scientific reports

Check for updates

OPEN

Population-based variance-reduced evolution over stochastic landscapes

Zelin Pei[1], Xiaoyu He[1]✉, Yi Pan[1], Baichun Peng[1], Wen Chen[2] & Yuren Zhou[1]

Black-box stochastic optimization involves sampling in both the solution and data spaces. Traditional variance reduction methods mainly designed for reducing the data sampling noise may suffer from slow convergence if the noise in the solution space is poorly handled. In this paper, we present a novel zeroth-order optimization method, termed Population-based Variance-Reduced Evolution (PVRE), which simultaneously mitigates noise in both the solution and data spaces. PVRE uses a normalized-momentum mechanism to guide the search and reduce the noise due to data sampling. A population-based gradient estimation scheme, a well-established evolutionary optimization technique, is incorporated to further reduce noise in the solution space. We show that PVRE exhibits the convergence properties of theory-backed optimization algorithms and the adaptability of evolutionary algorithms. In particular, PVRE achieves the best-known function evaluation complexity of $\mathcal{O}(n\epsilon^{-3})$ for finding an ϵ-accurate first-order optimal solution, up to a logarithmic factor, with any initial step-size. We assess the performance of PVRE through numerical experiments on benchmark problems as well as a real-world task involving adversarial attacks against neural image classifiers.

We consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) = \mathbb{E}_{\xi\sim\mathcal{D}}[F(x;\xi)], \tag{1}$$

where $x \in \mathbb{R}^n$ is the solution to be found, $\mathcal{D}$ is the data distribution, and ξ denotes a random data sample. As in most real-world scenarios, we assume that computing the expectation is intractable due to the large number of data samples or the lack of prior knowledge of the distribution.

Problems in the form of (1) are usually solved by stochastic methods[1] in which the objective function $f(x)$ is replaced by some ease-to-compute estimates $F(x;\xi)$. The stochastic gradient descent (SGD)[2] and its modern variants[3] are representative of these methods. They typically draw a random sample ξ from $\mathcal{D}$ at each iteration and evaluate the stochastic gradient $\nabla F(x;\xi)$ as an estimate of the true gradient $\nabla f(x)$. First-order information (i.e., the gradient oracle) plays a vital role in these methods since it gives the descent directions for exploring the solution space. These methods are not applicable, however, when gradients are not accessible. This situation, which we call the black-box setting, arises either when the gradient is nonexistent (e.g., the objective value is obtained from physical simulations), or when access to the intrinsic structure of the objective is restricted for security reasons[4]. Even in the ideal situation where automatic differentiation techniques[5] are available, the memory overheads and the strict software requirements can become a bottleneck[6] and limit the usability of gradient-based methods.

This work considers zeroth-order approaches to the problem (1) in the sense that we only have access to the stochastic objective value $F(x;\xi)$. Gaussian smoothing[7] is one of the most popular techniques for estimating gradients in zeroth-order optimization. The key idea is first to sample a Gaussian vector in the solution space and then output the finite difference along this vector as a stochastic gradient estimator. Gaussian smoothing is computationally efficient, theoretically sound, and easily integrated into most first-order methods[8]. However, Gaussian smoothing in the stochastic setting involves sampling in both data and solution spaces, inducing strong gradient noise which can slow down the convergence. Although various variance reduction methods[9] have been developed to address the noise issue, they are mostly restricted to the data sampling procedure, and their effectiveness can be limited due to the noise in the solution space. They also need extensive hyperparameter tuning or periodic gradient evaluations on mega-batches[10].

[1]School of Software Engineering, Sun Yat-Sen University, Guangzhou 510006, China. [2]Energy Development Research Institute, China Southern Power Grid Company Limited, Guangzhou, China. ✉email: hexy73@mail.sysu.edu.cn

Evolutionary algorithms (EAs)[11,12] are a family of meta-heuristics that act as a powerful alternative to Gaussian smoothing-based gradient methods for zeroth-order optimization. Compared to traditional gradient-based methods relying on single-point iterations, EAs evolve a population of solutions and possess inherent tolerance to the noise in the solution space[13]. In addition, modern EAs such as evolution strategies[14,15] are valued for their strong adaptation ability and robustness to complex landscapes[16]. These properties make them very suitable for handling complicated problems such as reinforcement learning[17], combinatorial optimization[18], and neural architecture search[19]. The limitation of EAs, however, is that they have no theoretical guarantees unless on a small number of benchmark problems.

In this paper, we combine the theoretical soundness of Gaussian smoothing-based gradient algorithms and the adaptability of EAs for tackling zeroth-order stochastic problems (1). The main novelty of our method lies in two aspects: 1) unlike existing variance-reduction approaches that primarily target data sampling noise in the data space, our method *simultaneously reduces both data sampling noise and solution space sampling noise* by integrating a recursive momentum rule with a population-based search strategy; 2) it introduces a normalization-based step-size adaptation mechanism that *ensures global and adaptive convergence with any initial step-size settings*, eliminating the need for delicate hyperparameter tuning. We demonstrate that the proposed Population-based Variance-Reduction (PVRE) algorithm attains an ϵ-accurate first-order optimal solution (i.e., a solution with gradient norm below ϵ) within $\mathcal{O}(n\epsilon^{-3})$ function evaluations, matching the best-known complexity bounds[10,20].

Notations $\|\cdot\|$ denotes ℓ_2 norm. $\mathcal{N}$ denotes the n-dimensional isotropic Gaussian distribution. $\mathbb{E}$ denotes taking expectation. $[\tau]$ denotes the set $\{1,\ldots,\tau\}$.

Preliminaries and assumptions

We review two related techniques, namely the Gaussian smoothing method[7] for zeroth-order gradient estimation and the STORM method[21] for data-induced gradient noise reduction. We also list the assumptions required for performance analyses.

Gaussian smoothing

We start by defining the smoothness of a function, as it impacts the accuracy of Gaussian smoothing-based gradient estimation. The detailed proof is provided in Appendix A.

Definition 1 (*Smoothness*) Let h be a differentiable function. We say h is L-smooth if its gradient is L-Lipschitz continuous, i.e.,

$$\|\nabla h(x) - \nabla h(y)\| \leq L\|x-y\|, \quad \forall x, y,$$

holds for some constant $L \in \mathbb{R}_+$.
The L-smoothness of a function h implies the following quadratic bounds:

$$|h(y) - h(x) - \langle \nabla h(x), y-x\rangle| \leq \frac{L}{2}\|x-y\|^2, \tag{2}$$

$$|h(x+v) - h(x-v) - 2\langle \nabla h(x), v\rangle| \leq L\|v\|^2, \tag{3}$$

for all x, y, and v.

