_b$ is the set of compared methods with valid results on that benchmark. The best-performing method receives an RAUC of 1.0, with larger values indicating worse relative performance. Methods are also ranked on each benchmark according to their mean AUC, with rank 1 assigned to the lowest AUC.

Aggregate performance across benchmarks is summarized using mean RAUC, the 25th–75th percentile interval of RAUC, mean rank, and the minimum–maximum rank range. The coefficient of variation (CV) of RAUC is additionally reported to characterize the variability of relative performance across benchmarks:

$$\text{CV}_m = \frac{\text{Std}_b\left(\text{RAUC}_{b,m}\right)}{\text{Mean}_b\left(\text{RAUC}_{b,m}\right)}. \tag{5}$$

The standard deviation is computed with the population convention. Wilcoxon signed-rank p-values are computed from the benchmark-wise deviations $\text{RAUC}_{b,m} - 1$ and $\text{Rank}_{b,m} - 1$. The tests assess whether these deviations are systematically greater than zero across benchmarks, indicating that a method consistently performs worse than the benchmark-wise best observed method.

B.3 Baseline Implementation Details

B.3.1 LMABO

For LMABO , we evaluate two prompt configurations. *LMABO-original* uses the prompt provided in the original implementation(Ngo et al., 2026). Since the prompt used in our framework contains more detailed optimization-state information and metadata, we additionally evaluate LMABO using a version of our prompt modified for direct AF selection, consistent with the LMABO framework. This variant is denoted as *LMABO-Elaborate*.

B.3.2 LLAMBO

For LLAMBO, we use our reimplementation under the same Sobol-based initial design used for the other methods, without the LLM-based warm-start initialization.

B.3.3 LABO

For LABO, we retain its LLM-based warm-start initialization in order to preserve the baseline's original initialization procedure. However, on the Griewank and Ackley benchmarks, we observed that the LLM warm-start included the known global optimum in the initial design. In particular, for Griewank, the warm-start included the center of the normalized search space, $x = (0.5, \ldots, 0.5)$, which corresponds to the global optimum in the original search space. As a result, the simple-regret AUC became zero before the BO iterations began, making the corresponding RAUC undefined.

For Griewank and Ackley, we disable the LABO warm-start and use the same Sobol-based initialization protocol used for the other methods. For all other benchmarks, the original LABO warm-start procedure is retained.

B.3.4 Classical Adaptive AF Portfolio Strategies.

GP-Hedge (Hoffman et al., 2011), No-PASt-BO (Vasconcelos et al., 2019), and SETUP-BO (Vasconcelos et al., 2022) were included. Each method is evaluated using both a full 12-AF portfolio and a curated portfolio of EI, LogEI, and TS, following LMABO (Ngo et al., 2026) to assess the effect of portfolio composition on adaptive selection performance.

C Details of Turbo Enhanced Sequential AF Selection

To implement the TuRBO-enhanced sequential AF selection, the following two steps are added to the original LPG-UCB algorithm. Trust region is updated after fitting the GP to define the local search region. After function evaluation, a trust-region transition is performed to determine an appropriate center for the next trust region. This formulation extends LPG-UCB from sequential AF selection within a fixed search regime to sequential selection over both AFs and search regimes.

C.1 Trust region update

$$\tau_t = (c_t, L_t) \in [0,1]^D \times \mathbb{R}_+,$$

$$L_{t,d} = \frac{\lambda_{t,d} L_t}{\left(\prod_{d'=1}^{D} \lambda_{t,d'}\right)^{1/D}}, \qquad d = 1, \ldots, D,$$

$$\mathcal{B}_{\mathrm{TuRBO}}^{[0,1]} = \prod_{d=1}^{D} \left[\max\left(c_{t,d} - \frac{L_{t,d}}{2}, 0\right),\ \min\left(c_{t,d} + \frac{L_{t,d}}{2}, 1\right)\right],$$

$$\mathcal{B}_{\mathrm{TuRBO}} = u^{-1}\left(\mathcal{B}_{\mathrm{TuRBO}}^{[0,1]}\right), \qquad \mathcal{B}_{\mathrm{Global}} = \mathcal{X}.$$

Here, c_t and L_t denote the trust-region center and base length, respectively, while $\lambda_{t,d}$ is the GP lengthscale for dimension d. The TuRBO region is scaled according to the GP lengthscales and mapped back to the original search space, whereas global arms search over the full domain $\mathcal{X}$.

C.2 Trust region state transition

Let b_t denote the best objective value associated with the current trust region, and let s_t and f_t denote the consecutive success and failure counters, respectively. When a TuRBO arm is selected, a successful iteration is defined as

$$y_t < b_t - \epsilon_{\mathrm{TR}}|b_t|.$$

The success and failure counters are updated as

$$(s_{t+1}, f_{t+1}) = \begin{cases} (s_t + 1, 0), & y_t < b_t - \epsilon_{\mathrm{TR}}|b_t|, \\ (0, f_t + 1), & \text{otherwise.} \end{cases}$$

The trust-region base length is then adjusted according to the consecutive success and failure tolerances,

$$L_{t+1} = \begin{cases} \min(2L_t, L_{\max}), & s_{t+1} = s_{\mathrm{tol}}, \\ L_t/2, & f_{t+1} = f_{\mathrm{tol}}, \\ L_t, & \text{otherwise.} \end{cases}$$

When the evaluated TuRBO candidate improves the current trust-region best, the center and best value are updated as

$$(c_{t+1}, b_{t+1}) = \begin{cases} \left(u(x_t^{(I_t)}), y_t\right), & y_t < b_t, \\ (c_t, b_t), & \text{otherwise.} \end{cases}$$

If the trust-region length falls below the minimum allowable length, $L_{t+1} < L_{\min}$, the trust region is restarted using a randomly selected previously evaluated point $(x_r, y_r) \in \mathcal{D}_t$,

$$(c_{t+1}, L_{t+1}, b_{t+1}) = \left(u(x_r), L_{\mathrm{init}}, y_r\right), \qquad s_{t+1} = f_{t+1} = 0.$$

When a global arm is selected, the trust-region state is normally unchanged. However, if the global candidate improves upon the current trust-region best, the trust region is restarted around the newly discovered point,

$$(c_{t+1}, L_{t+1}, b_{t+1}) = \left(u(x_t^{(I_t)}), L_{\mathrm{init}}, y_t\right), \qquad \text{if } m_{I_t} = \text{Global and } y_t < b_t,$$

with the success and failure counters reset to zero. Otherwise, the existing trust-region state is retained. This transition allows global search to relocate the local search region when a better region of the search space is identified, while TuRBO arms locally adapt the trust-region size based on sequential success and failure.

C.3 Pseudocode

The pseudocode of the proposed algorithm is provided in Algorithm 2.

D Details of Real-World Benchmark Experiments

D.1 Evaluation Metric

For the high-dimensional real-world benchmarks, the global optimum is unknown, preventing direct computation of simple regret. We therefore evaluate optimization performance using the area under the incumbent curve (AUC). Let f^t_{best} denote the best objective value observed up to iteration t. The AUC is computed as

$$\text{AUC} = \sum_{t=1}^{T-1} \frac{f^t_{\text{best}} + f^{t+1}_{\text{best}}}{2}.$$

Lower AUC values indicate better optimization performance, reflecting faster improvement of the incumbent over the evaluation budget. AUC values are averaged over 10 independent seeds.

D.2 TuRBO Parameter Setting

For TuRBO, a single trust region is maintained with an initial trust-region length parameter of 0.8, a maximum value of 1.6, and a minimum value of 0.5^7. The trust-region length is expanded by a factor of two after three consecutive iterations that improve the incumbent objective value, and contracted by a factor of two after five consecutive iterations without improvement. As static baselines, LogPI, LogEI, UCB, and TS are each evaluated independently under both the global search regime and the TuRBO trust-region regime.

Algorithm 2 TuRBO-Enhanced LLM-Prior Gaussian-UCB Strategy Selector

Require: $\mathcal{D}_0$, strategy arms $\{(m_i, \alpha_i)\}_{i=1}^{K}$, Ψ, $\{\sigma_{0,i}^2\}$, σ_r^2, c, β_t, L_{init}, $L_{\min}$, $L_{\max}$, s_{tol}, f_{tol}, ε_{TR}.

