ECPv2: Fast, Efficient, and Scalable Global Optimization of Lipschitz Functions

Fares Fourati[1], **Mohamed-Slim Alouini**[1], **Vaneet Aggarwal**[2]
[1] KAUST
[2]Purdue University
fares.fourati@kaust.edu.sa, slim.alouini@kaust.edu.sa, vannet@purdue.edu

Abstract

We propose ECPv2, a scalable and theoretically grounded algorithm for global optimization of Lipschitz continuous functions with unknown Lipschitz constants. Building on the Every Call is Precious (ECP) framework, which ensures that each accepted function evaluation is potentially informative, ECPv2 addresses key limitations of ECP, including high computational cost and overly conservative early behavior. ECPv2 introduces three innovations: (i) an adaptive lower bound that prevents vacuous acceptance regions, (ii) a memory mechanism that restricts comparisons to a fixed-size subset of past evaluations, and (iii) a fixed random projection that accelerates distance computations in high dimensions. We theoretically show that ECPv2 retains ECP's regret guarantees and expands the acceptance region with high probability. Extensive experiments and ablation studies empirically validate these findings. Using principled hyperparameter settings, we evaluate ECPv2 across a wide range of nonconvex optimization problems and find that it consistently matches or outperforms leading optimizers while significantly reducing wall clock time.

Code — https://github.com/fouratifares/ECP
Extended version — https://arxiv.org/abs/2511.16575

Introduction

Global optimization is a long-standing and fundamental challenge in optimization (Törn and Žilinskas 1989; Pardalos 2013; Floudas and Pardalos 2014; Zabinsky 2013; Stork, Eiben, and Bartz-Beielstein 2022). The goal is to identify the global maximum of a function that may be non-convex, non-smooth, and accessible only through black-box evaluations. Such settings render gradient-based and local methods insufficient, motivating algorithms that can efficiently explore the search space while maintaining broad coverage.

The challenge is further exacerbated in many real-world applications, where each function evaluation may be costly in terms of time, energy, or monetary resources, significantly limiting the number of allowable queries. Nevertheless, global optimization remains critical across a wide range of domains, including robotics (Antonova et al. 2023; D'Ambrosio et al. 2024), hyperparameter tuning for machine learning models (Lindauer et al. 2022a), and query optimization for black-box large language models (Chen et al. 2024; Lin et al. 2024; Kharrat, Fourati, and Canini 2025). These demands underscore the need for methods that are not only accurate and robust, but also highly *efficient*.

A natural structural assumption in global optimization is *Lipschitz continuity*, which posits that the objective function varies at a bounded rate. While many real-world functions are Lipschitz continuous (Törn and Žilinskas 1989), the true Lipschitz constant is typically unknown. Prior methods such as DIRECT (Jones, Perttunen, and Stuckman 1993), AdaLIPO (Malherbe and Vayatis 2017), and AdaLIPO+ (Serré et al. 2024) address this by either deterministically partitioning the search space or estimating the Lipschitz constant through uniform random sampling. However, these approaches often spend substantial evaluation budget on regions unlikely to contain the maximum, thereby reducing sample efficiency.

To address this, Every Call is Precious (ECP) algorithm (Fourati et al. 2025) was proposed as a conservative and principled alternative to previous strategies. ECP evaluates a point only if it lies within an adaptively defined region of potential maximizers. This acceptance region is initialized conservatively, possibly even empty, and is progressively expanded until a viable candidate is found. By avoiding uniformly random exploration, ECP ensures that every function evaluation is justified and potentially informative. The algorithm enjoys no-regret guarantees and has demonstrated strong empirical performance across a wide range of synthetic and real-world tasks, consistently outperforming state-of-the-art methods from Lipschitz, Bayesian, evolutionary, and bandit optimization paradigms.

Despite its strong theoretical guarantees and empirical performance, ECP has several practical limitations. Its per-iteration cost scales linearly with both the number of previously evaluated points and the problem dimension, which can become prohibitive in long-horizon or high-dimensional settings. Moreover, its optimistic acceptance rule may overly constrain exploration in early iterations, leading to the premature rejection of potentially promising candidates. These limitations motivate the refinements introduced in this work.

Contributions

1. We propose ECPv2, a scalable global optimization algorithm that enhances the efficiency and practical applicability of ECP (Fourati et al. 2025), particularly in high-dimensional settings and under modest or large evaluation budgets. ECPv2 introduces three key innovations:
   - An *adaptive lower bound* $\varepsilon_t^{\oslash}$, theoretically derived from the acceptance rule, which avoids infeasible searches and reduces unnecessary early rejections;
   - A *worst-m memory mechanism*, which compares candidates only against the m worst previous evaluations, reducing both computational and memory costs;
   - A *fixed random projection* applied in the acceptance criterion, reducing the dimensionality d of distance calculations from $\mathcal{O}(d)$ to $\mathcal{O}(\log(n))$, where n denotes the maximum number of evaluations.
2. We provide theoretical guarantees for ECPv2. We show that the adaptive lower bound avoids unnecessary rejections; the combined mechanisms define an acceptance region that strictly contains that of ECP with high probability; and the algorithm retains no-regret performance with optimal finite-time bounds, while offering significantly improved computational efficiency.
3. We empirically validate our theoretical findings through extensive experiments and ablations. We derive principled hyperparameter settings and evaluate ECPv2 on a broad suite of high-dimensional, non-convex optimization problems. Across benchmarks, ECPv2 matches or outperforms state-of-the-art methods, while reducing wall-clock time by a substantial margin.

Related Work

Global optimization has long been a foundational problem in optimization. Classical non-adaptive methods such as grid search (Zabinsky 2013) or Pure Random Search (PRS) (Brooks 1958; Zabinsky 2013) offer simple baselines by exhaustively or randomly exploring the domain. However, these methods are sample-inefficient as they ignore prior evaluations and the structure of the objective function.

Adaptive strategies like evolutionary algorithms such as CMA-ES (Hansen and Ostermeier 1996; Hansen 2006; Hansen, Akimoto, and Baudis 2019), simulated Annealing (Metropolis et al. 1953; Kirkpatrick, Gelatt Jr, and Vecchi 1983), and Dual Annealing (Xiang et al. 1997; Tsallis 1988) improve over random baselines through adaptive exploration. However, they lack no-regret guarantees and often require substantial tuning and large numbers of evaluations.

Bayesian optimization (BO) (Frazier 2018; Balandat et al. 2020) represents another prominent line of work, constructing a probabilistic model (typically Gaussian Processes) to guide the search via acquisition functions. BO performs well when model assumptions hold, but is sensitive to hyperparameter choices. Hyperparameter-free variants such as A-GP-UCB (Berkenkamp, Schoellig, and Krause 2019) and recent approaches such as SMAC3 (Lindauer et al. 2022b) have addressed these issues, but may still struggle in high-dimensional or low-budget regimes. Notably, ECP (Fourati et al. 2025) has been shown to match or even surpass the performance of these approaches on several problems.

