DRSOM: A Dimension Reduced Second-Order Method

Chuwen Zhang CHUWZHANG@GMAIL.COM
Dongdong Ge GE.DONGDONG@SHUFE.EDU.CN
Chang He CHANGHE@163.SHUFE.EDU.CN
Bo Jiang ISYEBOJIANG@GMAIL.COM
Yuntian Jiang YUNTIANJIANG07@163.SUFE.EDU.CN
Shanghai University of Finance and Economics

Yinyu Ye YINYU-YE@STANFORD.EDU
Stanford University

Abstract

In this paper, we propose a Dimension-Reduced Second-Order Method (DRSOM) for convex and nonconvex (unconstrained) optimization. Under a trust-region-like framework, our method preserves the convergence of the second-order method while using only Hessian-vector products in a few directions, which enables the computational overhead of our method remain comparable to the first-order such as the gradient descent method. Theoretically, we show that the method has a local quadratic convergence and a global convergence rate of $O(\epsilon^{-3/2})$ to satisfy the first-order and second-order conditions in certain subspace under a commonly adopted approximated Hessian assumption. We further show that this assumption can be removed if we perform a step of the Lanczos method periodically at the end-stage of the algorithm. The applicability and performance of DRSOM are exhibited by various computational experiments, particularly in machine learning and deep learning. For neural networks, our preliminary implementation seems to gain computational advantages in terms of training accuracy and iteration complexity over state-of-the-art first-order methods such as SGD and ADAM.

1. Introduction

In this paper, we consider the following unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where $f:\mathbb{R}^n\mapsto\mathbb{R}$ is twice differentiable and possibly nonconvex and $f_{\inf}:=\inf f(x)>-\infty$. We aim to find a "stationary point" x such that

$$\|\nabla f(x)\|\le\epsilon \tag{2}$$

and x approximately satisfies the second-order necessary conditions in certain subspace.

Historically speaking, various methods have been proposed to solve (1). Under the Lipschitz-continuous gradient condition, the gradient descent method is able to converge to a point that satisfies (2) in $O(\epsilon^{-2})$ iterations [15]. This bound can be improved to $O(\epsilon^{-1/2})$ for convex f using Nesterov's acceleration [17]. If we use second-order information in the algorithm, Nesterov and Polyak [16] show that solving (1) has an iteration complexity of $O(\epsilon^{-3/2})$ with cubic regularization (also see Cartis et al. [4, 5]). Trust-region method is another popular second-order method, where a

ball constraint is introduced and the cubic regularization is absent in the objective. Luenberger and Ye [14] establish the iteration complexity of $O(\epsilon^{-3/2})$ for a fixed-radius trust-region method, which matches the iteration bound of the aforementioned cubic regularized Newton method. However, the complexity analysis for trust-region method with adaptive radius seems to be more challenging. To our best knowledge, the first $O(\epsilon^{-3/2})$ bound for adaptive strategy is established only recently by Curtis et al. [10].

For second-order methods, both explicit computation of Hessian and solving trust region subproblem (TRS) could be costly. A practical implementation usually uses a Lanczos method to find inexact solutions [5, 8, 9], where the "inexactness" can be controlled by the quality of Hessian approximation in some sense [4, 10, 11, 21]. Recently, Carmon et al. [2, 3] use the negative curvature with the accelerated gradient method to achieve a complexity bound of $O(\epsilon^{-7/4}\log(\epsilon^{-1}))$ with a fast Lanczos method, however, the numerical experiment is not provided.

Motivated by previous research, our goal is to find a first-order method that incorporates the second-order information cheaply. Specifically, we introduce a *Dimension-Reduced Second-Order Method* (DRSOM) that restricts the iterates in the subspace spanned by the gradient and the momentum. Therefore, the computational cost of DRSOM is mostly due to solving a 2-dimensional trust-region subproblem and the Hessian-vector products (see [18]) to construct the 2-by-2 approximated Hessian of the subproblem at each iteration, which is quite cheap. We also propose a "Radius-Free" DRSOM with a quadratic regularization similar to the framework proposed in [10].

Theoretically, under a commonly adopted approximated Hessian assumption (cf. [4, AM.4]), we show that DRSOM has a local quadratic convergence and has an $O(\epsilon^{-3/2})$ complexity to globally converge to a point satisfying the first-order condition (2) and the second-order condition in a certain subspace. We identify that this assumption is only needed at the end stage of the algorithm and can be further removed if we perform a step of the Lanczos method periodically. Furthermore, comparing to methods using fast curvature computation [2, 3] and inexact solutions [4, 8, 10] all along, the frequency of invoking the Lanczos method by DRSOM is greatly reduced, which results in significant savings in computational time.

Finally, we perform comprehensive experiments on convex and nonconvex problems. We note DRSOM is comparable to a second-order method (e.g. the Newton-CG) in iteration. For deep learning, our preliminary implementation demonstrates notable advantages in terms of training accuracy and iteration complexity over state-of-the-art first-order methods such as SGD and Adam.

Our paper is organized as follows. In Section 2, we discuss the details of the DRSOM including its important building blocks. In Section 3, we give a suit of convergence results of DRSOM. Finally, the comprehensive numerical results of DRSOM are summarized in Section 4.

2. The Algorithm Design

To facilitate the discussion, we denote $g_k = \nabla f(x_k), H_k = \nabla^2 f(x_k)$, and $d_k = x_k - x_{k-1}$ throughout the paper. In each iteration of the *Dimension-Reduced Second-Order Method* (DRSOM) we update $x_{k+1} = x_k - \alpha_k^1 g_k + \alpha_k^2 d_k$ and the step size $\alpha_k = (\alpha_k^1, \alpha_k^2)$ is determined by solving the following 2-dimensional quadratic model $m_k(\alpha)$:

$$\begin{aligned} &\min_{\alpha\in\mathbb{R}^2} m_k(\alpha) := f(x_k) + (c_k)^T\alpha + \frac{1}{2}\alpha^T Q_k \alpha \\ &\text{s.t. } \|\alpha\|_{G_k} \le \Delta, \end{aligned} \tag{3}$$

where

$$Q_k = \begin{bmatrix} (g_k)^T H_k g_k & -(d_k)^T H_k g_k \\ -(d_k)^T H_k g_k & (d_k)^T H_k d_k \end{bmatrix} \in \mathcal{S}^2, c_k := \begin{bmatrix} -\|g_k\|^2 \\ (g_k)^T d_k \end{bmatrix} \in \mathbb{R}^2, G_k = \begin{bmatrix} (g_k)^T g_k & -(g_k)^T d_k \\ -(g_k)^T d_k & (d_k)^T d_k \end{bmatrix},$$

and $\|\alpha\|_{G_k} = \sqrt{\alpha^T G_k \alpha}$. The idea of considering function $m_k(\alpha)$ was previously studied in Ye [22], Yuan and Stoer [24]. The novel ingredient in this paper is to impose a 2×2 trust-region step to determine the step-sizes, which is necessary and sufficient for solving nonconvex problems.

Although problem (3) is a 2-dimensional trust region model, it can be equivalently transformed into a "full-scale" trust-region problem as shown below.

Lemma 1 *The subproblem (3) is equivalent to*

$$\min_{d \in \mathbb{R}^n} \ \tilde{m}_k(d) := f(x_k) + g_k^T d + \frac{1}{2} d^T \tilde{H}_k d, \ \text{s.t.} \ \|d\| \le \Delta, \tag{4}$$

where $\tilde{H}_k = V_k V_k^T H_k V_k V_k^T$ and V_k is the orthonormal bases for $\mathcal{L}_k := \mathsf{span}\{g_k, d_k\}$.

