Local Constrained Bayesian Optimization

Efficient high-dimensional constrained Bayesian optimization by alternating between gradient-based local exploitation and uncertainty-driven local exploration.

Jingzhe Jing · Zheyi Fan · Szu Hui Ng · Qingpei Hu
AMSS, Chinese Academy of Sciences · National University of Singapore

ICML 2026

PART 01

Introduction

Why Bayesian optimization is useful for expensive black-box functions.

Introduction: Black-Box and Expensive Objective

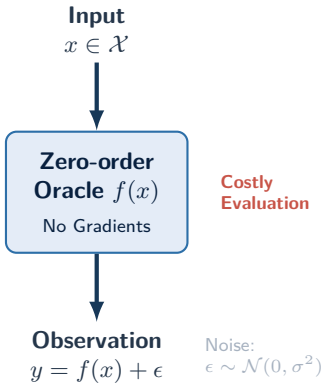
The optimization setting

We want to solve

$$\min_{x \in \mathcal{X}} f(x), \qquad \mathcal{X} \subset \mathbb{R}^d,$$

where **each evaluation is costly** and the function is only observed through noisy runs.

- **Black-box:** no closed form for f.
- **Expensive:** simulations, experiments, or rollouts dominate computation.
- **No analytic gradients:** finite differences are sample-inefficient and noise-sensitive.



Why BO?

Bayesian optimization replaces blind search with a **probabilistic surrogate** and an **acquisition rule** that actively selects the next experiment.

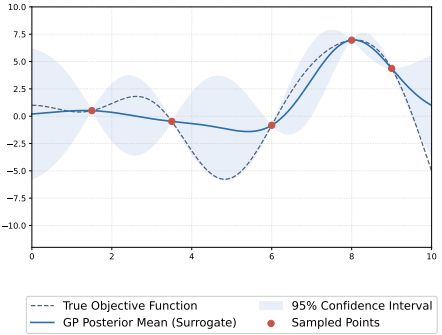
Surrogate Model

GP posterior view

After observing $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$, the model returns

$$f(x) \mid \mathcal{D}_n \sim \mathcal{N}\big(\mu_n(x), \sigma_n^2(x)\big).$$

- $\mu_n(x)$ is the fitted response surface.
- $\sigma_n(x)$ is calibrated epistemic uncertainty.
- Both are updated after every new query.



Sequential loop

Fit GP → score candidates → evaluate the best-scoring point → update the posterior.

Acquisition Function

Acquisition rule

An acquisition function $a_n(x)$ converts the GP posterior into a decision rule:

$$x_{n+1} \in \arg\max_{x \in \mathcal{X}} a_n(x).$$

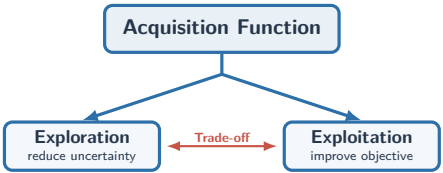
It rewards points that look promising **or** points that reduce uncertainty.

- **Exploitation:** low predicted objective value.
- **Exploration:** high posterior uncertainty.

Expected Improvement for minimization

Let $f^\star$ be the incumbent and $I(x) = \max\{f^\star - f(x), 0\}$. For $f(x) \mid \mathcal{D}_n \sim \mathcal{N}(\mu_n(x), \sigma_n^2(x))$,

$$\mathrm{EI}(x) = (f^\star - \mu_n(x))\Phi(z) + \sigma_n(x)\phi(z),$$
$$z = \frac{f^\star - \mu_n(x)}{\sigma_n(x)}.$$



Gaussian Process Preliminaries

Definition

A Gaussian process is a distribution over functions:

$$f(\cdot) \sim \mathcal{GP}\big(\mu(\cdot), k(\cdot,\cdot)\big).$$

For any finite inputs $X = (x_1, \ldots, x_n)$,

$$f(X) \sim \mathcal{N}\big(\mu(X), K(X,X)\big).$$

The kernel encodes smoothness and length-scale assumptions.