To leverage confidence bounds, NeuralUCB (Zhou, Li, and Gu 2020), have been adapted to global optimization via repeated sampling and retraining (Fourati et al. 2025). However, their reliance on data-hungry neural models limits practical efficiency.

To leverage smoothness, adaptive approaches have been proposed to improve efficiency. Tree-based algorithms such as Zooming (Kleinberg, Slivkins, and Upfal 2008), Hierarchical Optimistic Optimization (HOO) (Bubeck et al. 2011), and Deterministic Optimistic Optimization (DOO) (Munos 2011) require known smoothness parameters, whereas Simultaneous Optimistic Optimization (SOO) (Munos 2011; Preux, Munos, and Valko 2014; Kawaguchi, Maruyama, and Zheng 2016) and SequOOL (Bartlett, Gabillon, and Valko 2019) remove this requirement by performing optimistic exploration across a hierarchy of scales. However, these algorithms often underperform in practice due to their discrete space partitioning.

For continuous optimization under unknown Lipschitz constants, the DIRECT algorithm (Jones, Perttunen, and Stuckman 1993; Jones and Martins 2021) employs a deterministic space-partitioning strategy, iteratively refining regions with the highest potential upper bounds. While parameter-free, DIRECT is computationally demanding in high dimensions and may become overly conservative.

To address these limitations, AdaLIPO (Malherbe and Vayatis 2017) introduced an adaptive stochastic strategy that estimates the Lipschitz constant online via random sampling. It uses this estimate to construct upper confidence bounds and filter potentially optimal points, offering no-regret guarantees. AdaLIPO+ (Serré et al. 2024) improves practical performance by reducing exploration over time. However, their reliance on uniform exploration can result in inefficient use of evaluations.

Recently, Every Call is Precious (ECP) algorithm (Fourati et al. 2025) was proposed as a more conservative alternative: it evaluates points only when they are consistent with being global maximizers under a Lipschitz assumption. It avoids redundant evaluations by enforcing a strict geometric acceptance condition, achieving no-regret guarantees while often outperforming existing methods. However, ECP suffers from overly strict rejection of potentially valuable candidates. Moreover, like AdaLIPO and AdaLIPO+, it faces scalability bottlenecks due to the computational cost of distance calculations, which grows linearly with both the problem dimensionality and the evaluation budget. These limitations motivate our proposed algorithm, ECPv2, which improves the scalability and efficiency of ECP while preserving its theoretical guarantees.

In summary, ECPv2 extends the conservative and theoretically grounded framework of ECP by addressing its key scalability challenges. It offers a practical, efficient alternative to state-of-the-art global optimization approaches, suitable for problems where evaluations are expensive.

Global Optimization of Lipschitz Functions

We consider the problem of global optimization for a black-box, deterministic, non-convex function $f : \mathcal{X} \to \mathbb{R}$ that is expensive to evaluate. The function is defined over a compact, convex subset $\mathcal{X} \subset \mathbb{R}^d$ with non-empty interior.

In many practical applications, each function evaluation carries substantial computational, temporal, energetic, or monetary cost, making it essential to extract maximal value from every query. We assume access only to *zeroth-order* oracle calls, that is, we may evaluate $f(x)$ at chosen points x, but no gradient or higher-order information is available. The objective is to identify a global maximizer

$$x^\star \in \arg\max_{x \in \mathcal{X}} f(x)$$

within a finite budget of n function evaluations.

For the theoretical analysis, same as ECP (Fourati et al. 2025), we assume a single and minimal assumption:

Assumption 1. *We assume that f is Lipschitz-continuous with an **unknown** finite Lipschitz constant $k \geq 0$, i.e.,*

$$\forall x, x' \in \mathcal{X}, \quad |f(x) - f(x')| \leq k \cdot \|x - x'\|_2.$$

Let $\text{Lip}(k)$ denote the class of Lipschitz-continious functions with a Lipschitz constant k, and let $\text{Lip} := \bigcup_{k \geq 0} \text{Lip}(k)$ be the space of all Lipschitz functions.

Starting from an initial query $x_1 \in \mathcal{X}$ and observing its evaluation $f(x_1)$, an adaptive global optimization algorithm proposes a subsequent query x_2 based on this first evaluation. In general, at each step $i \geq 2$, it suggests the next point x_i to evaluate using all previous evaluations. After n steps, it outputs an index $\hat{\imath} \in \{1, \dots, n\}$ corresponding to the best evaluation observed:

$$\hat{\imath} \in \arg\max_{i=1,\dots,n} f(x_i).$$

To evaluate the performance of a global optimization algorithm $A \in \mathcal{G}$, where $\mathcal{G}$ is the class of global optimization algorithms, over a function f, we consider its final *regret* after n iterations:

$$\mathcal{R}_{A,f}(n) := \max_{x \in \mathcal{X}} f(x) - \max_{i=1,\dots,n} f(x_i), \qquad (1)$$

which measures the suboptimality of the best point queried so far relative to the unknown global maximum.

We are interested in algorithms that achieve *no-regret* over the Lip space, i.e., whose regret converges to zero in probability as the number of evaluations increases.

Definition 1 (No Regret)**.** *An algorithm $A \in \mathcal{G}$ is said to be no-regret over a class of functions $\mathcal{F}$ if:*

$$\forall f \in \mathcal{F}, \quad \mathcal{R}_{A,f}(n) \xrightarrow{p} 0 \quad \text{as } n \to \infty,$$

where $\mathcal{R}_{A,f}(n)$ denotes the regret of algorithm A with respect to function f after n rounds, and $\xrightarrow{p}$ denotes convergence in probability.

We also define the *radius* of the domain,

$$\text{rad}(\mathcal{X}) := \max\{r > 0 : \exists\, x \in \mathcal{X} \text{ such that } B(x,r) \subseteq \mathcal{X}\},$$

where $B(x,r) = \{x' \in \mathbb{R}^d : \|x - x'\|_2 \leq r\}$ is the Euclidean ball, and the *diameter*

$$\text{diam}(\mathcal{X}) := \sup_{x,x' \in \mathcal{X}} \|x - x'\|_2.$$

As established in prior work (Bull 2011), there is a fundamental limit on the best achievable regret when optimizing Lipschitz functions, even when the Lipschitz constant is known. From Theorem 1 in (Bull 2011):

Proposition 1 (Minimax Regret Lower Bound (Bull 2011))**.** *For any $f \in Lip(k)$, any $k \geq 0$, and any $n \in \mathbb{N}^\star$,*

$$\inf_{A \in \mathcal{G}} \sup_{f \in Lip(k)} \mathbb{E}[\mathcal{R}_{A,f}(n)] \geq \Omega(\text{rad}(\mathcal{X}) \cdot k \cdot n^{-1/d}).$$

where $\text{rad}(\mathcal{X})$ denotes the radius of the domain $\mathcal{X}$ and the expectation is taken over the sampling distribution induced by algorithm $\mathcal{A}$ when optimizing f.