Consequently, we don't require the constraint $d \in \mathcal{L}_k$ in problem (4) and shift the subspace restriction to the matrix $\tilde{H}_k$, which can be viewed as the projection of the original Hessian H_k in the subspace $\mathcal{L}_k$. Therefore, $\tilde{H}_k$ plays similar role of the approximated Hessian matrix in quasi-Newton method and DRSOM may also be regarded as a cheap quasi-Newton method. In fact, the reformulation (4) will be frequently used in the convergence analysis of DRSOM.

Although (4) is useful in the theoretical analysis, we actually solve the 2-dimensional quadratic problem (3) in the implementation of DRSOM. To efficiently compute the 2×2 matrix Q_k in (3), we make use of the decomposition $Q_k = \begin{bmatrix} -g_k & dk \end{bmatrix}^T \begin{bmatrix} -H_k \cdot g_k & H_k \cdot d_k \end{bmatrix}$. Thus, it remains to compute the two Hessian-vector products (see [18]): $H_k g_k$ and $H_k d_k$. We adopt the following two strategies to compute those products without request for the true Hessian H_k:

1. Finite difference: $H_k \cdot v \approx \frac{1}{\epsilon}[g(x_k + \epsilon \cdot v) - g_k]$.
2. Automatic differentiation (AD): $H_k g_k = \nabla(\frac{1}{2} g_k^T g_k), H_k d_k = \nabla(d_k^T g_k)$.

In practice, the finite-difference method should work in most cases and demonstrate reasonable efficiency of computation, except for deep-learning applications where efficient implementation of AD is realized.

Since we only need to solve a 2-dimensional TRS subproblem (3), per-iteration cost of DRSOM is very cheap. Furthermore, such low-dimensional TRS subproblem can be solved very efficiently [8, 18]; notably, Ye [23] shows that an ϵ-global primal-dual optimizer (α^*, λ^*) of TRS can be found in $O(\log\log(1/\epsilon))$ time. The complexity of the subproblem is thus affordable. We leave the details of this subroutine to Section A.

In the implementation, we also consider the "Radius-Free" DRSOM by dropping the ball constraint in (3) while imposing a quadratic regularization in the objective:

$$\min_{\alpha \in \mathbb{R}^2} \ m_k(\alpha) + \mu_k \|\alpha\|_{G_k}^2. \tag{5}$$

We present a conceptual DRSOM in Algorithm 1 by incorporating the two alternatives (3) and (4) to compute the step size α, where the adaptive strategy to update the radius of the trust-region constraint or the coefficient of the quadratic regularization is adopted. We leave the detailed description of the adjustment on Δ_k or μ_k in Section C.

Algorithm 1: A conceptual DRSOM algorithm

Data: Given $k_{\max}, \mu_1 = 0, \Delta_1 \in (0, \bar{\Delta})$, and $\eta \in [0, \zeta_1)$;
for $k = 1, ..., k_{\max}$ **do**
  Solve (3) or (5) for α_k, compute $d_{k+1} = -\alpha_k^1 g_k + \alpha_k^2 d_k$ and $\rho_k := \frac{f(x_k) - f(x_k + d_{k+1})}{m_k(0) - m_k(\alpha_k)}$;
  If $\rho_k > \eta$ Accept the step and update $x_{k+1} = x_k + d_{k+1}$; o.w. Adjust Δ_k in (3) or μ_k in (5).
end

3. Convergence Results

In this section, we provide a a suite of convergence results of DRSOM. The detailed proofs can be found in Section C.

3.1. Finite convergence for strongly convex quadratic programming

We first show DRSOM has finite convergence for convex quadratic programming.

$$\min f(x) = \frac{1}{2} x^T A x + a^T x, \tag{6}$$

where $A \succ 0$. For this case, we do not have to place a trust-region radius for DRSOM, i.e., Δ_k is sufficiently large, $\lambda_k = 0$ for all k. We have the following theorem.

Theorem 1 *If we apply DRSOM to (6) with no radius restriction, i.e., Δ is sufficiently large, then the DRSOM generates the same iterates of conjugate gradient method, if they start at the same point x_0.*

The equivalence of quadratic minimization over $\mathcal{L}_k$ and conjugate gradient method was also established in Yuan and Stoer [24]. We provide the proof of Theorem 1 for the completeness of the paper.

3.2. Global and local convergence rate

Before presenting the results on convergence rate, we make the following assumptions. The first one is standard for second order method; see Nesterov [15].

Assumption 1 *f has L-Lipschitz continuous gradient and M-Lipschitz continuous Hessian such that for $\forall x, y \in \mathbb{R}^n$,*

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\| \quad \textit{and} \quad \|\nabla^2 f(x) - \nabla^2 f(y)\| \le M\|x - y\|. \tag{7}$$

The second one is regarding the approximated Hessian $\tilde{H}_k$, which is commonly used in the literature; see [5, 10, 11, 21].

Assumption 2 *The approximated Hessian matrix $\tilde{H}_k$ along subspace $\mathcal{L}_k$ satisfies:*

$$\|(H_k - \tilde{H}_k) d_{k+1}\| \le C\|d_{k+1}\|^2 \tag{8}$$

Although we adopt the adaptive strategy to choose the radius in the implementation, our analysis is conducted under the fixed-radius strategy such that a step is always accepted for simplicity. In terms of the global convergence rate, we show that DRSOM has an $O\left(\epsilon^{-3/2}\right)$ complexity to converge to the first-order stationary point and second-order point in the subspace.

Theorem 2 *Under the fixed-radius strategy by setting $\Delta_k = \frac{2\sqrt{\epsilon}}{M}$, the DRSOM runs at most $O\left(\frac{3}{2}M^2(f(x_0) - f_{\inf})\epsilon^{-3/2}\right)$ iterations to reach an iterate x_{k+1} that satisfies the first-order condition (2), and the approximated second order condition in the subspace $\mathcal{L}_k$: $V_k V_k^T H_{k+1} V_k V_k^T \succeq -3\sqrt{\epsilon}I$, where V_k is the orthonormal bases for $\mathcal{L}_k$.*

Regarding the local rate of convergence, we have the following results.

Theorem 3 *Suppose x^* is a second-order stationary point such that $H(x^*) \succeq \mu I$ for some $\mu > 0$. Then if x_k is sufficiently close to x^*, DRSOM converges to x^* quadratically, namely: $\|x_{k+1} - x^*\| \leq O(\|x_k - x^*\|^2)$.*

3.3. Discussion on Assumption 2

In fact, the inequality (8) in Assumption 2 plays a crucial role in our convergence analysis. To ensure (8), a popular strategy is to apply a Lanczos method; see [5, 8, 10]. Although we have not rigorously established the validness of Assumption 2 yet, we manage to find out that it is only required when λ_k is small. Therefore, we apply the Lanczos method only when $\lambda_k \leq \sqrt{\epsilon}$ during the iteration process of DRSOM. Then the expanded subspace due to the Lanczos method, in return possibly produces a larger λ_k (and thus DRSOM proceeds). Therefore, Lanczos method is only called periodically when $\lambda_k \leq \sqrt{\epsilon}$, which corresponds to the late-stage of the algorithm. Finally, we terminate the algorithm as soon as $\lambda_k \leq \sqrt{\epsilon}$ and (8) hold simultaneously. We provide the detailed discussion in Section B.