Conditioning on dataset $\mathcal{D}$

For noisy observations $y = f(X) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2 I)$,

$$\mu_{\mathcal{D}}(x) = k(x,X)(K + \sigma_n^2 I)^{-1} y,$$

$$k_{\mathcal{D}}(x,x') = k(x,x') - k(x,X)(K+\sigma_n^2 I)^{-1} k(X,x').$$

The posterior is again Gaussian.

Key modeling payoff

The same posterior gives both a fitted surface $\mu_{\mathcal{D}}$ and the uncertainty $k_{\mathcal{D}}(x,x)$ used by acquisition functions.

The Gradient of a GP Is Again a GP

Why it is true

Differentiation is a linear operator. A linear transform of a jointly Gaussian vector remains Gaussian.

For differentiable kernels, such as RBF or Matérn $\nu > 1$,

$$\nabla f(x) \mid \mathcal{D} \sim \mathcal{N}\big(\nabla \mu_{\mathcal{D}}(x), \Sigma'(x \mid \mathcal{D})\big).$$

Closed form covariance

$$\Sigma'(x \mid \mathcal{D}) = \nabla_x k_{\mathcal{D}}(x, x') \nabla_{x'}^{\top}\big|_{x'=x}.$$

The mean and covariance of the gradient are obtained directly from derivatives of the posterior kernel.

Why LCBO cares

We can run gradient-based local search on a black-box objective without finite-differencing noisy function values.

PART 02

Problem Setting

From unconstrained BO to constrained and local optimization.

Constrained Bayesian Optimization

Formal problem definition

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, \qquad i = 1, \dots, m, \end{aligned}$$

with compact, convex $\mathcal{X} \subset \mathbb{R}^d$.

At iteration k, we query a noisy zero-order oracle:

$$y_{0,k} = f(x_k) + \xi_{0,k}, \qquad y_{i,k} = c_i(x_k) + \xi_{i,k},$$

where $\xi_{i,k}$ are independent Gaussian noises.

Modeling convention

Use $m + 1$ independent GPs:

$$f \sim \mathcal{GP}_f, \qquad c_i \sim \mathcal{GP}_i, \quad i = 1, \dots, m.$$

Each query can update both objective and constraint surrogates.

Not safe BO

LCBO allows infeasible queries because they are informative for the surrogates.

Inequalities can be unified through slack variables: $c_i(x) + s_i^2 = 0$, $s_i \in \mathbb{R}$.

Applications of Constrained BO

Quality and process	Engineering design	Sequential control
Process optimization	**Design under safety limits**	**Safe controller tuning**
▶ Maximize yield or quality. ▶ Stay within specification limits on measured properties.	▶ Minimize weight or cost. ▶ Enforce stress, displacement, or stability thresholds.	▶ Maximize reward. ▶ Avoid operating or energy-limit violations during search.

The high-dimensional bottleneck

The hard question is what to do when the design has $d > 20$ variables, where global CBO methods become sample-inefficient.

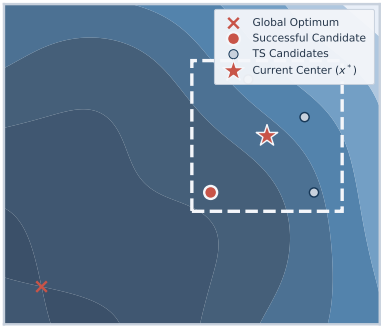
Local Bayesian Optimization

Global BO

- Searches the entire domain.
- Can target global optima.
- Regret guarantees often carry exponential dependence on d.

Local BO

- Focuses queries near the current iterate.
- Uses local model information efficiently.
- Trades global guarantees for dimension-polynomial behavior.



LCBO philosophy

Use local gradient information from GPs, but avoid rigid trust-region success/failure logic.