This result underscores the fundamental challenge of high-dimensional global optimization, where the optimal regret decreases only at a rate of $n^{-1/d}$ (Bull 2011). It highlights the crucial role of each function evaluation in driving meaningful progress, especially when the number of evaluations n is limited and the dimensionality d is large.

In this context, the ECP algorithm (Fourati et al. 2025) embraces a conservative optimization philosophy: it *evaluates only points that are likely to be potential maximizers*, employing Lipschitz-based acceptance conditions to filter candidate queries. This selection ensures that each evaluation has a potential to improve the current best solution. Notably, ECP achieves regret guarantees that match the theoretical lower bound, reflecting its strong theoretical foundation. However, its computational complexity grows linearly with both the number of evaluations and the problem dimensionality, which can significantly limit its scalability in large-scale settings. This limitation motivates the present work, which we discuss in detail in the following sections.

Background

The ECP algorithm (Fourati et al. 2025) was proposed as a principled solution to global optimization when function evaluations are costly. It builds on the idea that each query to the objective function should be a potential optimizer, through a *Lipschitz-based acceptance rule* that governs whether a candidate point $x \in \mathcal{X}$ should be evaluated.

Concretely, given a current archive $\{X_1, \dots, X_t\}$ of evaluated points and their values $\{f(X_1), \dots, f(X_t)\}$, a new candidate x sampled from the uniform distribution $\mathcal{U}(\mathcal{X})$ is accepted for evaluation only if:

$$\min_{i=1,\dots,t} (f(X_i) + \varepsilon_t \cdot \|x - X_i\|_2) \geq \max_{j=1,\dots,t} f(X_j), \qquad (2)$$

where $\varepsilon_t > 0$ is a time-dependent parameter.

Equation (2) determines whether a candidate point x remains *compatible* with being the global maximizer under a Lipschitz assumption. If ε_t were equal to the true (unkown) Lipschitz constant k, each quantity $f(X_i) + k\|x - X_i\|$ would represent the tightest upper bound on the unknown value $f(x)$, and the minimum over i would give the most optimistic value that x could still attain. The acceptance condition therefore evaluates x only when this best-case estimate exceeds the current maximum, ensuring that no query is spent on points that are provably suboptimal.

In practice, ε_t begins as a small lower bound, yielding a conservative acceptance region, but increases over time toward k, progressively relaxing the test while guaranteeing that true maximizers are never discarded on the long horizon. The parameter ε_t grows geometrically whenever the number of rejected candidates exceeds a threshold $C > 1$, by updating $\varepsilon_t \leftarrow \tau_{n,d} \cdot \varepsilon_t$, with $\tau_{n,d} > 1$.

Rejections are tracked using a counter h_{t+1}, which is initialized to zero and reset whenever ε_t is increased. At each iteration, h_t is set to the number of rejections from the previous step, and if the increase in rejections from one iteration to the next exceeds C, the update is triggered. This mechanism enlarges the effective search space over time, allowing broader exploration. If a candidate fails the acceptance test, it is rejected without evaluation and a new point is sampled from $\mathcal{U}(\mathcal{X})$.

Definition 2. (ECP ACCEPTANCE REGION) *The set of points potentially accepted by ECP at any time step $t \geq 1$, for any value $\varepsilon_t > 0$, is defined as:* $\mathcal{A}_{ECP}(\varepsilon_t, t) \triangleq \{x \in \mathcal{X} : \min_{i=1,\cdots,t}(f(X_i)+\varepsilon_t \cdot \|x-X_i\|_2) \geq \max_{j=1,\cdots,t} f(X_j)\}$.

The ECP strategy offers several attractive properties. First, ECP achieves no-regret over Lipschitz functions, ensuring convergence to a global maximizer. Second, it is *resource efficient*: by filtering out unpromising, ECP minimizes wasted calls to the objective function. Finally, ECP supports *adaptive exploration*: its acceptance criterion dynamically adapts to the current state of knowledge, promoting a conservative yet effective exploration of the search space. However, despite these strengths, ECP suffers from some practical challenges that limit its scalability in high-dimensional or long-horizon settings:

Rejection Overhead: When ε_t is too small, the acceptance region may be empty, causing the algorithm to repeatedly reject proposals before adapting ε_t. While this is sufficient to guarantee that the acceptance region is not larger than enough, it delays progress and increases wall-clock time, especially in the early search stages.

Computational Inefficiency: The acceptance condition in Equation (2) requires computing distances to *all* $t \leq n$ previous points, leading to at least $\Omega(n^2 d)$ overall computational complexity. In high-dimensional spaces or with large evaluation budgets, this becomes a bottleneck.

Toward a scalable alternative and to overcome these limitations while preserving ECP's conservative foundation, we propose ECPv2, which introduces key innovations, which make ECPv2 fast, efficient, and scalable, while retaining the theoretical guarantees. In the following sections, we formally describe ECPv2 and analyze its theoretical properties and empirical performance.

ECPv2

ECPv2, outlined in Algorithm 1, takes as input a budget n, a search space $\mathcal{X}$, and an objective function f. It builds upon the original ECP algorithm by retaining the core acceptance rule, including the geometric growth factor $\tau_{n,d} > 1$, the rejection threshold $C > 1$, and the initial parameter $\varepsilon_1 > 0$.

To improve scalability and performance in high-dimensional or complex settings, ECPv2 introduces three key modifications to the acceptance mechanism. These changes are briefly summarized below, with detailed explanations and theoretical guarantees provided in later sections.

(i) Adaptive lower bound $\varepsilon_t^{\oslash} \leq \varepsilon_t$. The size of the acceptance region at iteration t in ECP is governed by the parameter ε_t, which appears on the left-hand side of the acceptance condition (Equation (2)). When ε_t becomes too small, the acceptance region may become vanishingly small or even empty, leading to an excessive number of rejections.

To prevent such overly conservative behavior, ECPv2 introduces a time-dependent lower bound on ε_t:

$$\varepsilon_t^{\oslash} := \frac{\max_i f(x_i) - \min_j f(x_j)}{\mathrm{diam}(\mathcal{X})}. \tag{3}$$

This quantity ensures that the acceptance region is not trivially vacuous. At each step t, ECPv2 initializes ε_t (line 14):

$$\varepsilon_t = \max\left(\tau_{n,d} \cdot \varepsilon_{t-1}, \varepsilon_t^{\oslash}\right),$$

where $\tau_{n,d} > 1$, same as in ECP, controls the growth of ε_t over time. As shown in Lemma 1, this lower bound is necessary to guarantee that the acceptance region remains meaningfully sized, avoiding unnecessary rejections of promising candidates. The acceptance condition becomes:

$$\min_{i\in[t]}\left(f(x_i) + \max\{\varepsilon_t, \varepsilon_t^{\oslash}\} \cdot \|x - x_i\|_2\right) \geq \max_{j\in[t]} f(x_j). \tag{4}$$

The additional cost of maintaining $\varepsilon_t^{\oslash}$ is negligible, $\mathcal{O}(1)$ per iteration, since both $\max_i f(x_i)$ and $\min_j f(x_j)$ can be tracked incrementally during the optimization process. Lemma 2 further shows that the acceptance condition remains a superset of the previous acceptance condition.