4. Numerical experiments

In the numerical experiments, we first run DRSOM on the multinomial logistic regression model and the nonconvex $\mathcal{L}_2 - \mathcal{L}_p$ problem. The results show DRSOM is close to the authentic second-order methods in terms of the solution quality for both convex and nonconvex minimization with less computational time. We provide an intriguing example of the sensor network localization problem where DRSOM can provide better solutions than the first-order methods. The rest of the experiments focus on deep learning. Our preliminary implementation illustrates DRSOM outperforms Adam in training neural networks. We present implementation details and numerical results in Section C.

References

[1] Pratik Biswas and Yinyu Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Third International Symposium on Information Processing in Sensor Networks, IPSN 2004*, pages 46–54, 2004. ISBN 1-58113-846-6. doi: 10.1145/984622.984630.

[2] Yair Carmon, Oliver Hinder, John C. Duchi, and Aaron Sidford. "Convex Until Proven Guilty": Dimension-Free Acceleration of Gradient Descent on Non-Convex Functions, May 2017. URL http://arxiv.org/abs/1705.02766. arXiv:1705.02766 [math].

[3] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated Methods for NonConvex Optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, January 2018. ISSN 1052-6234, 1095-7189. doi: 10.1137/17M1114296. URL https://epubs.siam.org/doi/10.1137/17M1114296.

[4] Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Mathematical Programming*, 130(2):295–319, December 2011. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-009-0337-y. URL http://link.springer.com/10.1007/s10107-009-0337-y.

[5] Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, April 2011. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-009-0286-5. URL http://link.springer.com/10.1007/s10107-009-0286-5.

[6] Xiaojun Chen. Smoothing methods for nonsmooth, nonconvex minimization. *Mathematical Programming*, 134(1):71–99, August 2012. ISSN 1436-4646. doi: 10.1007/s10107-012-0569-0. URL https://doi.org/10.1007/s10107-012-0569-0.

[7] Xiaojun Chen, Dongdong Ge, Zizhuo Wang, and Yinyu Ye. Complexity of unconstrained $L_2 - L_p$ minimization. *Mathematical Programming*, 143(1-2):371–383, February 2014. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-012-0613-0. URL http://link.springer.com/10.1007/s10107-012-0613-0.

[8] Andrew R. Conn, Nicholas IM Gould, and Philippe L. Toint. *Trust region methods*. SIAM, 2000.

[9] Frank E. Curtis and Qi Wang. Worst-Case Complexity of TRACE with Inexact Subproblem Solutions for Nonconvex Smooth Optimization, April 2022. URL http://arxiv.org/abs/2204.11322. arXiv:2204.11322 [math].

[10] Frank E. Curtis, Daniel P. Robinson, and Mohammadreza Samadi. A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. *Mathematical Programming*, 162(1):1–32, March 2017. ISSN 1436-4646. doi: 10.1007/s10107-016-1026-2. URL https://doi.org/10.1007/s10107-016-1026-2.

[11] John E. Dennis and Jorge J. Moré. Quasi-Newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977. Publisher: SIAM.

[12] Dongdong Ge, Xiaoye Jiang, and Yinyu Ye. A note on the complexity of L_p minimization. *Mathematical Programming*, 129(2):285–299, 2011. doi: 10.1007/s10107-011-0470-2.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*, volume 228 of *International Series in Operations Research & Management Science*. Springer International Publishing, Cham, 2021. ISBN 978-3-030-85449-2 978-3-030-85450-8. doi: 10.1007/978-3-030-85450-8. URL https://link.springer.com/10.1007/978-3-030-85450-8.

[15] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[16] Yurii Nesterov and B.T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, August 2006. ISSN 1436-4646. doi: 10.1007/s10107-006-0706-8. URL https://doi.org/10.1007/s10107-006-0706-8.

[17] Yurii E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[18] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[19] Zizhuo Wang, Song Zheng, Yinyu Ye, and Stephen Boyd. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19(2):655–673, 2008. Publisher: SIAM.

[20] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, August 2017. arXiv: cs.LG/1708.07747.

[21] Peng Xu, Fred Roosta, and Michael W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *Mathematical Programming*, 184(1-2):35–70, November 2020. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-019-01405-z. URL http://link.springer.com/10.1007/s10107-019-01405-z.

[22] Yinyu Ye. Second Order Optimization Algorithms I. URL https://web.stanford.edu/class/msande311/lecture12.pdf.

[23] Yinyu Ye. A New Complexity Result on Minimization of a Quadratic Function with a Sphere Constraint. In *Recent Advances in Global Optimization*, volume 176, pages 19–31. Princeton University Press, 1991. ISBN 978-1-4008-6252-8. doi: 10.1515/9781400862528.19. URL https://www.degruyter.com/document/doi/10.1515/9781400862528.19/html.

[24] Y.-X. Yuan and J. Stoer. A subspace study on conjugate gradient algorithms. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 75(1):69–77, 1995. Publisher: Wiley Online Library.

[25] Hongchao Zhang and William W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM journal on Optimization*, 14(4):1043–1056, 2004. Publisher: SIAM.

Appendix A. Details for Solving TRS

Similar to full-dimensional TRS, in our method, the subproblem (3) can be solved efficiently. We introduce the following Lemma which is widely known:

Lemma 2 *The vector α^* is the global solution to trust-region subproblem (3) if it is feasible and there exists a Lagrange multiplier $\lambda^* \geq 0$ such that (α^*, λ^*) is the solution to the following equations:*

$$(Q_k + \lambda G_k)\alpha + c_k = 0, Q_k + \lambda G_k \succeq 0, \lambda(\Delta - \|\alpha\|_{G_k}) = 0. \tag{9}$$

From the construction of Q_k and G_k, we have that

$$Q_k + \lambda G_k = \begin{bmatrix} -g_k^T \\ d_k^T \end{bmatrix} (H_k + \lambda I) \begin{bmatrix} -g_k & d_k \end{bmatrix}. \tag{10}$$

Therefore, even if Q_k is indefinite, there always exists a sufficiently large λ such that condition (9) holds. Due to the fact that we only use 2 directions, the subproblems are easy to solve; for example, Ye [23] shows that an ϵ-global primal-dual optimizer (α^*, λ^*) satisfying (9) can be found in $O(\log\log(1/\epsilon))$ time. One may also find the optimal solutions by other standard methods in Conn et al. [8].

We also introduce the normalized problem to enable concise analysis. Let V_k be the orthonormal bases for $\mathcal{L}_k$,

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^2} \ & f(x_k) + \alpha^T V_k^T g_k + \frac{1}{2}\alpha^T V_k^T H_k V_k \alpha \\ \text{s.t.} \ & \|\alpha\| \leq \Delta_k. \end{aligned} \tag{11}$$

It is easy to (11) and (3) are equivalent under a linear transformation. With a slightly abuse of notation, letting α_k and λ_k be the solution and the associated Lagrangian multiplier with the trust region constraint to the normalized problem, we have the following lemma.

Lemma 3 *Let α_k and λ_k be the solution and the associated Lagrangian multiplier with the trust region constraint to the normalized problem. Construct $d_{k+1} = V_k\alpha_k$, then d_{k+1} is the solution to the full-scale problem (4) such that*

$$(\tilde{H}_k + \lambda_k I)d_{k+1} + g_k = 0, \ \tilde{H}_k + \lambda_k I \succeq 0, \ \lambda_k(\|d_{k+1}\| - \Delta_k) = 0, \tag{12}$$

where $\tilde{H}_k = V_k V_k^T H_k V_k V_k^T$.

