Hessian Estimation via Stein's Identity in Black-Box Problems

Jingyi Zhu JINGYI.ZHU@JHU.EDU
DAMO Academy, Alibaba Inc., Bellevue, WA

Abstract

When only the noisy zeroth-order (ZO) oracle is available, stochastic approximation algorithms are popular for estimating the root of the multivariate gradient equation. Inspired by Stein's identity, this work establishes a novel Hessian approximation scheme. We compare it with second-order simultaneous perturbation stochastic approximation (2SPSA) algorithm (Spall, 2000). On the basis of the almost sure convergence guarantee with the same convergence rate, 2SPSA requires *four* ZO queries, while ours requires *three* instead. Moreover, 2SPSA requires *two* statistically independent perturbations and *two* differencing stepsizes, while ours requires generating *one* perturbation vector and tuning *one* differencing stepsize only. Besides, the weighting mechanism for the Hessian estimate is generalized and the smoothness restriction on the loss function is relaxed compared to 2SPSA. Finally, we present numerical support for the reduced per-iteration ZO query complexity.

Keywords: stochastic optimization; Hessian estimation; Stein's identity; gradient-free methods

1. Introduction

Stochastic approximation (SA) has been widely applied in the problem of minimization. Let $\theta \in \mathbb{R}^d$ concatenate all the adjustable model parameters. Let the system stochasticity be represented by the random variable ω following a probability distribution $\mathbb{P}$ on its domain Ω. Consider finding the minimizer of a twice-differentiable bounded-from-below loss function $f(\cdot) : \mathbb{R}^p \to \mathbb{R}$:

$$\theta^* \equiv \arg\min_{\theta \in \mathbb{R}^d} f(\theta)\,, \text{ where } f(\theta) \equiv \mathbb{E}_{\omega \sim \mathbb{P}} \left[F(\theta, \omega) \right] \,. \tag{1}$$

In (1), the $\mathbb{R}^d \times \Omega \mapsto \mathbb{R}$ mapping $F(\cdot, \cdot)$ evaluated at (θ, ω) represents one *noisy* observation of $f(\theta)$ when a realization $\omega \sim \mathbb{P}$ is drawn from Ω. Evaluating these zeroth-order (ZO) queries $F(\cdot, \cdot)$ are usually *expensive*. Under this setup, the generic SA recursions are widely applied:

$$\hat{\theta}_{k+1} = \begin{cases} \hat{\theta}_k - a_k \hat{\boldsymbol{g}}_k \,, & \text{stochastic 1st-order method using ZO oracles,} \quad (2a) \\ \hat{\theta}_k - a_k \hat{\boldsymbol{H}}_k^{-1} \hat{\boldsymbol{g}}_k \,, & \text{stochastic 2nd-order method using ZO oracles,} \quad (2b) \end{cases}$$

where $\hat{\theta}_k$ denotes the recursive estimate at the kth iteration, a_k is positive stepsize (gain), $\hat{\boldsymbol{g}}_k$ represents an estimate for the gradient function $\boldsymbol{g}(\theta) \equiv \nabla f(\theta)$ evaluated at $\hat{\theta}_k$, and $\hat{\boldsymbol{H}}_k$ represents an estimate for the Hessian function $\boldsymbol{H}(\theta) \equiv \nabla^2 f(\theta)$ evaluated at $\hat{\theta}_k$. Except for the SA scheme (2), random search (including stochastic ruler, stochastic comparison, simulated annealing) is also useful in solving (1), but it will not be our focus.

1.1. Why Do We Care Second-Order SA in Black-Box Problems?

Setting the subtleties relating to both the noisy ZO oracles and the stepsizes aside, using SA algorithms (2) to solve (1) in part stems from the localized model $[f(\theta) + \boldsymbol{d}^T \boldsymbol{g}(\theta) + \boldsymbol{d}^T \boldsymbol{B}(\theta) \boldsymbol{d}/2]$

constructed within the region $\{\theta + \boldsymbol{d} : \|\boldsymbol{d}\| \leq \delta\}$, where $\boldsymbol{B}(\cdot)$ is a curvature matrix. In a nutshell, letting $\boldsymbol{B}(\theta) = \mathrm{L}_2\boldsymbol{I}$ (where L_2 *upper-bounds* the spectral norm of the second-order information $\boldsymbol{H}(\cdot)$ throughout this neighborhood) and $\boldsymbol{B}(\theta) = \boldsymbol{H}(\theta)$ inside that neighborhood motivate (2a) and (2b), respectively.

Currently, (2a) remains dominant over (2b), mainly because (2b) suffers from (i) the model-trust region/radius issue and (ii) expensive computational cost. Issue (i) becomes obvious when (2b) is naively implemented around a region with negative curvature for *nonconvex* loss functions. On the contrary, issue (i) does not arise in (2a) because the conservative localized model with $\boldsymbol{B}(\theta) = \mathrm{L}_2\boldsymbol{I}$ enforces *extremely small* updates. Besides, large parameter space d and/or large dataset size I renders (2b) infeasible in most big-data applications. The storage, update, and inversion of the d^2-entry curvature matrix $\boldsymbol{B}(\theta)$ across a total of I data points can be prohibitive[1] as in issue (ii). Fortunately, using the damping techniques in Levenberg-Marquardt method reviewed in Remark 1, issue (i) can be largely overcome. To address issue (ii), researchers may impose low-rank, diagonal, or other structures onto the curvature matrix so as to store, compute, and invert $\boldsymbol{B}(\cdot)$ easily.

Remark 1 *Computing* $\arg\min_{\boldsymbol{d}:\|\boldsymbol{d}\|\leq\delta} [f(\theta) + \boldsymbol{d}^T\boldsymbol{g}(\theta) + \boldsymbol{d}^T\boldsymbol{B}(\theta)\boldsymbol{d}/2]$ *is equivalent to computing* $\arg\min_{\boldsymbol{d}\in\mathbb{R}^d} [f(\theta) + \boldsymbol{d}^T\boldsymbol{g}(\theta) + \boldsymbol{d}^T(\boldsymbol{B}(\theta) + \epsilon\boldsymbol{I})\boldsymbol{d}/2] \equiv -(\boldsymbol{B}(\theta) + \epsilon\boldsymbol{I})^{-1}\boldsymbol{g}(\theta)$. *Although* ϵ *is a complicated function of* δ*, we can* directly *work with* ϵ *for convenience.*

After a sharp decline during early iterations, the estimates from (2a) suffer from slow convergence rate in later iterations. When $\hat{\theta}_k$ reaches the vicinity of θ^*, the algorithmic scheme (2b) offers multiple edges: (a) the resulting $\hat{\theta}_k$ remains intact under *linear*[2] mappings imposed upon θ; (b) they eliminate[3] the need for tuning *some* hyper-parameters; (c) the estimates enjoy faster convergence when the iterate $\hat{\theta}_k$ is close to the optimum defined as θ^* in (1) where the underlying loss function $f(\cdot)$ is *locally quadratic*; (d) local curvature exploitation (preconditioning) makes the loss surface more isotropic[4] and mitigate the ill-conditioning effects.

1.2. Perspectives on Randomness and Data

In anticipation of Sect. 1.3, Sect. 2.2, and Sect. 4.2, we comment on the randomness $\Omega \ni \omega \sim \mathbb{P}$ in (1) that inevitably arise when the function measurements are collected from either physical experiments or computer simulation.

Data-Stream (Fresh Samples) For online learning where fresh sample arrives sequentially, $\omega \in \Omega$ may represent some noises imposed on the underlying loss. When data come in a streaming fashion, it is natural to assume that all the sample points ω's are independently drawn from Ω and identically distributed according to the distribution $\mathbb{P}$.

[1] Large I makes it costly to evaluate Hessian, and large d makes it costly to invert the Hessian. Each Newton iteration requires $O(Id^2)$ to evaluate the *exact* Hessian of all the collected data points and $O(d^3)$ to invert it.

[2] For example, the value for θ changes with the unit of measurement or the choice of coordinate system.

[3] For (2a) using ZO queries, the gain sequence in the form of $a_k = a/k$ must satisfy $a > [3\lambda_{\min}(\boldsymbol{H}(\theta^*))]^{-1}$ (Spall, 1992) in order to attain the fastest convergence rate of $O(k^{-1/3})$. Second-order methods (2b) can achieve the *optimum* rate *without* knowing the minimum eigenvalue of $\boldsymbol{H}(\theta^*)$.

[4] In stochastic optimization, the asymptotic behaviors of the stochastic optimizer only cares about how many data you've seen. The optimization problem becomes an estimation problem. The asymptotic convergence behavior kicks in sooner when second-order information are approximated, since navigating the curved loss landscape stops being the bottleneck—but in well-conditioned problems it's already not the bottleneck.

Fixed Dataset (Collected Samples) For empirical risk minimization (ERM) where only a fixed number (say I) of (training) samples[5] are available, it is standard practice to *reuse* the given I samples using a (mini-)batch size of J ($1 \leq J \leq I$). When sampling *with* replacement is applied, all the J-elements subsets of $\{1, \cdots, I\}$ constitute Ω, and $\mathbb{P}$ places a probability mass of ${}^{J!(I-J)!}/{}_{I!}$ on each $\omega \in \Omega$. Unlike sampling *without* replacement (exhausting every sample within one epoch), this scenario where $\omega \in \Omega$ following uniform distribution $\mathbb{P}$ can still fit into (1).

