

Appendix A: Random Projection Guarantees

To support our analysis, we restate and adapt key results from (Arriaga and Vempala 2006) that characterize the behavior of random Gaussian projections. These results guarantee the concentration of norms and distances after dimensionality reduction, which is critical to our use of random projections in Lipschitz optimization.

Norm Preservation for a Single Vector. The following lemma shows that projecting a fixed vector using a random Gaussian matrix approximately preserves its squared norm, with high probability:

Lemma 6 (Adapted from Lemma 2 in (Arriaga and Vempala 2006)). *Let $R = (r_{ij})$ be a random $n \times d'$ matrix, where each entry r_{ij} is drawn independently from $\mathcal{N}(0, 1)$. For any fixed vector $u \in \mathbb{R}^n$ and any $\delta \in (0, 1)$, let $u' = \frac{1}{\sqrt{d'}} R^T u$. Then $\mathbb{E}[\|u'\|_2^2] = \|u\|_2^2$, and:*

$$\mathbb{P}[\|u'\|_2^2 > (1 + \delta)\|u\|_2^2] \leq e^{-(\delta^2 - \delta^3)\frac{d'}{4}},$$

$$\mathbb{P}[\|u'\|_2^2 < (1 - \delta)\|u\|_2^2] \leq e^{-(\delta^2 - \delta^3)\frac{d'}{4}}.$$

Distance Preservation Between Pairs of Vectors. Using the lemma above, we immediately obtain the following corollary, which states that pairwise Euclidean distances are preserved under the same random projection with high probability. This result is essential for our dimensionality reduction technique:

Corollary 2 (Adapted from Theorem 2 (Neuronal RP) in (Arriaga and Vempala 2006)). *Let R be a random $n \times d'$ matrix with entries drawn independently from $\mathcal{N}(0, 1)$. For any $u, v \in \mathbb{R}^n$, define the projections $u' = \frac{1}{\sqrt{d'}} R^T u$ and $v' = \frac{1}{\sqrt{d'}} R^T v$. Then for any $\delta \in (0, 1)$,*

$$\begin{aligned} \mathbb{P}[(1 - \delta)\|u - v\|_2^2 \leq \|u' - v'\|_2^2 \leq (1 + \delta)\|u - v\|_2^2] \\ \geq 1 - 2e^{-(\delta^2 - \delta^3)\frac{d'}{4}}. \end{aligned}$$

Distance Preservation for a Finite Set of Vectors. The following theorem shows that a random Gaussian projection approximately preserves all pairwise distances among a set of n vectors, with high probability. This is crucial when applying dimensionality reduction to all the sampled points by ECPv2.

Proof of Lemma 4

Proof. When $\delta = 0$, by design of the projection matrix in Equation (6), it is simply the identity, and the result holds trivially. For $\delta \in (0, 1)$, we proceed as follows. There are $\binom{n}{2} < \frac{n^2}{2}$ distinct pairs of vectors. From the Corollary 2, the probability that a single pair deviates outside the $(1 \pm \delta)$ bounds is at most $2 \exp\left(-(\delta^2 - \delta^3)\frac{d'}{4}\right) \leq 2 \exp(-2 \log(\beta \cdot n)) = 2 \exp(-\log(\beta^2 \cdot n^2))$. By the union bound over all pairs, we have the probability that any pair has a large distortion is at most $\frac{1}{\beta^2}$. \square

Proof of Lemma 5

Proof. Let $x \in \mathcal{A}_t(\varepsilon_t, t, m)$, which implies

$$\min_{i \in \mathcal{I}_t^m} (f(X_i) + \max\{\varepsilon_t, \varepsilon_t^\otimes\} \cdot \|x - X_i\|_2) \geq \max_j f(X_j).$$

By the Lemma 4, for all i , with a probability at least $1 - \frac{1}{\beta^2}$:

$$\|\mathbf{P}x - \mathbf{P}X_i\|_2 \leq \frac{\|x - X_i\|_2}{\sqrt{1 - \delta}}.$$

Multiplying both sides by $\tilde{\varepsilon}_t = \max\{\varepsilon_t, \varepsilon_t^\otimes\}/\sqrt{1 - \delta}$ gives

$$\tilde{\varepsilon}_t \cdot \|\mathbf{P}x - \mathbf{P}X_i\|_2 \geq \max\{\varepsilon_t, \varepsilon_t^\otimes\} \cdot \|x - X_i\|_2.$$

Thus, for all i , with probability at least $1 - \frac{1}{\beta^2}$:

$$f(X_i) + \tilde{\varepsilon}_t \cdot \|\mathbf{P}x - \mathbf{P}X_i\|_2 \geq f(X_i) + \max\{\varepsilon_t, \varepsilon_t^\otimes\} \cdot \|x - X_i\|_2.$$

Taking the minimum over $i \in \mathcal{I}_t^m$ yields with probability at least $1 - \frac{1}{\beta^2}$:

$$\begin{aligned} \min_{i \in \mathcal{I}_t^m} (f(X_i) + \tilde{\varepsilon}_t \cdot \|\mathbf{P}x - \mathbf{P}X_i\|_2) \geq \\ \min_{i \in \mathcal{I}_t^m} (f(X_i) + \max\{\varepsilon_t, \varepsilon_t^\otimes\} \cdot \|x - X_i\|_2). \end{aligned}$$

Hence,

$$\min_{i \in \mathcal{I}_t^m} (f(X_i) + \tilde{\varepsilon}_t \cdot \|\mathbf{P}x - \mathbf{P}X_i\|_2) \geq \max_j f(X_j).$$

Hence, $x \in \mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P})$. \square

Appendix B: No-Regret Guarantees

The acceptance region of ECPv2, similar to ECP, is motivated by the definition of consistent functions.

Definition 4. (CONSISTENT FUNCTIONS) *The active subset of Lipschitz functions, with a Lipschitz constant k , consistent with the black-box function f over $t \geq 1$ evaluated samples $(x_1, f(x_1)), \dots, (x_t, f(x_t))$ is: $\mathcal{F}_{k,t} \triangleq \{g \in Lip(k) : \forall i \in \{1, \dots, t\}, g(x_i) = f(x_i)\}$.*

Using the above definition of a consistent function, we define the subset of points that can maximize at least some function g within that subset of consistent functions and possibly maximize the target f .

Definition 5. (POTENTIAL MAXIMIZERS) *For a Lipschitz function f with a Lipschitz constant $k \geq 0$, let $\mathcal{F}_{k,t}$ be the set of consistent functions with respect to f , as defined in Definition 4. For any iteration $t \geq 1$, the set of potential maximizers is defined as follows: $\mathcal{P}_{k,t} \triangleq \left\{x \in \mathcal{X} : \exists g \in \mathcal{F}_{k,t} \text{ where } x \in \arg \max_{x \in \mathcal{X}} g(x)\right\}$.*

We can then show the relationship between the potential maximizers and our proposed acceptance region.

Lemma 7 (Lemma 8 in (Malherbe and Vayatis 2017)). *If $\mathcal{P}_{k,t}$ denotes the set of potential maximizers of the function f , as defined in Definition 4, then we have $\mathcal{P}_{k,t} = \{x \in \mathcal{X} : \min_{i=1, \dots, t} (f(x_i) + k \cdot \|x - x_i\|_2) \geq \max_{j=1, \dots, t} f(x_j)\}$.*

Proposition 2. (POTENTIAL OPTIMALITY) For any iteration t , if $\mathcal{P}_{k,t}$ denotes the set of potential maximizers of $f \in \text{Lip}(k)$, as in Definition 4, and $\mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P})$ denotes our acceptance region, defined in Equation (7), then with probability at least $1 - 1/\beta^2$: $\forall \varepsilon_t > k, \mathcal{P}_{k,t} \subseteq \mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P})$.

Proof. From Proposition 2 of (Fourati et al. 2025), we have $\mathcal{P}_{k,t} \subseteq \mathcal{A}_{ECP}(\varepsilon_t, t)$ for any $\varepsilon_t > k$. By Corollary 1, $\mathcal{A}_{ECP}(\varepsilon_t, t) \subseteq \mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P})$. Therefore, by transitivity, $\mathcal{P}_{k,t} \subseteq \mathcal{A}_{ECPv2}(\varepsilon_t, t, m, \mathbf{P})$ for any $\varepsilon_t > k$. \square

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:

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

Therefore, when ε_t reaches or exceeds k , i.e., $\varepsilon_t \geq k$, which is unavoidable with a growing number of evaluations, the acceptance space does not exclude any potential maximizer, as all potential maximizers remain within the acceptance condition, which is crucial to guarantee the no-regret property of ECPv2.

First, we define the i^* as the hitting time, after which ε_t reaches or overcomes the Lipschitz constant k .

Definition 6. (HITTING TIME) For the sequence $(\varepsilon_i)_{i \in \mathbb{N}}$ and the unknown Lipschitz constant $k > 0$, we can define $i^* \triangleq \min \{i \in \mathbb{N}^*: \varepsilon_i \geq k\}$.

In the following lemma, we upper-bound the time t after which ε_t is guaranteed to reach or exceed k . This shows that for sufficiently large t , the event of reaching k is inevitable, with proof provided in Appendix A.