Proof According to (9), we have that

$$(V_k^T H_k V_k + I)\alpha_k + V_k^T g_k = 0, V_k^T H_k V_k + \lambda_k I \succeq 0, \lambda_k(\Delta - \|\alpha_k\|) = 0. \tag{13}$$

Multiplying V_k to the left of both sides of the first equation in (13) yields that

$$V_k V_k^T H_k V_k V_k^T V_k \alpha_k + \lambda_k V_k \alpha_k = V_k V_k^T H_k V_k \alpha_k + \lambda_k V_k \alpha_k = -V_k V_k^T g_k = -g_k,$$

where the first equality is due to $V_k^T V_k = I$ and the last equality follows from $V_k V_k^T$ is the projection matrix of $\mathcal{L}_k$ and $g_k \in \mathcal{L}_k$. As a result, we have that:

$$(V_k V_k^T H_k V_k V_k^T + \lambda_k I)d_{k+1} + g_k = 0$$

proving the first equation in (12). Due to the second equation in (13), we have that

$$\alpha^T V_k^T H_k V_k \alpha + \lambda_k \alpha^T \alpha \geq 0, \ \forall \ \alpha \in \mathbb{R}^2.$$

By letting $d = V_k \alpha$, it is equivalent to

$$d^T H_k d + \lambda_k d^T d \geq 0, \ \forall \ d \in \mathcal{L}_k$$

due to V_k is the orthonormal bases for $\mathcal{L}_k$ and $d^T d = \alpha^T V_k^T V_k \alpha = \alpha^T \alpha$. The inequality above is further equivalent to

$$\tilde{d}^T (V_k V_k^T H_k V_k V_k^T + \lambda_k I) \tilde{d} \geq 0, \ \forall \ \tilde{d} \in \mathbb{R}^n$$

as $V_k V_k^T$ is the projection matrix of $\mathcal{L}_k$, and this proves $\tilde{H}_k + \lambda I \succeq 0$ in (12). Finally, since $d_{k+1}^T d_{k+1} = \alpha_k^T V_k^T V_k \alpha_k = \alpha_k^T \alpha_k$, the last equation in (12) follows from the last equation in (13). ■

Appendix B. Proofs of Main Results

B.1. Proof of Theorem 1

Proof To show its equivalence to the conjugate gradient method, we only have to prove the iterate x_k by DRSOM minimizes $f(x)$ in the subspace such that:

$$x_k \in \mathcal{L}_k = x_0 + \mathsf{span}\{d_1, ..., d_k\}.$$

Since there is no radius, the solution that minimizes m_k strictly corresponds to the optimizer of $f(x)$ in the subspace $x_k + \mathsf{span}\{g_k, d_k\}$. In other words, the iterate of DRSOM can also be retrieved by simply choosing the stepsizes such that $x_{k+1} = \arg\min f(x)$ and $x_{k+1} = x_k - \alpha_k^1 g_k + \alpha_k^2 d_k$. From such a perspective, we show that the x_k is equivalent to $\tilde{x}_k$ by the conjugate gradient method.

Note $\tilde{x_k}$ minimizes $f(x)$ over the subspace below:

$$x_0 + \mathsf{span}\{\tilde{d}_1, ..., \tilde{d}_k\},$$

where $\tilde{d}_1, ..., \tilde{d}_k$ are conjugate directions for CG. By construction, we see $x_1 = x_0 + \alpha_0^1 g_0$ and $d_1 = \alpha_0^1 g_0$, so that $x_1 = \tilde{x}_1$. Assume it holds for k, we know for CG:

$$\tilde{g_k} \in \mathsf{span}\{\tilde{d_k}, \tilde{d}_{k+1}\},$$

and since $g_k = \tilde{g_k}, d_k = \tilde{d_k}$, we have

$$\mathsf{span}\{\tilde{d_k}, \tilde{d}_{k+1}\} = \mathsf{span}\{d_k, g_k\}, \tag{14}$$

Now we know from the next minimizer $x_{k+1} \in x_k + \mathsf{span}\{d_k, g_k\}$:

$$x_{k+1} \in \tilde{x}_k + \mathsf{span}\{\tilde{d_k}, \tilde{d}_{k+1}\},$$

and since $\tilde{x}_{k+1}$ minimizes f over both $x_0 + \mathsf{span}\{d_1, ..., d_k, \tilde{d}_{k+1}\}$ and $x_k + \mathsf{span}\{\tilde{d}_{k+1}\}$, we have that $x_{k+1} = \tilde{x}_{k+1}$ as the desired result. ■

B.2. Proof of Theorem 2

Let us first inspect the properties of the reduction by the quadratic model.

Lemma 4 (Model reduction) *At iteration k, let d_{k+1} and λ_k be the solution and Lagrangian multiplier constructed in Lemma 3. If $\lambda_k > 0$, we have the following amount of decrease on $\tilde{m}_k$:*

$$\tilde{m}_k(d_{k+1}) - \tilde{m}_k(0) = -\frac{1}{2}\lambda_k \Delta_k^2. \tag{15}$$

Proof In view of Lemma 1 and Lemma 3, the optimal condition (12) holds. Then, we have $\|d_{k+1}\| = \Delta_k$ due to $\lambda_k > 0$ and that:

$$\begin{aligned}\tilde{m}_k(d_{k+1}) - \tilde{m}_k(0) &= g_k^T d_{k+1} + \frac{1}{2} d_{k+1}^T \tilde{H}_k d_{k+1} \\ &= -\frac{1}{2} d_{k+1}^T (\tilde{H}_k + \lambda_k I) d_{k+1} + \frac{1}{2} d_{k+1}^T \tilde{H}_k d_{k+1} \\ &= -\frac{1}{2}\lambda_k \Delta_k^2,\end{aligned} \tag{16}$$

which completes the proof. ■

Take $\Delta_k = \Delta = \frac{2\sqrt{\epsilon}}{M}$ and combine with the reduction of the quadratic model, we conclude that DRSOM generates sufficient decrease at every iteration k as long as $\lambda_k \geq \sqrt{\epsilon}$. The following analysis based on a *fixed* trust-region radius is motivated from Luenberger and Ye [14].

Lemma 5 (Sufficient decrease) *At iteration k, take $\Delta_k = \Delta = \frac{2\sqrt{\epsilon}}{M}$, and let d_{k+1} and λ_k be the solution and Lagrangian multiplier obtained in Lemma 3. If $\lambda_k \geq \sqrt{\epsilon}$, we have the following amount of function value decrease,*

$$f(x_{k+1}) \leq f(x_k) - \frac{2}{3M^2}\epsilon^{3/2}. \tag{17}$$

Proof Since $d_{k+1} \in \mathcal{L}_k$ and $V_k V_k^T$ is the projection matrix of $\mathcal{L}_k$, it holds that

$$d_{k+1}^T \tilde{H}_k d_{k+1} = d_{k+1}^T V_k V_k^T H_k V_k V_k^T d_{k+1} = d_{k+1}^T H_k d_{k+1}.$$

Moreover, with second-order Lipschitz continuity and tailor expansion, we immediately have:

$$\begin{aligned}f(x_{k+1}) &\leq f(x_k) + (g_k)^T d_{k+1} + \frac{1}{2}(d_{k+1})^T H_k (d_{k+1}) + \frac{M}{6}\|d_{k+1}\|^3 \\ &= f(x_k) + (g_k)^T d_{k+1} + \frac{1}{2}(d_{k+1})^T \tilde{H}_k (d_{k+1}) + \frac{M}{6}\|d_{k+1}\|^3 \\ &= f(x_k) - \frac{1}{2}\lambda_k \Delta^2 + \frac{1}{6} M \Delta^3 \\ &= f(x_k) - \frac{2\lambda_k \epsilon}{M^2} + \frac{4\epsilon^{3/2}}{3M^2}\end{aligned} \tag{18}$$

where the second last equality is due to Lemma 4 and $\|d_{k+1}\| = \Delta_k = \Delta$ from the optimality condition (12) with $\lambda_k > 0$. Noting that $\lambda_k \geq \sqrt{\epsilon}$, inequality (18) implies that

$$f(x_{k+1}) \leq f(x_k) - \frac{2}{3M^2}\epsilon^{3/2}.$$

■

The following result states that when $\lambda_k \le \sqrt{\epsilon}$ and the Hessian regularity condition hold, we can terminate the process at the next iterate x_{k+1} that approximated satisfies the first-order condition and the second-order condition in the subspace.