(ii) Worst-m memory mechanism. In addition to bounding ε_t, a major limitation of ECP is that the computational and memory cost of the acceptance condition grows linearly with the number of previous evaluations t. Moreover, as t increases, the acceptance condition becomes increasingly conservative, often rejecting candidate points unnecessarily, particularly when ε_t is still small.

To address these issues, and inspired by the stochastic maximization trick of (Fourati, Aggarwal, and Alouini 2024), which reduces maximization cost by retaining only the most informative actions, ECPv2 restricts the acceptance test to the m worst-performing previously evaluated points. Formally, define:

$$\mathcal{I}_t^m = \arg \min_{\substack{S\subseteq\{1,\ldots,t\} \\ |S|=m}} \sum_{i\in S} f(x_i),$$

where $\mathcal{I}_t^m$ indexes the m points with the lowest objective values among the t evaluated so far. The integer parameter

$m \geq 1$ controls the trade-off between computational efficiency and conservatism: setting $m \geq t$ recovers the ECP condition, while smaller values of m reduce both computational and memory costs.

The modified acceptance condition becomes:

$$\min_{i \in \mathcal{I}_t^m} \left(f(x_i) + \max\{\varepsilon_t, \varepsilon_t^{\oslash}\} \cdot \|x - x_i\|_2 \right) \geq \max_{j \in [t]} f(x_j). \tag{5}$$

(iii) Introduce random projection. Instead of computing distances in the original space $\mathbb{R}^d$, ECPv2 uses a fixed random projection $\mathbf{P} : \mathbb{R}^d \to \mathbb{R}^{d'}$ to reduce computational cost, with d' being function of the budget n and a controlled maximal distortion $\delta \in [0, 1)$. For $\delta = 0$ we consider an identity projection $\mathbf{I}_d$ (no projection) and revert to the case of ECP. For $\delta > 0$, we can have an acceleration where projection is applied though a random projection matrix as:

$$\mathbf{P} := \mathbf{1}_{\delta > 0} \cdot \frac{1}{\sqrt{d'}} R^\top + (1 - \mathbf{1}_{\delta > 0}) \cdot \mathbf{I}_d, \tag{6}$$

where $R \in \mathbb{R}^{d \times d'}$ is a matrix with i.i.d. entries drawn from $\mathcal{N}(0, 1)$, with $d' = 8 \log(\beta n)/(\delta^2 - \delta^3)$. The choice of this form ensures that each projected vector $x_i' = \mathbf{P}x_i \in \mathbb{R}^{d'}$ lies in a lower-dimensional subspace while preserving pairwise distances with high probability $(1 - 1/\beta^2)$. Here, $\mathbf{1}_{\delta > 0}$ is the indicator function, which equals 1 if $\delta > 0$, and 0 otherwise.

The acceptance condition is then evaluated in the lower-dimensional projected space using a scaled parameter $\tilde{\varepsilon}_t := \frac{\max\{\varepsilon_t, \varepsilon_t^{\oslash}\}}{\sqrt{1-\delta}}$. The modified acceptance rule becomes:

$$\min_{i \in \mathcal{I}_t^m} \left(f(x_i) + \tilde{\varepsilon}_t \cdot \|\mathbf{P}x - \mathbf{P}x_i\|_2 \right) \geq \max_{j \in [t]} f(x_j), \tag{7}$$

where δ is the distortion due to projection and $\mathcal{I}_t^m$ is the Worst-m memory set described above.

Definition 3. (ECPv2 Acceptance Region) *The set of points potentially accepted by ECPv2 at any time step* $t \geq 1$, *for any value* $\varepsilon_t > 0$, *for any controlled distortion* $\delta \in [0, 1)$, *any value of* $\beta > 1$, *resulting in a projection matrix* $\mathbf{P}$ *as define in Equation* (6), *and for any integer* $m \geq 1$ *is defined as:* $\mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P}) \triangleq \{x \in \mathcal{X} : \min_{i \in \mathcal{I}_t^m} (f(X_i) + \frac{\max\{\varepsilon_t, \varepsilon_t^{\oslash}\}}{\sqrt{1-\delta}} \cdot \|\mathbf{P}x - \mathbf{P}X_i\|_2) \geq \max_{j=1,\cdots,t} f(X_j)\}$.

The following sections detail the theoretical motivation and guarantees behind each modification.

Adaptive Lower Bound $\varepsilon_t^{\oslash}$

In ECP algorithm, a new point $x \in \mathcal{X}$ is evaluated only if it satisfies the acceptance condition in Equation (2). This mechanism controls the exploration by controlling the admissibility of new samples. However, when ε_t is too small, no candidate point may satisfy the acceptance condition, resulting in wasted proposals and inefficiency.

The ECP addresses this issue by gradually increasing ε_t multiplicatively (i.e., $\varepsilon_t \leftarrow \varepsilon_t \cdot \tau_{n,d}$), which guarantees progress but can incur a large number of rejections before reaching a viable threshold. To reduce unnecessary overhead, we derive a principled lower bound on ε_t to ensure that not every sampled point is trivially rejected.

Algorithm 1: ECPv2

Input: $n \in \mathbb{N}^\star$, $m \geq 1$, $\delta \in [0, 1)$, $\beta > 1$, $\varepsilon_1 > 0$, $\tau_{n,d} > 1$, $C > 1$, search space $\mathcal{X} \subset \mathbb{R}^d$, objective function f

Initialize projection matrix $\mathbf{P}$ as in Equation (6)
Sample $x_1 \sim \mathcal{U}(\mathcal{X})$, evaluate $f(x_1)$
$\hat{x}_1 \leftarrow \mathbf{P}x_1$, $t \leftarrow 1$, $h_1 \leftarrow 1$, $h_2 \leftarrow 0$
while $t < n$ **do**
  Sample $x_{t+1} \sim \mathcal{U}(\mathcal{X})$, $h_{t+1} \leftarrow h_{t+1} + 1$
  $\hat{x}_{t+1} \leftarrow \mathbf{P}x_{t+1}$ ▷ (Projection **if** $\delta > 0$)
  if $(h_{t+1} - h_t) > C$ **then** ▷ (Growth)
    $\varepsilon_t \leftarrow \tau_{n,d} \cdot \varepsilon_t$, $h_{t+1} \leftarrow 0$
  end if
  if $x_{t+1} \in \mathcal{A}_{\text{ECPv2}}(\varepsilon_t, t, m, \mathbf{P})$ **then** ▷ (Acceptance)
    Evaluate $f(x_{t+1})$
    $t \leftarrow t + 1$, $h_t \leftarrow h_{t+1}$
    $\varepsilon_{t+1}^{\oslash} \leftarrow \frac{\max_{i \leq t} f(x_i) - \min_{i \leq t} f(x_i)}{\text{diam}(\mathcal{X})}$
    $\varepsilon_{t+1} \leftarrow \max(\tau_{n,d} \cdot \varepsilon_t, \varepsilon_{t+1}^{\oslash})$, $h_{t+1} \leftarrow 0$
  end if
end while
return $x_{\hat{i}}$ where $\hat{i} \in \arg\max_{i=1,\ldots,n} f(x_i)$

