Efficient Cost-Aware LLM Evaluation via Bayesian Bandit Gittins Indices

Qian Xie[1], Yueli He[2], Nairen Cao[2]
[1]Cornell University [2]New York University

Motivation

One problem in LLM evaluation is using benchmark examples to search for a good task-specific configuration.

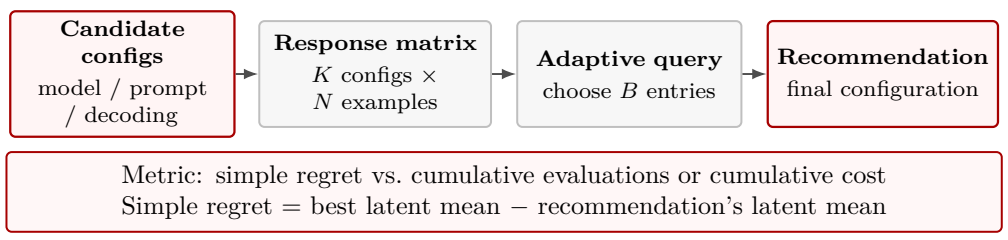
- LLM configuration: model, prompting technique, temperature, decoding strategy, etc.
- Brute-force evaluation scores every configuration on every benchmark example.
- Configuration-level full evaluation may query only selected candidates, but still uses all examples per query.
- Zhou et al. (2025) first study the bandit formulation, but UCB-E/LRF are frequentist-style and cost-unaware.

Goal: recommend a strong configuration with far lower evaluation cost.

Premise: the latent target is mean accuracy over all benchmark examples; batch means are noisy observations.

Problem Setup

Cost-aware LLM configuration recommendation



Setup: configurations have latent means; each iteration queries B response-matrix entries (B = batch size); c_k encodes query cost (e.g., token prices).

Evaluation ends at the 10% cap unless a stopping rule triggers earlier.

Bayesian Bandit Gittins Policy

Bandit model: arm = LLM configuration; pull = evaluations of one or a batch of examples for the chosen arm.

Bayesian updates: with new batch mean y_{n+1} of a chosen arm, update its posterior mean μ_n using:

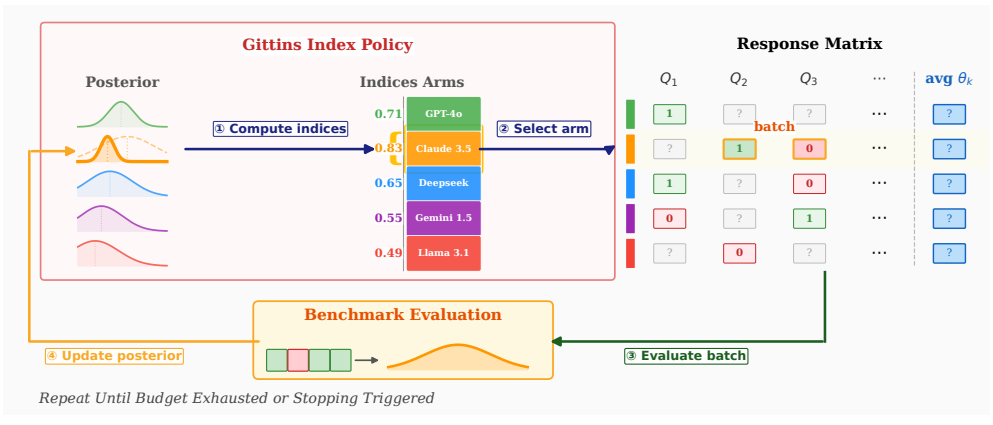
$$\mu_{n+1} = \mu_n + K_n(y_{n+1} - \mu_n), \qquad K_n = \frac{v_n}{v_n + \tau^2}, \qquad v_{n+1} = \left(v_n^{-1} + \tau^{-2}\right)^{-1}.$$

Efficient Gittins computation: Fixed observation noise τ^2 gives deterministic $\{K_n\}$, so we can precompute root schedule $\{r_n\}$ of dynamic program offline for each arm. Then the Gittins index at horizon n is:

$$G_n = \mu_n - r_n.$$

Policy: Evaluate the highest-index arm if uncompleted; otherwise, stop. Completed-arm index is posterior mean.

Gittins policy comes with both an arm-pull rule and a stopping rule.



Design Choices

Prior choices. Gittins-G: shared general prior $\mathcal{N}(0.5, 0.04)$.
Gittins-S: data-specific prior—easy $\mathcal{N}(0.8, 0.01)$; medium $\mathcal{N}(0.6, 0.02)$; hard $\mathcal{N}(0.4, 0.02)$; very hard $\mathcal{N}(0.2, 0.01)$.

Observation noise. Binary batch mean variance is $\theta_k(1-\theta_k)/B \le 1/(4B)$; default observation noise τ^2 uses this conservative bound. Plug-in noise or pilot-estimated noise can be alternatives.

Cost scaling. $c_k = \lambda \tilde{c}_k$; λ sets the simple-regret/cumulative cost trade-off.