Lemma 6 *At iteration k, if the Lagrangian multiplier λ_k associated with the trust region constraint in (3) satisfies $\lambda_k \le \sqrt{\epsilon}$ and Hessian regularity condition (8) holds, then the iterate x_{k+1} approximately satisfies the first-order condition, and the second-order condition in the subspace $\mathcal{L}_k$.*

Proof Suppose d_{k+1} is the solution obtained in Lemma 3. By second-order Lipschitz continuity and the first equation in the optimality condition (12), we have that:

$$\begin{aligned}\|g_{k+1}\| &\le \|g_{k+1} - g_k - H_k d_{k+1}\| + \|(H_k - \tilde{H}_k)d_{k+1}\| + \|(g_k + \tilde{H}_k d_{k+1})\| \\ &\le \left\| \int_0^1 \left[\nabla^2 f(x_k + \tau d_{k+1}) - H_k\right] d_{k+1} d\tau \right\| + \|(H_k - \tilde{H}_k)d_{k+1}\| + \lambda_k \|d_{k+1}\| \\ &\le \frac{1}{2} M \|d_{k+1}\|^2 + \lambda_k \|d_{k+1}\| + \|(H_k - \tilde{H}_k)d_{k+1}\|\end{aligned} \tag{19}$$

In view of Assumption 2, $\lambda_k \le \sqrt{\epsilon}$ and $\|d_{k+1}\| \le \Delta = \frac{2\sqrt{\epsilon}}{M}$, we immediately have

$$\begin{aligned}\|g_{k+1}\| &\le \left(\frac{1}{2}M + C\right) \|d_{k+1}\|^2 + \lambda_k \|d_{k+1}\| \\ &\le \left(\frac{1}{2}M + C\right) \frac{4\epsilon}{M^2} + \frac{2\epsilon}{M} \\ &\le \left(\frac{4}{M} + \frac{4C}{M^2}\right) \epsilon.\end{aligned} \tag{20}$$

As for the second-order condition, the second condition in (12) and $\lambda_k \le \sqrt{\epsilon}$ imply that

$$\begin{aligned}-\sqrt{\epsilon} I \preceq -\lambda_k I \preceq \tilde{H}_k &= V_k V_k^T H_{k+1} V_k V_k^T + \tilde{H}_k - V_k V_k^T H_{k+1} V_k V_k^T \\ &= V_k V_k^T H_{k+1} V_k V_k^T + V_k V_k^T (H_k - H_{k+1}) V_k V_k^T \\ &\preceq V_k V_k^T H_{k+1} V_k V_k^T + \|V_k V_k^T (H_k - H_{k+1}) V_k V_k^T\| I \\ &\preceq V_k V_k^T H_{k+1} V_k V_k^T + \|V_k V_k^T\| \|H_{k+1} - H_k\| \|V_k V_k^T\| I \\ &= V_k V_k^T H_{k+1} V_k V_k^T + \|H_{k+1} - H_k\| I \\ &\preceq V_k V_k^T H_{k+1} V_k V_k^T + M \|d_{k+1}\| I \\ &\preceq V_k V_k^T H_{k+1} V_k V_k^T + 2\sqrt{\epsilon} I,\end{aligned} \tag{21}$$

where the second last matrix inequality is due to the Lipschitz continuity of the Hessian and the last matrix inequality follows from $\|d_{k+1}\| \le \Delta = \frac{2\sqrt{\epsilon}}{M}$. Hence, it holds that

$$V_k V_k^T H_{k+1} V_k V_k^T \succeq -3\sqrt{\epsilon} I,$$

which indicates that H_{k+1} is approximately positive semi-definite in the subspace $\mathcal{L}_k$. ■

We now ready to show Theorem 2. According to Lemma 6, when $\lambda_k \leq \sqrt{\epsilon}$, we already obtain an iterate that approximately satisfies the first-order condition, and the second-order condition in certain subspace. On the other hand, when $\lambda_k > \sqrt{\epsilon}$, Lemma 5 indicates that the objective function has a amount of decrease $\frac{2}{3M^2}\epsilon^{3/2}$ at every iteration k. Note that the total amount of decrease cannot exceed $f(x_0) - f_{\inf}$. Therefore, the number of iterations with $\lambda_k > \sqrt{\epsilon}$ is upper bounded by

$$O\left(\frac{3}{2}M^2(f(x_0) - f_{\inf})\epsilon^{-3/2}\right),$$

which thus is also the iteration bound of our algorithm.

B.3. Proof of Theorem 3

Lemma 7 *Let V_k be the orthonormal bases for $\mathcal{L}_k$, suppose α_k is the solution to the normalized problem (11). Let $d_{k+1} = V_k\alpha_k$, then the following inequality holds:*

$$\|d_{k+1}^{SN} - d_{k+1}\| \leq \frac{1}{\mu}\lambda_k\|d_{k+1}\|, \tag{22}$$

where λ_k is the Lagrangian multiplier associated with the trust region constraint in (11), and $d_{k+1}^{SN} = V_k\alpha_k^{SN}$ is the subspace Newton step with α_k^{SN} defined by:

$$\alpha_k^{SN} = \arg\min_{\alpha\in\mathbb{R}^2} \; f(x_k) + \alpha^T V_k^T g_k + \frac{1}{2}\alpha^T V_k^T H_k V_k \alpha. \tag{23}$$

Proof Since α_k is a solution to the normalized problem (11), the optimality condition gives that

$$(V_k^T H_k V_k + \lambda_k I)\alpha_k = -g_k^T V_k.$$

As α_k^{SN} is a solution to problem (22), due to optimality condition it holds that

$$V_k^T H_k V_k \alpha_k^{SN} = -g_k^T V_k.$$

Combining the above two equations yields that

$$V_k^T H_k V_k(\alpha_k - \alpha_k^{SN}) = -\lambda_k\alpha_k. \tag{24}$$

Moreover, note that

$$\text{for any } \alpha \neq 0, \text{ we have } V_k\alpha \neq 0 \text{ and } \|V_k\alpha\| = \|\alpha\|, \tag{25}$$

which combined with $H_k \succeq \mu I$ implies that

$$\alpha^T V_k^T H_k V_k \alpha \geq \mu\|V_k\alpha\|^2 = \mu\|\alpha\|^2.$$

Therefore $V_k^T H_k V_k \succeq \mu I_2$ holds (also implies that $V_k^T H_k V_k$ is nonsingular), it follows that

$$\|\alpha_k^{SN} - \alpha_k\| \leq \|(V_k^T H_k V_k)^{-1}\|\|V_k^T H_k V_k(\alpha_k^{SN} - \alpha_k)\| \leq \frac{1}{\mu}\lambda_k\|\alpha_k\|, \tag{26}$$

where the second inequality is due to (24). Therefore, by combining the construction of d_{k+1}^{SN} and d_{k+1} with (25) we conclude that

$$\|d_{k+1}^{SN} - d_{k+1}\| = \|\alpha_k^{SN} - \alpha_k\| \leq \frac{1}{\mu}\lambda_k\|\alpha_k\| = \frac{1}{\mu}\lambda_k\|d_{k+1}\|. \tag{27}$$

■

We now ready to provide the following key result to analyze the local convergence rate of our algorithm, where we assume it converges to a strict local optimum x^* such that $H(x^*) \succeq \mu I$ for some $\mu > 0$.