$u(x) \leftarrow (x - lb)/(ub - lb)$; initialize $A_{1,i} = 0$.
$x_0^{best} \leftarrow \arg\min_{(x,y)\in\mathcal{D}_0} y$; $(c_1, b_1, L_1) \leftarrow (u(x_0^{best}), f(x_0^{best}), L_{\text{init}})$; $s_1 = f_1 = 0$.
for $t = 1, \ldots, T$ **do**
  Fit $\mathcal{G}_{t-1}$ to $\mathcal{D}_{t-1}$ and obtain GP lengthscales $\lambda_{t,d}$.
  $L_{t,d} \leftarrow \lambda_{t,d} L_t / \left(\prod_{d'=1}^{D} \lambda_{t,d'}\right)^{1/D}$.
  $\mathcal{B}_{\text{TuRBO}}^{[0,1]} \leftarrow \prod_{d=1}^{D} [\max(c_{t,d} - L_{t,d}/2, 0), \min(c_{t,d} + L_{t,d}/2, 1)]$.
  $\mathcal{B}_{\text{TuRBO}} \leftarrow u^{-1}(\mathcal{B}_{\text{TuRBO}}^{[0,1]})$; $\mathcal{B}_{\text{Global}} \leftarrow \mathcal{X}$.
  Construct S_t and obtain $q_{t,i}^{LLM} \leftarrow \Psi(S_t)_i$ for all i; initialize $\mathcal{A}_t \leftarrow \emptyset$.
  for $i = 1, \ldots, K$ **do**
    $\mathcal{B}_{t,i} \leftarrow \mathcal{B}_{\text{Global}}$ if $m_i = \text{Global}$, otherwise $\mathcal{B}_{\text{TuRBO}}$.
    $x_t^{(i)} \in \arg\max_{x\in\mathcal{B}_{t,i}} \alpha_i(x; \mathcal{G}_{t-1})$.
    if $x_t^{(i)}$ and $\mu_{t-1}^{GP}(x_t^{(i)})$ are valid **then**
      $\mathcal{A}_t \leftarrow \mathcal{A}_t \cup \{i\}$; $\widehat{v}_{t,i} \leftarrow \mu_{t-1}^{GP}(x_t^{(i)})$.
    end if
  end for
  for $i \in \mathcal{A}_t$ **do**
    $(\sigma_{t,i}^{post})^2 \leftarrow (\sigma_{0,i}^{-2} + A_{t,i}\sigma_r^{-2})^{-1}$; $w_{t,i} \leftarrow \sigma_r^2/(\sigma_r^2 + A_{t,i}\sigma_{0,i}^2)$.
    $\mu_{t,i}^{post} \leftarrow w_{t,i} q_{t,i}^{LLM} + (1 - w_{t,i})\widehat{\theta}_{t,i}$; $UCB_i(t) \leftarrow \mu_{t,i}^{post} + \beta_t \sigma_{t,i}^{post}$.
  end for
  $I_t \in \arg\max_{i\in\mathcal{A}_t} UCB_i(t)$; $y_t \leftarrow f(x_t^{(I_t)})$; $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(x_t^{(I_t)}, y_t)\}$.
  if $m_{I_t} = \text{TuRBO}$ **then**
    if $y_t < b_t - \varepsilon_{\text{TR}}|b_t|$ **then**
      $s_{t+1} \leftarrow s_t + 1$; $f_{t+1} \leftarrow 0$.
    else
      $s_{t+1} \leftarrow 0$; $f_{t+1} \leftarrow f_t + 1$.
    end if
    $L_{t+1} \leftarrow L_t$.
    if $s_{t+1} = s_{\text{tol}}$ **then**
      $L_{t+1} \leftarrow \min(2L_t, L_{\max})$; $s_{t+1} \leftarrow 0$.
    else if $f_{t+1} = f_{\text{tol}}$ **then**
      $L_{t+1} \leftarrow L_t/2$; $f_{t+1} \leftarrow 0$.
    end if
    if $y_t < b_t$ **then**
      $(c_{t+1}, b_{t+1}) \leftarrow (u(x_t^{(I_t)}), y_t)$.
    else
      $(c_{t+1}, b_{t+1}) \leftarrow (c_t, b_t)$.
    end if
    if $L_{t+1} < L_{\min}$ **then**
      Sample $(x_r, y_r) \in \mathcal{D}_t$; $(c_{t+1}, L_{t+1}, b_{t+1}) \leftarrow (u(x_r), L_{\text{init}}, y_r)$; $s_{t+1} = f_{t+1} = 0$.
    end if
  else if $m_{I_t} = \text{Global}$ and $y_t < b_t$ **then**
    $(c_{t+1}, L_{t+1}, b_{t+1}) \leftarrow (u(x_t^{(I_t)}), L_{\text{init}}, y_t)$; $s_{t+1} = f_{t+1} = 0$.
  else
    $(c_{t+1}, L_{t+1}, b_{t+1}, s_{t+1}, f_{t+1}) \leftarrow (c_t, L_t, b_t, s_t, f_t)$.
  end if
  for $i \in \mathcal{A}_t$ **do**
    $v_t^{(i)} \leftarrow y_t$ if $i = I_t$, otherwise $\widehat{v}_{t,i}$.
    $r_i^{(t)} \leftarrow \text{clip}((f_{\text{best}}^{t-1} - v_t^{(i)})/(\sigma_y^t + \epsilon), -c, c)$; $A_{t+1,i} \leftarrow A_{t,i} + 1$.
    $\widehat{\theta}_{t+1,i} \leftarrow [A_{t,i}\widehat{\theta}_{t,i} + r_i^{(t)}]/A_{t+1,i}$, with $\widehat{\theta}_{t+1,i} = r_i^{(t)}$ if $A_{t,i} = 0$.
  end for
  for $i \notin \mathcal{A}_t$ **do**
    $(A_{t+1,i}, \widehat{\theta}_{t+1,i}) \leftarrow (A_{t,i}, \widehat{\theta}_{t,i})$.
  end for
end for

E LLM PROMPT

The prompt design was adapted from LMABO (Ngo et al., 2026) and extended for the proposed framework. For comparison, LMABO is evaluated using both its original prompt and the extended prompt used in our method.

E.1 TASK DEFINITION AND SURROGATE-MODEL SPECIFICATION

```
You are an expert in Bayesian Optimization, specifically tasked with evaluating acquisition functions for the next iteration to minimize an objective function.

For context, we use a Gaussian Process as the surrogate model with an RBF (Squared Exponential) kernel with automatic relevance determination.
```

E.2 OPTIMIZATION-CONTEXT METADATA DESCRIPTION

```
I will provide you with a summary of the Bayesian Optimization process at
each step. This summary will include the following information:
- **N:** The total number of points evaluated so far.
- **D:** The dimensionality of the search space (number of input parameters).
- **Budget progress:** 0.0=start, 1.0=end. Low values favor exploration;
high values favor exploitation.
- **Remaining iterations:** Absolute iterations left in the optimization.
- **f_range:** Range, mean, and standard deviation of all observed
function values so far.
- **f_min:** The current best (lowest) observed objective value.
- **f_improvement:** How much f_min has improved relative to f_range spread.
High = significant progress; low = stagnation.
- **Shortest distance:** Distance from last point to nearest existing
observation. This signal is only meaningful together with budget progress:
small distance early in the run may suggest premature clustering and can
favor exploration, but small distance late in the run is often just normal
fine-grained refinement near a good optimum and should NOT by itself
trigger a switch to exploration.
- **Lengthscales:** GP kernel hyperparameters describing smoothness and
relevance of each input dimension. Provided as range, mean, and std.
- **Lengthscale CV:** Coefficient of variation of lengthscales
(std / mean). High CV means the landscape varies significantly across
dimensions -- some dimensions are more relevant than others.
- **Outputscale:** Overall magnitude of function variation.
- **Recent history (if available):** The last up to 3 selections, each with
the AF name, the actual observed value (y_value), the reward it received,
and an explicit outcome label -- (SUCCESS)/(FAILURE)/(NEUTRAL) -- that
already reflects the reward's sign. This is the ONLY historical
information you receive. A single data point can be noisy -- do not
overreact to just the most recent entry. Look for a pattern across the
available entries (e.g., an AF or category repeatedly receiving poor
rewards is much stronger evidence than one bad result). For the single
most recent entry specifically, do not narrate its outcome from memory or
```

```
general impression - re-check its explicit (SUCCESS)/(FAILURE)/(NEUTRAL)
label and its y_value directly before using it in your reasoning; small
reward magnitudes (e.g. -0.01) are still failures if labeled (FAILURE),
not near-neutral results.
```