The Gaussian smoothing of a function at solution x is the expected objective value over a Gaussian distribution with mean x and a predefined variance:

Definition 2 (*Gaussian smooth approximation*) Fix a function $h : \mathbb{R}^n \to \mathbb{R}$. The Gaussian smooth approximation of h is defined as

$$h_\eta(x) = \mathbb{E}_{v\sim\mathcal{N}}[h(x+\eta v)]$$

where $\eta \in \mathbb{R}_+$ is called the smoothing radius.
The approximation above has a nice property that it is differentiable even when the original function is not. Moreover, it has closed-form gradients depending only on objective function values. This property was given in Theorem 2 of the work[7], and for completeness we restate it below:

Proposition 1 (Gradient estimation based on Gaussian smoothing) *Assume a function $h : \mathbb{R}^n \to \mathbb{R}$ is differentiable. Let $v \in \mathbb{R}^n$ be a perturbation vector and $g \in \mathbb{R}^n$ the finite-difference of h along v as*

$$g = \frac{h(x+\eta v) - h(x-\eta v)}{2\eta} v, \tag{4}$$

where $\eta \in \mathbb{R}_+$. Then g is an unbiased estimator of $\nabla h_\eta(x)$ if v is Gaussian distributed and independent of x, i.e.,

$$\mathbb{E}_{v\sim\mathcal{N}}[g] = \nabla h_\eta(x).$$

Proposition 1 is critical as it provides a way to compute an unbiased gradient estimator via evaluating only two solutions. In the next section, we will apply Gaussian smoothing to the component function $F(x;\xi)$ in problem (1) to obtain a gradient estimator.

STORM

STORM, short for STOchastic Recursive Momentum[21], is a mechanism for reducing the variance due to data sampling in first-order settings. Considering the problem (1) and supposing that the stochastic gradient $\nabla F(x;\xi)$ is available, STORM applies the following descent step

$$x_{t+1} = x_t - \gamma_t d_t, \tag{5}$$

with the momentum term $d_t \in \mathbb{R}^n$ defined recursively as

$$d_t = \underbrace{(1-a_{t-1})d_{t-1} + a_{t-1}\nabla F(x_t;\xi_t)}_{\text{momentum}} + (1-a_{t-1})\underbrace{(\nabla F(x_t;\xi_t) - \nabla F(x_{t-1};\xi_t))}_{\text{error correction}}, \tag{6}$$

where $\gamma_t \in \mathbb{R}_+$ is the descent step-size, $a_{t-1} \in (0,1)$ is the momentum learning rate, and ξ_t is a data sample drawn from $\mathcal{D}$. The momentum d_t is intended to capture the deterministic gradient $\nabla f(x_t)$, provided that the component function $F(x;\xi)$ is smooth. One may view d_t as a combination of standard momentum and correction terms. The term $(1-a_{t-1})d_{t-1} + a_{t-1}\nabla F(x_t;\xi_t)$ acts as a classical heavy-ball momentum[22] that incorporates stochastic gradient estimations into previous descent step. The term $\nabla F(x_t;\xi_t) - \nabla F(x_{t-1};\xi_t)$ corrects the momentum term via including the gradient difference evaluated with the same data sample, aiming to reduce the variance introduced in previous data sampling. The effectiveness of STORM depends on a prerequisite that the error correction term must approach 0 when the iterations converge, in which case the momentum term acts as a Monte Carlo gradient estimator averaged over *t*.

Assumptions

We make the following assumptions regarding the problem (1).

Assumption 1 (*Boundedness of the objective value*) The objective value is lower bounded by some constant $f_* \in \mathbb{R}$, i.e.,

$$f(x) \geq f_*, \quad \forall x.$$

Assumption 2 (*Unbiasedness and boundedness of the data sampling*) The data sampling is unbiased, i.e.,

$$\mathbb{E}_{\xi\sim\mathcal{D}}[\nabla F(x;\xi)] = \nabla f(x), \quad \forall x.$$

In addition, there exists some constant $\sigma \in \mathbb{R}_+$ such that

$$\mathbb{E}_{\xi\sim\mathcal{D}}[\|\nabla F(x;\xi) - \nabla f(x)\|^2] \leq \sigma^2, \quad \forall x.$$

Assumption 3 (*Smoothness of the component function*) The component function $F(x;\xi)$ is L-smooth in x for all ξ.

Assumption 4 (*Boundedness of the gradient norm*) The norm of the gradient is upper bounded by some constant $G \in \mathbb{R}_+$, i.e.,

$$\|\nabla f(x)\| \leq G, \quad \forall x.$$

Assumptions 1 and 2 are customary in analyzing stochastic optimization algorithms. Assumption 3 is relatively restrictive, as it necessitates the smoothness of each component function. Nevertheless, it is the key in achieving the $\mathcal{O}(T^{-\frac{1}{3}})$ convergence rate and was typically made in recent studies relying on STORM. Notice that almost all existing works using Gaussian smoothing, to our knowledge, require this assumption to establish convergence theorems, even for getting a slower rate of $\mathcal{O}(T^{-\frac{1}{4}})$[23]. This means, compared to existing Gaussian smoothing based zeroth-order algorithms, we can achieve a speedup in convergence for free. Assumption 4 is used in guaranteeing the adaptability. We note that this is looser than the uniform bound (i.e., $\|\nabla F(x;\xi)\| \leq G$ for all ξ) typically used in STORM based methods[21,24].

Hereinafter we denote the Gaussian smooth approximations of f and F by f_η and F_η, respectively, i.e.,

$$f_\eta(x) = \mathbb{E}_{v\sim\mathcal{N}}[f(x+\eta v)], \quad F_\eta(x;\xi) = \mathbb{E}_{v\sim\mathcal{N}}[F(x+\eta v;\xi)].$$

PVRE: Population-based variance-reduced evolution

We present the PVRE method for handling problem (1) in black-box settings. PVRE combines Gaussian smoothing and STORM to reduce the sampling noise in both the data space and the solution space. It also uses a normalized descent rule to achieve adaptability.

Adaptive descent with normalized momentum

PVRE employs the following normalized descent rule

$$x_{t+1} = x_t - \gamma_t \frac{d_t}{\|d_t\|} \tag{7}$$

where $d_t \in \mathbb{R}^n$ is a descent step. The use of normalization is critical to achieving adaptability. To see this, denote ϵ_t as the difference between d_t and $\nabla f_{\eta_t}(x_t)$:

$$\epsilon_t = d_t - \nabla f_{\eta_t}(x_t). \tag{8}$$

Then we can bound the per-iteration progress using ϵ_t.