Lemma 8 *Suppose the iterate of DRSOM x_k converges to x^* which satisfies $H(x^*) \succeq \mu I$, when x_k is sufficiently close to x^*, then we have:*

$$\|x_{k+1} - x^*\| \leq \frac{M}{\mu}\|x_k - x^*\|^2 + \frac{1}{\mu}\|(H_k - \tilde{H}_k)d_{k+1}\| + \left(\frac{2L}{\mu^2} + \frac{1}{\mu}\right)\lambda_k\|d_{k+1}\|. \tag{28}$$

Proof We first write

$$\begin{aligned}\|x_{k+1} - x^*\| &= \|x_k + d_{k+1} - x^*\| \\ &\leq \|x_k + d_{k+1}^N - x^*\| + \|d_{k+1}^{SN} - d_{k+1}^N\| + \|d_{k+1} - d_{k+1}^{SN}\|,\end{aligned} \tag{29}$$

where $d_{k+1}^N = -H_k^{-1}g_k$ is the standard Newton step and $d_{k+1}^{SN} = V_k\alpha_k^{SN}$ is the subspace Newton step with α_k^{SN} defined by (23). The first term in (29) is upper bounded by $\frac{M}{\mu}\|x_k - x^*\|^2$ due to the standard analysis in Newton's method. To bound the second term, we note that $\alpha_k^{SN} = (V_k^T H_k V_k)^{-1}V_k^T g_k$ with $V_k^T V_k = I$ as it is a solution to problem (23), which implies that

$$\tilde{H}_k d_{k+1}^{SN} = \tilde{H}_k V_k (V_k^T H_k V_k)^{-1}V_k^T g_k = V_k V_k^T H_k V_k V_k^T V_k (V_k^T H_k V_k)^{-1}V_k^T g_k = V_k V_k^T g_k = g_k,$$

where the last equality is due to $V_k V_k^T$ is the projection matrix of the subspace $\mathcal{L}_k$ and $g_k \in \mathcal{L}_k$. Then, the second term can be further bounded above as follows

$$\begin{aligned}\|d_{k+1}^{SN} - d_{k+1}^N\| &= \|d_{k+1}^{SN} + H_k^{-1}g_k\| \\ &= \|d_{k+1}^{SN} - H_k^{-1}\tilde{H}_k d_{k+1}^{SN}\| \\ &= \|H_k^{-1}(H_k - \tilde{H}_k)d_{k+1}^{SN}\| \\ &\leq \|H_k^{-1}\|\|(H_k - \tilde{H}_k)d_{k+1}^{SN}\| \\ &\leq \frac{1}{\mu}\|(H_k - \tilde{H}_k)d_{k+1}^{SN}\| \\ &\leq \frac{1}{\mu}\left(\|(H_k - \tilde{H}_k)d_{k+1}\| + \|(H_k - \tilde{H}_k)(d_{k+1}^{SN} - d_{k+1})\|\right) \\ &\leq \frac{1}{\mu}\|(H_k - \tilde{H}_k)d_{k+1}\| + \frac{1}{\mu}\|H_k - \tilde{H}_k\|\|(d_{k+1}^{SN} - d_{k+1})\|,\end{aligned} \tag{30}$$

Combining the above inequalities, we have that

$$\begin{aligned}\|x_{k+1} - x^*\| &\leq \frac{M}{\mu}\|x_k - x^*\|^2 + \frac{1}{\mu}\|(H_k - \tilde{H}_k)d_{k+1}\| + \left(\frac{1}{\mu}\|H_k - \tilde{H}_k\| + 1\right)\|d_{k+1} - d_{k+1}^{SN}\| \\ &\leq \frac{M}{\mu}\|x_k - x^*\|^2 + \frac{1}{\mu}\|(H_k - \tilde{H}_k)d_{k+1}\| + \left(\frac{2L}{\mu^2} + \frac{1}{\mu}\right)\lambda_k\|d_{k+1}\|,\end{aligned} \tag{31}$$

where the last inequality follows from Lemma 7 and $\max\{\|H_k\|, \|\tilde{H}_k\|\} \leq L$ due to the Lipschitz continuity of the gradient. ■

Now we are ready to complete the proof of Theorem 3.

Proof It suffices to further upper bound (28). We only consider the scenario that $\lambda_k \leq \sqrt{\epsilon}$, as otherwise the objective function has a amount of decrease $\frac{2}{3M^2}\epsilon^{3/2}$ at every iteration k by Lemma 5 and this will occur in a very limited times when x_k is sufficiently close to x^*.

Since (8) holds, one has that

$$\frac{1}{\mu}\|(H_k - \tilde{H}_k)d_{k+1}\| \leq \frac{C}{\mu}\|d_{k+1}\|^2.$$

Moreover, as we adopt the fixed radius strategy, $\lambda_k = 0$ whenever $\|d_{k+1}\| < \Delta$. In the case of $0 < \lambda_k \leq \sqrt{\epsilon}$, we have $\lambda_k \leq \sqrt{\epsilon} = \frac{M}{2}\Delta = \frac{M}{2}\|d_{k+1}\|$. Therefore, in both cases, we have $\lambda_k\|d_{k+1}\| \leq \frac{M}{2}\|d_{k+1}\|^2$. Consequently, (28) can be bounded above by

$$\|x_{k+1} - x^*\| \leq \frac{M}{\mu}\|x_k - x^*\|^2 + \frac{C}{\mu}\|d_{k+1}\|^2 + \left(\frac{2L}{\mu^2} + \frac{1}{\mu}\right)\frac{M}{2}\|d_{k+1}\|^2.$$

Note that

$$\begin{aligned}
\|d_{k+1}\| &\leq \|x_k - x^* + d_{k+1}\| + \|x_k - x^*\| \\
&= \|x_{k+1} - x^*\| + \|x_k - x^*\| \\
&\leq \frac{M}{\mu}\|x_k - x^*\|^2 + \frac{C}{\mu}\|d_{k+1}\|^2 + \left(\frac{2L}{\mu^2} + \frac{1}{\mu}\right)\frac{M}{2}\|d_{k+1}\|^2 + \|x_k - x^*\| \\
&\leq \frac{M}{\mu}\|x_k - x^*\|^2 + \|x_k - x^*\| + O(\|d_{k+1}\|^2).
\end{aligned}$$

By rearranging the terms, we have

$$\|d_{k+1}\| - O(\|d_{k+1}\|^2) \leq \frac{M}{\mu}\|x_k - x^*\|^2 + \|x_k - x^*\|. \tag{32}$$

From the assumption x_k converges to x^*, it holds that $d_{k+1} \to 0$, when k is sufficiently large. Thus, inequality (32) implies $\|d_{k+1}\| \leq \|x_k - x^*\|$, i.e. $\|d_{k+1}\| = O(\|x_k - x^*\|)$, which in return shows that $\|x_{k+1} - x^*\| \leq O(\|x_k - x^*\|^2)$. ■

B.4. Detailed discussion on Assumption 2

According to Lemma 6 and the last part of the proof for Theorem 3, Assumption 2 is required only when $\lambda_k \leq \sqrt{\epsilon}$. When the violation of Assumption 2 is identified for some $\lambda_k \leq \sqrt{\epsilon}$, we apply the Lanczos method [5, 8, 10] to generate a larger subspace to form an approximated Hessian that satisfies Assumption 2 and the algorithm terminates if the resulting $\lambda_{k+1} \leq \sqrt{\epsilon}$. However, it is possible that the expanded subspace due to the Lanczos method, in return produces a larger $\lambda_{k+1} > \sqrt{\epsilon}$ and in this case we check Assumption 2 and repeat the above procedure when λ_k drops below $\sqrt{\epsilon}$ again. Therefore, we call Lanczos method periodically whenever $\lambda_k \leq \sqrt{\epsilon}$. Fortunately, for every iteration with $\lambda_k \leq \sqrt{\epsilon}$, the objective function has a amount of decrease $\frac{2}{3M^2}\epsilon^{3/2}$ by Lemma 5. Thus the number of times to invoke Lanczos method is very limited when x_k is close to convergence.

