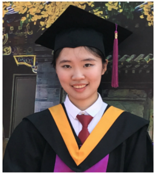
Cost-aware Stopping for Bayesian Optimization

Qian Xie[*,1] Linda Cai[*,2] Alexander Terenin[1] Peter I. Frazier[1] Ziv Scully[1]

[1]Cornell University [2]University of California, Berkeley

[*]Equal contribution

ICML 2026

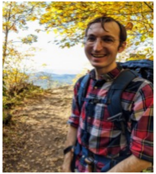


Qian Xie

Linda Cai

Alexander Terenin

Peter I. Frazier

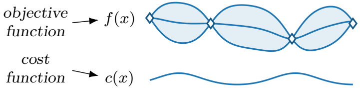


Ziv Scully

Cost-aware Bayesian optimization with adaptive stopping

We care about not only solution quality but also evaluation cost.

Cost-aware Bayesian optimization

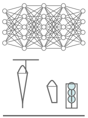


At evaluation t, fit a posterior model, choose x_t, observe $f(x_t)$, and incur evaluation cost $c(x_t)$.

Real-world examples

Hyperparameter tuning (AutoML): train time / compute cost.

Scientific discovery: lab or simulation cost.



Metric: cost-adjusted simple regret

$$R_c = \mathbb{E}\Big[\underbrace{\min_{1\le t\le\tau} f(x_t) - \inf_{x\in\mathcal{X}} f(x)}_{\text{solution quality}} + \underbrace{\sum_{t=1}^{\tau} c(x_t)}_{\text{evaluation cost}}\Big]$$

Balances final solution quality and cumulative evaluation cost up to stopping.

Adaptive stopping

The stopping time τ is chosen adaptively from data, rather than fixed in advance.

Active stopping can further improve cost-adjusted simple regret.

Overview of existing stopping rules

Existing stopping rules do not fully account for varying evaluation costs.

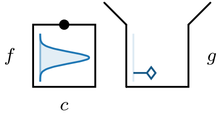
Stopping rule family	Examples	Cost limitation
Simple heuristics	Best value unchanged for a few iterations; no statistically significant improvement	Cost does not directly enter.
Acquisition-based	PI / EI / KG falls below some threshold	Usually unit-cost or heuristic.
Regret-based	UCB-LCB, SRGap-med, PRB	Mainly tracks solution quality.

How can we design a stopping rule that adapts to varying evaluation costs?

State-of-the-art cost-aware acquisition rules: PBGI and LogEIPC

Pandora's Box gives a principled way to incorporate evaluation costs.

Gittins-index intuition



Pandora's Box (Weitzman'79): open the closed box with cost or take the opened box?

The Gittins index g is the value in the opened box such that one is indifferent between the two options.

PBGI (Xie et al.'24)

$$\alpha_t^{\mathrm{PBGI}}(x) = g \quad \text{where} \quad \mathrm{EI}_t(x; g) = c(x).$$

The PBGI acquisition value is the fair value g such that expected improvement over g is worth the cost.

LogEIPC (Snoek et al.'12; Ament et al.'23)

$$\alpha_t^{\mathrm{LogEIPC}}(x; y_{1:t}^*) = \log(\mathrm{EI}_t(x; y_{1:t}^*)/c(x)).$$

The LogEIPC acquisition value is the logarithm of expected improvement per cost.

In Pandora's Box, the Gittins index naturally comes with a stopping rule. Can we bring this idea to Bayesian optimization?

PBGI/LogEIPC stopping rule

Stop when no point's expected improvement is worth its cost.

Equivalent stopping conditions

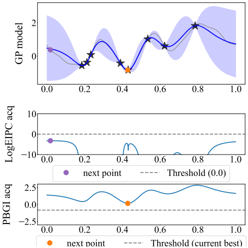
$$\min_{x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}} \alpha_t^{\mathrm{PBGI}}(x) \geq y_{1:t}^* \quad \Longleftrightarrow \quad \max_{x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}} \alpha_t^{\mathrm{LogEIPC}}(x; y_{1:t}^*) \leq 0$$

PBGI: point with best fair value already evaluated. **LogEIPC:** no point's expected improvement is worth cost.