Lemma 1 (Per-iteration progress) *The detailed proof is provided in Appendix B. Under Assumption 3, the iterations* $\{x_t\}$ *generated by (7) satisfy*

$$f_{\eta_{t+1}}(x_{t+1}) \le f_{\eta_t}(x_t) - \frac{\gamma_t}{3}\|\nabla f_{\eta_t}(x_t)\| + \frac{8}{3}\gamma_t\|\epsilon_t\| + \frac{L}{2}\left(\gamma_t^2 + (\eta_t^2 + \eta_{t+1}^2)n\right).$$

Lemma 1 states that, once the estimation error ϵ_t is small, sufficient descent in the smoothed objective f_{η_t} can be ensured. Moreover, this property is ensured adaptively in two aspects: 1) it is independent of the data noise (as it does not rely on Assumption 2), and 2) knowledge of the problem-dependent constant L is not required, as the L-related term in the bound can be made arbitrarily small by decreasing the hyperparameters γ_t, η_t, and η_{t+1}.

It is now clear that all we need is to make sure $d_t \approx \nabla f_{\eta_t}(x_t)$ so that ϵ_t can be bounded. We achieve this by using the following update rule:

$$\begin{cases} d_t = (1 - a_{t-1})d_{t-1} + a_{t-1}g_t + (1 - a_{t-1})(g_t - \tilde{g}_{t-1}) \\ d_0 = g_0 \end{cases} \tag{9}$$

where $a_{t-1} \in (0, 1)$ is a constant, and $g_t, \tilde{g}_{t-1} \in \mathbb{R}^n$ are intended to satisfy

$$\mathbb{E}[g_t] = \nabla f_{\eta_t}(x_t) \text{ and } \mathbb{E}[\tilde{g}_{t-1}] = \nabla f_{\eta_{t-1}}(x_{t-1}).$$

Comparing (9) with (6), it is found that the descent step d_t follows the momentum update rule of STORM, with the exception that we replaced the objective function f with its Gaussian smoothing. Our idea here is to obtain g_t and $\tilde{g}_{t-1}$ by applying the finite-difference rule (4) to the component function F and then inject them into (9). One core difficulty exists, however, in that whereas the correction term in (6) only reduces the noise due to data sampling, applying finite-differences would introduce additionally noise due to solution sampling. Technically speaking, the error correction term may not diminish (i.e., $\|g_t - \tilde{g}_{t-1}\| \not\to 0$) even when the iterations converge, which would impact the variance reduction effect. We overcome this by using a population mechanism and a perturbation reuse strategy in gradient estimation.

Population-based gradient estimation with perturbation reuse

We show how to compute g_t and $\tilde{g}_{t-1}$ in (9). To obtain g_t, we generate a set of Gaussian perturbations $\{v_{t,k} \sim \mathcal{N}\}_{k\in[\tau]}$ where τ is the population size. We also draw a set of data samples $\{\xi_{t,k} \sim \mathcal{D}\}_{k\in[\tau]}$. Then, apply the finite-difference rule (4) to F with pairs of $\xi_{t,k}$ and $v_{t,k}$ as

$$g_t = \frac{1}{\tau}\sum_{k=1}^{\tau} g_{t,k}, \quad g_{t,k} = \frac{F(x_t + \eta_t v_{t,k}; \xi_{t,k}) - F(x_t - \eta_t v_{t,k}; \xi_{t,k})}{2\eta_t} v_{t,k}, \tag{10}$$

where $\eta_t \in \mathbb{R}_+$ is the corresponding smoothing radius.

Lemma 2 (Population based variance reduction) *The detailed proof is provided in Appendix B. Suppose Assumptions 4 and 2 hold and let*

$$\rho_t = \frac{L^2}{2}\eta_t^2(n+6)^3 + 2(n+4)(G^2 + \sigma^2).$$

Then, we have

$$\mathbb{E}\left[\|g_t - \nabla f_{\eta_t}(x_t)\|^2\right] \le \frac{\rho_t}{\tau}.$$

We now proceed on computing $\tilde{g}_{t-1}$ for estimating $\nabla f_{\eta_{t-1}}(x_{t-1})$. Intuitively, one can draw another population of Gaussian perturbations and perform Gaussian smoothing similarly as in (10). But we propose that a better way is to *reuse the population* $\{v_{t,k}\}_{k\in[\tau]}$ that has been used for gradient estimation at x_t. Specifically, given the population $\{v_{t,k}\}_{k\in[\tau]}$ and the corresponding data samples $\{\xi_{t,k}\}_{k\in[\tau]}$ used in (10), we perform gradient estimation at solution x_{t-1} as

$$\tilde{g}_{t-1} = \frac{1}{\tau}\sum_{k=1}^{\tau}\tilde{g}_{t-1,k}, \quad \tilde{g}_{t-1,k} = \frac{F(x_{t-1}+\eta_{t-1}v_{t,k};\xi_{t,k}) - F(x_{t-1}-\eta_{t-1}v_{t,k};\xi_{t,k})}{2\eta_{t-1}}v_{t,k}, \tag{11}$$

where η_{t-1} is the smoothing radius that may differ from η_t.

The perturbation reuse in (11) is the key step of PVRE. At first glance, $\tilde{g}_{t-1}$ is simply an unbiased estimator of $\nabla f_{\eta_{t-1}}(x_{t-1})$, according to Proposition 1. But we notice that the underlying reason lies in that the difference between these two gradient estimators can be upper bounded when they are constructed with the same population of perturbations.

Lemma 3 (Boundedness of the correction term) *The detailed proof is provided in Appendix B. Under Assumption 3, the gradient estimations given in 10 and 11 satisfy*

$$\mathbb{E}\left[\|g_{t,i} - \tilde{g}_{t-1,i}\|^2\right] \le \frac{3L^2(\eta_t^2+\eta_{t-1}^2)}{4}(n+6)^3 + 3(n+4)L^2\gamma_{t-1}^2.$$

Lemma 3 implies that, when η_t, η_{t-1}, and γ_{t-1} are sufficiently small, the correction term in (9) (i.e., $g_t - \tilde{g}_{t-1}$) is negligible, and the iterations will converge (i.e., $x_t \approx x_{t-1}$). The momentum d_t then behaves as a Monte Carlo estimator of $\nabla f_{\eta_t}(x_t)$ whose variance reduces over iterations. This is confirmed by the following lemma.

Lemma 4 (Boundedness of gradient estimation errors) *The detailed proof is provided in Appendix B. Consider the iterations generated from (7) with the settings 9 to 11.*

Choose the hyperparameters as

$$\eta_t = \eta_0(t+1)^{-2/3}, \quad \gamma_t = \gamma_0(t+1)^{-2/3}, \quad a_t = (t+1)^{-2/3}, \quad \eta_0 = \frac{\gamma_0}{n+6}. \tag{12}$$

Under Assumptions 2 to 4, the gradient estimation error defined in (8) can be bounded as

$$\sum_{t=0}^{T-1}\gamma_t\mathbb{E}[\|\epsilon_t\|] \le 9\gamma_0(\gamma_0 L + G + \sigma)(1+\log T)\sqrt{\frac{n+6}{\tau}}.$$

Main results

We now present the PVRE method by putting all above details together. The pseudocode is given in Algorithm 1. Below we establish the convergence theorem of the proposed method. The detailed proof is provided in Appendix C.