1.3. Prior Work on Second-Order SA Using ZO Queries

Second-order estimation using ZO oracles can be traced back to Fabian (1971) which requires $O(d^2)$ ZO queries per iteration. A simultaneous perturbation version of the Hessian estimate, which costs *four* ZO queries per iteration was proposed in Spall (2000). Later, a similar estimation form using *three* ZO queries per iteration was considered in Prashanth et al. (2016), but it entails several contrived constants. We note that Martens and Grosse (2015); Wang et al. (2017); Agarwal et al. (2019) use first-order oracle, Agarwal et al. (2017) uses Hessian–vector-product oracle, and Sohl-Dickstein et al. (2014); Byrd et al. (2016); Saab and Shen (2019) use second-order oracle. Here we assume that only noisy ZO oracle is available, so methods not solely using ZO oracles are beyond the scope of our comparison.

Core Budget Indicator For problem (1), the *ZO query complexity* (to achieve certain level of accuracy) is usually the *key* budget indicator. When the number of ZO queries per iteration is fixed, the *iteration/runtime complexity* may be used interchangeably with oracle complexity in performance measure. Besides, the floating-point-operations (FLOPs) per iteration may also be important for experimenters in high-dimensional problems, see Zhu et al. (2020).

Convergence Notion When the loss function in ERM is composed of a finite number of summands, notions of convergence and rates of convergence are in line with those in deterministic optimization. For example, in Johnson and Zhang (2013); Martens and Grosse (2015); Byrd et al. (2016); Sohl-Dickstein et al. (2014); Schraudolph et al. (2007), rates of convergence are linear or quadratic as a measure of *iteration-to-iteration improvement* in the empirical risk function. In contrast, we follow the traditional SA notion, including applicability to general noisy loss functions (discussed in Sect. 1.2), and stochastic notions of convergence and rates of convergence based on *sample-points (almost surely, a.s.)* and convergence in distribution.

1.4. Overview and Contribution

The remainder of this paper is organized as follows. Sect. 2 conveys the motivation behind the newly-proposed Hessian approximation based on Stein's identity and presents implementation details. Sect. 3 provides theoretical justification for (2b) using damped Hessian estimate. Sect. 4 illustrates the numerical performance of this algorithm in comparison with 2SPSA. Sect. 5 includes some concluding remarks and envisions multiple further directions. Before proceeding, let us outline the key difference between ours and 2SPSA.

[5] Although the experimenter only has access to the *training* error (based on the fixed I samples) in black-box problems, the ultimate goal for ERM should be minimizing the *generalization* error. Note that *deterministic* optimization applied on given dataset (deeming the empirical risk loss as a deterministic loss function) does *not* work in terms of the *generalization* performance.

1) On the basis that the fastest[6] convergence rate remains to be $O(k^{-1/3})$, our estimator to appear in (12d) requires *three* ZO queries only, whereas 2SPSA requires *four*. Moreover, (12d) requires generating *one* perturbation vector via Monte Carlo and tuning *one* differencing stepsize only, while 2SPSA requires *two* statistically independent perturbations and *two* differencing stepsizes.

2) The smoothing rate w_k in (10b) is allowed to decrease as long as A.4 is met, whereas 2SPSA considers direct averaging only.

3) Our Hessian estimator is symmetric by construction, while 2SPSA in Spall (2000) requires an additional step to manually symmetrize its Hessian estimate. Moreover, our estimator has a rather clean and elegant form compared to 2RDSA in Prashanth et al. (2016).

4) The restriction of "four-times continuously differentiable with bounded fourth-order derivatives" in 2SPSA is relaxed to "thrice continuously differentiable with Lipschitz-continuous third-order derivatives."

5) Based on Stein's identity, the motivation behind our gradient/Hessian estimators and the corresponding derivation for bias/variance are simplified compared to 2SPSA.

2. Motivation, Description, and Implementation

2.1. Stein's Identity Motivates Our Gradient/Hessian Estimators

The key to implement (2) is to construct efficient estimators $\hat{\boldsymbol{g}}_k$ and $\hat{\boldsymbol{H}}_k$. Both Spall (2000) and Prashanth et al. (2016) resort to the fundamental Taylor's theorem[7], and they require "four-times continuously differentiable with bounded fourth-order derivatives" to achieve the optimal convergence rate of $O(k^{-1/3})$. On the contrary, Stein's identity allows us to construct gradient/Hessian estimators for a general class of smooth functions, and enables the optimal convergence rate possible when the underlying function is "thrice continuously differentiable with Lipschitz-continuous third-order derivatives." The following proposition describes the basic Stein's identity. Note Erdogdu (2015) also uses Stein's identity to estimate Hessian, but their work is restricted for *linear* predictor with Gaussian data.

Proposition 1 (First-Order and Second-Order Stein's Identity) *(Stein et al., 2003) Let $\boldsymbol{X} \in \mathbb{R}^d$ be a d-dimensional random vector whose underlying density function is $p(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$.*

i) Assume that the density function $p(\boldsymbol{x})$ is differentiable. *Let $q : \mathbb{R}^d \mapsto \mathbb{R}$ be a differentiable function such that $\mathbb{E}[\nabla q(\boldsymbol{X})]$* exists. *Then*

$$\mathbb{E}\left\{q(\boldsymbol{X})[p(\boldsymbol{X})]^{-1}\nabla p(\boldsymbol{X})\right\} = -\mathbb{E}[\nabla q(\boldsymbol{X})]\,. \tag{3}$$

ii) Assume that the density function $p(\boldsymbol{x})$ is twice differentiable. *Let $q : \mathbb{R}^d \mapsto \mathbb{R}$ be a twice differentiable function such that $\mathbb{E}[\nabla^2 q(\boldsymbol{X})]$* exists. *Then*

$$\mathbb{E}\left\{q(\boldsymbol{X})[p(\boldsymbol{X})]^{-1}\nabla^2 p(\boldsymbol{X})\right\} = \mathbb{E}[\nabla^2 q(\boldsymbol{X})]\,. \tag{4}$$

[6] Note that $O(k^{-1/3})$ in terms of root mean squared (RMS) error $(\mathbb{E}[\|\hat{\theta}_k - \theta^*\|^2])^{1/2}$ is the *fastest* rate possible for $a_k = O(k^{-1})$ and $c_k = O(k^{-1/6})$, when $f(\cdot)$ is thrice continuously differentiable and is not quadratic.

[7] Note that under the same condition as in 2SPSA and following the proofs therein will not give us the fastest convergence rate.

For the special case of multivariate standard normal vector $\boldsymbol{X} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{I})$, we have $\nabla p(\boldsymbol{x}) = -\boldsymbol{x}p(\boldsymbol{x})$ and $\nabla^2 p(\boldsymbol{x}) = (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{I})p(\boldsymbol{x})$. In this case, (3) and (4) reduce to

$$\begin{cases} \mathbb{E}[\boldsymbol{X}q(\boldsymbol{X})] = \mathbb{E}[\nabla q(\boldsymbol{X})]\,, \\ \mathbb{E}[(\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{I})q(\boldsymbol{X})] = \mathbb{E}[\nabla^2 q(\boldsymbol{X})]\,, \end{cases} \quad \text{when } \boldsymbol{X} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{I})\,.$$

For the case of $\boldsymbol{X} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$, we have $\nabla p(\boldsymbol{x}) = -\boldsymbol{\Sigma}^{-1}\boldsymbol{x}p(\boldsymbol{x})$, $\nabla^2 p(\boldsymbol{x}) = \left(\boldsymbol{\Sigma}^{-1}\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\right)p(\boldsymbol{x})$. In this case, (3) and (4) reduce to

$$\begin{cases} \mathbb{E}[\boldsymbol{\Sigma}^{-1}\boldsymbol{X}q(\boldsymbol{X})] = \mathbb{E}[\nabla q(\boldsymbol{X})]\,, \\ \mathbb{E}\left[\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\right)q(\boldsymbol{X})\right] = \mathbb{E}\left[\nabla^2 q(\boldsymbol{X})\right]\,, \end{cases} \quad \text{when } \boldsymbol{X} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})\,.$$

Remark 2 *The distribution for the* absolutely continuous *d-dimensional random vector $\boldsymbol{X}$ is allowed to come from the exponential family (with continuous density function) per Hudson et al. (1978), and the family of elliptical distributions (i.e., spherical distribution, hyperbolic distribution, logistic distribution, multivariate Laplace distribution, multivariate t-distribution) per Landsman and Nešlehová (2008).*

Remark 3 *Teerapabolarn (2013) extends (3–4) to several* discrete *distributions, including the binomial (Rademacher) and the Poisson distributions.*

Let us introduce the smoothed loss function $f_c(\theta)$ constructed through the convolution of the underlying loss function $f(\cdot)$ and the Gaussian kernel. The upcoming gradient/Hessian estimator with perturbation magnitude c will be the *unbiased* estimator for the gradient/Hessian of the *smoothed* loss function $f_c(\cdot)$. Even though $f(\cdot)$ may *not* be continuous, $f_c(\cdot)$ is *infinitely many times differentiable*, inheriting from the Gaussian kernel function.