Appendix C. Extended Numerical Results

To demonstrate the efficacy of DRSOM, we implement the algorithm using the Julia programming language for convenient comparisons to first- and second-order methods. Most of the experiments, except the neural networks, are handled by the Julia version on a desktop of Mac OS with a 3.2 GHz 6-Core Intel Core i7 processor. The competing algorithms, including the (accelerated) gradient descent method, LBFGS, and Newton trust-region method, are computed via a third-party package `Optim.jl`[1], including a set of line search algorithms in `LineSearches.jl`[2].

We also implement a version in PyTorch that enables experiments in neural network training. For this part, the DRSOM runs on a Ubuntu desktop with Intel Xeon CPU E5-2698 v4 processor and 1 NVIDIA Tesla V100. We provide a comparison of SGD and Adam. Note the SGD and Adam optimizer used in our experiments is provided by the official implementation of PyTorch[3].

A complete description of "Trust-region" style Algorithm 1 is given in Algorithm 2. We use the standard updating mechanism for Δ_k as in [8].

Algorithm 2: A DRSOM algorithm using trust-region updates

Data: Given $k_{\max}, \beta_1 < 1 < \beta_2, \zeta_1 < \zeta_2 \leq 1; \bar{\Delta} > 0, \Delta_0 \in (0, \bar{\Delta})$, and $\eta \in [0, \zeta_1)$;
for $k = 1, ..., k_{\max}$ **do**
  Solve (3) for α_k, obtain $d_{k+1} = -\alpha_k^1 g_k + \alpha_k^2 d_k$, and ρ_k;
  if $\rho_k \leq \zeta_1$ **then**
    Decrease $\Delta_{k+1} = \beta_1 \Delta_k$
  else
    if $\rho_k > \zeta_2$ and $\|d_{k+1}\| = \Delta_k$ **then**
      $\Delta_{k+1} = \min\left\{\beta_2 \Delta_k, \bar{\Delta}\right\}$
    else
      $\Delta_{k+1} = \Delta_k$
    end
  end
  if $\rho_k > \eta$ **then**
    $x_{k+1} = x_k + d_{k+1}$
  else
    $x_{k+1} = x_k$
  end
end

To use a "Radius-Free" DRSOM, we briefly describe a strategy to update μ_k. Note that the "Radius-Free" parameter μ_k is designed as a wild estimate to λ_k for (3). We adjust μ_k by the following rule:

1. For details, see `https://github.com/JuliaNLSolvers/Optim.jl`
2. For details, see `https://github.com/JuliaNLSolvers/LineSearches.jl`
3. For details, see `https://pytorch.org/docs/stable/generated/torch.optim.Adam.html`

Let $\mu_1 \le \mu_2$ be the eigenvalues of H_k in the subspace, consider a simple adaptive rule to put μ_k in a desired interval $[\underline{\mu}_k, \overline{\mu}_k]$:

$$\begin{aligned} \underline{\mu}_k &= \max\{0, -\mu_1\},\ \overline{\mu}_k = \max\{\underline{\mu}_k, \mu_2\} + \mu_M \\ \gamma_{k+1} &= \begin{cases} \beta_2 \gamma_k, & \rho_k \le \zeta_1 \\ \max\{\underline{\gamma}, \min\left\{\sqrt{\gamma_k}, \beta_1 \gamma_k\right\}, & \rho_k > \zeta_2\} \end{cases} \\ \mu_k &= \gamma_k \cdot \overline{\mu}_k + \max\left\{1 - \gamma_k, 0\right\} \cdot \underline{\mu}_k \end{aligned} \tag{33}$$

where $\gamma_k > 0$, μ_M is a big number to bound μ_k from above, and $\underline{\gamma}$ is the minimum level for γ_k. Furthermore, we also let $\beta_1 < 1 < \beta_2$ in the same spirit of Algorithm 2. Specifically, we increase γ_k if the model reduction is not accurate according to ρ_k.

In the above procedure (33), we adjust μ_k by γ_k instead, which is expected to be less affected by the eigenvalues. If γ_k approaches to 0, then μ_k is close to $\underline{\mu}_k$ which implies $\tilde{H}_k$ is almost positive semi-definite. Otherwise, γ_k induces a large μ_k so to give a small trust-region radius.

C.1. Logistic Regression

We consider a multinomial logistic regression model for the MNIST dataset. The training set contains $60,000$ pictures for handwritten digits; the test set has $10,000$ pictures. We present the classification performance of DRSOM for 10 and 40 epochs in comparison to a popular stochastic first-order method, SAGA, and a second-order method, LBFGS. Specifically, we run DRSOM and LBFGS in full-batch. Then, we collect the zero-one classification loss of training (training error) and test data (test error) in Table 1. These results show that DRSOM is comparable to SAGA and LBFGS.

Epoch	Method	Training error	Testing error
10	SAGA	0.0699	0.0779
10	LBFGS	0.1245	0.1175
10	DRSOM	0.1149	0.1076
40	SAGA	0.0690	0.0759
40	LBFGS	0.0750	0.0783
40	DRSOM	0.0760	0.0819

Table 1: Performance of DRSOM on MNIST classification compared to other algorithms

C.2. $\mathcal{L}_2 - \mathcal{L}_p$ Minimization

We next test the performance of DRSOM for nonconvex $\mathcal{L}_2 - \mathcal{L}_p$ minimization. Recall $\mathcal{L}_2 - \mathcal{L}_p$ minimization problem (Chen [6], Chen et al. [7], Ge et al. [12]):

$$\phi^* = \min_{x \in \mathbb{R}^m} \phi(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_p^p, \tag{34}$$

where $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n, 0 < p < 1$. To overcome nonsmoothness of $\|\cdot\|_p$, we apply the smoothing strategy mentioned in [6]:

$$f(x) = \|Ax - b\|_2^2 + \lambda \sum_{i=1}^{n} s(x_i, \varepsilon)^p, \tag{35}$$

where $s(x_i, \varepsilon)$ is a smoothed approximation of $|x_i|$ and ε is a small pre-defined constant $\varepsilon > 0$:

$$s(x, \varepsilon) = \begin{cases} |x| & \text{if } |x| > \varepsilon \\ \frac{x^2}{2\varepsilon} + \frac{\varepsilon}{2} & \text{if } |x| \le \varepsilon \end{cases} \tag{36}$$

We randomly generate datasets with different sizes n, m based on the following procedure. The elements of matrix A are generated by $A_{ij} \sim \mathcal{N}(0, 1)$ with 15% sparsity of 15%. To construct the true sparse vector $v \in \mathbb{R}^m$, we let for all i:

$$v_i \sim \begin{cases} 0 & \text{with probability } p = 0.5 \\ \mathcal{N}\left(0, \frac{1}{n}\right) & \text{otherwise} \end{cases}$$

Then we let $b = Av + \delta$ where δ is the noise generated as $\delta_i \sim \mathcal{N}(0, 1), \forall i$. The parameter λ is chosen as $\frac{1}{5}\|A^T b\|_\infty$. We generate instances for (n, m) from $(100, 10)$ to $(1000, 100)$. We set $p = 0.5$ and the smoothing parameter $\varepsilon = 1e^{-1}$.