Lemma 8. (HITTING TIME UPPER-BOUND) For any function $f \in \text{Lip}(k)$, for any coefficient $\tau_{n,d} > 1$, any initial value $\varepsilon_1 > 0$, any constant $C > 1$, any integer m , any distortion $\delta \in [0, 1]$, and any integer $m \geq 1$, the hitting time i^* is upperbounded as follows:

$$\forall \varepsilon_1 > 0, \quad i^* \leq \max \left(\left\lceil \log_{\tau_{n,d}} \left(\frac{k}{\varepsilon_1} \right) \right\rceil, 1 \right).$$

Proof. Notice that for all $t \geq 1$, for any choice of $C > 1$, $\varepsilon_1 > 0$, $\tau_{n,d} > 1$, $\delta \in [0, 1]$, and $m \geq 1$, ε_t grows throughout the iterations when the stochastic growth condition is satisfied and when a point is evaluated. While the growth condition may or may not occur, the latter happens deterministically after every evaluation. Therefore, $\varepsilon_t \geq \varepsilon_1 \tau_{n,d}^{t-1}$. Hence, the result follows from the non-decreasing and diverging geometric growth of $\varepsilon_1 \tau_{n,d}^{t-1}$. \square

Proof of Theorem 1

For any given function f , with some fixed unknown Lipschitz constant k , and for any chosen constants $\varepsilon_1 > 0$, $\tau_{n,d} > 1$, $C > 1$, $\delta \in [0, 1]$, and $m \geq 1$, as shown in Lemma 8, there exists a constant $L = \left\lceil \log_{\tau_{n,d}} \left(\frac{k}{\varepsilon_1} \right) \right\rceil$,

not depending on n , such that for $t \geq L$, we have $t \geq i^*$. Hence, by Definition 6, for $t \geq L$, ε_t reaches and exceeds k . Therefore, it is guaranteed that as t tends to infinity, ε_t surpasses k . Furthermore, by Proposition 2, we know that for all $\varepsilon_t > k$, $\mathcal{P}_{k,t} \subseteq \mathcal{A}_t(\varepsilon_t, t, m, \mathbf{P})$. Thus, as t tends to infinity, the search space uniformly recovers all the potential maximizers and beyond.

Lemma 9 (Corollary 13 in (Malherbe and Vayatis 2017)). Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact and convex set with non-empty interior and let $f \in \text{Lip}(k)$ be a k -Lipschitz functions defined on \mathcal{X} for some $k \geq 0$. Then, for any $n \in \mathbb{N}^*$ and $\xi \in (0, 1)$, we have with probability at least $1 - \xi$,

$$\max_{x \in \mathcal{X}} f(x) - \max_{i=1, \dots, n} f(x_i) \leq k \cdot \text{Diam}(\mathcal{X}) \cdot \left(\frac{\ln(1/\xi)}{n} \right)^{\frac{1}{d}}$$

where x_1, \dots, x_n denotes a sequence of n independent copies of $x \sim \mathcal{U}(\mathcal{X})$.

Proposition 3. (ECPv2 FASTER THAN PURE RANDOM SEARCH) Consider the ECPv2 algorithm tuned with any initial value $\varepsilon_1 > 0$, any constant $\tau_{n,d} > 1$, any constant $C > 1$, any integer $m \geq 1$, any $\delta \in [0, 1]$, and any $\beta > 1$. Then, for any $f \in \text{Lip}(k)$ and $n \geq i^*$, with probability $1 - 1/\beta^2$

$$\mathbb{P} \left(\max_{i=i^*, \dots, n} f(x_i) \geq y \right) \geq \mathbb{P} \left(\max_{i=i^*, \dots, n} f(x'_i) \geq y \right)$$

for all $y \geq \max_{i=1, \dots, i^*-1} f(x_i)$, where x_1, \dots, x_n are n evaluated points by ECPv2 and x'_1, \dots, x'_n are n independent uniformly distributed points over \mathcal{X} .

Proof. We proceed by induction. Let x_1, \dots, x_{i^*} be a sequence of evaluation points generated by ECPv2 after i^* iterations, and let x'_{i^*} be an independent point randomly sampled over \mathcal{X} . Consider any $y \geq \max_{i=1, \dots, i^*-1} f(x_i)$, and define the corresponding level set $\mathcal{X}_y = \{x \in \mathcal{X} : f(x) \geq y\}$. Assume, without loss of generality, that $\mu(\mathcal{X}_y) > 0$ (otherwise, $\mathbb{P}(f(x_{i^*}) \geq y) = 0$, and the result trivially holds).

For $t = i^*$, we have $\varepsilon_{i^*} \geq k$, therefore by Proposition 2 we have with probability at least $1 - 1/\beta^2$ $\mathcal{P}_{k,i^*} \subseteq \mathcal{A}_{ECPv2}(\varepsilon_{i^*}, i^*, m, \mathbf{P}) \subseteq \mathcal{X}$. Moreover, since $y \geq \max_{i=1, \dots, i^*-1} f(x_i)$, if \mathcal{X}_y is non-empty, its elements are potential maximizers. Hence, $\mathcal{X}_y \subseteq \mathcal{P}_{k,i^*}$. If \mathcal{X}_y is empty, the result holds trivially. Next, we compute the following probabilities:

$$\begin{aligned} \mathbb{P}(f(x_{i^*}) \geq y) &= \mathbb{E} [\mathbb{I}\{x_{i^*} \in \mathcal{X}_y\}] \\ &= \mathbb{E} \left[\frac{\mu(\mathcal{A}_{ECPv2}(\varepsilon_{i^*}, i^*, m, \mathbf{P}) \cap \mathcal{X}_y)}{\mu(\mathcal{A}_{ECPv2}(\varepsilon_{i^*}, i^*, m, \mathbf{P}))} \right] \\ &\geq \mathbb{E} \left[\frac{\mu(\mathcal{P}_{k,i^*} \cap \mathcal{X}_y)}{\mu(\mathcal{X})} \right] \\ &= \mathbb{E} \left[\frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X})} \right] \\ &= \mathbb{P}(f(x'_{i^*}) \geq y). \end{aligned}$$

Now, suppose the statement holds for some $n \geq i^*$. Let x_1, \dots, x_{n+1} be a sequence of evaluation points generated

by ECPv2 after $n + 1$ iterations, and let x'_1, \dots, x'_{n+1} be a sequence of $n + 1$ independent points sampled over \mathcal{X} .

As before, assume $\mu(\mathcal{X}_y) > 0$, and let $\mathcal{A}_{ECPv2}(\varepsilon_n, n, m, \mathbf{P})$ denote the sampling region of $x_{n+1} \mid x_1, \dots, x_n$. Then, on the event $\{\max_{i=i^*, \dots, n} f(x_i) < y\}$, we have, with probability $1 - 1/\beta^2$, $\mathcal{X}_y \subseteq \mathcal{P}_{k,n} \subseteq \mathcal{A}_{ECPv2}(\varepsilon_n, n, m, \mathbf{P}) \subseteq \mathcal{X}$. We note $\mathcal{T} = \mathbb{I}\{\max_{i=i^*, \dots, n} f(x_i) \geq y\}$ and for brevity we note $\mathcal{A}_{ECPv2,n} = \mathcal{A}_{ECPv2}(\varepsilon_n, n, m, \mathbf{P})$ and compute:

$$\begin{aligned} & \mathbb{P}\left(\max_{i=i^*, \dots, n+1} f(x_i) \geq y\right) \\ &= \mathbb{E}\left[\mathcal{T} + \mathbb{I}\left\{\max_{i=i^*, \dots, n} f(x_i) < y, x_{n+1} \in \mathcal{X}_y\right\}\right] \\ &= \mathbb{E}\left[\mathcal{T} + \mathbb{I}\left\{\max_{i=i^*, \dots, n} f(x_i) < y\right\} \frac{\mu(\mathcal{A}_{ECPv2,n} \cap \mathcal{X}_y)}{\mu(\mathcal{A}_{ECPv2,n})}\right] \\ &\geq \mathbb{E}\left[\mathcal{T} + \mathbb{I}\left\{\max_{i=i^*, \dots, n} f(x_i) < y\right\} \frac{\mu(\mathcal{P}_{k,n} \cap \mathcal{X}_y)}{\mu(\mathcal{A}_{ECPv2,n})}\right] \\ &\geq \mathbb{E}\left[\mathcal{T} + \mathbb{I}\left\{\max_{i=i^*, \dots, n} f(x_i) < y\right\} \frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X})}\right] \\ &\geq \mathbb{E}\left[\mathcal{T} + \mathbb{I}\left\{\max_{i=i^*, \dots, n} f(x'_i) < y\right\} \frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X})}\right] \\ &= \mathbb{E}\left[\mathcal{T} + \mathbb{I}\left\{\max_{i=i^*, \dots, n} f(x'_i) < y, x'_{n+1} \in \mathcal{X}_y\right\}\right] \\ &= \mathbb{P}\left(\max_{i=i^*, \dots, n+1} f(x'_i) \geq y\right) \end{aligned}$$

where the third inequality follows from the fact that $x \mapsto \mathbb{I}\{x \geq y\} + \mathbb{I}\{x < y\} \frac{\mu(\mathcal{X}_y)}{\mu(\mathcal{X})}$ is non-decreasing, and the induction hypothesis implies that $\max_{i=i^*, \dots, n} f(x_i)$ stochastically dominates $\max_{i=i^*, \dots, n} f(x'_i)$. Thus, by induction, the statement holds for all $n \geq i^*$, completing the proof. \square

Proof of Theorem 2

Fix any $\xi \in \left(0, 1 - \frac{1}{\beta^2}\right)$, and let i^* be defined as in Definition 6. Consider any iteration budget $n > i^*$.