Lemma 1 (Lower Bound on ε_t)**.** *Let* $\mathcal{X} \subset \mathbb{R}^d$ *be a compact search space with bounded diameter* $\text{diam}(\mathcal{X})$. *Let* $\{X_1, \ldots, X_t\} \subset \mathcal{X}$ *be the set of previously evaluated points. Define:* $f_{\max_t} := \max_{i=1,\ldots,t} f(X_i)$ *and* $f_{\min_t} := \min_{i=1,\ldots,t} f(X_i)$. *If there exists a point* $x \in \mathcal{X}$ *satisfying the ECP acceptance condition, then* ε_t *must satisfy:* $\varepsilon_t \geq \varepsilon_t^{\oslash} = \frac{f_{\max_t} - f_{\min_t}}{\text{diam}(\mathcal{X})}$.

This bound (proved in Appendix C) provides for each time step t a theoretically necessary initialization for ε_t (for not being empty). Instead of relying solely on geometric growth, this lower bound can be used to re-initialize ε_t adaptively, avoiding large rejection loops without sacrificing convergence or optimality.

For the same ε_t it is shown (Appendix C) that the acceptance condition with the lower-bound is a super-set of ECP.

Lemma 2. *Let* $\mathcal{A}_{ECP}(\varepsilon_t, t)$ *denote the acceptance region defined by Equation* (2), *and let* $\mathcal{A}_t(\varepsilon_t, t)$ *denote the region defined by Equation* (4). *Then, for any* ε_t, *we have*

$$\mathcal{A}_{ECP}(\varepsilon_t, t) \subseteq \mathcal{A}_t(\varepsilon_t, t).$$

This result highlights that ECPv2 does not miss any potential maximizer that would have been accepted by ECP, and may even allow more potential maximizers to be accepted, ones that might have been rejected due to the overly restrictive value of ε_t. Further empirical validation and discussion of this bound are provided in Appendix F.

Worst-m Memory Mechanism

In the ECP algorithm, each new candidate $x \in \mathcal{X}$ is evaluated against *all* previously queried points $\{X_1, \ldots, X_t\}$ using the acceptance condition in Equation (2). While theoretically sound, this condition becomes increasingly restrictive and computationally costly as t grows, particularly when many historical points are near-optimal or redundant.

This scalability bottleneck can hinder both efficiency and exploratory behavior.

To mitigate these issues, and inspired by the stochastic maximization strategy introduced in (Fourati, Aggarwal, and Alouini 2024), we propose the *Worst-m memory mechanism*, which limits the acceptance check to only the *m worst-performing* points in the archive. Formally, let

$$\mathcal{I}_t^m := \arg \min_{\substack{S \subseteq \{1,\ldots,t\} \\ |S|=m}} \sum_{i \in S} f(X_i)$$

denote the indices corresponding to the m lowest-valued function evaluations among $\{f(X_1), \ldots, f(X_t)\}$.

Intuitively, this strategy focuses rejection pressure on clearly suboptimal regions of the search space, thereby avoiding unnecessary constraints from high-performing points that might otherwise hinder exploration. By excluding these points, we obtain a broader acceptance region. Notably, when $m = n$, we exactly recover the ECP condition.

The Worst-m memory rule *relaxes* the original condition, inducing a larger acceptance region, as stated below.

Lemma 3. *Let $\mathcal{A}_t(\varepsilon_t, t)$ denote the acceptance region defined by Equation* (2)*, and let $\mathcal{A}_t(\varepsilon_t, t, m)$ denote the region defined by Equation* (5)*. Then, for any $m \geq 1$, we have*

$$\mathcal{A}_t(\varepsilon_t, t) \subseteq \mathcal{A}_t(\varepsilon_t, t, m).$$

See Appendix C for the proof and Appendix G for empirical analysis of the Worst-m strategy.

Random Projection-Based Acceleration

In high-dimensional settings, computing distances between candidate points and the archive of previously evaluated points becomes a dominant computational bottleneck in ECP. To alleviate this, we introduce a *projection-based acceleration* strategy that employs a fixed random projection $\mathbf{P} : \mathbb{R}^d \to \mathbb{R}^{d'}$ for which we show the following results that are proved in Appendix A.

Lemma 4. *Let $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$ be a set of n vectors. Fix a distortion parameter $\delta \in [0, 1)$. Let the projection matrix $\mathbf{P}$ be generated as in Equation* (6)*. Let $x_i' = \mathbf{P}x_i$ be the projection of each vector x_i. If $d' \geq \frac{8 \log(\beta n)}{\delta^2 - \delta^3}$, then with probability at least $1 - \frac{1}{\beta^2}$, the following inequality holds simultaneously for all $i, j \in \{1, \ldots, n\}$:*

$$(1 - \delta)\|x_i - x_j\|_2^2 \leq \|x_i' - x_j'\|_2^2 \leq (1 + \delta)\|x_i - x_j\|_2^2.$$

The lemma is obtained by adapting classical Johnson-Lindenstrauss (JL) results (Arriaga and Vempala 2006) describing the concentration properties of random Gaussian projections. These results guarantee that norms and pairwise distances are preserved up to controlled distortion after dimensionality reduction. The specific restated and adapted forms used in our analysis appear in Appendix A.

We now formalize the relationship between the acceptance regions in the original and projected spaces.

Lemma 5. *Let $\mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P})$ denote the acceptance region of ECPv2 in Definition 3 and let $\mathcal{A}_t(\varepsilon_t, t, m)$ denote the region defined by Equation* (5)*. Let $\mathbf{P}$ be the random projection matrix with reduction d' as in Lemma 4. Then, for any $m \geq 1$, we have with probability at least $1 - 1/\beta^2$:*

$$\mathcal{A}_t(\varepsilon_t, t, m) \subseteq \mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P}).$$

Extensive empirical analysis of the projection mechanism, including its sensitivity to the parameters δ and β, is provided in Appendix D, while principled values for these parameters are derived in a subsequent section.

Combining Lemmas 2, 3, and 5, we obtain:

Corollary 1. *Let $\mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P})$ denote the acceptance region of ECPv2, as defined in Definition 3, and let $\mathcal{A}_{ECP}(\varepsilon_t, t)$ denote the region defined in Definition 2. Let $\mathbf{P}$ be a random projection matrix with reduced dimension d', as in Lemma 4. Then, for any $m \leq n$, with probability at least $1 - 1/\beta^2$, the following inclusion holds:*

$$\mathcal{A}_{ECP}(\varepsilon_t, t) \subseteq \mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P}).$$

Corollary 1 establishes that the acceptance region of ECPv2 strictly contains that of ECP with high probability. In particular, the proposed techniques, including the lower bound, the worst-m mechanism, and the projection-based acceleration, guarantee that any point accepted by ECP will also be accepted by ECPv2 with high-probability, ensuring that no potential maximizers are erroneously discarded due to an overly restrictive acceptance region.