Require: $x_0 \in \mathbb{R}^n$: initial solution; $\eta_t \in \mathbb{R}_+$: smoothing radius; $\gamma_t \in \mathbb{R}_+$: step-sizes; $\alpha_t \in (0,1)$: momentum changing rates

Draw a set of data samples $\{\xi_{0,k} \sim \mathscr{D}\}_{k\in[\tau]}$
Draw a population of perturbations $\{v_{0,k} \sim \mathscr{N}\}_{k\in[\tau]}$
$g_{0,k} = \frac{F(x_0+\eta_0 v_{0,k};\xi_{0,k}) - F(x_0-\eta_0 v_{0,k};\xi_{0,k})}{2\eta_0}v_{0,k}, \quad k \in [\tau]$
$d_0 = g_0 = \frac{1}{\tau}\sum_{k=1}^{\tau} g_{0,k}$
for $t \leftarrow 0, \ldots, T-1$ **do**
  $x_{t+1} = x_t - \gamma_t \frac{d_t}{\|d_t\|}$
  Draw a set of data samples $\{\xi_{t+1,k} \sim \mathscr{D}\}_{k\in[\tau]}$
  Draw a population of perturbations $\{v_{t+1,k} \sim \mathscr{N}\}_{k\in[\tau]}$
  $g_{t+1,k} = \frac{F(x_{t+1}+\eta_{t+1}v_{t+1,k};\xi_{t+1,k}) - F(x_{t+1}-\eta_{t+1}v_{t+1,k};\xi_{t+1,k})}{2\eta_{t+1}}v_{t+1,k}, \quad k \in [\tau]$
  $\tilde{g}_{t,k} = \frac{F(x_t+\eta_t v_{t+1,k};\xi_{t+1,k}) - F(x_t-\eta_t v_{t+1,k};\xi_{t+1,k})}{2\eta_t}v_{t+1,k}, \quad k \in [\tau]$
  $g_{t+1} = \frac{1}{\tau}\sum_{k=1}^{\tau} g_{t+1,k}$
  $\tilde{g}_t = \frac{1}{\tau}\sum_{k=1}^{\tau} \tilde{g}_{t,k}$
  $d_{t+1} = (1-a_t)d_t + a_t g_{t+1} + (1-a_t)(g_{t+1} - \tilde{g}_t)$
end for

Algorithm 1. Population-based Variance-Reduced Evolution (PVRE).

Theorem 1 *Consider solving problem (1) using Algorithm 1 with settings in (12). Suppose Assumptions 4 to 3 hold. Then the iterations satisfy*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|] \leq 72T^{-\frac{1}{3}}\left(\frac{\Delta_f}{\gamma_0}+(\gamma_0 L+G+\sigma)(1+\log T)\sqrt{\frac{n+6}{\tau}}+\gamma_0 L\right)+2L\sqrt{n+6}\gamma_0 T^{-\frac{2}{3}},$$

where $\Delta_f = f(x_0) - f_*$.

Remark (Function evaluation complexity) The bound given above can be tightened when choosing $\gamma_0 = \Theta\left(\sqrt{\frac{\Delta_f}{L\left(1+\sqrt{\frac{n+6}{\tau}}\right)}}\right)$, which yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|] = \tilde{\mathcal{O}}\left(\frac{1+\log T}{T^{\frac{1}{3}}}\left(\sqrt{\Delta_f L}\left(1+\left(\frac{n}{\tau}\right)^{\frac{1}{4}}\right)+(G+\sigma)\sqrt{\frac{n}{\tau}}\right)\right)$$

where the $\tilde{\mathcal{O}}$ notation hides the negligible $\log T$ term. Now choose $\tau = \Theta(n)$ and consider L, Δ_f, G, and σ as constants. Then PVRE guarantees $\min_{0\leq t<T}\mathbb{E}[\|\nabla f(x_t)\|] \leq \epsilon$ within $T = \tilde{\mathcal{O}}\left(\frac{1}{\epsilon^3}\right)$ iterations. As 4τ function evaluations are performed at each iteration, we conclude that the function evaluation complexity for PVRE to output an ϵ-accurate first-order optimal solution is $\tilde{\mathcal{O}}\left(\frac{n}{\epsilon^3}\right)$. This aligns with the best known result given in[25], up to a $\log T$ factor, which is caused by the diminishing step-size rule. We note that the main advantage of PVRE over existing methods lies in its adaptive convergence, which requires no prior knowledge of the problem landscape characteristics (e.g., L and σ).

Experiments

We evaluate the performance of PVRE on several benchmark problems and a real-world adversarial attack task. Additional studies on the impact of population size and initial step size are also present.

Binary classification based benchmark problems

We evaluate PVRE on three binary classification models, including logistic regression (LR), non-convex support vector machine (NSVM), and linear support vector machine (LSVM). The objective function is given by:

$$F(x;\xi) = loss(x;z,y) = \begin{cases} \log(1+\exp(-y(x^\top z))) & \text{(LR)} \\ 1-\tanh(y(x^\top z)) & \text{(NSVM)} \\ \max\left\{0, 1-y(x^\top z)\right\} & \text{(LSVM)} \end{cases}$$

where each data sample ξ is a pair of input feature $z \in \mathbb{R}^n$ and the associated label $y \in \{-1, 1\}$. LR is the simplest model as it is convex and smooth; we choose it to test the algorithm's exploitation ability. NSVM is smooth but nonconvex, and we use it to verify the global exploration performance of an algorithm. LSVM deviates from the smoothness assumption, and is used mainly for testing whether an algorithm is robust to irregularities on the landscape.

The data distribution $\mathcal{D}$ consists of N data samples, i.e., $\mathcal{D} = \{\xi_1, \ldots, \xi_N\}$, and is constructed using four widely-used benchmark datasets: `covtype`, `gisste`, `rcv1`, and `real-sim`. Table 1 provides a brief summary of these statistics.

We compare PVRE with five state-of-the-art algorithms, including ZO-SGD[23], ZO-SignSGD[26], ZO-AdaGrad[27], ZO-Adam[28], and ZO-STORM[21]. Since the original STORM method was not designed for a zeroth-order setting, we construct its zeroth-order variant, referred to as ZO-STORM, by replacing its gradient oracles with a Gaussian smoothing-based gradient estimator. All algorithms use the finite-difference gradient estimator (10) with $\tau = 20$. The minibatch size is set to 1000 for all algorithms. In Gaussian smoothing, we set the initial smoothing radius as $\eta_0 = 10^{-6}$, and adopt a diminishing schedule given by $\eta_t = \eta_0/(1+t)^{2/3}$. All algorithms are initialized at $x_0 = (0, \ldots, 0)^\top$ and employ a diminishing step-size schedule $\gamma_t = \gamma_0/(1+t)^{2/3}$ throughout the optimization process. The initial step-size γ_0 is fine-tuned by a grid-search in $\{10^{-4}, 10^{-3}, \ldots, 10^4\}$. Every algorithm is run independently 11 times on each dataset. The results from the median run (whose final training loss is the median among the 11 outputs) are reported.

dataset	*n*	*N*
`covtype`	54	581,012
`gisste`	5,000	6,000
`rcv1`	47,236	677,399
`real-sim`	20,958	72,309

Table 1. Statistical summary of the datasets used.