By Proposition 3, for all $t \geq i^*$, the precision threshold $\varepsilon_t \geq k$ with probability at least $1 - \frac{1}{\beta^2}$, where k is the (unknown) Lipschitz constant of f . This implies that, starting from iteration i^* , ECPv2 behaves at least as efficiently as Pure Random Search (PRS) in improving over the best value observed so far.

Let $x^* \in \arg \max_{x \in \mathcal{X}} f(x)$ and define the simple regret after n evaluations as $\mathcal{R}_{ECPv2,f}(n) := f(x^*) - \max_{i=1, \dots, n} f(x_i)$. From Lemma 9, with total probability at

least $1 - \frac{1}{\beta^2} - \xi$, ECPv2 enjoys the same regret bound:

$$\begin{aligned} \mathcal{R}_{ECPv2,f}(n) &\leq k \cdot \text{diam}(\mathcal{X}) \cdot \left(\frac{\ln(1/\xi)}{n - i^* + 1}\right)^{1/d} \\ &= k \cdot \text{diam}(\mathcal{X}) \cdot \left(\frac{n}{n - i^* + 1}\right)^{\frac{1}{d}} \cdot \left(\frac{\ln(\frac{1}{\xi})}{n}\right)^{\frac{1}{d}} \\ &\leq k \cdot \text{diam}(\mathcal{X}) \cdot (i^*)^{\frac{1}{d}} \left(\frac{\ln(1/\xi)}{n}\right)^{\frac{1}{d}} \\ &\leq k \cdot \text{diam}(\mathcal{X}) \cdot \log_{\tau_{n,d}}\left(\frac{k}{\varepsilon_1}\right)^{\frac{1}{d}} \left(\frac{\ln(1/\xi)}{n}\right)^{\frac{1}{d}} \end{aligned}$$

where the last two inequalities follow from Lemma 8, which ensures $\frac{n}{n - i^* + 1} \leq i^*$ for all $n > i^*$.

For the case $n \leq i^*$, the bound is trivially satisfied since:

$$\mathcal{R}_{ECPv2,f}(n) \leq k \cdot \text{diam}(\mathcal{X}).$$

\square

Appendix C: Missing Proofs

Proof of Lemma 1

Proof. Suppose there exists $x \in \mathcal{X}$ such that:

$$\min_{i=1, \dots, t} (f(X_i) + \varepsilon_t \cdot \|x - X_i\|_2) \geq f_{\max_t}.$$

Then for all $i \in \{1, \dots, t\}$, we must have:

$$f(X_i) + \varepsilon_t \cdot \|x - X_i\|_2 \geq f_{\max_t},$$

which implies:

$$\varepsilon_t \cdot \|x - X_i\|_2 \geq f_{\max_t} - f(X_i), \quad \forall i \in [t].$$

Hence:

$$\varepsilon_t \geq \frac{f_{\max_t} - f(X_i)}{\|x - X_i\|_2}, \quad \forall i \in [t].$$

Taking the maximum over all $i \in [t]$:

$$\varepsilon_t \geq \max_{i=1, \dots, t} \left(\frac{f_{\max_t} - f(X_i)}{\|x - X_i\|_2} \right).$$

Finally, since $\|x - X_i\|_2 \leq \text{diam}(\mathcal{X})$ for all $x \in \mathcal{X}$, we get:

$$\varepsilon_t \geq \max_{i=1, \dots, t} \left(\frac{f_{\max_t} - f(X_i)}{\text{diam}(\mathcal{X})} \right) = \frac{f_{\max_t} - f_{\min_t}}{\text{diam}(\mathcal{X})}.$$

\square

Proof of Lemma 2

Proof. Note that for a given value of ε_t , ECPv2 instead uses $\max\{\varepsilon_t, \varepsilon^\circ\} \geq \varepsilon_t$. Thus the result follows directly from Lemma 1 in (Fourati et al. 2025). \square

Proof of Lemma 3

Proof. When $t \leq m$, we have $\mathcal{I}_t^m = \{1, \dots, t\}$, so the Worst- m condition reduces exactly to the condition in Equation (4), implying $\mathcal{A}_{ECP}(\varepsilon_t, t) \subseteq \mathcal{A}_t(\varepsilon_t, t, m)$, as shown by Lemma 2.

When $t > m$, the set \mathcal{I}_t^m is a strict subset of $\{1, \dots, t\}$ with cardinality m . Thus, any point x that satisfies Equation (4) automatically satisfies Equation (5), establishing $\mathcal{A}_t(\varepsilon_t, t) \subseteq \mathcal{A}_t(\varepsilon_t, t, m)$. \square

Appendix D: Analysis of Projection Dimension Sensitivity (impact of δ and β)

The Lemma 4 states that any set of n points in high-dimensional space can be embedded into $\mathbb{R}^{d'}$ with pairwise distances preserved within a multiplicative factor of $1 \pm \delta$, provided:

$$d' \geq \frac{8 \log(\beta n)}{\delta^2 - \delta^3}$$

where $\beta > 1$ controls the failure probability. The term $\delta^2 - \delta^3$ governs the required projection dimension d' in terms of the distortion tolerance δ . To understand its behavior, we define the distortion scaling function:

$$f(\delta) = \frac{1}{\delta^2 - \delta^3} = \frac{1}{\delta^2(1 - \delta)} \quad \text{for } \delta \in (0, 1)$$

As $\delta \rightarrow 0$, the required projection dimension diverges: $f(\delta) \sim 1/\delta^2$. Similarly, as $\delta \rightarrow 1$, the denominator vanishes and $f(\delta) \rightarrow \infty$. Thus, there exists an intermediate value of δ that minimizes the bound.

On the choice of δ

We analytically compute the minimum of $f(\delta)$ by solving:

$$f(\delta) = \frac{1}{\delta^2(1 - \delta)} \Rightarrow f'(\delta) = \frac{-2(1 - \delta) + \delta}{\delta^3(1 - \delta)^2}$$

Setting $f'(\delta) = 0$ yields:

$$-2(1 - \delta) + \delta = 0 \Rightarrow 3\delta = 2 \Rightarrow \delta^* = \frac{2}{3}$$

At this optimal distortion level:

$$f\left(\frac{2}{3}\right) = \frac{1}{\left(\frac{2}{3}\right)^2 - \left(\frac{2}{3}\right)^3} = \frac{1}{\frac{4}{9} - \frac{8}{27}} = \frac{1}{\frac{4}{27}} = \frac{27}{4} = 6.75$$

Hence, the projection dimension is minimized when $\delta = \frac{2}{3}$, and the bound simplifies to:

$$d' \geq \frac{8 \log(\beta n)}{6.75}$$

Visualization Figure 3 shows the scaling term $f(\delta)$ over the interval $(0.01, 0.99)$. The y-axis is plotted logarithmically to highlight the divergence at the boundaries. The unique minimum at $\delta = \frac{2}{3}$ is marked on the curve.

Projection Dimension Scaling

Figure 4 illustrates how the required projection dimension d' scales with the number of evaluations n for various values of the allowed distortion δ :

$$d' = \frac{8 \log(\beta n)}{\delta^2 - \delta^3},$$

where β is a constant (set to 5 in our case to ensure the guarantee holds with probability above 96%). Smaller values of δ lead to larger required projection dimensions due to the tighter approximation constraints. The plot uses a log-log scale to capture behavior over several orders of magnitude in n , ranging from 10^1 to 10^4 .

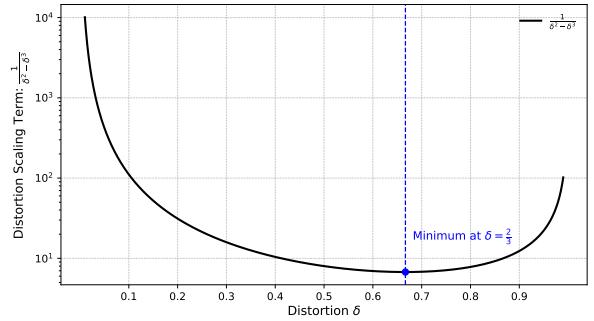


Figure 3: JL scaling term $f(\delta) = \frac{1}{\delta^2 - \delta^3}$ as a function of distortion δ . The function attains a minimum at $\delta = \frac{2}{3}$, yielding the most distortion-efficient embedding.