Importantly, this enlargement is controlled: the distortion δ introduced by the projection can be made arbitrarily small with high probability by choosing a sufficiently large projection dimension, specifically $d' = \mathcal{O}\left(\log(\beta n)/(\delta^2 - \delta^3)\right)$, while the worst-$m$ mechanism can be adjusted via the choice of m. Consequently, the worst-m and projection techniques provide mechanisms for accelerating ECP while preserving its conservative decision criteria. In high-dimensional settings, this yields substantial computational gains.

Computational Complexity Gains

A central motivation for ECPv2 is to reduce the high computational overhead of Lipschitz-based global optimization, especially in high-dimensional settings or under large evaluation budgets. Table 1 compares the theoretical complexity of ECPv2 with existing methods, including ECP, AdaLIPO, and AdaLIPO+. The analysis highlight substantial reductions in both runtime and memory requirements.

Runtime Complexity Gains. In ECP, AdaLIPO, and AdaLIPO+, each iteration requires checking the acceptance condition against all previously evaluated points. This involves computing $t \leq n$ distances in $\mathbb{R}^d$, leading to a per-iteration complexity of $\Omega(td)$. Over n evaluations, this results in an overall runtime of $\Omega(n^2 d)$, which becomes prohibitive as either n or d increases. Rather than considering all $t = \mathcal{O}(n)$ past evaluations, ECPv2 computes the acceptance condition using only the m worst-performing archive points, reducing the number of distance computations from t to $m \ll n$. Furthermore, before computing distances, the candidate is projected from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$ and the distance is then computed in $\mathbb{R}^{d'}$. Projection incurs a cost of $\Omega(dd')$ per iteration, and each projected distance is computed in $\Omega(d')$

Lipschitz Method	Memory	Runtime	Regret Upper Bound
AdaLIPO	$\mathcal{O}(nd)$	$\Omega(n^2 d)$	$\mathcal{O}_{1-\frac{1}{\xi}}\left(c^{\ddagger} \cdot p^{-\frac{1}{d}} \cdot (\ln(3/\xi))^{\frac{1}{d}} \cdot k \cdot n^{-\frac{1}{d}}\right)$
AdaLIPO+	$\mathcal{O}(nd)$	$\Omega(n^2 d)$	—
ECP	$\mathcal{O}(nd)$	$\Omega(n^2 d)$	$\mathcal{O}_{1-\frac{1}{\xi}}\left(c^{\star} \cdot (\ln(1/\xi))^{\frac{1}{d}} \cdot k \cdot n^{-\frac{1}{d}}\right)$
ECPv2 (this work)	$\mathcal{O}((m+d)\log(\beta n))$	$\Omega(n(m+d)\log(\beta n))$	$\mathcal{O}_{1-\frac{1}{\beta^2}-\frac{1}{\xi}}\left(c^{\star} \cdot (\ln(1/\xi))^{\frac{1}{d}} \cdot k \cdot n^{-\frac{1}{d}}\right)$
Lower Bound (Bull 2011)	—	—	$\Omega\left(\mathrm{rad}(\mathcal{X}) \cdot k \cdot n^{-\frac{1}{d}}\right)$

Table 1: Comparison of Lipschitz global optimization methods with unknown Lipschitz constants. d: dimension, n: evaluation budget, $m \ll n$: number of worst-performing points retained, β: ECPv2 hyperparameter, p: AdaLIPO sampling probability. $c^{\star} = \mathrm{diam}(\mathcal{X}) \cdot \log_{\tau_{n,d}}\left(\frac{k}{\varepsilon_1}\right)^{1/d}$, $c^{\ddagger} = \mathrm{diam}(\mathcal{X}) \cdot \left(5 + \frac{2\ln(\xi/3)}{\ln(1-\Gamma(f,k_{i^*-1}))}\right)^{1/d}$.

time. With these techniques, the per-iteration complexity becomes $\Omega(dd' + md')$, where we set $d' = \mathcal{O}(\log(\beta n))$. The total runtime complexity over n evaluations is then $\Omega\big(n(m+d)\log(\beta n)\big)$, which is linear in n (up to a log factor) and only linear in d when m and β are held constant.

Memory Complexity Gains. In addition to runtime savings, ECPv2 also provides substantial improvements in memory efficiency. Prior Lipschitz-based methods such as ECP, AdaLIPO, and AdaLIPO+ store the entire evaluation history, each of dimensionality d, yielding a memory complexity of $\mathcal{O}(nd)$. ECPv2, by contrast, only stores: A projection matrix of size $\mathcal{O}(d\log(\beta n))$ and the m worst-performing projected points, requiring $\mathcal{O}(m\log(\beta n))$ memory. Thus, the total memory complexity of ECPv2 is $\mathcal{O}((m+d)\log(\beta n))$, which scales and only linearly with the dimension d and only logarithmically with the budget n.

No-Regret Guarantees

Although the proposed mechanisms in ECPv2 significantly reduce computational cost and minimize unnecessary rejections, the algorithm retains theoretical guarantees. In particular, ECPv2 preserves the no-regret behavior over Lipschitz objectives. We present both asymptotic and finite-time guarantees below, with full proofs available in Appendix B.

Theorem 1 (Asymptotic No-Regret)**.** *Let $f \in Lip(k)$ for some unknown Lipschitz constant $k > 0$, and let ECPv2 be run with any $\delta \in [0,1)$, $\varepsilon_1 > 0$, integer $m \geq 1$, $\beta > 1$, $\tau_{n,d} > 1$, and $C > 1$. Then the simple regret satisfies:*

$$\mathcal{R}_{ECPv2,f}(n) \xrightarrow{p} 0.$$

Theorem 2 (Finite-Time Regret Bound)**.** *Let $f \in Lip(k)$ be a non-constant function over a compact domain $\mathcal{X} \subset \mathbb{R}^d$, and let ECPv2 be tuned with any $\delta \in [0,1)$, integer $m \geq 1$, $\beta > 1$, $\varepsilon_1 > 0$, $\tau_{n,d} > 1$, and $C > 1$. Then for any budget $n \in \mathbb{N}^\star$ and confidence level $\xi \in (0, 1-\frac{1}{\beta^2})$, with probability at least $1 - \frac{1}{\beta^2} - \xi$, the simple regret satisfies:*

$$\mathcal{R}_{ECPv2,f}(n) \leq k \cdot \mathrm{diam}(\mathcal{X}) \cdot \log_{\tau_{n,d}}\left(\frac{k}{\varepsilon_1}\right)^{\frac{1}{d}}\left(\frac{\ln(1/\xi)}{n}\right)^{\frac{1}{d}}.$$