Lemma 1 (Stein's Identity Based Estimator) *Let $\boldsymbol{u} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{I})$. Let $f_c(\theta) \equiv \mathbb{E}_{\boldsymbol{u}}[f(\theta + c\boldsymbol{u})]$ be the* smoothed *loss function. Assume the left-hand-sides of (5–9)* exist*, we have the following estimators.*

i) One-measurement estimator for $\nabla f_c(\theta)$ and $\nabla^2 f_c(\theta)$:

$$\mathbb{E}_{\boldsymbol{u}}\left[c^{-1}\boldsymbol{u}f(\theta + c\boldsymbol{u})\right] = \mathbb{E}_{\boldsymbol{u}}\left[\boldsymbol{g}(\theta + c\boldsymbol{u})\right] \equiv \nabla f_c(\theta)\,, \tag{5}$$

$$\mathbb{E}_{\boldsymbol{u}}\left[c^{-2}\left(\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I}\right)f(\theta + c\boldsymbol{u})\right] = \mathbb{E}_{\boldsymbol{u}}\left[\boldsymbol{H}(\theta + c\boldsymbol{u})\right] \equiv \nabla^2 f_c(\theta)\,. \tag{6}$$

ii) Two-measurements estimator for $\nabla f_c(\theta)$ and $\nabla^2 f_c(\theta)$:

$$\nabla f_c(\theta) = \begin{cases} \mathbb{E}_{\boldsymbol{u}}\left\{c^{-1}[f(\theta + c\boldsymbol{u}) - f(\theta)]\boldsymbol{u}\right\}\,, & \text{(7a)} \\ \mathbb{E}_{\boldsymbol{u}}\left\{(2c)^{-1}[f(\theta + c\boldsymbol{u}) - f(\theta - c\boldsymbol{u})]\boldsymbol{u}\right\}\,, & \text{(7b)} \end{cases}$$

$$\nabla^2 f_c(\theta) = \begin{cases} \mathbb{E}_{\boldsymbol{u}}\left\{c^{-2}[f(\theta + c\boldsymbol{u}) - f(\theta)](\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I})\right\}\,, & \text{(8a)} \\ \mathbb{E}_{\boldsymbol{u}}\left\{(2c^2)^{-1}[f(\theta + c\boldsymbol{u}) + f(\theta - c\boldsymbol{u})](\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I})\right\}\,. & \text{(8b)} \end{cases}$$

iii) Three-measurements estimator for $\nabla^2 f_c(\theta)$:

$$\nabla^2 f_c(\theta) = \mathbb{E}_{\boldsymbol{u}}\left\{(2c^2)^{-1}[f(\theta + c\boldsymbol{u}) + f(\theta - c\boldsymbol{u}) - 2f(\theta)](\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I})\right\}\,. \tag{9}$$

Although Lemma 1 does *not* provide an unbiased gradient/Hessian estimators for the true loss function $f(\cdot)$, they are instrumental to construct an *asymptotically unbiased* estimators in (11–12) with its bias vanishing at certain rate to appear in Lemmas 2–3.

2.2. Algorithmic Form

Recall that the model-trust issue discussed in Sect. 1.1 can be handled by Levenberg-Marquardt *damping* technique per Remark 1. Therefore, we modify (2b) as follows:

$$\begin{cases} \hat{\theta}_{k+1} = \hat{\theta}_k - a_k[\boldsymbol{p}_k(\overline{\boldsymbol{H}}_k)]^{-1}\hat{\boldsymbol{g}}_k\,, & (10a) \\ \overline{\boldsymbol{H}}_{k+1} = \overline{\boldsymbol{H}}_k - w_k(\overline{\boldsymbol{H}}_k - \hat{\boldsymbol{H}}_k)\,, & (10b) \end{cases}$$

for $k \geq 0$. The general adaptive SA algorithm (10) is comprised of two recursions: (10a) estimates θ via a stochastic analogue of the Newton method, and (10b) produces estimate for $\boldsymbol{H}(\cdot)$ through a weighted average of the seen Hessian estimates $\hat{\boldsymbol{H}}_k$'s. The stepsize w_k governing the smoothing rate for Hessian estimate places a crucial role in the convergence of (10), and one common choice[8] is $w_k = {}^{1}\!/_{(k+1)}$ for $k \geq 0$. When $w_0 > 0$, the initialization $\overline{\boldsymbol{H}}_0$ can be a scaling matrix, so that *early iteration* of (10) resembles that of (2a). In (10), $\boldsymbol{p}_k(\cdot)$ is a mapping from $\mathbb{R}^{d\times d}$ to the set of symmetric and positive definite matrices. The straightforward form for $\boldsymbol{p}_k$ is the approximate damped Newton (see Remark 1): $\boldsymbol{p}_k(\overline{\boldsymbol{H}}_k) = \overline{\boldsymbol{H}}_k + \epsilon_k\boldsymbol{I}$, for $\epsilon_k > \lambda_{\min}(\overline{\boldsymbol{H}}_k)$. Another popular mapping is $\boldsymbol{p}_k(\overline{\boldsymbol{H}}_k) = (\overline{\boldsymbol{H}}_k^T\overline{\boldsymbol{H}}_k + \epsilon_k\boldsymbol{I})^{1/2}$ for $\epsilon_k \to 0$, where the square root here is the unique positive definite square root (implementable via `sqrtm` in MATLAB).

For succinctness, write $F_k^{\pm} \equiv F(\hat{\theta}_k \pm c_k\boldsymbol{u}_k, \omega_k^{\pm})$ and $F_k \equiv F(\hat{\theta}_k, \omega_k)$. Write $\varepsilon_k^{\pm} \equiv F_k^{\pm} - f(\hat{\theta}_k \pm c_k\boldsymbol{u}_k)$ and $\varepsilon_k \equiv F(\hat{\theta}_k, \omega_k) - f(\hat{\theta}_k)$. We focus on the uncontrolled[9] noise scenario where $\omega_k^{\pm}$ and ω_k are i.i.d. Since only noisy evaluation of the loss function is accessible, the gradient estimate can be as (11) using one or two noisy ZO oracles

$$\hat{\boldsymbol{g}}_k = \begin{cases} c_k^{-1}F_k^{+}\boldsymbol{u}_k\,, & (11a) \\ c_k^{-1}(F_k^{+} - F_k)\boldsymbol{u}_k\,, & (11b) \\ (2c_k)^{-1}(F_k^{+} - F_k^{-})\boldsymbol{u}_k\,, & (11c) \end{cases}$$

and the Hessian estimate can be as (12) using one to three noisy ZO oracles

$$\hat{\boldsymbol{H}}_k = \begin{cases} c_k^{-2}F_k^{+}(\boldsymbol{u}_k\boldsymbol{u}_k^T - \boldsymbol{I})\,, & (12a) \\ c_k^{-2}(F_k^{+} - F_k)(\boldsymbol{u}_k\boldsymbol{u}_k^T - \boldsymbol{I})\,, & (12b) \\ (2c_k^2)^{-1}(F_k^{+} + F_k^{-})(\boldsymbol{u}_k\boldsymbol{u}_k^T - \boldsymbol{I})\,, & (12c) \\ (2c_k^2)^{-1}(F_k^{+} + F_k^{-} - 2F_k)(\boldsymbol{u}_k\boldsymbol{u}_k^T - \boldsymbol{I})\,. & (12d) \end{cases}$$

The Hessian estimators (12) are guaranteed to be symmetric. This contrasts with 2SPSA (Spall, 2000) where manual symmetrization is required.

2.3. Implementation Aspects

The entire second-order algorithms (10) is summarized in Algorithm 1.

$\hat{\theta}_0$ initialization Following the last paragraph in Sect. 1.1, it is recommended to first run the first-order scheme (2a), and then switch to the second-order scheme (2b) after $\hat{\theta}_k$ has reached the vicinity of θ^* where the loss function is *locally quadratic*.

[8] This is the stepsize enforced in 2SPSA. Generally, we allow w_k to go to zero at a rate such that $\sum_k w_k^2 c_k^{-4} < \infty$, see A.4 to appear.

[9] For the controllable noise scenario where $\omega_k^{\pm} = \omega_k$, frequently encountered in simulation optimization or the fixed-dataset discussed in Sect. 1.2.

Algorithm 1 Second-Order SA Using Normal Perturbation in Black-Box Problems

```
Input: initialization $\hat{\theta}_0$, $\overline{\boldsymbol{H}}_0 = \boldsymbol{I}$; coefficients $a_k, w_k$ and $c_k$ for $0 \le k \le K$.
set iteration counter $k = 0$.
for $k = 0, 1, \cdots, K$ do
    generate $\boldsymbol{u}_k \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ and collect noisy loss observations.
    estimate $\hat{\boldsymbol{g}}_k$ via (11) and $\hat{\boldsymbol{H}}_k$ via (12).
    update $\hat{\theta}_{k+1}$ and $\overline{\boldsymbol{H}}_{k+1}$ via (10).
end
Output: terminal estimate $\hat{\theta}_{K+1}$ (or iterate average).
```

Positive-definite mapping The crucial step before applying second-order estimation to the parameter update for $\hat{\theta}_k$ within an optimization context is enforcing the Hessian estimate to be positive-definite. Note that the estimation errors in $\hat{\boldsymbol{H}}_k$ *may* result in negative eigenvalues of $\overline{\boldsymbol{H}}_k$, which is inevitable due to the noisy ZO oracles. Generally, it costs $O(d^3)$ FLOPs to guarantee a symmetric matrix to be positive-definite (Zhu and Spall, 2002). Zhu et al. (2020) proposed to utilize the symmetric indefinite matrix decomposition to reduce the per-iteration FLOPs to $O(d^2)$.

Gradient/Hessian averaging It is desirable to average several $\hat{\boldsymbol{H}}_k$ and $\hat{\boldsymbol{g}}_k$ values despite the additional query cost, especially in *high-noise* environment.

Blocking We may enforce to set $\hat{\theta}_{k+1} = \hat{\theta}_k$ if the evaluation of the noisy evaluation at $\hat{\theta}_{k+1}$ is substantially higher than that at $\hat{\theta}_k$ by a user-specified constant.