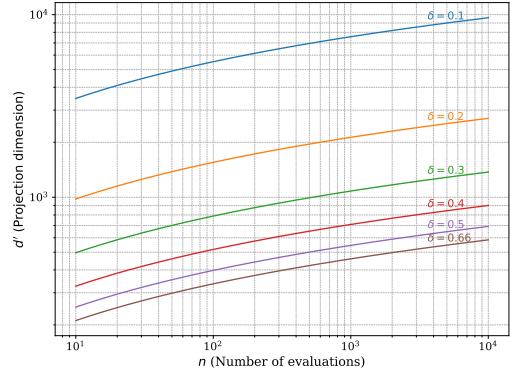


Figure 4: Projection dimension d' as a function of the number of evaluations n for various values of $\delta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, with $\beta = 5$ fixed.

On the choice of β

The Lemma provides the following guarantee on the success probability of the embedding:

$$\mathbb{P}[\text{pairwise distances are preserved within } (1 \pm \delta)] \geq 1 - \frac{1}{\beta^2}$$

To achieve a success probability of at least $p = 0.95$, we solve:

$$1 - \frac{1}{\beta^2} \geq 0.95 \implies \frac{1}{\beta^2} \leq 0.05 \implies \beta^2 \geq 20$$

This gives the threshold:

$$\beta \geq \sqrt{20} \approx 4.47$$

In practice, we round up for simplicity and robustness. Choosing $\beta = 5$ yields:

$$\Pr[\text{Success}] \geq 1 - \frac{1}{25} = 0.96$$

Thus, we adopt $\beta = 5$ in our empirical evaluations, which ensures that the embedding succeeds with at least 96% probability.

Effect of β on Distortion Concentration

To empirically validate the bound in Lemma 4, we investigate how the choice of the confidence parameter β affects the preservation of pairwise distances after random projection. Recall that a larger β results in a larger projection dimension d' , thereby increasing the probability that all pairwise distortions fall within the acceptable range $[1-\delta, 1+\delta]$.

We generate $n = 100$ data points in \mathbb{R}^{1000} from a uniform distribution over $[0, 1]^d$, and project them to $\mathbb{R}^{d'}$, where d' is determined according to the JL-type bound:

$$d' \geq \frac{8 \log(\beta n)}{\delta^2 - \delta^3}, \quad \text{with } \delta = 0.5.$$

For each value of $\beta \in \{0.05, 0.5, 5\}$, we repeat the projection process over 1000 independent trials and collect all pairwise distortion ratios.

Figure 5 presents the resulting histograms. As expected, the distortion distributions become more concentrated around 1 as β increases. A horizontal line segment indicates the theoretical distortion bound $[1 - \delta, 1 + \delta]$, which serves as a visual reference for acceptable approximation quality. These results empirically support the theoretical guarantee: larger β leads to more reliable distance preservation due to increased projection dimension.

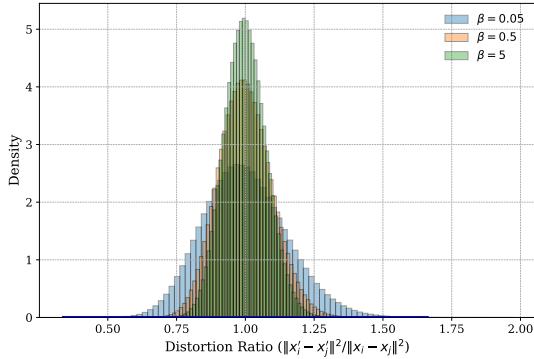


Figure 5: Histogram of pairwise distance distortions aggregated over 1000 random trials. Each histogram corresponds to a different value of β , with $n = 100$, $d = 1000$, and distortion bound $\delta = 0.5$. A thick blue horizontal line marks the acceptable distortion interval $[1 - \delta, 1 + \delta]$. As β increases (and with it the projection dimension d'), the distortion distribution becomes more concentrated around 1, consistent with Lemma 4.

Empirical Success Rates vs. Projection Dimension

To further assess the validity of the theoretical projection bound from Lemma 4, we empirically estimate the success rate of random projections across varying target dimensions d' .

We fix $n = 100$, $d = 1000$, $\delta = \frac{2}{3}$, and vary d' around the theoretical minimum:

$$d'_{\text{bound}} = \left\lceil \frac{8 \log(\beta n)}{\delta^2 - \delta^3} \right\rceil, \quad \text{with } \beta = 5.$$

For each d' , we generate 1000 random projections and compute the proportion of trials in which all pairwise distances are preserved within the interval $[1 - \delta, 1 + \delta]$.

Figure 6 reports the empirical success rate as a function of d' . The horizontal dashed line marks the theoretical success threshold $1 - 1/\beta^2 = 0.96$, while the vertical dashed line marks the theoretical projection bound. The plot confirms that success rates converge to the theoretical threshold around the predicted dimension d'_{bound} , supporting the tightness of the bound.

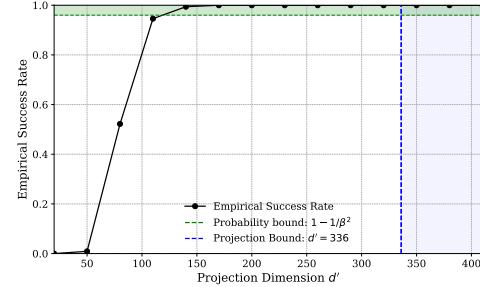


Figure 6: Empirical success rate of random projections preserving all pairwise distances within $(1 \pm \delta)$, with $\delta = \frac{2}{3}$, $n = 100$, and $\beta = 5$. Each point averages 1000 trials. The theoretical projection bound and success threshold are marked with dashed lines.

Appendix E: Analysis of the Worst- m Approximation Strategy

Effect of m on the Minimization

To alleviate the increasing conservatism in the acceptance condition as the number of evaluations grows, ECPv2 adopts a subsampling approach: it considers only the m worst (i.e., lowest) function values from the evaluation history. This subset is defined as:

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

The hyperparameter $m \geq 1$ controls the aggressiveness of the approximation. Notably, for $m \geq n$, we recover the standard ECP acceptance condition. For smaller m , the approximation becomes less conservative and computationally cheaper. The modified acceptance condition becomes:

$$\min_{i \in \mathcal{I}_t^m} (f(x_i) + \max\{\varepsilon_t, \varepsilon_t^\phi\} \cdot \|x - X_i\|_2) \geq \max_j f(x_j).$$

We denote the left-hand side of this inequality as $\phi_m(x)$ and compare it to its full-data counterpart, $\phi_{\text{full}}(x)$:

$$\phi_{\text{full}}(x) = \min_i [f(x_i) + \epsilon_t \cdot \|x - x_i\|_2],$$

$$\phi_m(x) = \min_{i \in \mathcal{I}_m} [f(x_i) + \epsilon_t \cdot \|x - x_i\|_2],$$

where \mathcal{I}_m is the index set of the m lowest function values, and we fix $\epsilon_t = 1$.

To quantify the quality of the Worst- m approximation, we evaluate the ratio $\phi_{\text{full}}(x)/\phi_m(x)$ as a function of the sampling fraction m/n . Specifically, we sample $n = 100$ points uniformly within each function’s domain, select a random test point x_{test} , and compute the ratio for varying values of $m \in [1, 100]$. Each experiment is repeated over 1000 independent trials per function.

We fit the observed mean ratios to a logarithmic model of the form:

$$\frac{\phi_{\text{full}}(x)}{\phi_m(x)} \approx a \cdot \log(m/n) + b,$$

where (a, b) are fitted parameters for each benchmark function. This functional form reflects the intuition that adding more points provides diminishing returns in surrogate quality. We perform this evaluation across four standard non-convex test functions.

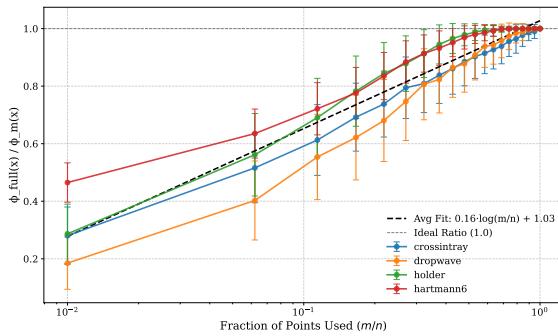


Figure 7: Ratio $\phi_{\text{full}}(x)/\phi_m(x)$ as a function of m/n , the fraction of retained evaluations, for four benchmark functions. Each curve represents the average over 1000 trials using $n = 100$ points. Error bars indicate half the standard deviation across trials. The black dashed line shows the average fit $\bar{a} \cdot \log(m/n) + \bar{b}$, with $(\bar{a}, \bar{b}) \approx (0.16, 1.03)$. The horizontal dashed line marks the ideal ratio of 1.0.

This empirical study reveals a consistent logarithmic relationship between $\phi_{\text{full}}(x)/\phi_m(x)$ and m/n across all benchmark functions. The key insight is that even a small, well-chosen subset of evaluations ($m \ll n$) retains most of the quality of the full acceptance condition. This observation supports the use of the Worst- m strategy to balance surrogate fidelity and computational efficiency, with practical implications for large-scale and real-time optimization scenarios.