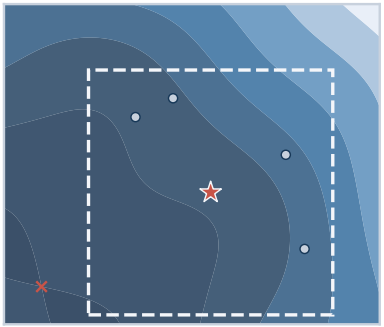
Local Bayesian Optimization

Global BO

- Searches the entire domain.
- Can target global optima.
- Regret guarantees often carry exponential dependence on d.

Local BO

- Focuses queries near the current iterate.
- Uses local model information efficiently.
- Trades global guarantees for dimension-polynomial behavior.



LCBO philosophy

Use local gradient information from GPs, but avoid rigid trust-region success/failure logic.

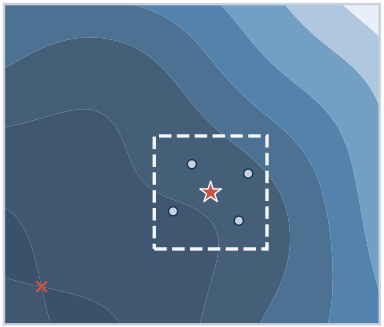
Local Bayesian Optimization

Global BO

- Searches the entire domain.
- Can target global optima.
- Regret guarantees often carry exponential dependence on d.

Local BO

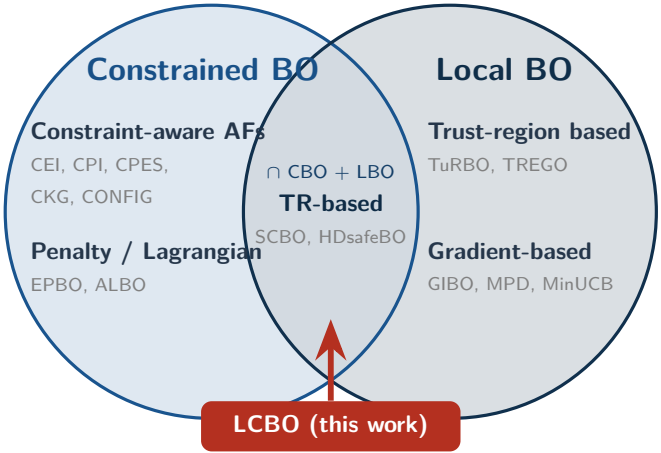
- Focuses queries near the current iterate.
- Uses local model information efficiently.
- Trades global guarantees for dimension-polynomial behavior.



LCBO philosophy

Use local gradient information from GPs, but avoid rigid trust-region success/failure logic.

LCBO Positioning



Our position

- First to extend GIBO's TR-free local exploration-exploitation framework to CBO
- First convergence guarantee for this class of local CBO methods.

PART 03

Algorithm

Gradient-information exploration and penalty-gradient exploitation.

The Penalized Surrogate

Quadratic penalty · penalty factor ρ · objective · constraint vector

$$Q_\rho(x) \;=\; f(x) \;+\; \frac{\rho}{2}\|c(x)\|_2^2$$

Why penalty?	Why quadratic?	Schedule
Converts the constrained problem into an unconstrained surface whose gradients can be estimated from the GPs.	Smoothness propagates from c to $\|c(x)\|_2^2$, enabling GP-gradient analysis and step-size theory.	Grow ρ_k slowly with the step size: $\rho_k = k^{1/4}, \qquad \eta_k = \frac{1}{2L_m} k^{-1/2}.$

Each iteration

Local exploration sharpens the gradient at x_k; local exploitation takes one projected gradient step on Q_{ρ_k}.

Need for Active Sampling

Why sample actively?