E.3 ACQUISITION-FUNCTION TAXONOMY

```
================================================
ACQUISITION FUNCTION TAXONOMY -- READ CAREFULLY
================================================
EXPLOITATION-focused:
- PI (Probability of Improvement): computes the probability that the next
point improves over the current best. Tends to be risk-averse and
exploitation-focused.
- LogPI (Log Probability of Improvement): log-transformed variant of PI
for numerical stability. Same behavior as PI.
- EI (Expected Improvement): computes the expected magnitude of improvement
over the current best. Balances probability and magnitude.
- LogEI (Log Expected Improvement): log-transformed variant of EI for
numerical stability. Same behavior as EI.
- PosMean (Posterior Mean): selects the point where the GP predicts the
lowest function value. Pure exploitation - ignores uncertainty entirely.
Empirically, this and other exploitation-focused AFs have
performed strongly across a wide range of benchmarks, including several
that are theoretically considered difficult -- do not assume
exploitation-focused AFs are only appropriate in "easy" or late-stage
settings.

BALANCED:
- TS (Thompson Sampling): draws a random sample from the GP posterior and
optimizes it. Exploration and exploitation are balanced stochastically.

EXPLORATION-focused:
- JES (Joint Entropy Search): selects points that maximally reduce joint
entropy about both the location and value of the global optimum.
- PES (Predictive Entropy Search): selects points that maximally reduce
entropy about the location of the global optimum.
- MES (Max-value Entropy Search): selects points that maximally reduce
entropy about the optimal function value.
- UCB (Upper Confidence Bound): optimizes mean + beta * std. Explicitly
trades off exploitation (mean) and exploration (std) via a tunable parameter.
- PosSTD (Posterior Standard Deviation): directly maximizes GP posterior
uncertainty. Pure exploration -- no exploitation component.
- KG (Knowledge Gradient): computes the expected improvement in the optimal
posterior mean after one observation. Considers the value of information
gained.
```

E.4 ACQUISITION-FUNCTION USAGE GUIDANCE

```
IMPORTANT: Exploration-oriented AFs (see category above) are best understood
```

```
as tools for quickly locating a promising region early in the search, not
as tools for escaping a local optimum late in the search. Empirically, these AFs can produce large, valuable jumps in incumbent quality when used
early, but continuing to favor them once budget progress is high and no further improvement is materializing is usually a mistake - at that point
exploitation-focused AFs tend to perform better. Favor exploration-oriented
AFs primarily when budget progress is low; be increasingly skeptical of them
as budget progress increases, regardless of stagnation.
```

E.5 SCORING INSTRUCTIONS

```
================================================
SCORING INSTRUCTIONS
================================================
At each step, weigh the available signals as follows:
- Budget progress, shortest distance, f_improvement, and lengthscale CV remain the primary signals for judging the current optimization state.
- Recent history (up to 3 past selections with their rewards) is an additional, secondary check -- it is useful mainly for catching cases where you are about to repeat reasoning that has demonstrably not been working, but it should not override multiple structural signals that agree with each other. Because it is only 1-3 data points, treat it as noisy: react to a repeated pattern across entries, not to any single entry in isolation.
- Your score should reflect a nuanced judgment based on the full context,
not a mechanical rule applied to one variable.
- CRITICAL -- avoid repeating a demonstrated failure pattern: if recent history shows the same AF (or AFs from the same category, e.g. exploration-focused AFs) appearing multiple times with consistently poor
rewards, do not keep assigning that AF or category the highest score based
on the same reasoning that already failed to help. A single poor reward, on
the other hand, is weak evidence on its own and should only modestly lower
your confidence, not reverse your judgment entirely.
- CRITICAL - for the single most recent history entry, explicitly re-check
its (SUCCESS)/(FAILURE)/(NEUTRAL) label and y_value before reasoning about
it. Do not summarize or paraphrase its outcome from a general impression
of "things have been going well/poorly" -- verify the label and number directly. A (FAILURE) label means failure regardless of how small the reward magnitude looks.

Then output a suitability score between 0.0 and 1.0 for EACH of the 12 AFs listed below.

CRITICAL SCORING RULES:
- Scores MUST be meaningfully differentiated based on the current context.
- Do NOT assign similar scores to all AFs.
- Exploration-oriented AFs should receive high scores primarily when budget
progress is low (early phase); as budget progress increases, their scores should generally decrease even under stagnation, in favor of
```

```
exploitation-focused AFs.
```

E.6 Output-Format Constraints

```
================================================
CRITICAL OUTPUT FORMAT RULE -- READ CAREFULLY
================================================
You MUST respond with ONLY a single valid JSON object. Nothing else.
- NO markdown headers (no ###, no **bold**, no horizontal rules ---)
- NO bullet points, NO tables, NO prose before or after the JSON
- NO code fences (no ```json or ```)
- Your entire response must be parseable directly as JSON, starting with
{ and ending with }

The JSON object must have EXACTLY these 12 keys. The example below shows
ONLY the required format/structure - the score value 0.5 is a placeholder
with no meaning and must not be treated as a suggested or default score
for any AF:

{"JES": {"score": 0.5, "justification": "brief reason"}, "PES": {"score": 0.5, "justification": "brief reason"}, "MES": {"score": 0.5, "justification": "brief reason"}, "KG": {"score": 0.5, "justification": "brief reason"}, "PI": {"score": 0.5, "justification": "brief reason"}, "LogPI": {"score": 0.5, "justification": "brief reason"}, "EI": {"score": 0.5, "justification": "brief reason"}, "LogEI": {"score": 0.5, "justification": "brief reason"}, "UCB": {"score": 0.5, "justification": "brief reason"}, "PosMean": {"score": 0.5, "justification": "brief reason"}, "PosSTD": {"score": 0.5, "justification": "brief reason"}, "TS": {"score": 0.5, "justification": "brief reason"}}

This rule applies to EVERY response, for every iteration, no matter how
many iterations have passed. Always output ONLY the JSON object with all
12 AF keys.

Firstly, just give a brief confirmation that you understand the task.
Even this confirmation message does not need to be JSON, but starting
from the NEXT message (the first real optimization state), every single
response must be ONLY the JSON object described above.
```

E.7 Iteration-Wise Format Reminder

```
[Reminder: respond with ONLY the JSON object, no markdown,
no headers, no prose, no code fences - just the raw JSON with all 12 AF keys.
Ensure score spread of at least 0.4 between highest and lowest.
If recent history shows a repeated pattern of an AF/category failing,
do not keep repeating that same reasoning -- but do not overreact to a
single data point either.]
```

E.8 Iteration-Wise Optimization State Prompt

At each BO iteration, the current optimization state is serialized and provided to the LLM together with the scoring prompt. The following template shows the state representation used in our experiments; the bracketed fields are populated with the corresponding values at each iteration.

```
Current optimization state:
- N: {N}
- D: {dim}
- Budget progress: {budget_progress:.2f} (0.0=start, 1.0=end) - early phase favors exploration; late phase favors exploitation
- Remaining iterations: {remaining}
- f_range: Range [{f_min:.4f}, {f_max:.4f}], Mean {f_mean:.4f} (Std Dev {f_std:.4f})
- f_min: {f_min:.4f}
- f_improvement: {f_improvement:.3f} - higher means optimization has progressed significantly
- Shortest distance: {shortest_dist:.4f} - small means clustering/over-exploitation; large means sufficient exploration
- Lengthscales: Range [{ls_min:.4f}, {ls_max:.4f}], Mean {ls_mean:.4f} (Std Dev {ls_std:.4f})
- Lengthscale CV: {ls_cv:.3f} - high means landscape varies significantly across dimensions
- Outputscale: {outputscale:.4f}
- Available AFs: {af_list}{af_history_text}{recent_history_text}
```

E.9 Recent AF Selection History

When previous AF-selection results are available, up to the three most recent selections are appended to the optimization-state prompt in the following format.

```
- Recent history (most recent first, up to last 3 selections, their actual observed values, and outcome labels - the (SUCCESS)/(FAILURE)/(NEUTRAL) label reflects the reward's sign exactly; do not re-interpret the sign yourself, trust the label):
  iter {iteration}: {af} selected, y_value={y_value:.4f}, reward={reward:.4f} ({label})
  ...
```

F LLM Ablation Study

To examine the sensitivity of LPG-UCB to the choice and scale of the underlying LLM, we conduct an LLM ablation study on 10 representative benchmarks selected from the full synthetic benchmark suite, as listed in Table 4. We evaluate LPG-UCB using Gemma 3 4B, Gemma 3 12B, Ministral 3 14B, and Qwen3 14B while keeping all other experimental settings unchanged. We also include the random-prior variant as a reference to assess the contribution of LLM-generated prior information.