Effect of m in ECPv2

To isolate the impact of m in the ECPv2 optimizer, we conducted an ablation study with fixed parameters $\delta = \frac{2}{3}$ and $\beta = 5$.

We benchmarked ECPv2 across 12 diverse test functions, varying in dimension and landscape. For each function, we ran 100 independent optimization trials, each for $n = 1000$ evaluations. The tested values for m were:

$$m \in \{2, 4, 8, 16, 32, 64, 128, 256, 512, 1000\}.$$

We recorded the best-so-far function value over time and averaged across runs. Each figure reports the final best-so-far mean (with \pm half standard deviation error bars) as a function of total CPU time.

Remark 1. *ECPv2 only performs projection when the function’s intrinsic dimensionality d exceeds d' : $d > d' = \left\lceil \frac{8 \log(5n)}{\delta^2 - \delta^3} \right\rceil$, where n is the evaluation budget. When $d \leq d'$, projection is skipped regardless of the m setting.*

Figures 10, 8, 9 illustrate the impact of varying m on optimization performance across all functions. As expected, increasing m leads to longer computational times due to the higher cost of minimization (especially for very large values of m). At the same time, larger m values tend to achieve better optima on average (not always), though with diminishing returns beyond a certain point.

In almost all experiments across diverse functions and dimensionalities, the three choices of $m \in \{256, 512, 1000\}$ are the slowest, with larger values of m generally leading to slower performance. Notably, setting $m = 1000$ effectively disables the worst- m projection since $n = 1000$, thereby recovering the ECP minimization. This observation highlights the impact of using larger projection dimensions on wall-clock time.

These results support the findings in Appendix E, where the Worst- m strategy closely approximated the exact solution even for modest values of m . We conclude that very large values of m are not necessary in practice: smaller settings can achieve comparable or even superior results in significantly less time (up to 4×–5× faster in CPU seconds).

Interestingly, in some cases, smaller values of m outperform larger ones or even the exact method. For example, in the Eggholder problem, $m = 128$ achieves the best score while being 2.5× faster than exact minimization. In the Rosenbrock function (500 dimensions), $m = 16$ yields the best performance while being 1.5× faster. We attribute this to the relaxed acceptance condition used when m is small: it is more permissive, allowing better candidate points to be accepted when the acceptance region, controlled by ε_t , is still small. In contrast, with larger m , the acceptance region tightens—especially when coupled with lower values of ε_t —potentially rejecting high-quality candidates prematurely.

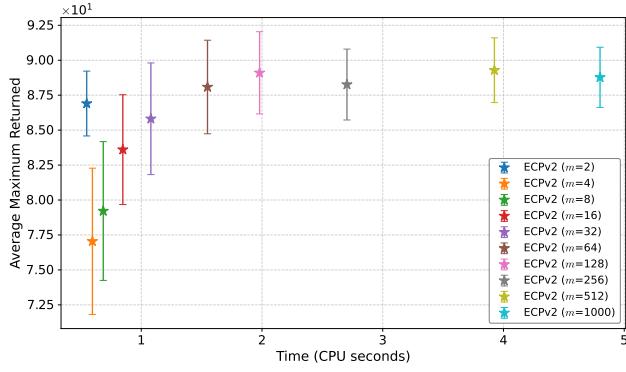
These insights further motivate the use of small projection dimensions $m \ll n$, not only for computational efficiency but also for effective optimization performance.

Appendix F: Empirical Validation of the Adaptive Lower Bound ε_t^\otimes

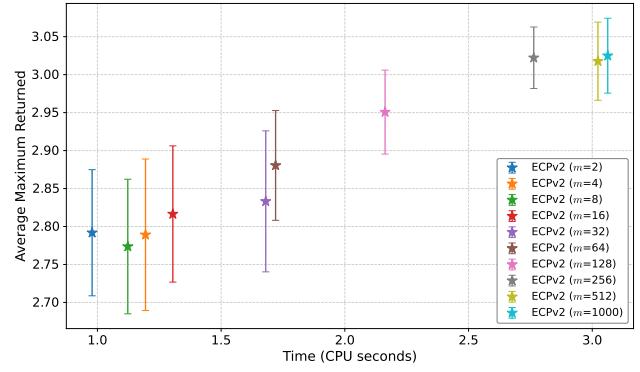
Validation of Necessity

To empirically validate the necessity and effectiveness of the adaptive lower bound ε_t^\otimes introduced in ECPv2, we conduct a focused experiment using the Ackley function.

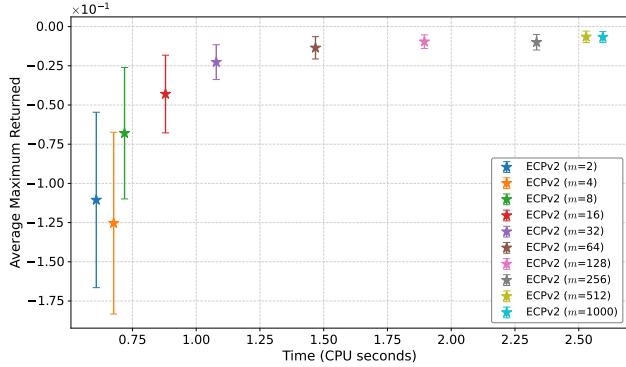
We fix the algorithm at iteration $t = 5$ and simulate a scenario where five previously accepted points $\{x_1, \dots, x_5\}$ are sampled uniformly from the domain \mathcal{X} . We evaluate these points to obtain their function values $\{f(x_1), \dots, f(x_5)\}$ and denote the current maximum as f_{\max} .



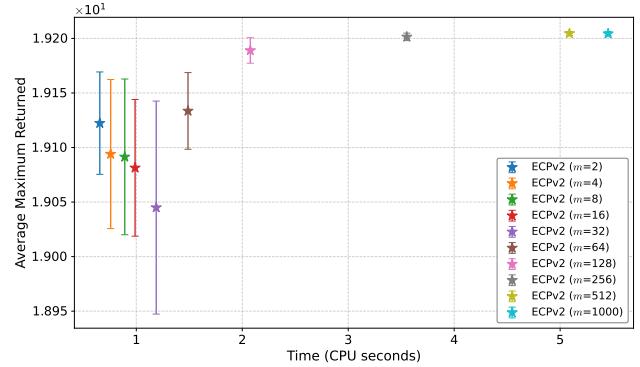
(a) Eggholder



(b) Hartmann (6D)



(c) Himmelblau



(d) Holder

Figure 8: Ablation study on the projection dimension m in ECPv2 (Part 1 of 3). Each point represents the final average of the best score after $n = 1000$ evaluations, averaged over 100 runs, with \pm half the standard deviation. Projection is skipped as the function dimension $d < d' = \left\lceil \frac{8 \log(5n)}{\delta^2 - \delta^3} \right\rceil$.

For a grid of ε values in the range $[0.01, 1.5]$, we draw 1000 independent candidate points $\tilde{x} \sim \mathcal{U}(\mathcal{X})$ and compute the empirical *acceptance ratio*, i.e., the fraction of points satisfying the ECP acceptance condition:

$$\min_{i=1, \dots, t} (f(x_i) + \varepsilon \cdot \|\tilde{x} - x_i\|_2) \geq f_{\max}.$$

As shown in Lemma 1, a necessary condition for any point to be accepted is that $\varepsilon \geq \varepsilon_t^\circlearrowleft$, where:

$$\varepsilon_t^\circlearrowleft = \frac{f_{\max_t} - f_{\min_t}}{\text{diam}(\mathcal{X})}.$$

This value ensures that the acceptance region is not trivially empty.

Figure 12 illustrates the empirical acceptance ratio as a function of ε . We observe a sharp phase transition: for $\varepsilon < \varepsilon_t^\circlearrowleft$, the acceptance ratio is virtually zero, indicating that all candidate points are rejected. Once ε surpasses the threshold, the acceptance ratio increases rapidly, confirming that $\varepsilon_t^\circlearrowleft$ effectively marks the boundary of a non-trivial search region.

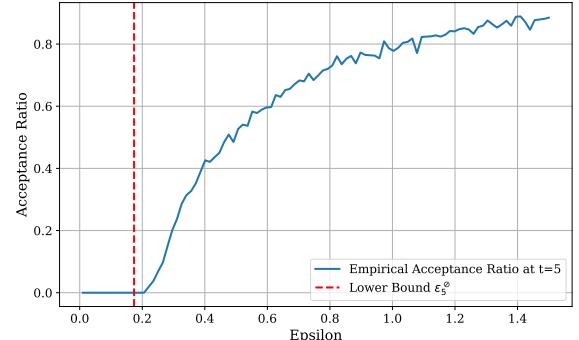


Figure 12: Empirical acceptance ratio at iteration $t = 5$ on the 10D Ackley function. The red dashed line denotes the theoretical lower bound $\varepsilon_t^\circlearrowleft$. This confirms the bound's role in preventing empty acceptance regions.