The next iterate follows an estimated descent direction:

$$x_{k+1} = \mathcal{P}_{\mathcal{X}}\big(x_k - \eta_k \nabla Q_{\rho_k}(x_k)\big).$$

A noisy gradient means a wasted step, especially in high dimensions. Therefore, LCBO spends a small batch of queries to make the gradient estimate at x_k more precise.

Gradient uncertainty

For each model $i \in \{f, c_1, \ldots, c_m\}$,

$$\nabla g_i(x_k) \mid \mathcal{D} \sim \mathcal{N}\big(\nabla \mu_i(x_k), \Sigma_i'(x_k \mid \mathcal{D})\big),$$

where

$$\Sigma_i'(x_k \mid \mathcal{D}) = \nabla_x k_{\mathcal{D}}^{(i)}(x_k, x') \nabla_{x'}^{\top}\big|_{x'=x_k}.$$

The scalar proxy is $\operatorname{tr} \Sigma_i'(x_k \mid \mathcal{D})$.

Goal

Choose a candidate batch Z that most reduces the posterior uncertainty of the gradient at the current iterate.

Gradient Information Acquisition

Expected reduction in gradient uncertainty

For model i, define

$$\alpha_i^{\mathrm{GI}}(x_k, Z) = \mathbb{E}\left[\underbrace{\operatorname{tr}\Sigma_i'(x_k \mid \mathcal{D}_{k-1})}_{\text{before}} - \underbrace{\operatorname{tr}\Sigma_i'(x_k \mid \mathcal{D}_{k-1} \cup Z)}_{\text{after}}\right].$$

Simplification

The before term does **not** contain Z. Hence maximizing reduction is equivalent to minimizing the after term:

$$\alpha_i(x_k, Z) = \mathbb{E}\left[\operatorname{tr}\Sigma_i'(x_k \mid \mathcal{D}_{k-1} \cup Z)\right].$$

LCBO guards the worst case:

$$Z = \arg\min_{Z} \max_{i} \alpha_i(x_k, Z), \quad i \in \{f, c_1, \ldots, c_m\}.$$

Alternative interpretation

An **A-optimal design** criterion for the local gradient: minimize the trace of the gradient posterior covariance at x_k.

GP free lunch

GP posterior covariance depends only on **where** we sample, not the observed values. The expectation is therefore computed before collecting outcomes.

Local Exploitation Phase

GP-gradient of the penalty surrogate

Using posterior means for objective and constraints,

$$\nabla Q_{\rho_k}(x_k) = \nabla \mu^{(f)}_{\mathcal{D}_k}(x_k) + \rho_k \, \nabla \mu^{(c)}_{\mathcal{D}_k}(x_k)^\top \mu^{(c)}_{\mathcal{D}_k}(x_k).$$

Then take

$$x_{k+1} = \mathcal{P}_{\mathcal{X}}\big(x_k - \eta_k \nabla Q_{\rho_k}(x_k)\big).$$

Local counterpart of EI

EI asks *where* the function value might improve globally. LCBO asks *which local direction* decreases the penalized objective fastest at x_k.

Tractable approximation

The exact posterior of $\nabla c(x)^\top c(x)$ is non-Gaussian because ∇c and c are dependent. LCBO uses a tractable approximation.