As shown in Table 5, Gemma 3 12B, Ministral 3 14B, and Qwen3 14B achieve closely comparable performance, with mean RP values of 1.37, 1.33, and 1.36, respectively, and mean ranks ranging from 5.10 to 6.50. Their coefficients of variation are also similar, ranging from 0.26 to 0.35. Ministral 3 14B achieves the best overall performance among these models. The smaller Gemma 3 4B model shows a noticeable performance decrease, with a mean RP of 2.37 and a mean rank of 8.80. Nevertheless, it still outperforms the random-prior variant, which obtains a mean RP of 2.56 and a mean rank of 10.50. This result suggests that even a substantially smaller LLM can provide useful prior information for LPG-UCB beyond an uninformed random prior.

Table 5: Summary statistics for LLM-based methods and ablation variants across benchmarks.

Method	MeanRAUC (IQR)	P-value (RAUC)	MeanRank (Min-Max)	P-value (Rank)	CV of RAUC
Gemma3-12b	1.37 (1.13-1.40)	1.95e-03	5.60 (2-16)	1.95e-03	0.35
Ministral-3-14b	1.33 (1.21-1.32)	7.81e-03	5.10 (1-12)	7.81e-03	0.26
Qwen3-14b	1.36 (1.06-1.40)	3.91e-03	6.50 (1-19)	3.91e-03	0.33
Gemma3-4b	2.37 (1.19-1.70)	3.91e-03	8.80 (1-18)	3.91e-03	1.08
LLM_random_prior	2.56 (1.28-2.28)	1.95e-03	10.50 (3-20)	1.95e-03	1.02

Table 4: Synthetic benchmark functions used in the ablation study.

COCO-BBOB		BoTorch	
Benchmark	**Dim.**	**Benchmark**	**Dim.**
Bent Cigar	5	Ackley	5
Büche–Rastrigin	5	Hartmann6	6
Weierstrass	5	Six-Hump Camel	2
Gallagher 101	5	Griewank	9
Lunacek bi-Rastrigin	5	Rosenbrock	4

G Details of LPG-UCB strategy-selection dynamics analysis

Table 6: Overall AF selection frequency across all benchmarks and runs.

AF	Selection Count	Selection Frequency (%)
PosMean	9,629	55.02
PI	3,473	19.85
LogPI	2,797	15.98
JES	512	2.93
EI	422	2.41
LogEI	263	1.50
MES	157	0.90
KG	108	0.62
TS	82	0.47
UCB	31	0.18
PES	26	0.15
Total	17,500	100.00

Table 7: Successful LLM preference overrides across optimization stages.

Stage	Iterations	Improvements	Override Improvements	Ratio (%)
Early (0–30%)	5,250	1,156	615	53.20
Mid (30–60%)	5,250	1,009	631	62.54
Late (60–100%)	7,000	1,082	757	69.96
Total	17,500	3,247	2,003	61.69

H Regret Analysis

H.1 Problem Setting and Notation

Let

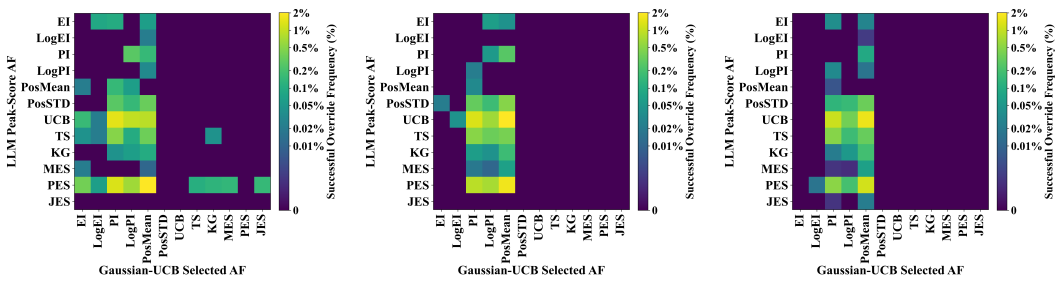
$$[K] := \{1, \dots, K\}$$

denote the finite set of AF arms. At round t, $\mathcal{A}_t \subseteq [K]$ denotes the active set and $I_t \in \mathcal{A}_t$ the arm selected by the Gaussian-UCB selector. The number of selector-reward updates received by arm i before round t is

$$A_{t,i} := \sum_{s<t} \mathbf{1}\{i \in \mathcal{A}_s\}.$$

The regret analysis considers the full active-set setting,

$$\mathcal{A}_t = [K], \qquad \forall t,$$



(a) Early stage (0–30%). (b) Middle stage (30–60%). (c) Late stage (60–100%).

Figure 3: Successful LLM preference overrides across representative stages of the optimization process. Each heatmap reports the frequency of decisions in which the Gaussian-UCB selector selected an AF outside the set of highest-scoring LLM preferences and improved the incumbent solution.

under which every arm receives either observed or GP-based pseudo-feedback at each completed round and therefore

$$A_{t,i} = t - 1.$$

Let $\mathcal{F}_{t-1}$ denote the information available before round t. The conditional expected selector reward of arm i is

$$\theta_{t,i} := \mathbb{E}\left[r_i^{(t)} \mid \mathcal{F}_{t-1}\right].$$

Since the selector reward is clipped to $[-c, c]$, we have

$$\theta_{t,i} \in [-c, c].$$

At each round, define the dynamic comparator

$$i_t^\star \in \arg\max_{i \in [K]} \theta_{t,i}.$$

The instantaneous and cumulative dynamic regrets are defined as:

$$\Delta_t := \theta_{t,i_t^\star} - \theta_{t,I_t}, \qquad R_T := \sum_{t=1}^{T} \Delta_t.$$

H.2 Assumptions

Assumption 1 (Restless reward evolution)**.** *The conditional expected selector reward of each arm may evolve with the shared BO state even when that arm is not selected. Specifically,*

$$\theta_{t,i} = \Theta_i(S_t),$$

where S_t denotes the BO state at round t. Consequently, a new objective observation may alter the expected selector rewards of multiple arms through the resulting change in S_t.

Assumption 2 (Rotting expected selector rewards)**.** *For each arm i, the conditional expected selector reward is nonincreasing:*

$$\theta_{t+1,i} \le \theta_{t,i}, \qquad \forall t.$$

Equivalently, defining the one-step decay

$$d_{t,i} := \theta_{t,i} - \theta_{t+1,i},$$

we have

$$d_{t,i} \ge 0.$$

This assumption models the decreasing remaining improvement potential as the optimization progresses.

Assumption 3 (Conditional sub-Gaussian noise)**.** *For each arm i and round t, the selector reward satisfies*

$$r_i^{(t)} = \theta_{t,i} + \xi_{t,i},$$

where

$$\mathbb{E}[\xi_{t,i} \mid \mathcal{F}_{t-1}] = 0.$$

Conditionally on $\mathcal{F}_{t-1}$, the residual is assumed to be σ_r-sub-Gaussian:

$$\mathbb{E}[\exp(\lambda \xi_{t,i}) \mid \mathcal{F}_{t-1}] \le \exp\left(\frac{\lambda^2 \sigma_r^2}{2}\right), \qquad \forall \lambda \in \mathbb{R}.$$

Here, σ_r^2 denotes a common variance proxy across arms and rounds.