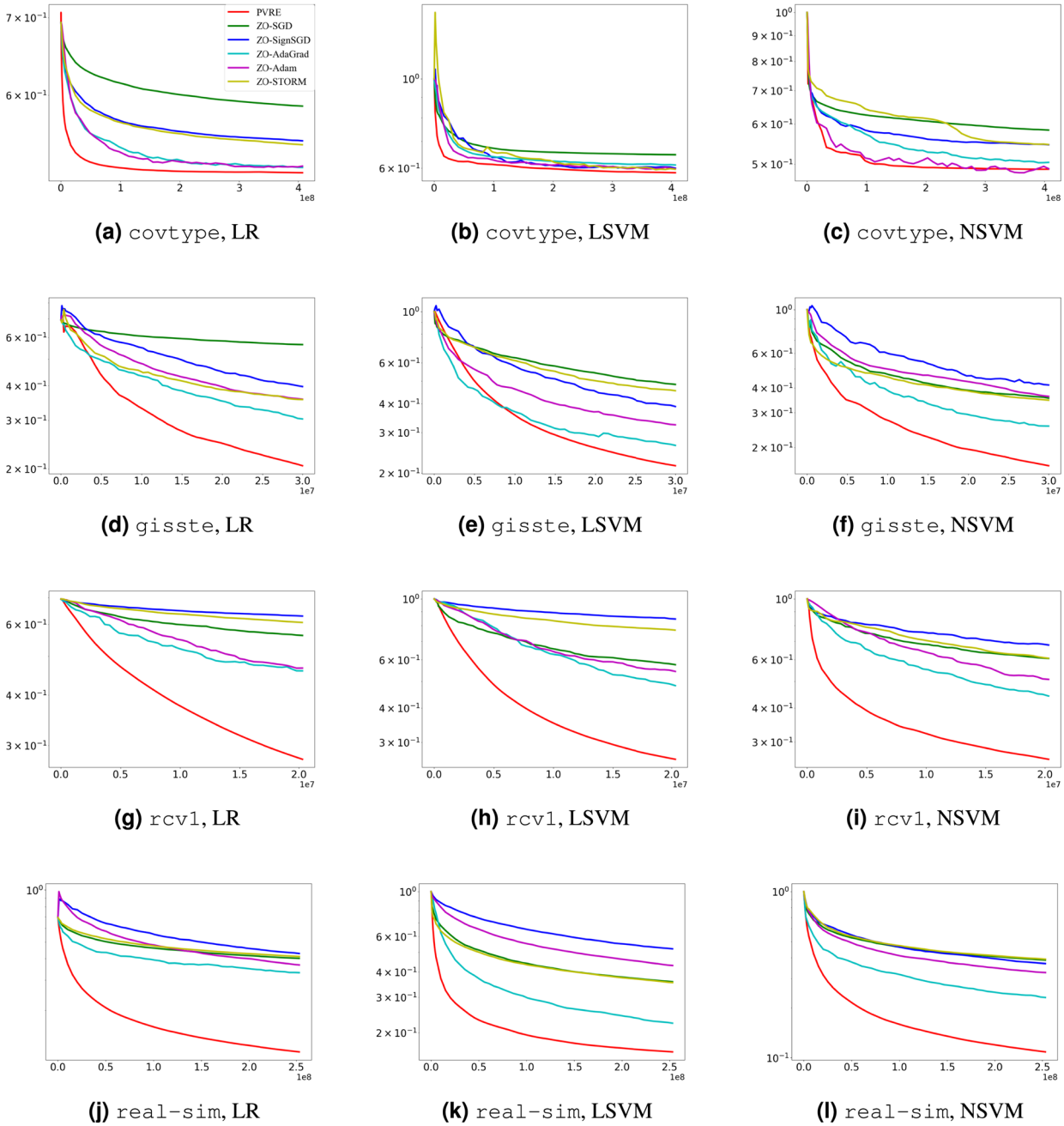


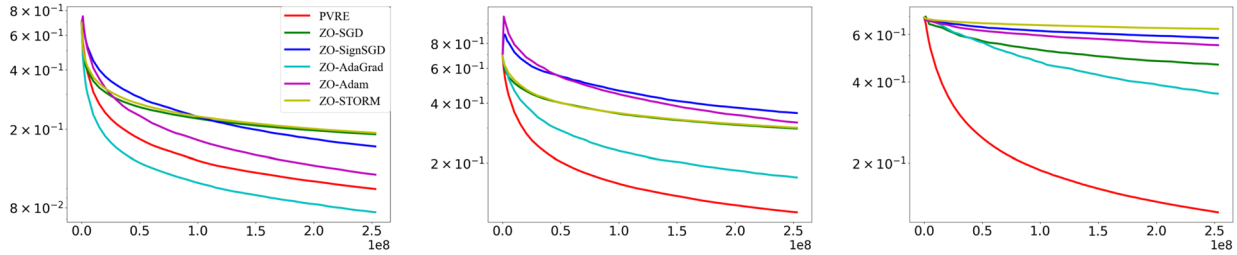
Fig. 1. Comparison on benchmark problems. The curve displays the training loss versus the number of function evaluations.

Figure 1 shows the convergence performance of the algorithms. PVRE is clearly the best performer and enjoys faster convergence speed. We also notice that PVRE performs consistently well on NSVM even though this problem does not satisfy the componentwise smoothness assumption. This implies PVRE is robust against irregular landscape features.

Impact of population size

PVRE uses a population-based search mechanism to guide the optimization process. To evaluate the impact of this mechanism, we conduct experiments to examine how the population size parameter τ affects performance. Specifically, we test the LR model on the `real-sim` dataset where the population size parameter τ is set to $\{1, 10, 100\}$. All other settings are the same as those in Section "Binary classification based benchmark problems".

Figure 2 shows the experimental results. When the population size is set to 1, the PVRE algorithm essentially degenerates into a strategy similar to a single-point search. In this case, due to the lack of support for population



(a) $\tau = 1$ **(b)** $\tau = 10$ **(c)** $\tau = 100$

Fig. 2. Comparison under different population size settings. The curve displays the training loss versus the number of function evaluations.