LCBO Algorithm at a Glance

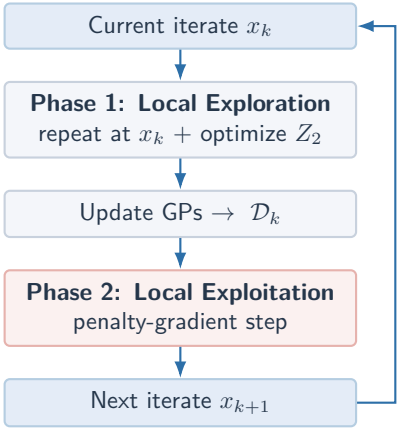
ALGORITHM 1

LCBO

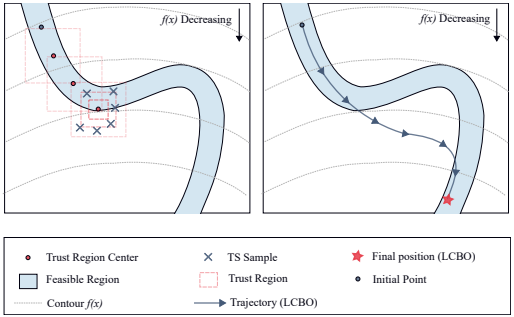
input: $\mathbf{X}$, oracles f, c
init: $\boldsymbol{x}_1 \in \mathbf{X}$, D_0, schedules $\{\boldsymbol{\eta}_k\}, \{\boldsymbol{\rho}_k\}$

for $k = 1, \ldots, K$ **do**
  // local exploration
  $Z_1 \leftarrow [x_k, \ldots, x_k]^\top$
  $Z_2 \leftarrow \arg\min_Z \boldsymbol{\alpha}(x_k, Z)$
  $Z \leftarrow [Z_1^\top, Z_2^\top]^\top$
  evaluate f, c at $Z \rightarrow$ update $\mathcal{D}_k$

  // local exploitation
  $\widehat{\nabla} Q_{\rho_k}(x_k)$ from $\mathcal{D}_k$
  $x_{k+1} \leftarrow \mathcal{P}_{\mathcal{X}}\big(x_k - \eta_k \widehat{\nabla} Q_{\rho_k}(x_k)\big)$
end for



Why Trust Regions Stall



TR structural dilemma (hard constraint)

- Descent-aligned candidates can be infeasible.
- Feasible points inside the region may not improve.
- Both count as local failures, shrinking the radius.

LCBO mechanism (soft constraint)

Penalty gradients combine objective descent with constraint-violation correction, allowing iterates to slide along or briefly cross the boundary while still progressing.

PART 04

Theory

KKT residuals and dimension-polynomial rates.

Preliminary: KKT Residuals

For the constrained optimization problem:

$$\min_{x \in \mathcal{X}} f(x) \quad \text{s.t.} \quad c(x) = 0, \qquad c : \mathcal{X} \to \mathbb{R}^m,$$

we characterize its optimality using two KKT residuals: **feasibility** and **stationarity**.

KKT Residuals

Stationarity:

$$r_s(x, \lambda) = \operatorname{dist}\big(\nabla f(x) + \nabla c(x)^\top \lambda, -N_{\mathcal{X}}(x)\big)$$

Feasibility:

$$r_f(x) = \|c(x)\|_2$$

Both residuals vanish simultaneously if and only if (x, λ) satisfies the KKT conditions, i.e., x is a first-order stationary point of the constrained problem (Nocedal & Wright, 2006).

Main Convergence Theorem

Assumptions 5.1–5.3

Stationary C^4 bounded positive-definite kernel, e.g. RBF or Matérn $\nu > 2$; compact convex $\mathcal{X}$; and regularity condition (Lin et al.,2022):

$$\operatorname{dist}\left(\nabla c(x_k)^\top c(x_k), -N_{\mathcal{X}}(x_k)\right) \geq \gamma \|c(x_k)\|.$$

Theorem 5.5

With $\eta_k = \frac{1}{2L_m} k^{-1/2}$ and $\rho_k = k^{1/4}$, for $K > \widetilde{K}_m$, with probability at least $1 - \delta$,

$$\frac{1}{K} \sum_{k=1}^{K} r_s(x_{k+1}, \lambda_{k+1}) \leq \frac{C_{1,m} + C_2 E_K}{\sqrt{K}},$$

$$\frac{1}{K} \sum_{k=1}^{K} r_f(x_{k+1}) \leq \frac{C_3}{\sqrt{K}} + \frac{C_{4,m} + C_5 E_K}{K}.$$

E_K aggregates the GP gradient/function estimation errors. Its dimension dependence is what we control next via Corollary 5.6.