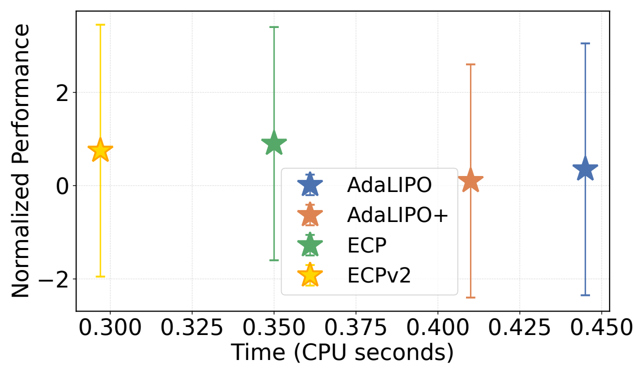


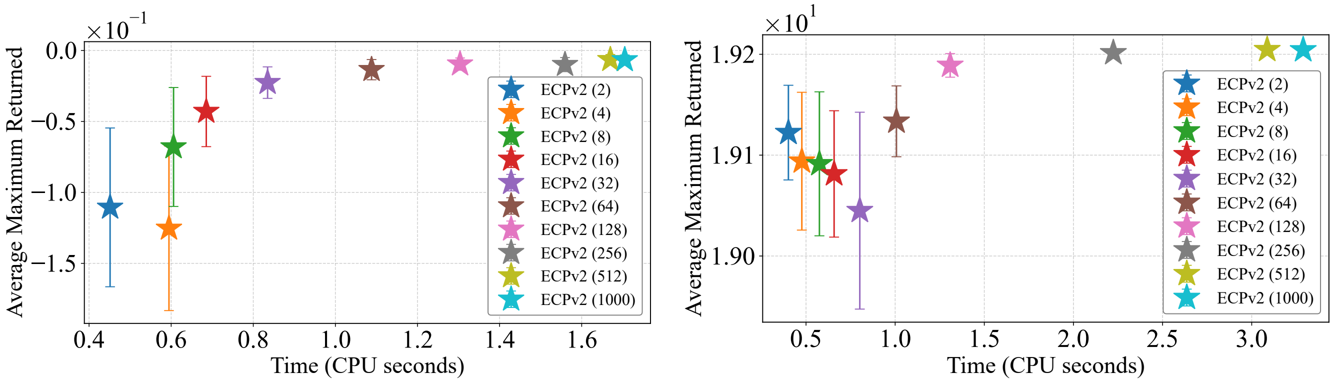
Figure 1: Comparison of Lipschitz optimization methods on Rosenbrock with $d \in \{3, 100, 200, 300, 500\}$. Each star shows the mean performance across dimensions after $n = 200$ evaluations, averaged over 100 runs. ECPv2 uses hyperparameters ($\beta = 5$, $\delta = 2/3$, $m = 8$).

Discussion. Theorem 1 establishes that ECPv2 is no-regret. Theorem 2 further provides a finite-time performance guarantee, bounding the simple regret with high probability. Specifically, ECPv2 achieves the minimax optimal convergence rate of $\mathcal{O}(kn^{-1/d})$, matching the lower bound $\Omega(kn^{-1/d})$ in Proposition 1. Notably, this is obtained with significantly reduced runtime and memory cost compared to the previous Lipschitz-based optimizers (see Table 1).

In fact, ECPv2 attains the same regret upper bound as ECP, but with a slightly lower confidence level. This trade-off is explicitly governed by the parameter β, which balances three factors: computational complexity, memory usage, and the confidence level of the regret bound. A larger β improves the confidence guarantee (bringing it closer to 1), but potentially increases the overhead dealing with larger dimensions. Conversely, a smaller β accelerates execution at the cost of a looser high-probability guarantee.

Principled Hyperparameter Choices (β, δ)

The random projection component introduces the distortion tolerance δ and the confidence scaling factor β. These pa-



(a) Himmelblau

(b) Holder

Figure 2: Ablation study on the projection dimension m in ECPv2, using fixed parameters $\delta = 2/3$ and $\beta = 5$. For each benchmark function, every method is allocated $n = 1000$ evaluations and performance is averaged over 100 independent runs.

rameters jointly determine the reduced projection dimension d' via the bound established in Lemma 4. The parameter β governs the probability that all pairwise distances are preserved within the specified distortion interval.

To ensure high-confidence embeddings, with a 96% success rate, we fix $\beta = 5$. Furthermore, we analytically derive the value of δ that minimizes the required projection dimension, and find the optimal setting to be $\delta = \frac{2}{3}$. This choice yields the most dimension-efficient embedding. Full derivations and sensitivity analysis are provided in Appendix D.

Empirical Evaluation

We evaluate the performance of ECPv2 on high-dimensional, non-convex global optimization tasks and compare it against Lipschitz-based global optimization methods (Table 1), including ECP (Fourati et al. 2025), AdaLIPO (Malherbe and Vayatis 2017), and AdaLIPO+ (Serré et al. 2024).

Comparisons with general-purpose optimizers such as SMAC3 (Lindauer et al. 2022a), DIRECT (Jones and Martins 2021), and Dual Annealing (Xiang et al. 1997) are provided in Appendix I, implementation details in Appendix J, examination of the sensitivity to the projection dimension and the influence of δ and β in Appendix D, and extensive ablations in Appendices E–G.

Figure 1 presents an ablation of the projection dimension m on Himmelblau and Holder using fixed parameters $\delta = 2/3$ and $\beta = 5$, illustrating how m influences the behavior of ECPv2. Appendix E extends this study to ten additional benchmark functions and shows that the Worst-m approximation provides substantial computational savings: evaluating candidates against only a small subset of the worst-performing points reduces the acceptance-test cost by roughly 4–5×. Remarkably, small values such as $m \in 8, 16, 32, 64, 128$ often match and sometimes even exceed the performance of the full method by allowing more permissive early exploration. Larger choices of m yield diminishing returns and eventually recover both the behavior and the runtime overhead of the original ECP.

Appendix F demonstrates that the adaptive lower bound $\varepsilon_t^{\oslash}$ is crucial for preventing empty acceptance regions. Enabling it alone yields roughly $2\times$ faster runtimes at low evaluation budgets and can improve performance.

Appendix G shows that combining random projections with the Worst-m rule produces the fastest ECPv2 variant. On the 500-dimensional Rosenbrock function, all ECPv2 configurations outperform ECP in runtime, with the full version achieving the best speed–quality tradeoff.

Figure 2 reports results on Rosenbrock functions with dimensions $d \in \{3, 100, 200, 300, 500\}$, comparing ECPv2 (using the default hyperparameters $\beta = 5$, $\delta = 2/3$, and $m = 8$) to Lipschitz-based baselines from Table 1. ECPv2 attains competitive performance while significantly reducing runtime (Appendix H), even at moderate budgets ($n = 200$). Additional results on high-dimensional settings—Rosenbrock with $d = 500$ and Powell with $d = 1000$—are presented in Appendix H. In both cases, ECPv2 not only converges roughly $2\times$ faster than ECP but also reaches higher optimization scores. These gains grow more pronounced with increasing dimensionality, illustrating the scalability and efficiency of our approach.