This experiment offers empirical support for the theoretical lower bound $\varepsilon_t^\circlearrowleft$, demonstrating its critical role in maintaining optimization progress. Without it, the algorithm may experience excessive early rejections, stalling exploration. ECPv2's adaptive enforcement of this threshold ensures that

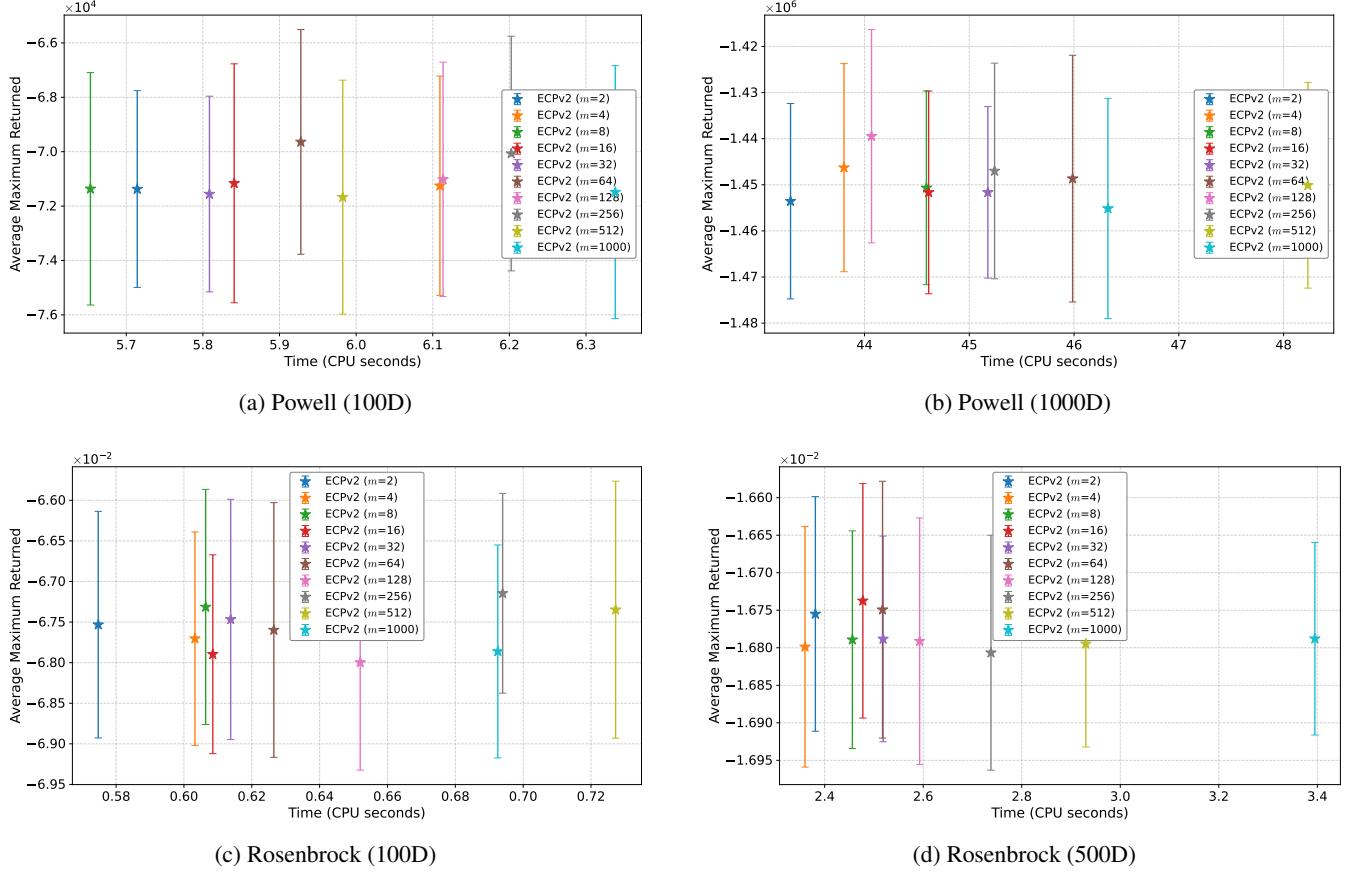


Figure 9: Ablation study on the projection dimension m in ECPv2 (Part 2 of 3). Each point represents the final average of the best score after $n = 1000$ evaluations, averaged over 100 runs, with $\pm \frac{1}{2}$ standard deviation.

each iteration maintains a viable and meaningful acceptance region.

Effect of the Lower Bound on ECPv2

To better understand the contribution of the lower bound enhancement in ECPv2, we performed an ablation study comparing two variants: ECP and ECPv2 (LB only): a variant where only the lower bound enhancement is enabled (`Worst-m = False`, `Random Projections = False`, `Lower Bound = True`).

Both algorithms were benchmarked on four benchmark functions over 100 independent runs, each with a budget of 100 evaluations. Figure 11 reports the final average performance and runtime.

We observe that enabling the lower bound mechanism alone yields a significant runtime improvement, approximately a $2\times$ speedup and even outperforming the optimization performance. This demonstrates that even without projection-based exploration or Worst- m sampling, incorporating the lower bound check provides a strong computational advantage, especially with lower-budgets (such as $n = 100$).

Appendix G: On the Combination of Random Projection and Worst- m

To understand the impact of combining the proposed Random Projection and Worst- m components in **ECPv2**, we perform an ablation study by enabling their different combinations.

Each configuration is evaluated on the rosenbrock function (with 500 dimensions), with 100 independent runs and a budget of $n = 5000$ function evaluations. The figure below shows the final performance (average maximum value returned) plotted against CPU time (seconds), with error bars denoting $\pm \frac{1}{2}$ standard deviation.

Findings.

- All ECPv2 variants are significantly faster than the ECP, while maintaining comparable or better solution quality.
- **ECPv2 (full)** (all true) is the *fastest* across runs, showing that combining all components leads to the fastest performance.
- The *Worst- m selection* is especially effective in high-dimensional problems like Rosenbrock (300D and 400D) and substantially reducing computational cost.

These results confirm that ECPv2 is a robust, scalable, and computationally efficient extension of ECP, with strong empirical performance.

Appendix H: Additional Results on ECPv2 vs. ECP

Statistical Significance

To assess whether the performance improvements in ECPv2 over ECP are statistically significant, we conducted a Wilcoxon signed-rank test on the per-objective rankings of average runtimes across 13 benchmark functions (michalewicz, perm, perm10, powell100, powell1000, rastrigin, rosenbrock, rosenbrock100, rosenbrock200, rosenbrock300, rosenbrock500, schaffer, and schubert). This non-parametric test is suitable for comparing paired samples when the assumption of normality may not hold.

The test yielded a statistic of $W = 7.0$ with a p-value of 0.0034, indicating a statistically significant difference ($p < 0.01$) in favor of ECPv2. The improved method consistently achieved faster average runtimes across most objectives. These results suggest that the enhancements introduced in ECPv2 contribute to a meaningful reduction in computational cost.

High-dimensional Problems

We evaluated ECPv2 against the ECP on two challenging high-dimensional benchmark problems: Rosenbrock (500 dimensions) and Powell (1000 dimensions). In both cases, ECPv2 not only demonstrated approximately 2× faster convergence compared to ECP, but also achieved higher optimization scores. These results highlight ECPv2’s scalability and efficiency in very high-dimensional settings. Detailed results are shown in Figure 14.

Appendix I: Additional Comparison with Other Global Optimization Algorithms

While the primary focus of this work is on algorithms tailored for Lipschitz optimization, along with AdaLIPO, AdaLIPO+, PRS, and DIRECT, we also evaluate several widely used global optimization baselines including SMAC3 and Dual Annealing under the same experimental setup as described in the main paper.

Among the evaluated methods, SMAC3 demonstrates strong optimization capability but suffers from significant overhead due to its model-based nature, making it much slower than other approaches. DIRECT and PRS by contrast, are extremely fast but tends to perform poorly. DIRECT suffers with high-dimensional settings.

Dual Annealing, ECP, and ECPv2 all strike a good balance between computational efficiency and optimization performance. Notably, ECPv2 achieves the best overall trade-off, consistently identifying high-quality solutions with minimal wall-clock time. ECPv2 is remarkably much faster than ECP in the rastrigin case, while providing a much competitive wall-clock time.

Both AdaLIPO and AdaLIPO+ also perform competitively, especially on the Rosenbrock family, with a slightly lower performance on rastrigin. Full quantitative results for these comparisons are provided in Figure 15.

Appendix J: Experimental Details

Default Hyper-parameters

We provide recommended default values for the main hyper-parameters of ECPv2, based on both theoretical analysis and empirical validation. These choices are robust across a wide range of optimization problems and were used in all our experiments unless otherwise stated.

To ensure a fair comparison, we avoid fine-tuning the hyper-parameters that are shared with the ECP algorithm. Instead, we retain the same values as those studied in ECP (Fourati et al. 2025), thereby preventing any performance advantage that could arise from tuning them.