Assumption 4 (Relative LLM-prior quality)**.** *Let*

$$U_{t,i} \sim \mathrm{Unif}(0,1)$$

be independent of $\mathcal{F}_{t-1}$. We assume that there exists $\gamma \in [0,1]$ such that

$$\mathbb{E}\left[\left|q_{t,i}^{\mathrm{LLM}} - \theta_{t,i}\right| \mid \mathcal{F}_{t-1}\right] \le \gamma\, \mathbb{E}[|U_{t,i} - \theta_{t,i}| \mid \mathcal{F}_{t-1}].$$

Thus, the LLM prior is assumed to provide more accurate arm-specific guidance than a uniform random prior in expectation.

H.3 Historical Mean and Reward-Error Decomposition

Under the full active-set algorithmic setting defined above, every AF arm receives either actual feedback or GP-based pseudo-feedback at every completed round. Hence, before round t, the empirical reward estimate for arm i uses $t-1$ feedback values:

$$\widehat{\theta}_{t,i} = \frac{1}{t-1} \sum_{s=1}^{t-1} r_i^{(s)}. \tag{6}$$

Using Assumption 3,

$$r_i^{(s)} = \theta_{s,i} + \xi_{s,i},$$

so

$$\begin{aligned} \widehat{\theta}_{t,i} &= \frac{1}{t-1} \sum_{s=1}^{t-1} (\theta_{s,i} + \xi_{s,i}) \\ &= \underbrace{\frac{1}{t-1} \sum_{s=1}^{t-1} \theta_{s,i}}_{\text{historical mean of true expected rewards}} + \underbrace{\frac{1}{t-1} \sum_{s=1}^{t-1} \xi_{s,i}}_{\text{average feedback noise}}. \end{aligned}$$

Define the historical average of the true expected selector rewards by

$$\bar{\theta}_{t,i} := \frac{1}{t-1} \sum_{s=1}^{t-1} \theta_{s,i}. \tag{7}$$

Then

$$\widehat{\theta}_{t,i} - \bar{\theta}_{t,i} = \frac{1}{t-1} \sum_{s=1}^{t-1} \xi_{s,i}. \tag{8}$$

This is the statistical part of the estimation error. It measures how far the empirical reward average deviates from the historical true-mean average due to feedback noise.

The quantity we ultimately need for regret analysis, however, is the error to the current expected reward $\theta_{t,i}$ rather than to the historical mean. Add and subtract $\bar{\theta}_{t,i}$:

$$\widehat{\theta}_{t,i} - \theta_{t,i} = \left(\widehat{\theta}_{t,i} - \bar{\theta}_{t,i}\right) + \left(\bar{\theta}_{t,i} - \theta_{t,i}\right).$$

Under Assumption 2,

$$\theta_{1,i} \ge \theta_{2,i} \ge \cdots \ge \theta_{t,i},$$

so all historical means are at least as large as the current mean and therefore

$$\bar{\theta}_{t,i} \geq \theta_{t,i}. \tag{9}$$

Define the resulting stale-history, or rotting bias as

$$b^{\mathrm{rot}}_{t,i} := \bar{\theta}_{t,i} - \theta_{t,i} \geq 0. \tag{10}$$

Combining the two pieces gives the central reward-error decomposition

$$\widehat{\theta}_{t,i} - \theta_{t,i} = \underbrace{\frac{1}{t-1}\sum_{s=1}^{t-1} \xi_{s,i}}_{\text{statistical fluctuation}} + \underbrace{b^{\mathrm{rot}}_{t,i}}_{\text{nonstationary rotting bias}}. \tag{11}$$

The statistical fluctuation is controlled using Assumption 3, while the rotting bias is handled using the nonstationary reward structure.

H.4 Deriving the Statistical Confidence Multiplier

Assumption 3 directly provides the conditional moment-generating-function (MGF) bound

$$\mathbb{E}[\exp(\lambda \xi_{s,i}) \mid \mathcal{F}_{s-1}] \leq \exp\left(\frac{\lambda^2 \sigma_r^2}{2}\right), \qquad \lambda \in \mathbb{R}. \tag{12}$$

This bound is sufficient to control the failure probability of the statistical confidence event. For $u > 0$ and $\lambda > 0$, Markov's inequality gives

$$\begin{aligned}
\Pr(\xi_{s,i} \geq u \mid \mathcal{F}_{s-1}) &= \Pr(\exp(\lambda \xi_{s,i}) \geq \exp(\lambda u) \mid \mathcal{F}_{s-1}) \\
&\leq \exp(-\lambda u)\mathbb{E}[\exp(\lambda \xi_{s,i}) \mid \mathcal{F}_{s-1}] \\
&\leq \exp\left(-\lambda u + \frac{\lambda^2 \sigma_r^2}{2}\right).
\end{aligned}$$

Choosing

$$\lambda^\star = \frac{u}{\sigma_r^2}$$

yields

$$\Pr(\xi_{s,i} \geq u \mid \mathcal{F}_{s-1}) \leq \exp\left(-\frac{u^2}{2\sigma_r^2}\right).$$

Applying the same argument to $-\xi_{s,i}$ gives the two-sided sub-Gaussian tail bound

$$\Pr(|\xi_{s,i}| \geq u \mid \mathcal{F}_{s-1}) \leq 2\exp\left(-\frac{u^2}{2\sigma_r^2}\right). \tag{13}$$

For the empirical mean, define the accumulated residual

$$Z_{t-1,i} := \sum_{s=1}^{t-1} \xi_{s,i}. \tag{14}$$

Because $\{\xi_{s,i}\}$ is a martingale-difference sequence satisfying the conditional MGF bound, iterated conditioning gives

$$\mathbb{E}[\exp(\lambda Z_{t-1,i})] \leq \exp\left(\frac{\lambda^2 (t-1)\sigma_r^2}{2}\right). \tag{15}$$

Thus $Z_{t-1,i}$ is sub-Gaussian with variance proxy $(t-1)\sigma_r^2$, which gives

$$\Pr(|Z_{t-1,i}| \geq (t-1)u) \leq 2\exp\left(-\frac{(t-1)u^2}{2\sigma_r^2}\right).$$

Using

$$\widehat{\theta}_{t,i} - \bar{\theta}_{t,i} = \frac{Z_{t-1,i}}{t-1},$$

we obtain, for each fixed (t, i),

$$\Pr\left(\left|\widehat{\theta}_{t,i} - \bar{\theta}_{t,i}\right| > u\right) \leq 2\exp\left(-\frac{(t-1)u^2}{2\sigma_r^2}\right). \tag{16}$$

Here $\delta \in (0, 1)$ denotes the total allowed failure probability for the simultaneous confidence statement over all rounds and all arms. We allocate a smaller failure budget $\delta_{t,i}$ to each pair (t, i) and require

$$\sum_{t=1}^{\infty}\sum_{i=1}^{K}\delta_{t,i} \leq \delta.$$

A convenient choice is

$$\delta_{t,i} = \frac{6\delta}{K\pi^2 t^2}. \tag{17}$$

Indeed, using $\sum_{t=1}^{\infty} t^{-2} = \pi^2/6$,

$$\sum_{t=1}^{\infty}\sum_{i=1}^{K}\delta_{t,i} = K\sum_{t=1}^{\infty}\frac{6\delta}{K\pi^2 t^2} = \delta.$$

For a fixed pair (t, i), choose the confidence radius so that the tail probability is at most $\delta_{t,i}$. Setting the tail bound equal to $\delta_{t,i}$,

$$2\exp\left(-\frac{(t-1)u^2}{2\sigma_r^2}\right) = \delta_{t,i},$$

and solving for u gives

$$u_{t,i} := \frac{\sigma_r}{\sqrt{t-1}}\sqrt{2\log\left(\frac{2}{\delta_{t,i}}\right)}. \tag{18}$$