2.4. Notation Conventions

Matrix and vector operations Let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ be a matrix and $\boldsymbol{x} \in \mathbb{R}^d$ be a vector. $\|\boldsymbol{x}\|$ returns the Euclidean norm of $\boldsymbol{x}$, and $\|\boldsymbol{A}\|$ returns the spectral norm of $\boldsymbol{A}$. If all eigenvalues of $\boldsymbol{A}$ are real, $\lambda_{\min}(\boldsymbol{A})$ returns its smallest eigenvalue. $\boldsymbol{A} \succ \boldsymbol{0} (\boldsymbol{A} \succeq \boldsymbol{0})$ means $\boldsymbol{A}$ is symmetric and positive definite (semi-definite).

Probability Let $\mathcal{F}_k$ represents the history[10] of the process (2). If $\hat{\theta}_0$ is random, $\mathcal{F}_0$ is spanned by $\hat{\theta}_0$; if otherwise, $\mathcal{F}_0$ is trivial. Note that $\hat{\theta}_{k+1}$ is a random variable that depends on the filtration $\mathcal{F}_k$ generated by the recursive algorithm (2) before $\hat{\theta}_{k+1}$ is realized. Let $\mathbb{E}_k(\cdot)$ denote the conditional expectation $\mathbb{E}[\,\cdot\,|\,\mathcal{F}_k]$. "Independent and identically distributed" is abbreviated as i.i.d. "Infinitely often" is abbreviated as i.o. "Almost surely" is abbreviated as a.s. The equality of two random variables almost surely *may* be written as equality for clarity.

Miscellaneous The binary operator $\otimes$ denotes the Kronecker product. In addition to $\boldsymbol{g}(\theta) \equiv \nabla f(\theta) \in \mathbb{R}^{d\times 1}$ and $\boldsymbol{H}(\theta) \equiv \nabla^2 f(\theta) \in \mathbb{R}^{d\times d}$, we also let $\nabla^3 f(\theta) \in \mathbb{R}^{1\times d^3}$ (as a *row* vector) be the third-order derivative of the loss function $f(\cdot)$.

[10] The precise definition for the filtration $\mathcal{F}_k$ may vary, depending on the estimator forms (11–12) and the corresponding ZO queries.

3. Convergence Theory

For clarity and for the reason explained in Remarks 4–5, we analyze $\hat{\boldsymbol{g}}_k$ in (11c) (using two ZO queries) and $\hat{\boldsymbol{H}}_k$ as (12d) using three ZO queries subsequently.

3.1. Model Assumptions

We present conditions under which $\hat{\theta}_k \to \theta^*$ and $\overline{\boldsymbol{H}}_k \to \boldsymbol{H}(\theta^*)$ almost surely as $k \to \infty$. Write the "conditioned" gradient estimate as $\overline{\boldsymbol{g}}_k \equiv [\boldsymbol{p}_k(\overline{\boldsymbol{H}}_k)]^{-1}\boldsymbol{g}(\hat{\theta}_k)$.

Assumption A.1 (Conditions on $f(\cdot)$) *The loss function $f(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$ is thrice continuous differentiable, and $\nabla^3 f(\cdot)$ is L_3-Lipschitz continuous*[11] *and bounded.*

Assumption A.2 (Noisy ZO Queries) *For the gradient-estimator (11c), assume $\mathbb{E}[\,\varepsilon_k^+ - \varepsilon_k^- \,|\, \hat{\theta}_k, \boldsymbol{u}_k] = 0$ and $\mathbb{E}[\,(\varepsilon_k^+ - \varepsilon_k^-)^2 \,|\, \hat{\theta}_k, \boldsymbol{u}_k]$ is uniformly bounded for all k. For the Hessian-estimator (12d), assume $\mathbb{E}[\,\varepsilon_k^+ + \varepsilon_k^- - 2\varepsilon_k \,|\, \hat{\theta}_k, \boldsymbol{u}_k] = 0$ and $\mathbb{E}[\,(\varepsilon_k^+ + \varepsilon_k^- - 2\varepsilon_k)^2 \,|\, \hat{\theta}_k, \boldsymbol{u}_k]$ is uniformly bounded for all k.*

Assumption A.3 (Random Perturbation) *$\boldsymbol{u}_k \overset{\text{i.i.d.}}{\sim} \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{u}_k$ is independent of $\mathcal{F}_k$. Moreover, both $\mathbb{E}\{[f(\hat{\theta}_k \pm c_k \boldsymbol{u}_k)]^2\}$ and $\mathbb{E}\{[f(\hat{\theta}_k)]^2\}$ are uniformly bounded.*

Assumption A.4 (Stepsizes) *The positive stepsizes satisfy $a_k, w_k, c_k \overset{k\to\infty}{\longrightarrow} 0$, $\sum_k a_k = \infty$, $\sum_k (a_k/c_k)^2 < \infty$, and $\sum_k (w_k/c_k^2)^2 < \infty$.*

Assumption A.5 $\limsup_{k\to\infty} \|\hat{\theta}_k\| < \infty$ *a.s. Moreover, for any $\rho > 0$ and for all $i \in \{1, \cdots, d\}$,*

$$\mathbb{P}\left(E_k \text{ i.o.}\right) = 0 \textit{ where the event } E_k \equiv \left\{ \operatorname{sign}(\overline{g}_{k,i}) \neq \operatorname{sign}(g_i(\hat{\theta}_k)) \textit{ and } |(\hat{\theta}_k)_i - (\theta^*)_i| \geq \rho \right\}. \tag{13}$$

Assumption A.6 (Sufficient Curvature) *For each k and for all θ, there exists a $\mathrm{C} > 0$ such that $(\hat{\theta}_k - \theta^*)^T \overline{\boldsymbol{g}}_k \geq \mathrm{C}\|\hat{\theta}_k - \theta^*\|$.*

Assumption A.7 (Conditions for $\boldsymbol{p}_k(\cdot)$) *$c_k^2[\boldsymbol{p}_k(\overline{\boldsymbol{H}}_k)]^{-1} \to \boldsymbol{0}$ almost surely as $k \to \infty$. For some $\delta > 0$, $\mathbb{E}[\|(\boldsymbol{p}_k(\hat{\boldsymbol{H}}_k))^{-1}\|^{2+\delta}]$ is bounded uniformly for all k. $\|\boldsymbol{p}_k(\overline{\boldsymbol{H}}_k) - \overline{\boldsymbol{H}}_k\| \to 0$ almost surely as $k \to \infty$.*

Comments on assumptions A.2–A.3 are standard in SA algorithms. $\sum_k w_k^2 c_k^{-4} < \infty$ in A.4 is to ensure the bounded variance of the Hessian estimator. Kushner and Clark (2012, pp. 40–41) explains why "$\limsup_{k\to\infty} \|\hat{\theta}_k\| < \infty$ a.s." in A.5 is not a stringent condition and could be expected to hold in most applications. Condition (13) in A.5 ensures that $\hat{\theta}_k$ cannot be bouncing around to cause the signs of the conditioned gradient elements to be changing infinitely often when $\hat{\theta}_k$ is strictly bounded away from θ^*. A.1 and A.6 ensure the smoothness[12] and steepness of $f(\cdot)$. Note that experimenter has full control over the stepsize and the mapping, A.4 and A.7 can be met easily.

[11] It means that $\|\nabla^3 f(\theta) - \nabla^3 f(\zeta)\| \leq \mathrm{L}_3\|\theta - \zeta\|$, where the third-order derivative $\nabla^3 f(\cdot)$ and the Euclidean norm $\|\cdot\|$ were defined in Subsection 2.4.

[12] Note that 2SPSA requires four-times continuously differentiable with bounded fourth-order derivatives. We change it to thrice continuously differentiable with Lipschitz-continuous third-order derivatives.

3.2. Asymptotic Unbiasedness

Let us leverage Lemma 1 to construct an asymptotically unbiased gradient and Hessian estimators. This is instrumental in establishing convergence and achieving optimal convergence rate later on.

Lemma 2 (Bias/Variance in Gradient Estimate) *Under A.1–A.3, the gradient estimators (11) satisfy* $\mathbb{E}_k(\hat{\boldsymbol{g}}_k) \overset{\text{a.s.}}{=} \boldsymbol{g}(\hat{\theta}_k) + O(c_k^2)$. *Furthermore,* $\mathbb{E}_k(\|\hat{\boldsymbol{g}}_k\|^2) \overset{\text{a.s.}}{=} O(c_k^{-2})$.

Remark 4 *Despite Lemma 2 applies to all three estimators in (11), it is recommended to use (11c) over (11a–11b). The rationale behind it is discussed in Appendix A.*

Lemma 3 (Bias/Variance in Hessian Estimate) *Under A.1–A.3, the Hessian estimator (12) satisfy* $\mathbb{E}_k(\hat{\boldsymbol{H}}_k) \overset{\text{a.s.}}{=} \boldsymbol{H}(\hat{\theta}_k) + O(c_k^2)$ *and* $\mathbb{E}_k(\|\hat{\boldsymbol{H}}_k\|^2) \overset{\text{a.s.}}{=} O(c_k^{-4})$.

Remark 5 *For the same reason in Remark 4, it is recommended to use (12d) over (12a–12c).*

3.3. Almost Sure Convergence

With the gain sequence properly weighting the bias and variance in the gradient and Hessian estimators, we can establish the almost surely convergence $\hat{\theta}_k \xrightarrow{\text{a.s.}} \theta^*$ as $k \to \infty$.