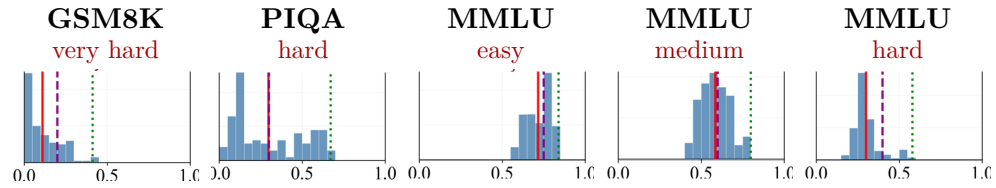
Recommendation. The original Gittins policy is required-completion; for LLM evaluation, optional-completion can be considered. Regret curves use posterior-mean recommendations before required-completion stopping.

Experimental Setup

Response matrices from three binary-correctness benchmarks:

GSM8K	PIQA	MMLU
very hard math	hard commonsense	57 subjects
122 configs	**103 configs**	**1500 configs**
1000 examples	1000 examples	varied examples

Gittins-S prior buckets:



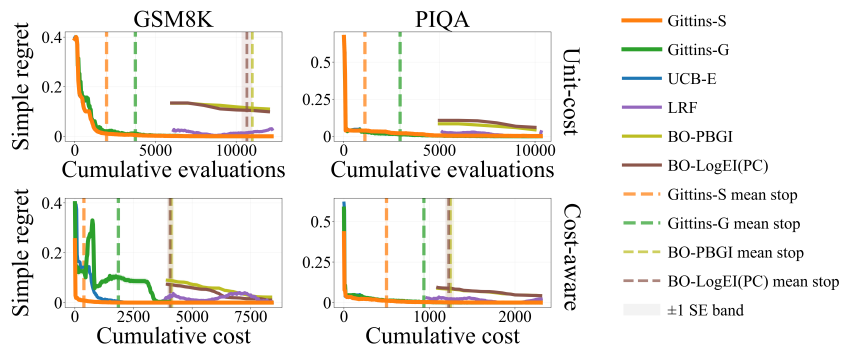
red: empirical mean, purple: prior mean, green: best configuration

Methods: Gittins-G/S; UCB-E, LRF (Zhou et al. 2025; frequentist-style bandit); PBGI, LogEI(PC) (Bayesian optimization)

Evaluation cap: 10% of brute-force full evaluations/cost

GSM8K and PIQA Results

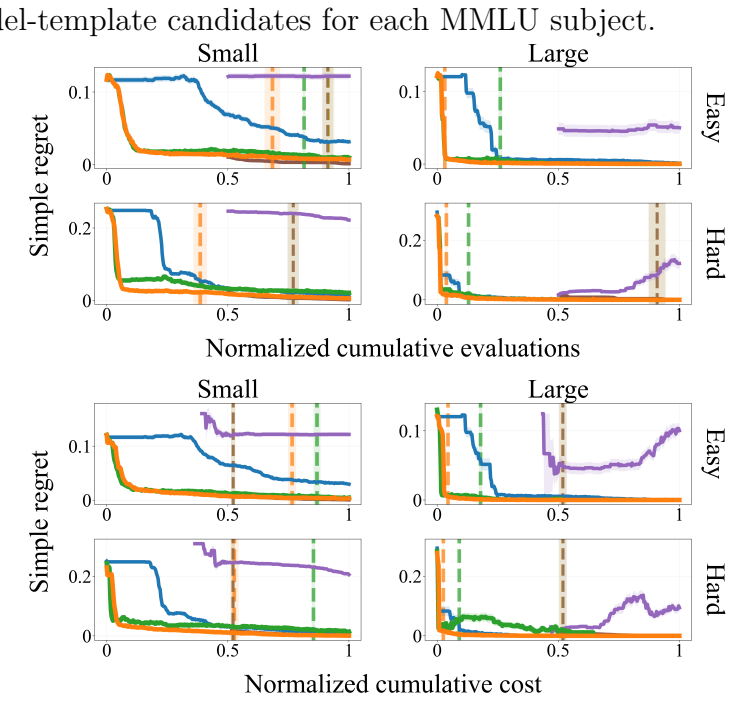
Many examples amplify savings: 1000 examples in each benchmark.



Takeaway: Gittins substantially outperforms full-evaluation-based Bayesian optimization on many-example benchmarks.

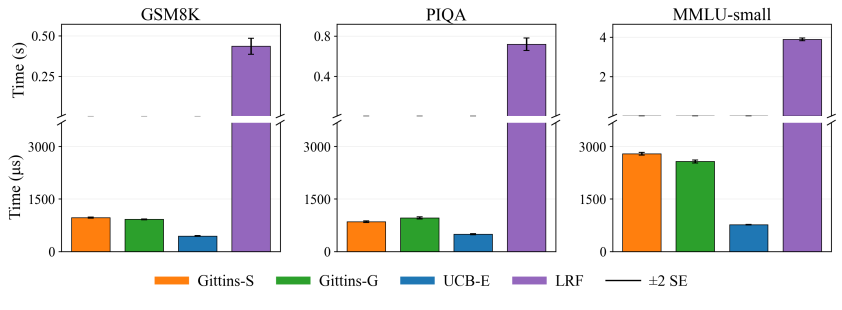
MMLU Results

Many candidates widen method gaps: DOVE provides evaluations of 1500 model-template candidates for each MMLU subject.



Takeaway: Gittins outperforms frequentist-style baselines on many-candidate tasks; Gittins usually stops much earlier than the 10% cap.

Computational Efficiency



Gittins computes faster than LRF which exploits arm correlations.