Then

$$\Pr\left(\left|\widehat{\theta}_{t,i} - \bar{\theta}_{t,i}\right| > u_{t,i}\right) \leq \delta_{t,i}.$$

Substituting $\delta_{t,i} = 6\delta/(K\pi^2 t^2)$, we define

$$\alpha_t := \sqrt{2\log\left(\frac{K\pi^2 t^2}{3\delta}\right)}. \tag{19}$$

Therefore

$$u_{t,i} = \alpha_t\frac{\sigma_r}{\sqrt{t-1}}. \tag{20}$$

Finally, the union bound over all (t, i) pairs implies that, with probability at least $1 - \delta$, simultaneously for all arms and rounds,

$$\left|\widehat{\theta}_{t,i} - \bar{\theta}_{t,i}\right| \leq \alpha_t\frac{\sigma_r}{\sqrt{t-1}}. \tag{21}$$

H.5 Exact Rotting-Bias Representation

Because

$$d_{u,i} = \theta_{u,i} - \theta_{u+1,i},$$

for $s < t$,

$$\theta_{s,i} - \theta_{t,i} = \sum_{u=s}^{t-1} d_{u,i}.$$

Therefore

$$\begin{aligned} b_{t,i}^{\mathrm{rot}} &= \frac{1}{t-1}\sum_{s=1}^{t-1}(\theta_{s,i} - \theta_{t,i}) \\ &= \frac{1}{t-1}\sum_{s=1}^{t-1}\sum_{u=s}^{t-1} d_{u,i} \\ &= \frac{1}{t-1}\sum_{u=1}^{t-1} u\, d_{u,i}. \end{aligned}$$

Thus a full-history estimator carries a weighted memory of all previous reward decays. Old decay events are not forgotten automatically.

On the high-probability event,

$$-\alpha_t \frac{\sigma_r}{\sqrt{t-1}} \le \widehat{\theta}_{t,i} - \bar{\theta}_{t,i} \le \alpha_t \frac{\sigma_r}{\sqrt{t-1}}, \tag{22}$$

where the rotting bias satisfies

$$\bar{\theta}_{t,i} = \theta_{t,i} + b^{\mathrm{rot}}_{t,i}, \qquad b^{\mathrm{rot}}_{t,i} \ge 0.$$

The left side of the confidence event gives

$$\widehat{\theta}_{t,i} \ge \bar{\theta}_{t,i} - \alpha_t \frac{\sigma_r}{\sqrt{t-1}} = \theta_{t,i} + b^{\mathrm{rot}}_{t,i} - \alpha_t \frac{\sigma_r}{\sqrt{t-1}}.$$

Rearranging yields the tighter upper bound

$$\theta_{t,i} \le \widehat{\theta}_{t,i} + \alpha_t \frac{\sigma_r}{\sqrt{t-1}} - b^{\mathrm{rot}}_{t,i}.$$

Since $b^{\mathrm{rot}}_{t,i} \ge 0$, dropping the nonpositive term $-b^{\mathrm{rot}}_{t,i}$ gives the simpler optimistic upper bound

$$\theta_{t,i} \le \widehat{\theta}_{t,i} + \alpha_t \frac{\sigma_r}{\sqrt{t-1}}. \tag{23}$$

For the lower bound, use the right side of the confidence event:

$$\widehat{\theta}_{t,i} \le \bar{\theta}_{t,i} + \alpha_t \frac{\sigma_r}{\sqrt{t-1}} = \theta_{t,i} + b^{\mathrm{rot}}_{t,i} + \alpha_t \frac{\sigma_r}{\sqrt{t-1}}.$$

Rearranging gives

$$\theta_{t,i} \ge \widehat{\theta}_{t,i} - \alpha_t \frac{\sigma_r}{\sqrt{t-1}} - b^{\mathrm{rot}}_{t,i}. \tag{24}$$

Unlike the upper bound, the lower bound retains the rotting-bias term because the historical mean may exceed the current expected reward.

H.6 Posterior Error Bounds

Under the full active-set setting,

$$A_{t,i} = t-1, \qquad w_{t,i} = \frac{a_i}{a_i + t - 1}, \qquad \sigma^{\mathrm{post}}_{t,i} = \frac{\sigma_r}{\sqrt{t-1+a_i}}.$$

Using

$$\mu^{\mathrm{post}}_{t,i} = w_{t,i} q^{\mathrm{LLM}}_{t,i} + (1 - w_{t,i}) \widehat{\theta}_{t,i},$$

together with

$$\widehat{\theta}_{t,i} - \theta_{t,i} = \left(\widehat{\theta}_{t,i} - \bar{\theta}_{t,i}\right) + b^{\mathrm{rot}}_{t,i},$$

the posterior error can be written as

$$\begin{aligned} \mu^{\mathrm{post}}_{t,i} - \theta_{t,i} = w_{t,i} \left(q^{\mathrm{LLM}}_{t,i} - \theta_{t,i}\right) \\ + (1 - w_{t,i}) \left(\widehat{\theta}_{t,i} - \bar{\theta}_{t,i}\right) + (1 - w_{t,i}) b^{\mathrm{rot}}_{t,i}. \end{aligned} \tag{25}$$

Define the LLM prior error relative to the current expected selector reward as

$$e^{\mathrm{LLM}}_{t,i} := \left| q^{\mathrm{LLM}}_{t,i} - \theta_{t,i} \right|. \tag{26}$$

On the simultaneous confidence event,

$$\left| \widehat{\theta}_{t,i} - \bar{\theta}_{t,i} \right| \le \alpha_t \frac{\sigma_r}{\sqrt{t-1}}.$$

Moreover,

$$(1-w_{t,i})\frac{\sigma_r}{\sqrt{t-1}} = \sqrt{1-w_{t,i}}\,\sigma_{t,i}^{\mathrm{post}} \leq \sigma_{t,i}^{\mathrm{post}}. \tag{27}$$

Therefore,

$$(1-w_{t,i})\left|\widehat{\theta}_{t,i}-\bar{\theta}_{t,i}\right| \leq \alpha_t \sigma_{t,i}^{\mathrm{post}}. \tag{28}$$

Applying this bound to (25), and using $b_{t,i}^{\mathrm{rot}} \geq 0$, yields the upper bound

$$\theta_{t,i} \leq \mu_{t,i}^{\mathrm{post}} + \alpha_t \sigma_{t,i}^{\mathrm{post}} + w_{t,i} e_{t,i}^{\mathrm{LLM}}. \tag{29}$$

Similarly, the lower bound is

$$\theta_{t,i} \geq \mu_{t,i}^{\mathrm{post}} - \alpha_t \sigma_{t,i}^{\mathrm{post}} - w_{t,i} e_{t,i}^{\mathrm{LLM}} - (1-w_{t,i}) b_{t,i}^{\mathrm{rot}}. \tag{30}$$

Unlike the upper bound, the lower bound retains the rotting-bias term because the historical mean may exceed the current expected reward.

H.7 Gaussian-UCB Regret Decomposition

Because I_t maximizes the Gaussian-UCB score over the active arms,

$$\mathrm{UCB}_{I_t}(t) \geq \mathrm{UCB}_{i_t^\star}(t).$$

Using

$$\mathrm{UCB}_i(t) = \mu_{t,i}^{\mathrm{post}} + \beta_t \sigma_{t,i}^{\mathrm{post}},$$

we have

$$\mu_{t,I_t}^{\mathrm{post}} + \beta_t \sigma_{t,I_t}^{\mathrm{post}} \geq \mu_{t,i_t^\star}^{\mathrm{post}} + \beta_t \sigma_{t,i_t^\star}^{\mathrm{post}}.$$

Rearranging gives

$$\mu_{t,i_t^\star}^{\mathrm{post}} - \mu_{t,I_t}^{\mathrm{post}} \leq \beta_t \left(\sigma_{t,I_t}^{\mathrm{post}} - \sigma_{t,i_t^\star}^{\mathrm{post}}\right). \tag{31}$$