Theorem 1 (Strong Convergence of Parameter) *Under A.1–A.7,* $\hat{\theta}_k \to \theta^*$ *a.s. as* $k \to \infty$.

Theorem 2 (Strong Convergence of Hessian) *Under A.1–A.7,* $\overline{\boldsymbol{H}}_k \to \boldsymbol{H}(\theta^*)$ *a.s. as* $k \to \infty$, *where* $\overline{\boldsymbol{H}}_k$ *is governed by (10b) and* $\hat{\boldsymbol{H}}_k$ *is computed per (12d).*

3.4. Rate of Convergence

Let us find the convergence rate here. We enforce the form of the stepsizes: $a_k = {}^{a}\!/_{(k+1)^\alpha}$, $c_k = {}^{c}\!/_{(k+1)^\gamma}$, and w_k satisfies A.4. Let $\tau \equiv \alpha - 2\gamma$ and $\tau_+ \equiv \tau \cdot \mathbb{I}_{\{\alpha=1\}}$

Theorem 3 (Asymptotic Normality) *Assume A.1–A.7 hold. Pick* $a > {}^{\tau_+}\!/_{[2\lambda_{\min}\{\boldsymbol{H}(\theta^*)\}]}$ *and* $\alpha \le 6\gamma$, *we have*

$$k^{\tau/2}(\hat{\theta}_k - \theta^*) \xrightarrow{\text{dist.}} \mathrm{N}(\mu, \boldsymbol{\Lambda}), \tag{14}$$

where

$$\begin{cases} \mu = \dfrac{ac^2}{3\tau_+ - 6a}[\boldsymbol{H}(\theta^*)]^{-1}\mathbb{E}[f^{(3)}(\theta^*)(\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u})\boldsymbol{u}], & \text{(15a)} \\ \boldsymbol{\Lambda} = \dfrac{a^2\mathrm{Var}[F(\theta^*,\omega)]}{2c^2(2a-\tau_+)}[\boldsymbol{H}(\theta^*)]^{-2}, & \text{(15b)} \end{cases}$$

with $\boldsymbol{u} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$.

Remark 6 *Note that when* $\omega \sim \mathbb{P}$ *is drawn i.i.d. from* Ω, *we have* $\mathbb{E}[(\varepsilon_k^+ - \varepsilon_k^-)^2 \,|\, \hat{\theta}_k, \boldsymbol{u}_k] \to 2\mathrm{Var}[F(\theta^*,\omega)]$ *a.s., where the variance is taken over* $\omega \in \Omega$. *This is due to* $\hat{\theta}_k \xrightarrow{\text{a.s.}} \theta^*$ *shown in Theorem 1 and* $c_k \to 0$ *assumed in A.4.*

Remark 7 *Although the possible fastest rate remains to be* $O(k^{-1/3})$ *for* $a_k = O(k^{-1})$, $c_k = O(k^{-1/6})$, *and* w_k *such that* $\sum_k w_k^2 c_k^{-4} < \infty$, *(12d) only requires* three *ZO queries to form one single Hessian estimate whereas* 2SPSA *requires* four *ZO queries. Moreover, (12d) only requires generating* one *perturbation* $\boldsymbol{u}_k$ *(via Monte Carlo) and differencing stepsize* c_k *(which requires hyper-parameter tuning), while* 2SPSA *requires* two *statistically independent perturbations and* two *differencing stepsizes (Zhu and Spall, 2020).*

3.5. Comparison with 2SPSA

We can now explain Sect. 1.4 in more details. Specifically, 2SPSA for per-iteration Hessian estimate requires sampling two random perturbations $\boldsymbol{\Delta}_k$ and $\widetilde{\boldsymbol{\Delta}}_k$ (commonly each component of the random perturbations are i.i.d. Rademacher distributed) and tuning two perturbation magnitudes c_k and $\widetilde{c}_k$. It has the following form:

$$\begin{cases} \hat{\boldsymbol{H}}_k \leftarrow (2c_k\widetilde{c}_k)^{-1}(F_k^{+,+} - F_k^{-,+} + F_k^{-} - F_k^{+})(\widetilde{\boldsymbol{\Delta}}_k^{-1}\boldsymbol{\Delta}_k^{-T}) \,, \\ \hat{\boldsymbol{H}}_k \leftarrow {}^1\!/_2(\hat{\boldsymbol{H}}_k + \hat{\boldsymbol{H}}_k^T) \,, \qquad \text{for symmetrization} \,, \end{cases} \tag{16}$$

where $F_k^{\pm} = F(\hat{\theta}_k \pm c_k\boldsymbol{\Delta}_k, \omega_k^{\pm})$ and $F_k^{\pm,+} = F(\hat{\theta}_k \pm c_k\boldsymbol{\Delta}_k + \widetilde{c}_k\widetilde{\boldsymbol{\Delta}}_k, \omega_k^{\pm,+})$, $\boldsymbol{\Delta}^{-1}$ indicates component-wise inverse[13]. Comparing (16) and (12d) side-by-side, we shall see the bullet-point comparison in Sect. 1.4, which is further summarized in Table 1. Note that the bias and variance for the gradient/Hessian estimates for 2SPSA requires the bound of the fourth-order derivative, while those in Lemma 2/3 requires the Lipschitz continuity constant of the third-order derivatives. Moreover, the fourth-order continuity in 2SPSA cannot be relaxed, as it is required in Taylor-expansion derivation.

	Optimal Rate*	Queries	Injected Randomness	Hyper-para	Weighting w_k
2SPSA	$O(k^{-1/3})$	4	Rademacher $\boldsymbol{\Delta}_k$, $\widetilde{\boldsymbol{\Delta}}_k$	c_k, $\widetilde{c}_k$	${}^1\!/_{k+1}$
Algo 1		3	Standard MVN $\boldsymbol{u}_k$	c_k	$\sum_k w_k^2 c_k^{-4} < \infty$

Table 1: Comparison between 2SPSA and Algorithm 1. * explained: the convergence rate in 2SPSA requires "four-times continuously differentiable with bounded fourth-order derivatives", while Algorithm 1 requires "thrice continuously differentiable with Lipschitz-continuous third-order derivatives."

To further dissect the big-O optimal rate in Table 1, we interpret Theorem 3 following Zhu (2020b, Sect. 3): $\lim_{k\to\infty}[\mathbb{E}(\|\hat{\theta}_k - \theta^*\|^2)]^{1/2} = k^{-\tau/2}\left[\|\boldsymbol{\mu}\|^2 + \mathrm{tr}(\boldsymbol{\Lambda})\right]$. The estimates generated by 2SPSA similarly satisfies $k^{\tau/2}(\hat{\theta}_k^{\text{2SPSA}} - \theta^*) \xrightarrow{\text{dist.}} \mathrm{N}(\boldsymbol{\mu}^{\text{2SPSA}}, \boldsymbol{\Lambda}^{\text{2SPSA}})$, where $\boldsymbol{\Delta}$ and $\boldsymbol{\Lambda}^{\text{2SPSA}}$ coincides with (15b). Let each component of $\boldsymbol{\Delta}$ be i.i.d. Rademacher distributed, then $\boldsymbol{\mu}^{\text{2SPSA}} = {}^{ac^2}\!/_{(3\tau_+ - 6a)}[\boldsymbol{H}(\theta^*)]^{-1}\mathbb{E}[f^{(3)}(\theta^*)\,(\boldsymbol{\Delta}\otimes\boldsymbol{\Delta}\otimes\boldsymbol{\Delta})\,\boldsymbol{\Delta}^{-1}]$. Unfortunately, it is difficult to make a concrete comparison between $\boldsymbol{\mu}$ in (15b) and $\boldsymbol{\mu}^{\text{2SPSA}}$, as it depends on the third-order derivative of the underlying *unknown* loss function $f(\cdot)$.

4. Numerical Experiment

To support that Stein's identity helps reducing the per-iteration ZO-queries, this section provides one synthetic illustration and one real-data example.

[13] Spall (2000) requires that both $\boldsymbol{\Delta}_k$ and $\tilde{\boldsymbol{\Delta}}_k$ have bounded inverse-second-moment. For Rademacher distributed random perturbation, component-wise inverse is well-defined as every component has zero probability of being zero.

4.1. Skewed-Quartic Function

This section compares Algorithm 1 with the counterpart 2SPSA. The loss function here is the skew-quartic function

$$f(\theta) = \theta^T \boldsymbol{A}^T \boldsymbol{A}\theta + 0.1\sum_{i=1}^{d}(\boldsymbol{A}\theta)_i^3 + 0.01\sum_{i=1}^{d}(\boldsymbol{A}\theta)_i^4 ,$$

where $(\cdot)_i$ is the ith component of the argument vector, and $\boldsymbol{A}$ is such that $d\boldsymbol{A}$ is an upper-triangular matrix of all ones. The additive noise in $F(\cdot)$ is independent $\mathrm{N}(0, 0.1)$, i.e., $F(\theta, \omega) = f(\theta) + \varepsilon(\omega)$, where $\varepsilon(\omega) \sim \mathrm{N}(0, 0.1)$. Though $\boldsymbol{H}(\theta) \succ {}^5\!/_4\boldsymbol{A}^T\boldsymbol{A}$, the Hessian function is ill-conditioned.