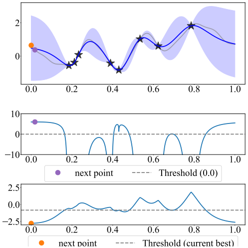
This recovers and generalizes an EI stopping rule under the unit-cost setting (Nguyen et al.'17).

Illustration unit-cost example

Large cost-per-sample: stop sooner



Small cost-per-sample: continue longer



Theoretical (safeguard) guarantee

Is our stopping rule theoretically sound, and when?

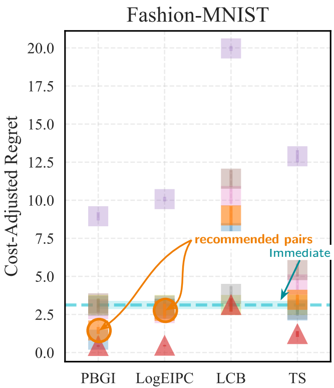
Theorem 3.2: no worse than immediate stopping

$$\underbrace{\mathbb{E}\Big[y^*_{1:\tau} - \min_{x\in\mathcal{X}} f(x) + \sum_{t=1}^{\tau} c(x_t)\Big]}_{\text{our recommended pairs}} \le \underbrace{\mathbb{E}\Big[y_1 - \min_{x\in\mathcal{X}} f(x) + c(x_1)\Big]}_{\text{immediate stopping}}$$

Implication

- Recommended pairs: our stopping rule paired with PBGI or LogEIPC.
- Matches the best achievable performance in the worst case, when evaluations are all very costly.
- Avoids over-spending—a property many cost-unaware stopping rules lack.

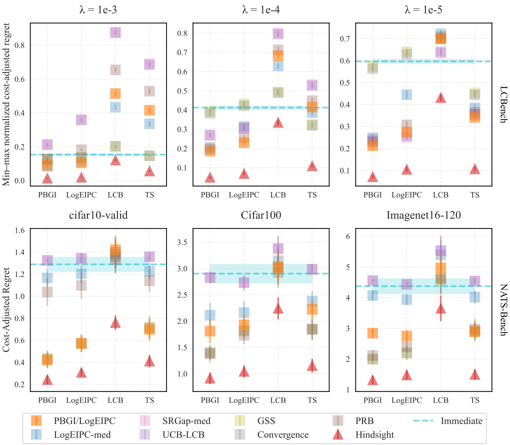
Illustration



Empirical results: strong cost-performance trade-offs

Is our stopping rule empirically good, and when?

AutoML benchmarks: LCBench & NATS-Bench



LCBench rank frequency

λ	Acq.	Top-3 large	Top-3 all
10^{-3}	PBGI	80.0%	60.0%
10^{-3}	LogEIPC	70.0%	62.8%
10^{-4}	PBGI	80.0%	62.9%
10^{-4}	LogEIPC	85.0%	68.7%
10^{-5}	PBGI	90.0%	74.4%
10^{-5}	LogEIPC	85.0%	71.5%

Large = datasets with more than 10,000 instances.

Empirical takeaway

Our recommended pairs—our stopping rule with PBGI or LogEIPC—rank **Top-3 in 70–90%** of large LCBench tasks; PBGI is the most robust default under misspecification.

Summary

We propose a cost-aware stopping rule for Bayesian optimization.

Three takeaways

- Cost-adjusted simple regret makes the solution-quality/evaluation-cost trade-off explicit.
- Connection to Pandora's box gives a shared stopping rule for PBGI and LogEIPC: stop when no point's expected improvement is worth its cost.
- Our stopping rule paired with PBGI or LogEIPC has a safeguard guarantee and strong empirical performance.

A practical recipe: acquire with PBGI or LogEIPC, stop by the PBGI/LogEIPC rule.

Paper / code