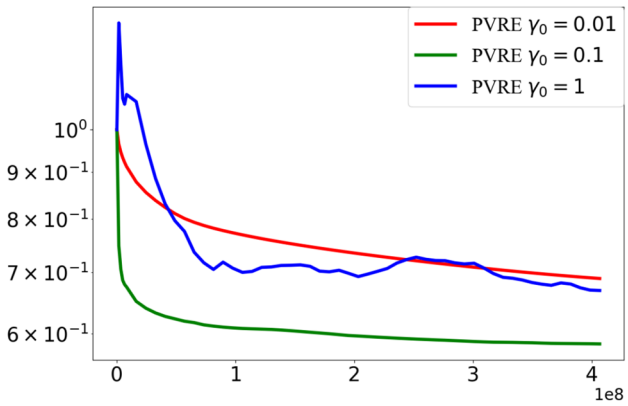


Fig. 3. Comparison under different initial step sizes. The curve displays the training loss versus the number of function evaluations.

diversity, PVRE cannot effectively use the differences and complementarities within the population to guide the search direction. With the increase in population size, PVRE clearly outperforms the comparison algorithm. It is probably due to that a larger population size increases the diversity of search directions, therefore enhancing exploration of the search space and solution quality.

Impact of initial step size

Here we examine the influence of the initial step size on the performance of the PVRE algorithm, where the model is LR and the experiments are conducted on the `covtype` dataset. The initial step size γ_0 is set to $\{0.01, 0.1, 1\}$, while keeping all other experimental settings identical to those in Section "Binary classification based benchmark problems".

Figure 3 presents the results on all test instances using performance profiles measured by the loss values. It is evident that the choice of initial step size significantly affects the convergence behavior of PVRE. When $\gamma_0 = 0.01$, the algorithm converges steadily but relatively slowly, resulting in stable yet gradual loss reduction across test instances. Increasing the initial step size to $\gamma_0 = 0.1$ leads to faster initial progress and lower loss values. However, when the initial step size is set to $\gamma_0 = 1$, the loss curve rises initially before decreasing and exhibits noticeable fluctuations throughout the optimization process.

Black-box adversarial attacks

We consider a real-world task for adversarial perturbation generalization[29]. Given a neural network-based image classifier, the task is to find a perturbation that, when universally applied, degrades the classifier's accuracy. The data sample ξ in this task consists of an image z and its true label t. The objective of this task reads:

$$F(x;\xi) = \underbrace{\max\{0, \log \pi_t(x \oplus z) - \max_{j \neq t} \pi_j(x \oplus z)\}}_{\text{attack loss}} + \underbrace{\frac{\lambda}{2}\|x \oplus z - z\|^2}_{\text{distortion}},$$

where λ is a regularization coefficient, $x \oplus z$ denotes applying the perturbation x to an image z, and $\pi_j(x \oplus z)$ is the corresponding probability that the perturbed image is predicted into class j. The objective penalizes the top-1 prediction accuracy and therefore encouraging misclassification due to the perturbation. The use of an ℓ_2 regularization is to prevent the magnitude of the image distortion from growing too fast.

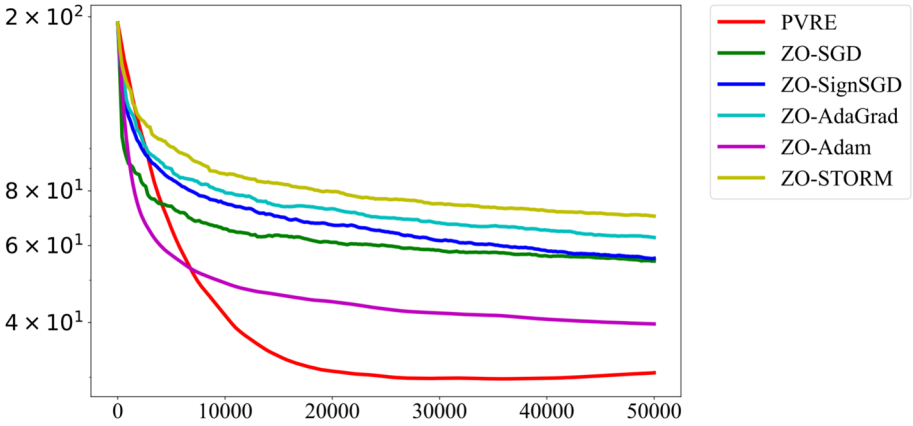


Fig. 4. Attack loss versus the number of function evaluations.

Algorithm	Attack success rate	Total loss	Averaged distortion
PVRE	79.60%	30.72	21.20
ZO-SGD	43.10%	55.29	7.40
ZO-SignSGD	44.10%	56.11	7.10
ZO-AdaGrad	39.70%	62.60	6.17
ZO-Adam	58.70%	39.72	10.40
ZO-STORM	33.40%	70.05	5.86

Table 2. Final results on universal adversarial perturbation after using a budget of 50000 function evaluations.

We use the `CIFAR-10` dataset[30] in the experiment and choose 1000 images to build the data distribution. The regularization coefficient λ is fixed to 10. The VGG16 model[31] is used as the image classifier. Suppose this model only receives images in the range $[-0.5, 0.5]^n$ where n is the dimension of the image space. We can then use the following perturbation operator to make sure the solution space becomes $\mathbb{R}^n$:

$$x \oplus z = \frac{1}{2}\tanh\left(\tanh^{-1}(2z) + x\right).$$

The five algorithms considered in the previous experiment are chosen here for comparison. We set η_0 to 10^{-5} and tune γ_0 in $\{10^{-7}, 10^{-6}, \ldots, 10^{2}\}$. In the data sampling, we use a minibatch of size 5. All other settings are kept the same as in Section "Binary classification based benchmark problems".

Figure 4 shows the trajectory of the attack loss versus the number of function evaluations. It is found that PVRE exhibits the fastest convergence rate among all algorithms. We also present the attack success rate, the final loss, and the averaged distortion (measured by $\frac{1}{N}\sum_{i=1}^{N}\|x \oplus z_i - z_i\|^2$) in Table 2, obtained within a budget of 50000 function evaluations. PVRE obtains the highest attack success rate and the lowest objective value. The perturbation obtained by PVRE exhibits the largest distortion magnitude, but we note that the obtained perturbation is almost imperceptible when applied to the images. In Fig. 5 we present several example images to confirm this. It is found that there is no significant difference between the perturbed images output by different algorithms.

Related work

First-order stochastic algorithms such as SGD achieve a gradient evaluation complexity of $\mathcal{O}(\epsilon^{-4})$ when the objective function f is smooth[32]. When the component objective F, the complexity of zeroth-order methods may align with their first-order counterparts in terms of the ϵ-dependence, but suffer an n-dependence slowdown—this being the price paid for not knowing the gradient.