We use $d = 20$ and initialize $\hat{\theta}_0$ within $[-5, 5]^d$. Both algorithms are run with iteration $K = 10000$ with 50 replicates (i.e., all the performance will be averaged over 50 replicates). We use $c_k = {}^1\!/_{(k+1)^{0.101}}$ and $a_k = {}^1\!/_{(k+1+A)}{}^{0.602}$ and A equals 10% of the total iteration number K for both (10) and 2SPSA. For fair comparison, we use $w_k = {}^1\!/_{k+1}$ for both methods. During the implementation, both algorithms use exactly twelve[14] ZO queries per iteration, so that the query complexity *aligns* with the iteration complexity. We see from Figure 1 that Algorithm 1 is more query-efficient than 2SPSA, in terms of the attainable accuracy using the *same* ZO queries.

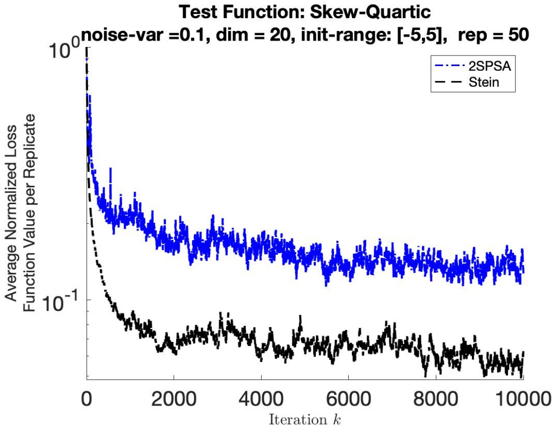

Figure 1: Performance of Algorithm 1 and 2SPSA in terms of normalized distance ${}^{[f(\hat{\theta}_k)-f(\theta^*)]}\!/_{[f(\hat{\theta}_0)-f(\theta^*)]}$ average across 50 independent replicates. Both algorithms use twelve ZO queries per iteration, so query complexity aligns with iteration complexity. The underlying loss function is the skew-quartic function with $d = 20$, and the noisy observation is corrupted by $\mathrm{N}(0, 0.1)$ noise.

4.2. Black-Box Binary Classification

We now apply both 2SPSA and the proposed algorithm to solve the black-box binary classification task Huang et al. (2020). The dataset PHISHING has $I = 11055$ samples $(\boldsymbol{x}_i, y_i)_{1\le i\le I}$, where the features $\boldsymbol{x}_i$ are 68-dimensional, and $y_i \in \{-1, 1\}$. Consider the following nonconvex corr-entropy

[14] Using twelve queries, four $\hat{\boldsymbol{H}}_k$ can be produced for Algorithm 1, and three $\hat{\boldsymbol{H}}_k$ can be produced for 2SPSA.

induced loss

$$f(\theta) = \frac{1}{I}\sum_{i=1}^{I} \frac{\kappa^2}{2}\left\{1-\exp\left[-\frac{(y_i - \boldsymbol{x}_i^T\theta)^2}{\kappa^2}\right]\right\}, \tag{17}$$

where θ is 68-dimensional, and the κ is some penalty parameter. Note that $f(\theta) = 0$ when all samples are correctly classified. The noisy observation $F(\theta, \omega)$ is

$$F(\theta, \omega) = \frac{1}{J}\sum_{j=1}^{J} \frac{\kappa^2}{2}\left\{1-\exp\left[-\frac{(y_{i_j(\omega)} - \boldsymbol{x}_{i_j(\omega)}^T\theta)^2}{\kappa^2}\right]\right\}, \tag{18}$$

for $J \leq I$, and the J indexes $\{i_1(\omega), \cdots, i_J(\omega)\}$ are i.i.d. uniformly drawn from $\{1, \cdots, I\}$ (without replacement as discussed in Sect. 1.2).

Consider (17) with penalty parameter $\kappa = 10$ and (18) with mini-batch size $J = 10$. Both 2SPSA and Algorithm 1 are initialized at a 68-dimensional vector of all ones. The ZO-query per iteration for both algorithm is twelve[15], so the query complexity *aligns* with the iteration complexity. We perform 25 independent replicates, each with $K = 5000$ iterations. Figure 2 shows the convergence of the two algorithms on the black-box binary classification problem. We see that Algorithm 1 is more query-efficient than 2SPSA in terms of the attainable accuracy using the *same* ZO queries.

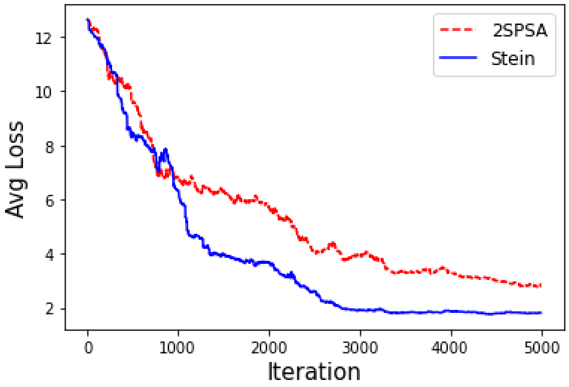

Figure 2: Performance of Algorithm 1 and 2SPSA in terms of the true loss function $f(\hat{\theta}_k)$ average across 25 independent replicates. Both algorithms use twelve ZO queries per iteration, so query complexity aligns with iteration complexity. A zero loss function is equivalent to 100% classification correctness.

5. Summary and Concluding Remarks

This work proposes (1) in solving the minimization problem when noisy ZO oracle is available. The key improvement from (12d) to 2SPSA is the reduced per-iteration ZO query number. Besides, ours

[15] With 12 queries, 2SPSA can construct the average of *three* Hessian estimators, as each of 2SPSA Hessian estimators costs 4 ZO-queries. Similarly, Algorithm 1 can construct the average of *four* Hessian estimators, each of which costs 3 queries.

and 2SPSA differ in the number of functions sampled, the sampling scheme for random perturbations, and the way in which the gradient/Hessian approximations are derived, as summarized in Sect. 1.4 and revisited in Sect. 3.5.

Several potential future work includes (i) extension to scenarios where unbiased direct measurements of $\boldsymbol{g}(\theta)$ are available; (ii) extension for possible distribution for $\boldsymbol{u}$, e.g., other continuous distributions per Remark 2 and discrete distributions per Remark 3; and (iii) leveraging importance sampling techniques in Monte Carlo sampling our estimators once the distribution for perturbation $\boldsymbol{u}$ is revealed.

Acknowledgment

The author would like to thank Dr. Zhenliang Zhang, Dr. Jian Tan, and Dr. Wotao Yin for inspirational discussion.

References

Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, 18(1):4148–4187, 2017.

Naman Agarwal, Brian Bullins, Xinyi Chen, Elad Hazan, Karan Singh, Cyril Zhang, and Yi Zhang. Efficient full-matrix adaptive regularization. In *International Conference on Machine Learning*, pages 102–110, 2019.

Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

Kai Lai Chung. *A course in probability theory*. Academic press, 2001.

Murat A Erdogdu. Newton-stein method: a second order method for glms via stein's lemma. In *Advances in Neural Information Processing Systems*, pages 1216–1224, 2015.

V Fabian. Stochastic approximation, optimization methods in statistics. In J. S. and Rustigi, editor, *Optimizing Methods in Statistics*, pages 439–470. Academic Press, New York, 1971.

Feihu Huang, Lue Tao, and Songcan Chen. Accelerated stochastic gradient-free and projection-free methods. In *International Conference on Machine Learning*, pages 4519–4530. PMLR, 2020.

H Malcolm Hudson et al. A natural identity for exponential families with applications in multiparameter estimation. *The Annals of Statistics*, 6(3):473–484, 1978.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Harold Joseph Kushner and Dean S Clark. *Stochastic approximation methods for constrained and unconstrained systems*, volume 26. Springer Science & Business Media, 2012.

Zinoviy Landsman and Johanna Nešlehová. Stein's lemma for elliptical random vectors. *Journal of Multivariate Analysis*, 99(5):912–927, 2008.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417, 2015.

LA Prashanth, Shalabh Bhatnagar, Michael Fu, and Steve Marcus. Adaptive system optimization using random directions stochastic approximation. *IEEE Transactions on Automatic Control*, 62 (5):2223–2238, 2016.

Samer S Saab and Dong Shen. Multidimensional gains for stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5):1602–1615, 2019.

Nicol N Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-newton method for online convex optimization. In *Artificial intelligence and statistics*, pages 436–443, 2007.

Jascha Sohl-Dickstein, Ben Poole, and Surya Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In *Proceedings of the International Conference on Machine Learning*, pages 604–612, Beijing, China, 21–26 June 2014.

James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

James C Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE transactions on automatic control*, 45(10):1839–1853, 2000.

Charles Stein, Persi Diaconis, Susan Holmes, Gesine Reinert, et al. Use of exchangeable pairs in the analysis of simulations. In *Stein's Method*, pages 1–25. Institute of Mathematical Statistics, 2003.

K Teerapabolarn. Stein's identity for discrete distributions. *International Journal of pure and applied mathematics*, 83(4):565, 2013.

Xiao Wang, Shiqian Ma, Donald Goldfarb, and Wei Liu. Stochastic quasi-newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization*, 27(2):927–956, 2017.

Jingyi Zhu. *Error Bounds and Applications for Stochastic Approximation with Non-Decaying Gain*. PhD thesis, Johns Hopkins University, 2020a.

Jingyi Zhu. Hessian inverse approximation as covariance for random perturbation in black-box problems. In *12th Annual Workshop on Optimization for Machine Learning*, 11 December 2020b.

Jingyi Zhu and James C. Spall. Stochastic approximation with non-decaying gain: Error bound and data-driven gain-tuning. *International Journal of Robust and Nonlinear Control*, 30(15): 5820–5870, 2020.