Recall that the instantaneous dynamic regret is

$$\Delta_t = \theta_{t,i_t^\star} - \theta_{t,I_t}.$$

The one-sided posterior bounds derived above give, for the comparator arm,

$$\theta_{t,i_t^\star} \leq \mu_{t,i_t^\star}^{\mathrm{post}} + \alpha_t \sigma_{t,i_t^\star}^{\mathrm{post}} + w_{t,i_t^\star} e_{t,i_t^\star}^{\mathrm{LLM}},$$

whereas for the selected arm,

$$\theta_{t,I_t} \geq \mu_{t,I_t}^{\mathrm{post}} - \alpha_t \sigma_{t,I_t}^{\mathrm{post}} - w_{t,I_t} e_{t,I_t}^{\mathrm{LLM}} - (1-w_{t,I_t}) b_{t,I_t}^{\mathrm{rot}}.$$

Subtracting the lower bound on θ_{t,I_t} from the upper bound on $\theta_{t,i_t^\star}$ therefore yields

$$\begin{aligned}\Delta_t &= \theta_{t,i_t^\star} - \theta_{t,I_t} \\ &\leq (\mu_{t,i_t^\star}^{\mathrm{post}} - \mu_{t,I_t}^{\mathrm{post}}) + \alpha_t \left(\sigma_{t,i_t^\star}^{\mathrm{post}} + \sigma_{t,I_t}^{\mathrm{post}}\right) \\ &\quad + w_{t,i_t^\star} e_{t,i_t^\star}^{\mathrm{LLM}} + w_{t,I_t} e_{t,I_t}^{\mathrm{LLM}} + (1-w_{t,I_t}) b_{t,I_t}^{\mathrm{rot}}.\end{aligned}$$

Substituting (31) and collecting terms yields

$$\begin{aligned}\Delta_t \leq\ & (\alpha_t+\beta_t)\sigma_{t,I_t}^{\mathrm{post}} + (\alpha_t-\beta_t)\sigma_{t,i_t^\star}^{\mathrm{post}} \\ & + w_{t,I_t} e_{t,I_t}^{\mathrm{LLM}} + w_{t,i_t^\star} e_{t,i_t^\star}^{\mathrm{LLM}} + (1-w_{t,I_t}) b_{t,I_t}^{\mathrm{rot}}.\end{aligned} \tag{32}$$

Using $(\alpha_t-\beta_t) \leq (\alpha_t-\beta_t)_+$ and $(1-w_{t,I_t}) b_{t,I_t}^{\mathrm{rot}} \leq b_{t,I_t}^{\mathrm{rot}}$, we obtain

$$\begin{aligned}\Delta_t \leq\ & (\alpha_t+\beta_t)\sigma_{t,I_t}^{\mathrm{post}} + (\alpha_t-\beta_t)_+\sigma_{t,i_t^\star}^{\mathrm{post}} \\ & + w_{t,I_t} e_{t,I_t}^{\mathrm{LLM}} + w_{t,i_t^\star} e_{t,i_t^\star}^{\mathrm{LLM}} + b_{t,I_t}^{\mathrm{rot}},\end{aligned} \tag{33}$$

where $(x)_+ := \max\{x, 0\}$.

Summing over $t = 2, \dots, T$ gives

$$R_T \le \Delta_1 + R_T^{\text{selector}} + R_T^{\text{rot}} + R_T^{\text{LLM}}, \tag{34}$$

where

$$R_T^{\text{selector}} := \sum_{t=2}^{T} \left[(\alpha_t + \beta_t)\sigma_{t,I_t}^{\text{post}} + (\alpha_t - \beta_t)_+ \sigma_{t,i_t^\star}^{\text{post}} \right],$$

$$R_T^{\text{rot}} := \sum_{t=2}^{T} b_{t,I_t}^{\text{rot}},$$

and

$$R_T^{\text{LLM}} := \sum_{t=2}^{T} \left[w_{t,I_t} e_{t,I_t}^{\text{LLM}} + w_{t,i_t^\star} e_{t,i_t^\star}^{\text{LLM}} \right].$$

Since $\Delta_1 \le 2c$, the first-round contribution is $O(1)$.

H.8 Selector-Uncertainty Contribution

Under the full active-set setting, every arm has

$$\sigma_{t,i}^{\text{post}} = \frac{\sigma_r}{\sqrt{t - 1 + a_i}} \le \frac{\sigma_r}{\sqrt{t-1}}.$$

Therefore, for either the selected arm I_t or the dynamic comparator $i_t^\star$,

$$\sum_{t=2}^{T} \sigma_{t,\cdot}^{\text{post}} \le \sigma_r \sum_{t=2}^{T} \frac{1}{\sqrt{t-1}}.$$

Reindexing with $n = t - 1$ gives

$$\sum_{t=2}^{T} \frac{1}{\sqrt{t-1}} = \sum_{n=1}^{T-1} \frac{1}{\sqrt{n}}.$$

Since $x^{-1/2}$ is positive and decreasing,

$$\sum_{n=1}^{T-1} \frac{1}{\sqrt{n}} \le 1 + \int_1^{T-1} x^{-1/2}\, dx = 2\sqrt{T-1} - 1 = O(\sqrt{T}).$$

Hence

$$\sum_{t=2}^{T} \sigma_{t,I_t}^{\text{post}} = O(\sigma_r \sqrt{T}), \qquad \sum_{t=2}^{T} \sigma_{t,i_t^\star}^{\text{post}} = O(\sigma_r \sqrt{T}). \tag{35}$$

Because α_t is nondecreasing and $0 \le \beta_t \le \beta_{\max}$,

$$\alpha_t + \beta_t \le \alpha_T + \beta_{\max}, \qquad (\alpha_t - \beta_t)_+ \le \alpha_T.$$

Therefore

$$R_T^{\text{selector}} = O\Big((\alpha_T + \beta_{\max}) \sigma_r \sqrt{T} \Big). \tag{36}$$

Since $\alpha_T = O(\sqrt{\log(KT/\delta)})$, this contribution is sublinear in T for fixed K.

H.9 Asymptotic Rotting Contribution

Define

$$b_t^{\max} := \max_i b_{t,i}^{\text{rot}}.$$

Then

$$R_T^{\text{rot}} \le \sum_{t=2}^{T} b_t^{\max}.$$

For each arm, $\theta_{t,i}$ is monotone nonincreasing and bounded below by $-c$. Therefore

$$\theta_{t,i} \to \theta_{\infty,i}. \tag{37}$$

By Cesàro convergence,

$$\bar{\theta}_{t,i} \to \theta_{\infty,i}.$$

Hence

$$b_{t,i}^{\mathrm{rot}} = \bar{\theta}_{t,i} - \theta_{t,i} \to 0. \tag{38}$$

Since K is finite,

$$b_t^{\max} \to 0,$$

and therefore

$$\sum_{t=2}^{T} b_t^{\max} = o(T). \tag{39}$$

Consequently,

$$R_T^{\mathrm{rot}} = o(T). \tag{40}$$

This establishes pointwise asymptotic no regret for any fixed reward trajectory satisfying bounded monotone rotting.

Remark 1 (Important uniformity caveat)**.** *The result above is an asymptotic statement for a fixed infinite reward sequence. It does not provide a uniform minimax rate over all horizon-dependent rotting environments.*

To see the distinction, consider a horizon-dependent sequence of the form

$$\theta_t = \begin{cases} 1, & t \le T/2, \\ 0, & t > T/2. \end{cases}$$

For any round $t > T/2$, the current reward level is already

$$\theta_t = 0.$$

However, the full-history mean still contains all earlier observations for which the expected reward was equal to 1*. In particular,*

$$\bar{\theta}_t = \frac{1}{t-1} \sum_{s=1}^{t-1} \theta_s \approx \frac{T/2}{t-1}.$$

Therefore the rotting bias is approximately

$$b_t^{\mathrm{rot}} = \bar{\theta}_t - \theta_t \approx \frac{T/2}{t-1}.$$

For example, at $t = 3T/4$,

$$b_t^{\mathrm{rot}} \approx \frac{T/2}{3T/4} = \frac{2}{3},$$

which is still of constant order. Hence a fixed-size drop occurring around $T/2$ can leave the full-history estimator with an $O(1)$ stale bias for $O(T)$ subsequent rounds.