- **Initial precision** ε_1 : Set to 0.01 (same as ECP). This defines the initial threshold for the acceptance region. A smaller value yields a more conservative start.
- **Growth factor** $\tau_{n,d}$: Set as $\tau_{n,d} = \max(1 + \frac{1}{nd}, 1.001)$ (same as ECP). This controls how quickly the acceptance threshold expands after a series of rejections.
- **Exploration patience** C : Set to 1000 (same as ECP). This parameter determines the number of consecutive rejections allowed before increasing ε_t .
- **Worst- m memory** m : Set to $m = 8$.
- **Distortion parameter** δ : Set to $\delta = \frac{2}{3}$. This value minimizes the required projection dimension d' for a given distortion tolerance, as discussed in Appendix D.
- **Projection confidence parameter** β : Set to $\beta = 5$, ensuring that pairwise distances are preserved under random projection with at least 96% probability.

These settings were chosen to provide strong performance without requiring problem-specific tuning. Sensitivity analyses for δ and β are presented in Appendix D, and analysis for m are presented in Appendix E and Appendix F.

Optimization Algorithms

For ECP, we use the implementation provided by (Fourati et al. 2025) with the default parameters (<https://github.com/fouratifares/ECP>).

For DIRECT and Dual Annealing, we use the implementations from SciPy (Virtanen et al. 2020), with standard hyperparameters and necessary modifications to adhere to the specified budgets.

For AdaLIPO and AdaLIPO+, we use the implementation provided by (Serré et al. 2024). To ensure fairness, we run AdaLIPO+ without stopping, thereby maintaining the same budget across all methods. Furthermore, since AdaLIPO requires an exploration probability p , we fix it at 0.1, as done by the authors (Malherbe and Vayatis 2017).

Optimization Objectives

We evaluate the proposed method on various global optimization problems. The problems were designed to challenge global optimization methods due to their highly non-convex curvatures (Molga and Smutnicki 2005; Surjanovic and Bingham 2013; Fourati et al. 2025).

Comparison Protocol

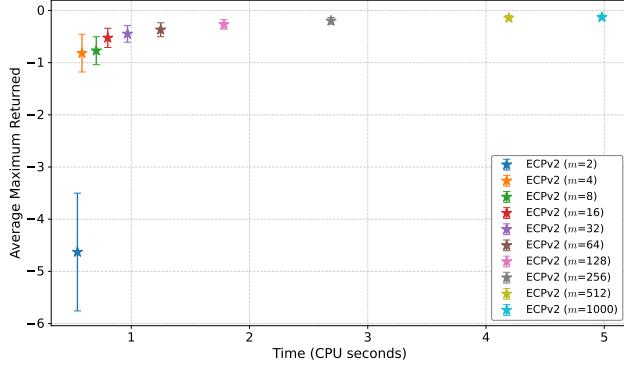
We use the same hyperparameters across all optimization tasks without fine-tuning them for each task, as this may not be practical when dealing with expensive functions.

We allocate a fixed budget of function evaluations, denoted by n , for all methods. The maximum value over the n iterations is recorded for each algorithm. This maximum is then averaged over 100 repetitions, and both the mean and standard deviation are tracked. Furthermore, the wall-clock time (seconds) spent to finish the budget n is further tracked and averaged over the runs. We then report the average time spent to achieve the average maximum.

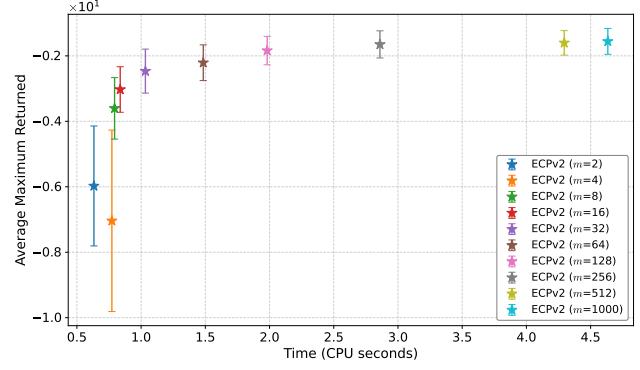
More evaluations increase the likelihood of finding better points. To ensure that all methods fully utilize the budget, we eliminate any unnecessary stopping conditions, such as waiting times. For example, in AdaLIPO+, we use the variant AdaLIPO+(ns), which continues running even when large rejections occur.

Compute and Implementations

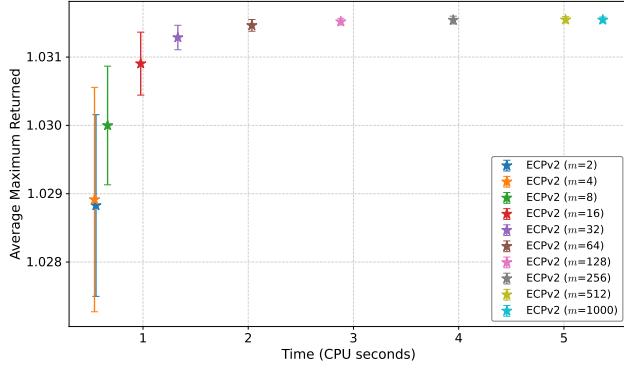
We implemented our method using open-source libraries in Python 3.9 and NumPy 1.23.4. Experiments were conducted on two systems: a local machine with an 11th Gen Intel® Core™ i7-1165G7 CPU @ 2.80 GHz and 16 GB RAM, and a high-performance computing (HPC) system with an Intel® Xeon® Gold 6148 CPU @ 2.40 GHz and 16 GB RAM. All experiments were run on CPUs without the use of GPUs. In each plot, all methods are tested under exactly the same settings to ensure fairness.