On functions with smooth components, various variance reduction methods exist that improve the complexity of first-order algorithms to $\mathcal{O}(\epsilon^{-3})$. Remarkable examples include STORM[21], SVRG[33], and SPIDER[10]. Extending these methods to zeroth-order settings, however, is not straightforward. For example, it is found that the convergence rate of zeroth-order SVRG methods cannot match the original SVRG in terms of the ϵ-dependence[34,35]. The zeroth-order SPIDER requires periodic gradient evaluations with a megabatch of data samples and a huge population of perturbations along the coordinate directions[10]. The best known result achieved by zeroth-order methods is found in the work[20] where the authors proved an $n^{3/4}\epsilon^{-3}$ complexity with a stronger assumption. Our result matches theirs if using the same assumption.

How to achieve adaptability is an active research topic in first-order stochastic optimization. Representative methods such as AdaGrad[27] and Adam[36] employ a second-order momentum to perform coordinate-wise

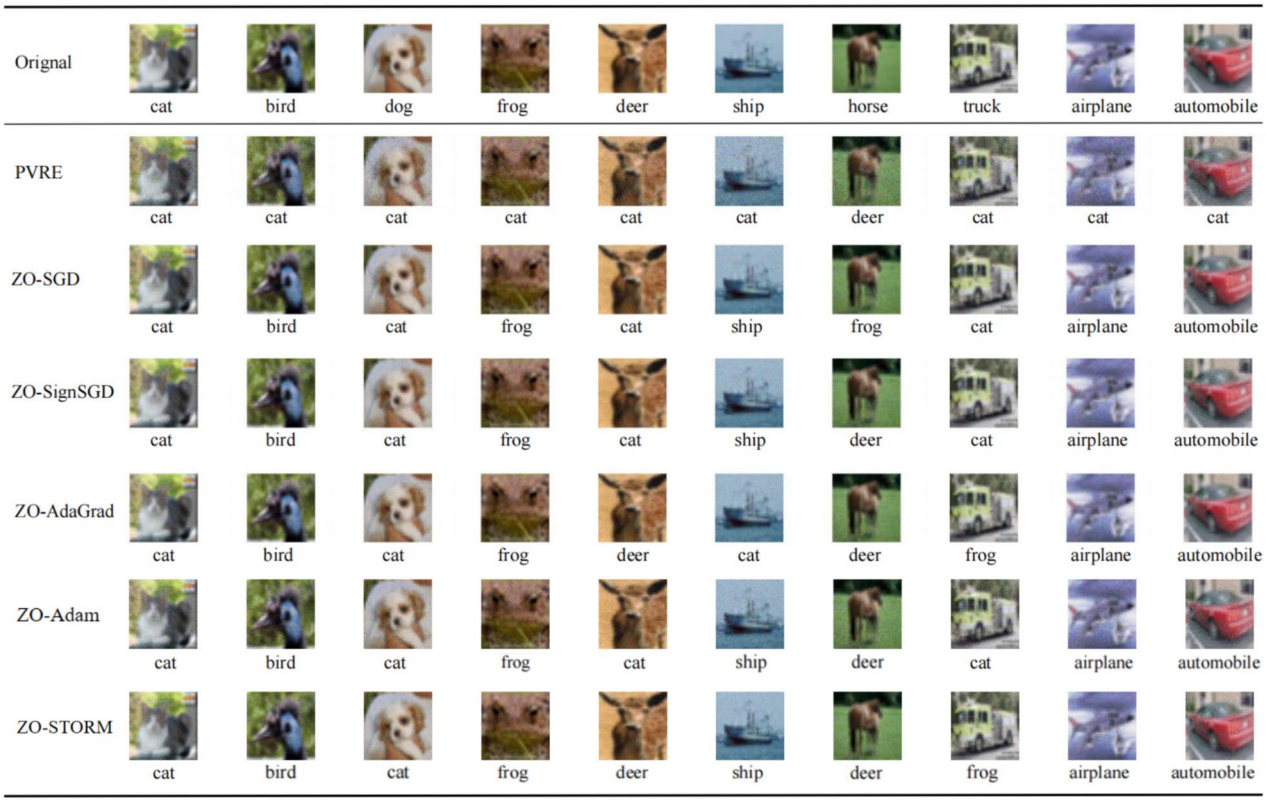


Fig. 5. Examples of perturbation images on the CIFAR-10 image dataset. The first row shows the original images and their true labels. The subsequent rows show the perturbed images with the corresponding falsely predicted labels.

normalization, aiming to address the ill-conditioning issue. The STORM method[24] incorporates an AdaGrad-style momentum rule and simultaneously achieves adaptive convergence and variance reduction. Adam and AdaGrad has also been extended to zeroth-order settings[28,37], yet they lack employ a variance-reduction mechanism.

EAs in the literature are seldom used for solving stochastic problems, possibly due to the fact that the standard comparison-based evolution operator may yield bias in gradient estimation[38]. We addressed this in a previous work[39] by using megabatch sampling in distributed settings. It has also been seen that classical evolution gradient search can work well in reinforcement learning and enjoy strong scalability[40]. The main advantage of using EAs is that they typically have better exploration ability on nonconvex landscapes compared to those employing single-solution iterations[41].

Conclusion

In this paper, we propose a method called PVRE, which combines Gaussian smoothing and population-based variance reduction techniques. We show that PVRE reaches the objective function complexity of $\tilde{\mathcal{O}}(n\epsilon^{-3})$ in finding an ϵ-accurate first-order optimum while ensuring the convergence with any initial step-size. PVRE demonstrates promising performance in several experiments on both benchmark problems and real-world applications.

Data availability

The datasets used in this study are all publicly available. Specifically, we use four benchmark datasets from the LIBSVM repository—covtype, gisste, rcv1, and real-sim—available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, as well as the CIFAR-10 dataset, which is accessible from https://www.cs.toronto.edu/~kriz/cifar.html.

Received: 30 May 2025; Accepted: 4 September 2025
Published online: 07 October 2025

References

1. Bottou, L., Curtis, F. E. & Nocedal, J. Optimization methods for large-scale machine learning. *SIAM review* **60**, 223–311 (2018).
2. Robbins, H. & Monro, S. A stochastic approximation method. *Annals Math. Stat.* **22**, 400–407 (1951).
3. Li, X. & Orabona, F. On the convergence of stochastic gradient descent with adaptive stepsizes. *Proc. Mach. Learn. Res.* **89**, 983–992 (2019).