Polynomial Dependence on Dimension

Setup

For RBF or Matérn $\nu = 2.5$ kernels and batch sizes $b_k^{(1)} = k$, $b_k^{(2)} = dk^2$

Stationarity

$$\frac{1}{K}\sum_{k=1}^{K} r_s(x_{k+1}, \lambda_{k+1}) = \widetilde{\mathcal{O}}(m^{\frac{5}{2}} d^{\frac{1}{6}} n^{-\frac{1}{6}} + \sigma m^{\frac{13}{6}} d^{\frac{7}{6}} n^{-\frac{1}{6}})$$

Feasibility

$$\frac{1}{K}\sum_{k=1}^{K} r_f(x_{k+1}) = \widetilde{\mathcal{O}}\big(m^{\frac{1}{6}} d^{\frac{1}{6}} n^{-\frac{1}{6}} + m^{\frac{8}{3}} d^{\frac{1}{3}} n^{-\frac{1}{3}} + \sigma m^{\frac{7}{3}} d^{\frac{4}{3}} n^{-\frac{1}{3}}\big).$$

Headline

The dependence is **polynomial in** d, not exponential.

Convergence Bounds: Local BO vs. Global BO

Global CBO

Targets global optimum.

- Global regret bounds can scale exponentially with dimension.
- Example RBF-kernel form for EPBO (Lu & Paulson, 2022):

$$\widetilde{\mathcal{O}}\big((\log n)^{d/2} n^{-\beta}\big).$$

- Inefficient once d is moderately large.

Local BO / LCBO

Targets KKT stationarity.

- Gives a local optimum guarantee.
- Dimension dependence is polynomial:

$$\widetilde{\mathcal{O}}\big(d^{\alpha} n^{-\beta}\big)$$

- Tractable in the $d = 25$ to $100+$ regime.

PART 05

Experiments

Synthetic, engineering, and control benchmarks up to 100D.

Experimental Setup

Baselines

Method	Family
CEI	Global CBO: EI $\times$ feasibility
EPBO	Global CBO: exact penalty
SCBO	Trust-region + Thompson sampling
HDsafeBO	Safe trust-region, high-d
LCBO	Penalty-gradient method

Synthetic tasks: 25D / 50D / 100D

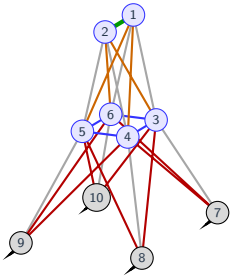
- Objective and constraints are random functions drawn from a GP.
- Realized through Random Fourier Features.
- The truth lies inside the GP model class.
- Constraints are tightened to keep the feasible region hard.

Real-World Benchmarks I

25-bar truss design: 25D

Choose the cross-sectional areas of 25 aluminum space-truss members.

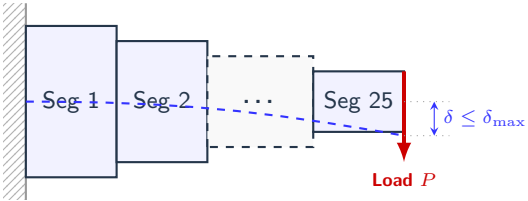
- Objective: minimize total weight.
- Constraints: member stress and nodal displacement under load.



Stepped cantilever beam: 50D

Choose width and height for 25 segments.

- Objective: minimize material volume.
- Constraints: tip deflection and bending stress limits.



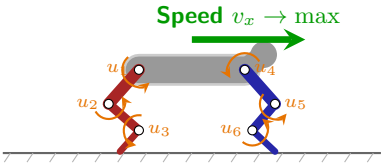
Real-World Benchmarks II

Planar running robot: 102D

Tune a linear control policy mapping a 17-dimensional observation to a 6-dimensional action:

$$a = Wo, \qquad W \in \mathbb{R}^{6\times 17}, \qquad d = 102.$$

- Objective: maximize average forward speed; equivalently minimize its negative.
- Constraint: keep total control effort under a fixed budget.