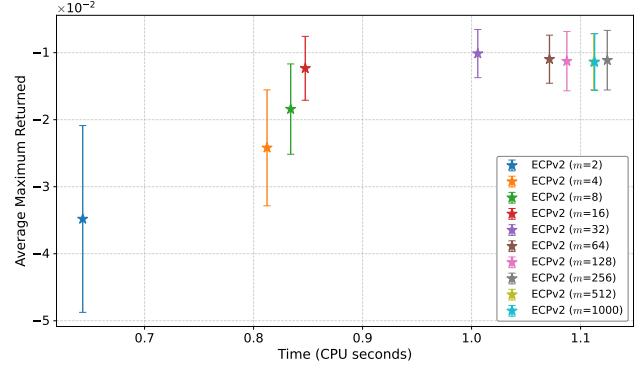
(a) Ackley



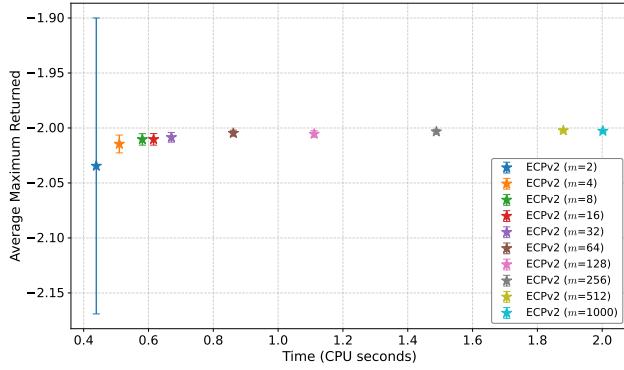
(b) Bukin



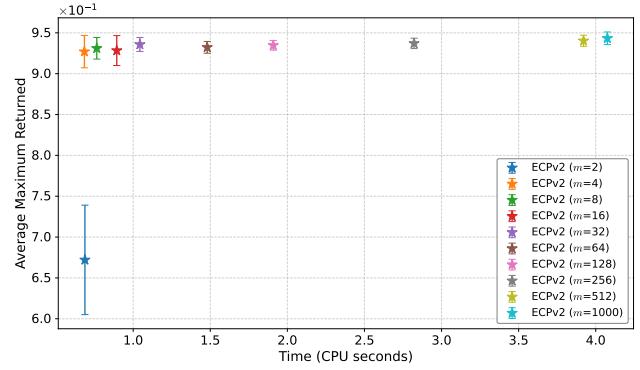
(c) Camel



(d) Colville

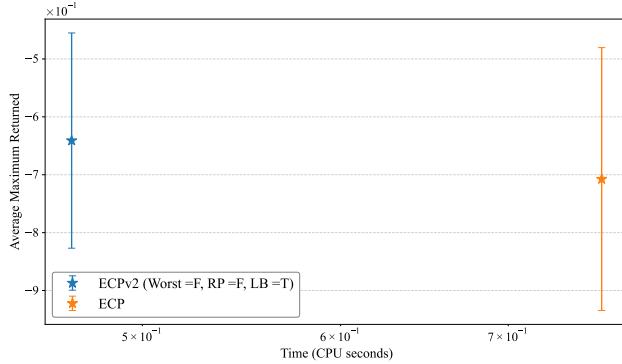


(e) Damavandi

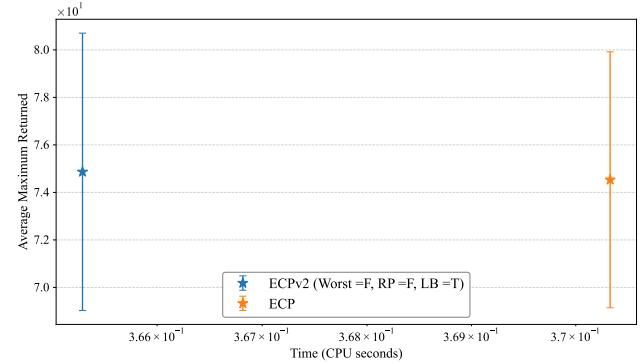


(f) Drop-Wave

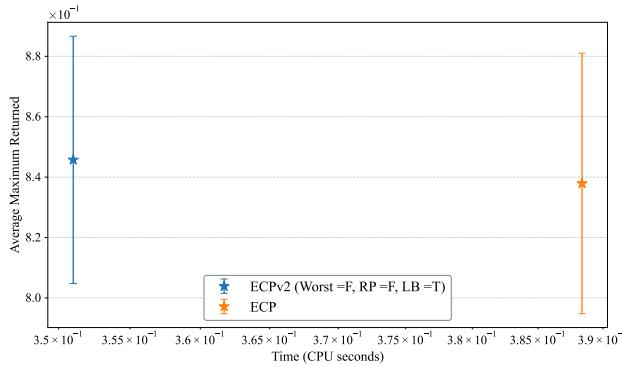
Figure 10: Ablation study on the projection dimension m in ECPv2 (Part 3 of 3). Each point represents the final average of the best score after $n = 1000$ evaluations, averaged over 100 runs, with \pm half the standard deviation. Projection is skipped as the function dimension $d < d' = \left\lceil \frac{8 \log(5n)}{\delta^2 - \delta^3} \right\rceil$.



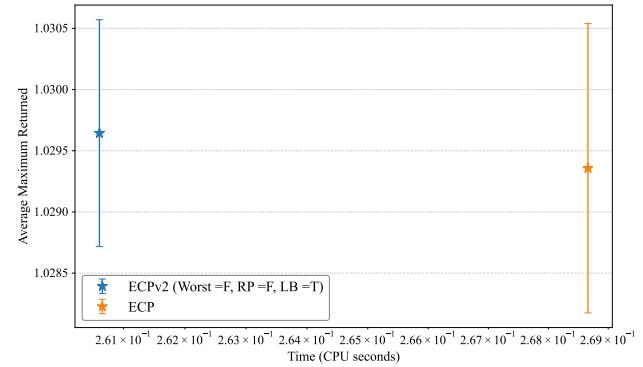
(a) Ackley



(b) Eggholder

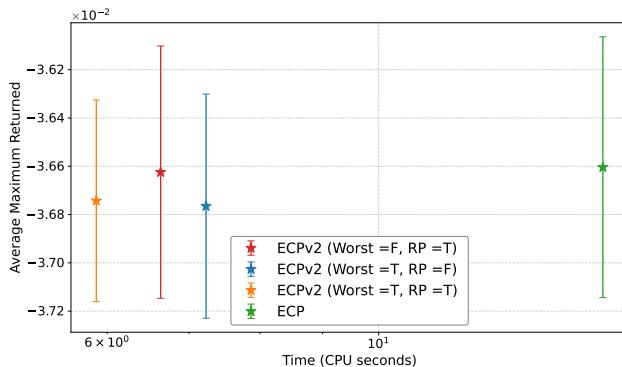


(c) Dropwave

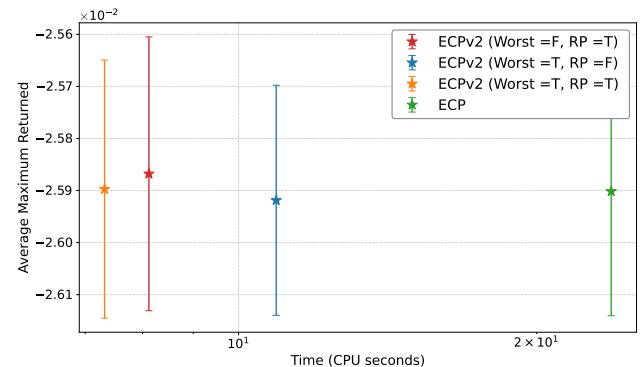


(d) Three-Hump Camel

Figure 11: Ablation comparison between ECP and the ECPv2 (LB only) variant on four benchmark functions. Each point shows the final average of the best score after 100 evaluations, averaged over 100 runs, with $\pm \frac{1}{2}$ standard deviation. The lower-bound enhancement alone provides up to 2 \times speedups with consistent solution quality.



(a) Rosenbrock (200D)



(b) Rosenbrock (300D)

Figure 13: Final performance vs. CPU time for ECPv2 ablations on Rosenbrock in higher dimensions. Error bars represent $\pm \frac{1}{2}$ standard deviation across 100 runs.

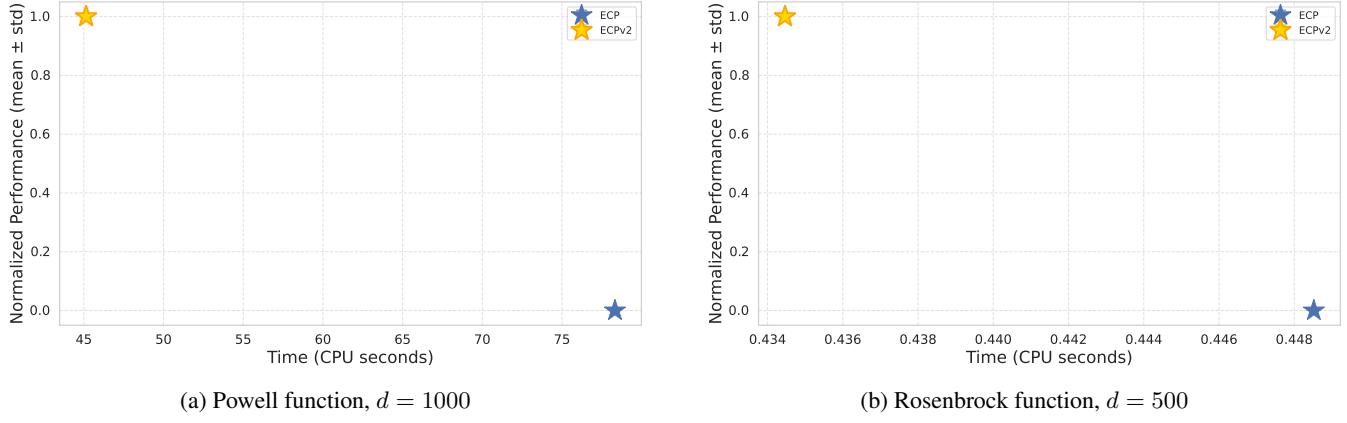


Figure 14: Performance comparison of ECP and ECPv2 on high-dimensional optimization problems with a budget $n = 200$, averaged over 100 repetitions.

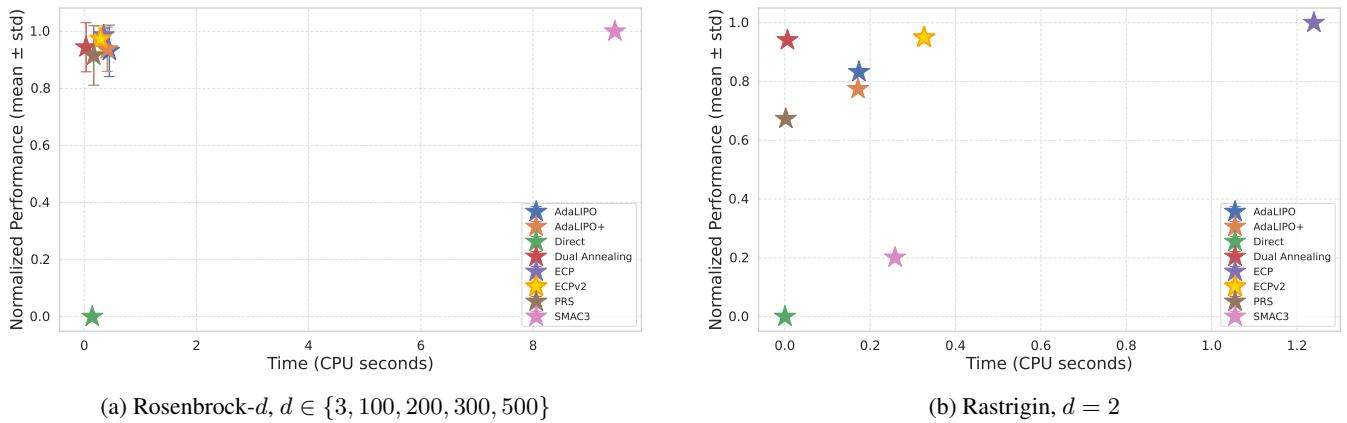


Figure 15: Comparison of additional global optimization baselines on two benchmark functions: (a) Rosenbrock across increasing dimensionalities, where dimensionality reduction improves performance; (b) Rastrigin in 2D, where dimensionality reduction is skipped since $d = 2 < d' = \left\lceil \frac{8 \log(5n)}{\delta^2 - \delta^3} \right\rceil$. In this case, acceleration primarily comes from the use of Worst- m and the lower-bound term ε_t^\otimes . Results are aggregated over 100 runs per optimizer per problem. Each point represents the average final best objective value after a modest budget of $n = 200$ evaluations.