4. Guidotti, R. et al. A survey of methods for explaining black box models. *ACM Comput. Surv.* **51**, 1–42. https://doi.org/10.1145/3236009 (2018).
5. Margossian, C. C. A review of automatic differentiation and its efficient implementation. *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.* **9**, e1305 (2019).
6. Sung, M., Palakonda, V., Kim, I.-M., Yun, S. & Kang, J.-M. Deco-mesc: Deep compression-based memory-constrained split computing framework for cooperative inference of neural network. *IEEE Trans. Veh. Technol.* **74** (8), 13319-13324 (2025).
7. Nesterov, Y. & Spokoiny, V. Random Gradient-Free Minimization of Convex Functions. *Foundations Comput. Math.* **17**, 527–566 (2017).
8. Gao, K. & Sener, O. Generalizing gaussian smoothing for random search. In *International Conference on Machine Learning*, 7077–7101 (PMLR, 2022).
9. Gower, R. M., Schmidt, M., Bach, F. & Richtarik, P. Variance-Reduced Methods for Machine Learning. *Proc. IEEE* **108**, 1968–1983. https://doi.org/10.1109/JPROC.2020.3028013 (2020).
10. Fang, C., Li, C. J., Lin, Z. & Zhang, T. SPIDER: Near-Optimal Non-Convex Optimization via Stochastic Path-Integrated Differential Estimator. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.*, 687–697 (2018).
11. Eiben, A. E. & Schoenauer, M. Evolutionary computing. *Inf. Process. Lett.* **82**, 1–6 (2002).
12. Holland, J. H. Genetic algorithms. *Sci. Am.* **267**, 66–73 (1992).
13. Hellwig, M. & Beyer, H.-G. On the steady state analysis of covariance matrix self-adaptation evolution strategies on the noisy ellipsoid model. *Theor. Comput. Sci.* **832**, 98–122 (2020).
14. Beyer, H.-G. & Schwefel, H.-P. Evolution strategies-a comprehensive introduction. *Nat. Comput.* **1**, 3–52 (2002).
15. Hansen, N., Müller, S. D. & Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evol. Comput.* **11**, 1–18 (2003).
16. He, X., Zhou, Y. & Chen, Z. Evolutionary Bilevel Optimization based on Covariance Matrix Adaptation. *IEEE Trans. Evol. Comput.* **23**, 258–272. https://doi.org/10.1109/TEVC.2018.2849000 (2019).
17. Diouane, Y., Lucchi, A. & Patil, V. P. A globally convergent evolutionary strategy for stochastic constrained optimization with applications to reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 836–859 (PMLR, 2022).
18. Blum, C., Puchinger, J., Raidl, G. R. & Roli, A. Hybrid metaheuristics in combinatorial optimization: A survey. *Appl. Soft Comput.* **11**, 4135–4151. https://doi.org/10.1016/j.asoc.2011.02.032 (2011).
19. Elsken, T., Metzen, J. H. & Hutter, F. Neural Architecture Search: A Survey. *J. Mach. Learn. Res.* **20**, 1–21 (2019).
20. Huang, F., Gao, S., Pei, J. & Huang, H. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *J. Mach. Learn. Res.* **23**, 1–70 (2022).
21. Cutkosky, A. & Orabona, F. Momentum-Based Variance Reduction in Non-Convex SGD. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*, 15210–15219 (2019).
22. Ghadimi, E., Feyzmahdavian, H. R. & Johansson, M. Global convergence of the Heavy-ball method for convex optimization. In *2015 European Control Conference (ECC)*, 310–315, https://doi.org/10.1109/ECC.2015.7330562 (2015).
23. Ghadimi, S. & Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.* **23**, 2341–2368 (2013).
24. Levy, K., Kavis, A. & Cevher, V. Storm+: Fully adaptive sgd with momentum for nonconvex optimization. In *35th Conference on Neural Information Processing Systems* (PMLR, 2021).
25. Huang, F., Tao, L. & Chen, S. Accelerated stochastic gradient-free and projection-free methods. In *International conference on machine learning*, 4519–4530 (PMLR, 2020).
26. Liu, S., Chen, P.-Y., Chen, X. & Hong, M. signsgd via zeroth-order oracle. In *International Conference on Learning Representations* (ICLR, 2018).
27. Duchi, J., Hazan, E. & Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **12**, 2121–2159 (2011).
28. Chen, X. et al. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. *Adv. Neural Inf. Process. Syst.* **32**, 7204–7215 (2019).
29. Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O. & Frossard, P. Universal Adversarial Perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 86–94, https://doi.org/10.1109/CVPR.2017.17 (IEEE, Honolulu, HI, 2017).
30. Krizhevsky, A. & Hinton, G. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases* **1** (2009).
31. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *Computer Science* arXiv:1409.1556 (2014).
32. Khaled, A. & Richtárik, P. Better Theory for SGD in the Nonconvex World. *Transact. Mach. Learn. Res.* arXiv:2002.03329 (2022).
33. Reddi, S. J., Hefny, A., Sra, S., Poczos, B. & Smola, A. Stochastic Variance Reduction for Nonconvex Optimization. In *International Conference on Machine Learning*, 314–323 (PMLR, 2016).
34. Ji, K., Wang, Z., Zhou, Y. & Liang, Y. Improved Zeroth-Order Variance Reduced Algorithms and Analysis for Nonconvex Optimization. In *International Conference on Machine Learning*, 3100–3109 (PMLR, 2019).
35. Liu, S. et al. Zeroth-order stochastic variance reduction for nonconvex optimization. *Adv. Neural Inf. Process. Syst.* **31** arXiv:1805.10367 (2018).
36. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *Computer Science* arXiv:1412.6980(2014).
37. He, X., Zheng, Z., Chen, Z. & Zhou, Y. Adaptive evolution strategies for stochastic Zeroth-order optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* **6**, 1271–1285. https://doi.org/10.1109/TETCI.2022.3146330 (2022).
38. Astete-Morales, S., Cauwet, M.-L. & Teytaud, O. Evolution Strategies with Additive Noise: A Convergence Rate Lower Bound. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, FOGA '15, 76–84, https://doi.org/10.1145/2725494.2725500 (Association for Computing Machinery, New York, NY, USA, 2015).
39. He, X. et al. Distributed evolution strategies for black-box stochastic optimization. *IEEE Trans. Parallel Distrib. Syst.* **33**, 3718–3731. https://doi.org/10.1109/TPDS.2022.3168873 (2022).
40. Salimans, T., Ho, J., Chen, X., Sidor, S. & Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint* arXiv:1703.03864 (2017).
41. Dang, D. C. et al. Escaping local optima using crossover with emergent diversity. *IEEE Trans. Evol. Comput.* **22**, 484–497. https://doi.org/10.1109/TEVC.2017.2724201 (2018).

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2022YFF0610003).

Author contributions

Z.P. designed and implemented the core methodology, conducted key experiments, and wrote the initial manu-

script. X.H. (corresponding author) co-designed and implemented the core methodology, supervised the overall research, proposed the main framework, and revised the manuscript. Y.P. contributed to algorithm optimization and result analysis. B.P. contributed to the experimental setup and data processing. W.C. provided domain-specific expertise. Y.Z. supervised and verified the overall research process. All authors reviewed and approved the final manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-18876-0.

Correspondence and requests for materials should be addressed to X.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.