Jingyi Zhu, Long Wang, and James C Spall. Efficient implementation of second-order stochastic approximation algorithms in high-dimensional problems. *IEEE transactions on neural networks and learning systems*, 31(8):3087–3099, 2020.

Xun Zhu and James C Spall. A modified second-order spsa optimization algorithm for finite samples. *International Journal of Adaptive Control and Signal Processing*, 16(5):397–409, 2002.

Appendix A. Supplementary Proofs

Proof [Proof for Lemma 1]

i) Let us discuss the *general* setting where the underlying loss function $f(\cdot)$ convolutes with a Gaussian distribution with a symmetric positive-definite covariance matrix $\boldsymbol{\Sigma} \succ \boldsymbol{0}$:

$$f(\theta|\,\boldsymbol{\Sigma}) \equiv \int_{\mathbb{R}^d} f(\zeta)\varphi(\zeta|\,\theta, \boldsymbol{\Sigma})\mathrm{d}\zeta\,, \tag{19}$$

where $\varphi(\zeta|\,\theta, \boldsymbol{\Sigma}) \equiv [(2\pi)^d\,|\boldsymbol{\Sigma}|]^{-1/2} \exp\left[-1/2(\zeta-\theta)^T\boldsymbol{\Sigma}^{-1}(\zeta-\theta)\right]$ is the probability density function of the multivariate normal distribution $\mathrm{N}(\theta, \boldsymbol{\Sigma})$. By Leibniz's rule,

$$\left\{\begin{array}{l} \nabla f(\theta|\,\boldsymbol{\Sigma}) = \int_{\mathbb{R}^d} \boldsymbol{\Sigma}^{-1}(\zeta-\theta)\varphi(\zeta|\,\theta,\boldsymbol{\Sigma})f(\zeta)\mathrm{d}\zeta\,, \\ \nabla^2 f(\theta|\,\boldsymbol{\Sigma}) = \int_{\mathbb{R}^d} \left[\boldsymbol{\Sigma}^{-1}\left(\zeta-\theta\right)\left(\zeta-\theta\right)^T\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\right]\varphi(\zeta|\,\theta,\boldsymbol{\Sigma})f(\zeta)\mathrm{d}\zeta\,. \end{array}\right. \tag{20}$$

We can rewrite (19–20) compactly as follows:

$$\left\{\begin{array}{l} f(\theta|\,\boldsymbol{\Sigma}) = \mathbb{E}\left[f(\theta+\boldsymbol{C}\boldsymbol{u})\right]\,, \\ \nabla f(\theta|\,\boldsymbol{\Sigma}) = \mathbb{E}\left[\boldsymbol{C}^{-1}\boldsymbol{u}f(\theta+\boldsymbol{C}\boldsymbol{u})\right]\,, \\ \nabla^2 f(\theta|\,\boldsymbol{\Sigma}) = \mathbb{E}\left\{[\boldsymbol{C}^{-1}\left(\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I}\right)\boldsymbol{C}^{-1}]f(\theta+\boldsymbol{C}\boldsymbol{u})\right\}\,, \end{array}\right. \tag{21}$$

where the expectation is taken with respect to the random variable $\boldsymbol{u} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{C}$ is a symmetric matrix[16] such that $\boldsymbol{\Sigma} = \boldsymbol{C}^2$. Recall that $f_c(\theta) = \mathbb{E}_{\boldsymbol{u}\sim\mathrm{N}(\boldsymbol{0},\boldsymbol{I})}\left[f(\theta + c\boldsymbol{u})\right]$. Letting $\boldsymbol{C} = c\boldsymbol{I}$ (equivalently $\boldsymbol{\Sigma} = c^2\boldsymbol{I}$) in (21) gives (5–6), which is a special case of Proposition 1.

ii) (7) can be obtained using $\mathbb{E}(\boldsymbol{u}) = \boldsymbol{0}$ and $\nabla f_c(\theta) = -\mathbb{E}\left[c^{-1}\boldsymbol{u}f(\theta - c\boldsymbol{u})\right]$. (8) can be obtained using $\mathbb{E}(\boldsymbol{u}\boldsymbol{u}^T) = \boldsymbol{I}$ and $\nabla^2 f_c(\theta) = \mathbb{E}\left[c^{-2}\left(\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I}\right)f(\theta - c\boldsymbol{u})\right]$.

iii) (9) can be directly derived from (8).

■

Proof [Proof of Lemma 2]

- **Bias for (11c)**

$$\mathbb{E}_k(\hat{\boldsymbol{g}}_k) \overset{(11c)}{=} (2c_k)^{-1}\mathbb{E}_k\left\{[f(\hat{\theta}_k + c_k\boldsymbol{u}_k) + \varepsilon_k^+ - f(\hat{\theta}_k - c_k\boldsymbol{u}_k) - \varepsilon_k^-]\boldsymbol{u}_k\right\}$$

$$\overset{\text{a.s.}}{=} (2c_k)^{-1}\left(\mathbb{E}_k\{[f(\hat{\theta}_k + c_k\boldsymbol{u}_k) - f(\hat{\theta}_k - c_k\boldsymbol{u}_k)]\boldsymbol{u}_k\} + \mathbb{E}_k[\boldsymbol{u}_k\mathbb{E}(\varepsilon_k^+ - \varepsilon_k^-\,|\,\hat{\theta}_k, \boldsymbol{u}_k)]\right) \tag{22}$$

$$\overset{\text{a.s.}}{=} (2c_k)^{-1}\mathbb{E}_k\left(\mathbb{E}\left\{[f(\hat{\theta}_k + c_k\boldsymbol{u}_k) - f(\hat{\theta}_k - c_k)]\boldsymbol{u}_k\,\middle|\,\hat{\theta}_k\right\}\right) \tag{23}$$

$$\overset{\text{a.s.}}{=} \nabla f_{c_k}(\hat{\theta}_k)\,, \tag{24}$$

[16] Suppose $\boldsymbol{\Sigma}$ has eigen-decomposition form as $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ for some unitary matrix $\boldsymbol{U}$ and diagonal matrix $\boldsymbol{D}$, then $\boldsymbol{C} = \boldsymbol{U}\sqrt{\boldsymbol{D}}\boldsymbol{U}^T$ is symmetric and satisfies $\boldsymbol{\Sigma} = \boldsymbol{C}^2$.

where (22) is due to Chung (2001, Thm. 9.1.3 on p. 315), (23) is valid given A.2, and (24) is due to (7b). When A.1 holds, we have $|f(\zeta) - f(\theta) - (\zeta-\theta)^T \boldsymbol{g}(\theta) - 2^{-1}(\zeta-\theta)^T \boldsymbol{H}(\theta)(\zeta-\theta)| = O(\|\zeta-\theta\|^3)$ by Taylor expansion. Moreover,

$$\begin{aligned} 2c[\nabla f_c(\theta) - \boldsymbol{g}(\theta)] &\overset{(7b)}{=} \mathbb{E}_{\boldsymbol{u}}\{[f(\theta + c\boldsymbol{u}) - f(\theta - c\boldsymbol{u})]\boldsymbol{u}\} - 2c\boldsymbol{g}(\theta) \\ &= \mathbb{E}\left\{[f(\theta + c\boldsymbol{u}) - f(\theta) - c\boldsymbol{u}^T\boldsymbol{g}(\theta) - 2^{-1}c^2\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u}]\boldsymbol{u}\right\} \\ &\quad - \mathbb{E}\left\{[f(\theta - c\boldsymbol{u}) - f(\theta) + c\boldsymbol{u}^T\boldsymbol{g}(\theta) - 2^{-1}c^2\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u}]\boldsymbol{u}\right\}, \end{aligned} \tag{25}$$

where (25) uses $\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}^T\boldsymbol{g}(\theta)\boldsymbol{u}] = \boldsymbol{g}(\theta)$ for $\boldsymbol{u} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ per A.3. Using the triangle inequality and the higher-moments for multivariate chi-squared distribution, we have

$$\|\nabla f_c(\theta) - \boldsymbol{g}(\theta)\| \leq \frac{c^2 O(1)}{6}\mathbb{E}_{\boldsymbol{u}}(\|\boldsymbol{u}\|^5) = c^2 \times O(d^{5/2}).$$

Combining above with (24), we have $\mathbb{E}_k(\hat{\boldsymbol{g}}_k) \overset{\text{a.s.}}{=} \boldsymbol{g}(\hat{\theta}_k) + O(c_k^2)$ for (11c).

- **Bias for (11b)** Similarly, $\mathbb{E}_k(\hat{\boldsymbol{g}}_k) \overset{\text{a.s.}}{=} \nabla f_{c_k}(\hat{\theta}_k)$ for estimator (11b) thanks to (7a). Moreover,

$$\begin{aligned} c[\nabla f_c(\theta) - \boldsymbol{g}(\theta)] &\overset{(7a)}{=} \mathbb{E}_{\boldsymbol{u}}\{[f(\theta + c\boldsymbol{u}) - f(\theta)]\boldsymbol{u}\} - c\boldsymbol{g}(\theta) \\ &= \mathbb{E}\left\{[f(\theta + c\boldsymbol{u}) - f(\theta) - c\boldsymbol{u}^T\boldsymbol{g}(\theta) - 2^{-1}c^2\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u}]\boldsymbol{u}\right\}, \end{aligned} \tag{26}$$

  where (26) uses $\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}^T\boldsymbol{g}(\theta)\boldsymbol{u}] = \boldsymbol{g}(\theta)$ and $\mathbb{E}_{\boldsymbol{u}}\{[\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u}]\boldsymbol{u}\} = \boldsymbol{0}$ for $\boldsymbol{u} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$. For estimator (11b), $\mathbb{E}_k(\hat{\boldsymbol{g}}_k) \overset{\text{a.s.}}{=} \boldsymbol{g}(\hat{\theta}_k) + O(c_k^2)$ can be similarly derived.