Conclusion

We introduced ECPv2, a fast and scalable algorithm for global optimization of Lipschitz-continuous functions with unknown constants. The method builds on the ECP framework, which already provides strong empirical performance and theoretical guarantees with minimal tuning. ECPv2 retains these guarantees while reducing computational and memory costs through an adaptive lower bound, a Worst-m selective memory, and a projection-based acceleration. These components, used individually or in combination, offer additional benefits in large-dimensional settings and under large evaluation budgets. Across a wide range of benchmarks, ECPv2 matches or exceeds the performance of several black-box optimization methods while delivering significant wall-clock speedups, demonstrating the robustness and scalability of the ECP family for black-box optimization.

References

Antonova, R.; Yang, J.; Jatavallabhula, K. M.; and Bohg, J. 2023. Rethinking optimization with differentiable simulation from a global perspective. In *Conference on Robot Learning*, 276–286. PMLR.

Arriaga, R. I.; and Vempala, S. 2006. An algorithmic theory of learning: Robust concepts and random projection. *Machine learning*, 63(2): 161–182.

Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A. G.; and Bakshy, E. 2020. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in neural information processing systems*, 33: 21524–21538.

Bartlett, P. L.; Gabillon, V.; and Valko, M. 2019. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, 184–206. PMLR.

Berkenkamp, F.; Schoellig, A. P.; and Krause, A. 2019. No-Regret Bayesian Optimization with Unknown Hyperparameters. *Journal of Machine Learning Research*, 20(50): 1–24.

Brooks, S. H. 1958. A discussion of random methods for seeking maxima. *Operations research*, 6(2): 244–251.

Bubeck, S.; Munos, R.; Stoltz, G.; and Szepesvári, C. 2011. X-Armed Bandits. *Journal of Machine Learning Research*, 12(5).

Bull, A. D. 2011. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10).

Chen, L.; Chen, J.; Goldstein, T.; Huang, H.; and Zhou, T. 2024. InstructZero: Efficient Instruction Optimization for Black-Box Large Language Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 6503–6518. PMLR.

D'Ambrosio, D. B.; Abeyruwan, S. W.; Graesser, L.; Iscen, A.; Amor, H. B.; Bewley, A.; Reed, B.; Reymann, K.; Takayama, L.; Tassa, Y.; et al. 2024. Achieving human level competitive robot table tennis. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*.

Floudas, C. A.; and Pardalos, P. M. 2014. *Recent Advances in Global Optimization*. Princeton, NJ: Princeton University Press.

Fourati, F.; Aggarwal, V.; and Alouini, M.-S. 2024. Stochastic Q-learning for Large Discrete Action Spaces. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 13734–13759. PMLR.

Fourati, F.; Kharrat, S.; Aggarwal, V.; and Alouini, M.-S. 2025. Every Call is Precious: Global Optimization of Black-Box Functions with Unknown Lipschitz Constants. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, 5176–5184. PMLR.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Hansen, N. 2006. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, 75–102.

Hansen, N.; Akimoto, Y.; and Baudis, P. 2019. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634.

Hansen, N.; and Ostermeier, A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation*, 312–317. IEEE.

Jones, D. R.; and Martins, J. R. 2021. The DIRECT algorithm: 25 years Later. *Journal of global optimization*, 79(3): 521–566.

Jones, D. R.; Perttunen, C. D.; and Stuckman, B. E. 1993. Lipschitzian optimization without the Lipschitz constant. *Journal of optimization Theory and Applications*, 79: 157–181.

Kawaguchi, K.; Maruyama, Y.; and Zheng, X. 2016. Global continuous optimization with error bound and fast convergence. *Journal of Artificial Intelligence Research*, 56: 153–195.

Kharrat, S.; Fourati, F.; and Canini, M. 2025. ACING: Actor-Critic for Instruction Learning in Black-Box LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 19086–19113. Association for Computational Linguistics. ISBN 979-8-89176-332-6.

Kirkpatrick, S.; Gelatt Jr, C. D.; and Vecchi, M. P. 1983. Optimization by simulated annealing. *science*, 220(4598): 671–680.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2008. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 681–690.

Lin, X.; Wu, Z.; Dai, Z.; Hu, W.; Shu, Y.; Ng, S.-K.; Jaillet, P.; and Low, B. K. H. 2024. Use Your INSTINCT: INSTruction optimization for LLMs usIng Neural bandits Coupled with Transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 30317–30345. PMLR.

Lindauer, M.; Eggensperger, K.; Feurer, M.; Biedenkapp, A.; Deng, D.; Benjamins, C.; Ruhkopf, T.; Sass, R.; and Hutter, F. 2022a. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54): 1–9.

Lindauer, M.; Eggensperger, K.; Feurer, M.; Biedenkapp, A.; Deng, D.; Benjamins, C.; Ruhkopf, T.; Sass, R.; and Hutter, F. 2022b. SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. *Journal of Machine Learning Research*, 23(54): 1–9.

Malherbe, C.; and Vayatis, N. 2017. Global optimization of Lipschitz functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 2314–2323. PMLR.

Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; and Teller, E. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092.

Munos, R. 2011. Optimistic optimization of a deterministic function without the knowledge of its smoothness. *Advances in neural information processing systems*, 24.

Pardalos, P. M. 2013. *Handbook of global optimization*, volume 2. Springer Science & Business Media.

Preux, P.; Munos, R.; and Valko, M. 2014. Bandits attack function optimization. In *2014 IEEE congress on evolutionary computation (CEC)*, 2245–2252. IEEE.

Serré, G.; Beja-Battais, P.; Chirrane, S.; Kalogeratos, A.; and Vayatis, N. 2024. LIPO+: Frugal Global Optimization for Lipschitz Functions. *arXiv preprint arXiv:2406.19723*.

Stork, J.; Eiben, A. E.; and Bartz-Beielstein, T. 2022. A new taxonomy of global optimization algorithms. *Natural Computing*, 21(2): 219–242.

Törn, A.; and Žilinskas, A. 1989. *Global optimization*, volume 350. Springer.

Tsallis, C. 1988. Possible generalization of Boltzmann-Gibbs statistics. *Journal of statistical physics*, 52: 479–487.

Xiang, Y.; Sun, D.; Fan, W.; and Gong, X. 1997. Generalized simulated annealing algorithm and its application to the Thomson model. *Physics Letters A*, 233(3): 216–220.

Zabinsky, Z. B. 2013. *Stochastic adaptive search for global optimization*, volume 72. Springer Science & Business Media.

Zhou, D.; Li, L.; and Gu, Q. 2020. Neural Contextual Bandits with UCB-based Exploration. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 11492–11502. PMLR.