Implementation Details

Efficiency tricks from GIBO

- Train local GP on the N_m most recent points.
- Normalize ∇Q_{ρ_k} by Euclidean norm.
- Restrict acquisition search to a dynamic local box $[x_k - \delta, x_k + \delta]$.
- Use fixed mini-batches per iteration.

Optimizing the acquisition

Use a LogSumExp smooth maximum as a differentiable surrogate for $\max_i \alpha_i$.

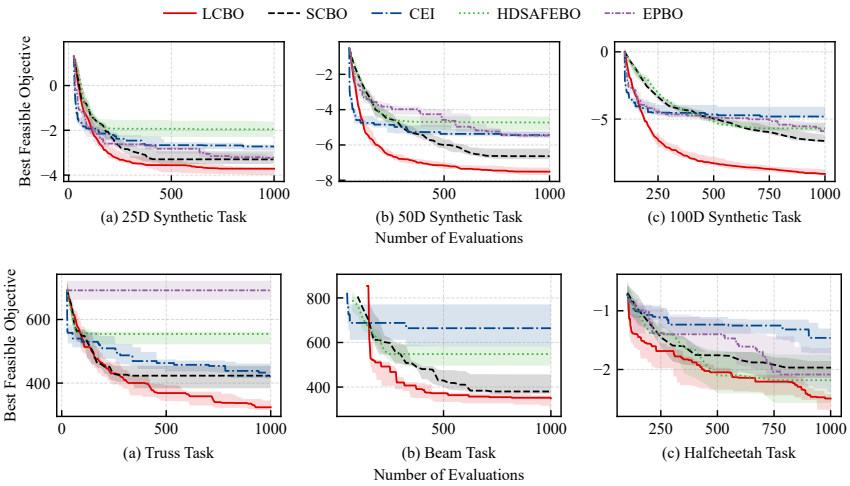
Theory-to-practice gap

Growing theoretical batches can starve the evaluation budget. Fixed mini-batches act like a stochastic approximation: more frequent noisy updates beat rare accurate updates under a finite budget.

Protocol

- Cold start: d initial random points.
- Budget: 1,000 oracle calls per task.
- Noise: $\xi \sim \mathcal{N}(0, 0.1^2)$ on f and c.
- 10 seeds; report median and IQR of best feasible objective.

Empirical Results



Ablation Studies

Study	Purpose	Design	Conclusion
Batch schedule	Bridge theory-practice gap for fixed mini-batches.	Compare fixed mini-batch, fixed large-batch, and growing-batch proxy on synthetic tasks.	Fixed mini-batches give better sample efficiency under early budgets.
Local GP / search box / gradient norm	Verify that gains are not caused by auxiliary engineering choices alone.	Remove each component on Truss 25D and Synthetic 50D.	Ablated variants remain competitive; components are practical stabilizers.
Penalty factor schedule	Test whether growing ρ_k requires task-specific tuning.	Compare $\rho_k = 10k^{1/4}$ to fixed $\rho \in \{1, 10, 100, 1000\}$.	Growing schedule navigates infeasibility-vs-progress without per-task tuning.

Takeaway

LCBO's advantage is not explained by a single engineering trick; the local penalty-gradient mechanism is robust.

Conclusion

- **01 Local exploration-exploitation**
  Gradient-based local descent and uncertainty-driven local exploration.

- **02 Single-loop framework**
  Carefully designed parameter updates guarantee convergence for our efficient single-loop framework.

- **03 Polynomial-in-d KKT rate**
  LCBO escapes exponential-in-d global regret behavior by targeting local KKT stationarity.

Thank you

Jingzhe Jing · Zheyi Fan · Szu Hui Ng · Qingpei Hu

Contact: fanzheyi@amss.ac.cn · qingpeihu@amss.ac.cn