Thus, although $b_{t,i}^{\mathrm{rot}} \to 0$ for each fixed infinite reward sequence, this pointwise convergence does not imply a horizon-uniform sublinear rate. An explicit bound on the cumulative rotting contribution requires an additional quantitative condition on how quickly the one-step decays $d_{t,i}$ vanish.

H.10 LLM PRIOR CONTRIBUTION

Let

$$U \sim \mathrm{Unif}(0,1)$$

and define

$$G(\theta) := \mathbb{E}_U|U-\theta| = \int_0^1 |u-\theta|\,du.$$

A direct calculation gives

$$G(\theta) = \begin{cases} \frac{1}{2} - \theta, & \theta < 0, \\ \frac{\theta^2+(1-\theta)^2}{2}, & 0 \le \theta \le 1, \\ \theta - \frac{1}{2}, & \theta > 1. \end{cases} \tag{41}$$

Since the clipped selector reward satisfies $\theta_{t,i} \in [-c, c]$, define

$$C_{\mathrm{rand}} := \sup_{\theta \in [-c,c]} G(\theta). \tag{42}$$

For $c \ge 1$,

$$C_{\mathrm{rand}} = c + \frac{1}{2}. \tag{43}$$

In particular, for the implemented clipping level $c = 2$,

$$C_{\mathrm{rand}} = \frac{5}{2}. \tag{44}$$

By Assumption 4,

$$\mathbb{E}\left[e_{t,i}^{\mathrm{LLM}} \mid \mathcal{F}_{t-1}\right] \le \gamma\,\mathbb{E}\left[|U_{t,i} - \theta_{t,i}| \mid \mathcal{F}_{t-1}\right] = \gamma G(\theta_{t,i}) \le \gamma C_{\mathrm{rand}}.$$

Thus γC_{rand} provides a uniform expected-error bound for the LLM prior under Assumption 4.

Because the selected arm I_t depends on the LLM scores, we do not factor the conditional expectation through the random selected index. Instead, use the pathwise bound

$$w_{t,I_t} e_{t,I_t}^{\mathrm{LLM}} \le \sum_{i=1}^{K} w_{t,i} e_{t,i}^{\mathrm{LLM}},$$

and the same bound for the comparator term. Hence

$$R_T^{\mathrm{LLM}} \le 2 \sum_{t=2}^{T} \sum_{i=1}^{K} w_{t,i} e_{t,i}^{\mathrm{LLM}}.$$

Taking conditional expectation and using Assumption 4 gives

$$\mathbb{E}[R_T^{\mathrm{LLM}}] \le 2\gamma C_{\mathrm{rand}} \sum_{t=2}^{T} \sum_{i=1}^{K} w_{t,i}.$$

Under the full active-set setting,

$$w_{t,i} = \frac{a_i}{a_i + t - 1}.$$

Therefore

$$\sum_{t=2}^{T} w_{t,i} = \sum_{n=1}^{T-1} \frac{a_i}{a_i + n}.$$

Since $a_i/(a_i + x)$ is positive and decreasing,

$$\sum_{n=1}^{T-1} \frac{a_i}{a_i + n} \le \int_0^{T-1} \frac{a_i}{a_i + x}\,dx = a_i \log\left(1 + \frac{T-1}{a_i}\right).$$

Consequently,

$$\mathbb{E}[R_T^{\mathrm{LLM}}] = O\left(\gamma C_{\mathrm{rand}} \sum_{i=1}^{K} a_i \log\left(1 + \frac{T}{a_i}\right)\right). \tag{45}$$

If $a_i = a$ for all arms, this simplifies to

$$\mathbb{E}[R_T^{\mathrm{LLM}}] = O\left(\gamma C_{\mathrm{rand}} K a \log\left(1 + \frac{T}{a}\right)\right). \tag{46}$$

Thus the LLM contribution grows only logarithmically in T under Assumption 4 and scales linearly with the relative quality coefficient γ. In particular, a smaller γ yields a smaller bound on the LLM-prior contribution. No uniform arm-coverage assumption is needed in the full active-set analysis because every active arm receives an update at each completed round.

H.11 Regret Bound

Since $\theta_{t,i} \in [-c, c]$, the instantaneous regret satisfies

$$0 \le \Delta_t = \theta_{t,i_t^\star} - \theta_{t,I_t} \le 2c.$$

In particular,

$$\Delta_1 \le 2c.$$

Therefore, on the simultaneous confidence event $\mathcal{E}_\delta$, the regret decomposition for $t \ge 2$ gives

$$R_T \le 2c + R_T^{\mathrm{selector}} + R_T^{\mathrm{rot}} + R_T^{\mathrm{LLM}}.$$

Suppose that Assumptions 1–4 hold under the full active-set setting $A_{t,i} = t - 1$, and let

$$\Pr(\mathcal{E}_\delta) \ge 1 - \delta.$$

Combining the above bounds and accounting for the complement event $\mathcal{E}_\delta^c$ yields the expected cumulative regret bound stated in Theorem 1:

$$\begin{aligned}\mathbb{E}[R_T] = O\Bigg((\alpha_T + \beta_{\max})\sigma_r\sqrt{T} + \mathbb{E}[R_T^{\mathrm{rot}}] \\ + \gamma C_{\mathrm{rand}} \sum_{i=1}^{K} a_i \log\left(1 + \frac{T}{a_i}\right) + 2cT\delta\Bigg).\end{aligned} \tag{47}$$

Proof. On $\mathcal{E}_\delta$,

$$R_T \le 2c + R_T^{\mathrm{selector}} + R_T^{\mathrm{rot}} + R_T^{\mathrm{LLM}},$$

where

$$R_T^{\mathrm{selector}} = O\Big((\alpha_T + \beta_{\max})\sigma_r\sqrt{T}\Big)$$

and

$$\mathbb{E}[R_T^{\mathrm{LLM}}] = O\left(\gamma C_{\mathrm{rand}} \sum_{i=1}^{K} a_i \log\left(1 + \frac{T}{a_i}\right)\right).$$

On the complement $\mathcal{E}_\delta^c$, the deterministic bound $R_T \le 2cT$ gives

$$\mathbb{E}\big[R_T \mathbf{1}_{\mathcal{E}_\delta^c}\big] \le 2cT\delta.$$

Combining the two events yields (47). □

For fixed K, C_{rand}, a_i, γ, and $\beta_{\max}$, choosing $\delta = T^{-2}$ gives

$$2cT\delta = O(T^{-1}), \qquad \alpha_T = O(\sqrt{\log(KT)}).$$

Under the bounded monotone rotting assumption, $R_T^{\mathrm{rot}}/T \to 0$ for every fixed reward trajectory. Since $0 \le R_T^{\mathrm{rot}}/T \le 2c$, dominated convergence gives

$$\frac{\mathbb{E}[R_T^{\mathrm{rot}}]}{T} \to 0.$$

Consequently, every term in the expected regret bound is sublinear in T, and

$$\mathbb{E}[R_T] = o(T).$$

This establishes the sublinear expected cumulative regret result stated in Corollary 1.

Corollary 2 (Explicit rotting-regret rates)**.** *Suppose, in addition to the assumptions of Theorem 1, that the one-step reward decay satisfies*

$$d_{t,i} \le C_i t^{-a}, \qquad a > 1.$$

Let $C = \max_{i\in[K]} C_i$*. Then*

$$b_{t,i}^{\mathrm{rot}} = \begin{cases} O(t^{1-a}), & 1 < a < 2, \\ O((\log t)/t), & a = 2, \\ O(1/t), & a > 2, \end{cases}$$

and consequently

$$R_T^{\mathrm{rot}} = \begin{cases} O(T^{2-a}), & 1 < a < 2, \\ O((\log T)^2), & a = 2, \\ O(\log T), & a > 2. \end{cases}$$

Proof. Using

$$b_{t,i}^{\mathrm{rot}} = \frac{1}{t-1} \sum_{u=1}^{t-1} u\, d_{u,i},$$

the decay envelope gives

$$b_{t,i}^{\mathrm{rot}} \le \frac{C}{t-1} \sum_{u=1}^{t-1} u^{1-a}.$$

The stated rates follow from the standard asymptotic behavior of this power sum for $1 < a < 2$, $a = 2$, and $a > 2$, followed by summation over t. □