- **Bias for (11a)** Still, $\mathbb{E}_k(\hat{\boldsymbol{g}}_k) \overset{\text{a.s.}}{=} \nabla f_{c_k}(\hat{\theta}_k)$ for estimator (11a) thanks to (5). Moreover,

$$\begin{aligned} c[\nabla f_c(\theta) - \boldsymbol{g}(\theta)] &\overset{(5)}{=} \mathbb{E}_{\boldsymbol{u}}\{f(\theta + c\boldsymbol{u})\boldsymbol{u}\} - c\boldsymbol{g}(\theta) \\ &= \mathbb{E}\left\{[f(\theta + c\boldsymbol{u}) - f(\theta) - c\boldsymbol{u}^T\boldsymbol{g}(\theta) - 2^{-1}c^2\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u}]\boldsymbol{u}\right\}, \end{aligned} \tag{27}$$

  where (27) uses $\mathbb{E}_{\boldsymbol{u}}[f(\theta)\boldsymbol{u}] = \boldsymbol{0}$, $\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}^T\boldsymbol{g}(\theta)\boldsymbol{u}] = \boldsymbol{g}(\theta)$, and $\mathbb{E}_{\boldsymbol{u}}\{[\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u}]\boldsymbol{u}\} = \boldsymbol{0}$.

The dominant contributor to the asymptotic variance of each element in $\hat{\boldsymbol{g}}_k$ is the $O(c_k^{-1})$ times the noise term, leading to a variance that is asymptotically proportional to c_k^{-2} with constant of proportionality independent of k because $\mathrm{Var}(\varepsilon_k^+ - \varepsilon_k^-)$ is asymptotically constant in k assumed in A.2. Furthermore, each elements at an arbitrary position in the $O(c_k^{-1})$ vector, as derived from (11), are uncorrelated across k by the independence assumptions on $\{\boldsymbol{u}_k\}$. Consequently, $\mathbb{E}_k(\|\hat{\boldsymbol{g}}_k\|^2) \overset{\text{a.s.}}{=} O(c_k^{-2})$. ■

Proof [Proof of Lemma 3] When A.2 and A.3 hold, we have $\mathbb{E}_k(\hat{\boldsymbol{H}}_k) \overset{\text{a.s.}}{=} \nabla^2 f_{c_k}(\hat{\theta}_k)$ for estimator (12d) following the proof of Lemma 2. When A.1 holds, i.e., $\|\nabla^3 f(\theta) - \nabla^3 f(\zeta)\| \leq \mathrm{L}_3\|\theta - \zeta\|$, by mean-value theorem we have $|f(\zeta) - f(\theta) - (\zeta-\theta)^T\boldsymbol{g}(\theta) - 2^{-1}(\zeta-\theta)^T\boldsymbol{H}(\theta)(\zeta-\theta) - 6^{-1}\nabla^3 f(\theta)\left[(\zeta-\theta)\otimes(\zeta-\theta)\otimes(\zeta-\theta)\right]| \leq 24^{-1}\mathrm{L}_3\|\zeta-\theta\|^4$. Moreover,

$$\begin{aligned} &2c^2[\nabla^2 f_c(\theta) - \boldsymbol{H}(\theta)] \\ &= \mathbb{E}_{\boldsymbol{u}}\left\{[f(\theta + c\boldsymbol{u}) + f(\theta - c\boldsymbol{u}) - 2f(\theta)]\left(\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I}\right)\right\} - 2c^2\boldsymbol{H}(\theta) \\ &= \mathbb{E}\left\{\left[f(\theta + c\boldsymbol{u}) - f(\theta) - c\boldsymbol{u}^T\boldsymbol{g}(\theta) - 2^{-1}c^2\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u} - 6^{-1}c^3\nabla^3 f(\theta)\left(\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}\right)\right]\left(\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I}\right)\right\} \\ &+ \mathbb{E}\left\{\left[f(\theta - c\boldsymbol{u}) - f(\theta) + c\boldsymbol{u}^T\boldsymbol{g}(\theta) - 2^{-1}c^2\boldsymbol{u}^T\boldsymbol{H}(\theta)\boldsymbol{u} + 6^{-1}c^3\nabla^3 f(\theta)\left(\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}\right)\right]\left(\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I}\right)\right\}, \end{aligned} \tag{28}$$

where (28) uses $\mathbb{E}_{\boldsymbol{u}}\left[\boldsymbol{u}^T \boldsymbol{g}(\boldsymbol{\theta}) \times (\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I})\right] = \boldsymbol{0}$, $\mathbb{E}_{\boldsymbol{u}}\left[2^{-1}\boldsymbol{u}^T \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{u} \times (\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I})\right] = \boldsymbol{H}(\boldsymbol{\theta})$, and $\mathbb{E}_{\boldsymbol{u}}\left[\nabla^3 f(\boldsymbol{\theta})\left(\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}\right) \times (\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I})\right] = \boldsymbol{0}$ when $\boldsymbol{u}$ follows A.3. Combined with triangle inequality and the higher-moments for multivariate chi-squared distribution, we have

$$\|\nabla^2 f_c(\boldsymbol{\theta}) - \boldsymbol{H}(\boldsymbol{\theta})\| \leq \frac{c^2 \mathrm{L}_3}{24}\mathbb{E}_{\boldsymbol{u}}\left(\|\boldsymbol{u}\|^4 \|\boldsymbol{u}\boldsymbol{u}^T - \boldsymbol{I}\|\right) = \frac{\mathrm{L}_3}{24}c^2 \times O(d^{7/2}).$$

Therefore, $\mathbb{E}_k(\hat{\boldsymbol{H}}_k) \overset{\text{a.s.}}{=} \boldsymbol{H}(\hat{\boldsymbol{\theta}}_k) + O(c_k^2)$ for (12d). We can similarly derive $\mathbb{E}_k(\hat{\boldsymbol{H}}_k) \overset{\text{a.s.}}{=} \boldsymbol{H}(\hat{\boldsymbol{\theta}}_k) + O(c_k^2)$ for the other three estimators (12a–12c).

The dominant contributor to the asymptotic variance of each element in $\hat{\boldsymbol{H}}_k$ is the $O(c_k^{-2})$ times the noise term, leading to a variance that is asymptotically proportional to c_k^{-4} with constant of proportionality independent of k because $\mathrm{Var}(\varepsilon_k^+ + \varepsilon_k^- - 2\varepsilon_k)$ is asymptotically constant in k and because of A.2. Furthermore, each elements at an arbitrary position in the $O(c_k^{-2})$ matrix, as derived from (12), are uncorrelated across k by the independence assumptions on $\{\boldsymbol{u}_k\}$. Consequently, $\mathbb{E}_k(\|\hat{\boldsymbol{H}}_k\|^2) \overset{\text{a.s.}}{=} O(c_k^{-4})$. ■

Proof [Proof of Thm. 1] From Lemma 2, we have $\mathbb{E}_k(\hat{\boldsymbol{g}}_k) = \boldsymbol{g}(\hat{\boldsymbol{\theta}}_k) + \boldsymbol{\beta}_k$ with $c_k^{-2}\|\boldsymbol{\beta}_k\|$ being uniformly bounded for sufficiently large k. The rest follows from Spall (2000, Thm. 1a). Note that the proof for Spall (2000, Thm. 1a) does not assume any particular form of the Hessian estimate; it only requires A.7 and the result in Lemmas 2–3. ■

Proof [Proof of Thm. 2] The proof proceeds similarly as Zhu (2020a, Appendix A). Let $\boldsymbol{W}_k \equiv \hat{\boldsymbol{H}}_k - \mathbb{E}_k(\hat{\boldsymbol{H}}_k)$, which satisfies $\mathbb{E}\boldsymbol{W}_k = \boldsymbol{0}$ for all k. Thanks to Lemma 3, $\mathbb{E}(c_k^4\|\hat{\boldsymbol{H}}_k\|^2) < \infty$ uniformly for all k. When A.4 holds, we have $\sum_k \left(\mathbb{E}\|w_k \boldsymbol{W}_k\|^2\right) < \infty$. By martingale convergence theorem, we have $\sum_k w_k \boldsymbol{W}_k \to \boldsymbol{0}$ a.s. Now that Lemma 3 gives $\mathbb{E}_k(\hat{\boldsymbol{H}}_k) = \boldsymbol{H}(\hat{\boldsymbol{\theta}}_k) + O(c_k^2)$, we have $\sum_k [w_k \mathbb{E}_k(\hat{\boldsymbol{H}}_k)] = \sum_k \{w_k[\boldsymbol{H}(\hat{\boldsymbol{\theta}}_k) + O(c_k^2)]\} \to \boldsymbol{H}(\boldsymbol{\theta}^*)$ a.s., using A.1 (Hessian is continuous around $\hat{\boldsymbol{\theta}}_k$), Theorem 1 ($\hat{\boldsymbol{\theta}}_k$ converges a.s. to $\boldsymbol{\theta}^*$), and A.4 ($w_k c_k^2 \to 0$). ■

Proof [Proof of Thm. 3] The proof can be derived following the proofs of the general result Zhu (2020b, Thm.3) after specifying the conditioner and the distribution for the perturbation. We skip the sketch here as it involves quite a few notations not introduced here. ■