Next, we test the performance of DRSOM and competing algorithms, including a first-order representative AGD, and two second-order methods, including LBFGS and the Newton trust-region method (Newton-TR). The AGD is facilitated with the Zhang-Hager line-search algorithm (see [25]). We report the iteration number needed to reach a first-order stationary point at a precision of $1e^{-6}$, precisely,

$$|\nabla f(x_k)| \le \epsilon := 1e^{-6}$$

The iterations needed for a set of methods are reported in the Table 2. These results show that

n	m	DRSOM	AGD	LBFGS	Newton TR
100	10	18	43	14	6
100	20	31	72	23	7
100	100	47	136	42	10
200	10	21	27	15	5
200	20	23	45	21	6
200	100	40	131	39	9
1000	10	13	16	9	4
1000	20	16	23	13	5
1000	100	19	32	16	5

Table 2: Performance of DRSOM on (34) compared to other algorithms: iterations needed for precision $\epsilon = 1e^{-6}$

the DRSOM is fairly close to the full-dimensional second-order methods, especially the original Newton trust-region method; it is far better than AGD in most test cases.

C.3. Sensor Network Localization

We next visit another nonconvex optimization problem, namely, the Sensor Network Localization (SNL). The SNL problem is to find coordinates of ad hoc wireless sensors given pairwise distances in the network. Fruitful research has been found for this problem, among which the approach based on Semidefinite Programming Relaxation (SDR) has witnessed great success; see, for example, Biswas and Ye [1], Wang et al. [19], and many others. We here use the notations in [19].

Let n sensors be points in $\mathbb{R}^d$, besides, assume another set of m *known* points (usually referred to as anchors) whose exact positions are $a_1, ..., a_m$. Let d_{ij} be the distance between sensor i and j, and $\bar{d}_{ik}$ be the distance from the sensor i to anchor point k. We can then define the set of distances as edges in the network:

$$N_x = \{(i,j) : \|x_i - x_j\| = d_{ij} \le r_d\}, N_a = \{(i,k) : \|x_i - a_k\| = d_{ik} \le r_d\}, \tag{37}$$

where r_d is a fixed parameter known as the *radio range*. The SNL problem considers the following quadratic constrained quadratic programming (QCQP) feasibility problem,

$$\begin{aligned} \|x_i - x_j\|^2 &= d_{ij}^2, \forall (i,j) \in N_x \\ \|x_i - a_k\|^2 &= \bar{d}_{ik}^2, \forall (i,k) \in N_a \end{aligned} \tag{38}$$

Since the problem is nonconvex, the SDR approaches the above problem by a two-stage strategy. In the first step, we use semidefinite programming to solve a lifted convex relaxation:

$$\begin{aligned} \min\ & 0 \bullet Z \\ \text{s.t.}\ & Z_{[1:2,1:2]} = I, \\ & (0; e_i - e_j)(0; e_i - e_j)^T \bullet Z = d_{ij}^2 \quad \forall (i,j) \in N_x, \\ & (-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2 \quad \forall (i,k) \in N_a \\ & Z \succeq 0. \end{aligned} \tag{39}$$

We let I be the identity matrix of dimension 2, e_i be a n-vector of zeros except for a one at i-th entry. Z is the positive semidefinite matrix, such that,

$$Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix}, \tag{40}$$

which is equivalent to state $Y \succeq X^T X$. If $\mathsf{rank}(Y) = 2$, the SDR solves the original problem; otherwise (Y, X) provides initial solution and needs further refinement. For example, at the second stage, we can solve the following nonlinear least-square problem (NLS):

$$\min_X \sum_{(i<j,j) \in N_x} (\|x_i - x_j\|^2 - d_{ij}^2)^2 + \sum_{(k,j) \in N_a} (\|a_k - x_j\|^2 - \bar{d}_{kj}^2)^2. \tag{41}$$

We may find local solutions by a gradient descent method (GD); see, for example, [1]. Alternatively, we apply the DRSOM to the sensor network localization problem. We here provide a randomly generated example with 80 points, 5 of which are anchors. We set the radio range to 0.5 and the random distance noise $n_f = 0.05$. We add a line search for GD that guarantees the strong Wolfe condition, see [18, p. 60]. We terminate at an iterate x_k if $\|g(x_k)\| \le 1e^{-6}$.

Our results basically show that: if we initialize the NLS problem (41) for SNL by the SDR (39), then DRSOM and GD are comparable. However, if we do not have the SDR solution at hand, the DRSOM may usually provides better solutions than GD, which shows the benefit of the second-order optimality condition.

Figure 1 illustrates the realization results of GD and DRSOM with the SDR initialization. In this case, both algorithms are able to guarantee convergence to the ground truth.

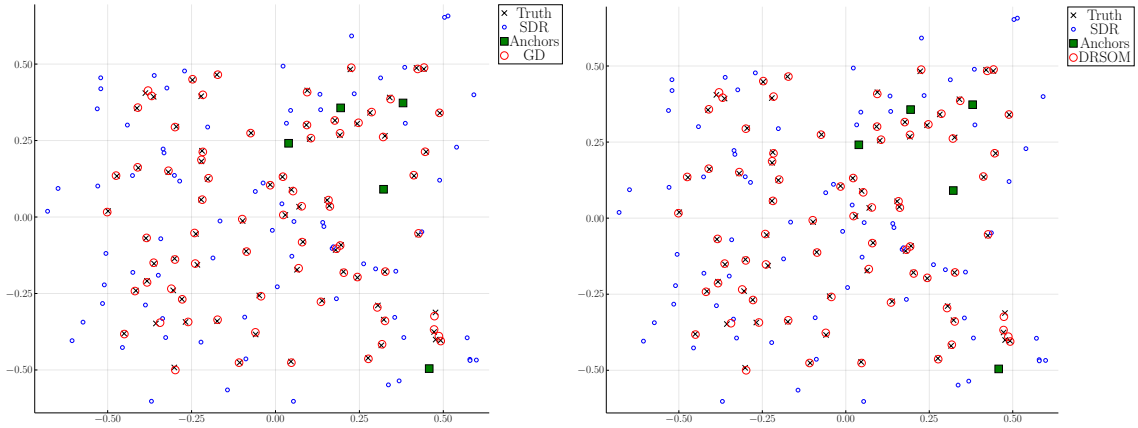

Figure 1: Comparison of localization by GD and DRSOM with SDR initialization. The rectangles and **crosses** represent the anchors and true locations, respectively. The blue circles are solutions by SDR, and the red circles are final solutions of GD/DRSOM.

As a comparison, Figure 2 depicts the case without solving SDR first. We use the same parameter settings as for the previous case. The GD and DRSOM are initialized by $X_i := 0, i = 1, ..., n$. In this particular case, the GD fails to recover true positions; to our experience, it converges to a strict local minimum x^* such that $H(x^*) \succ 0$. However, the DRSOM can sometimes provide accurate solutions even without the SDR initialization. In this example, we rigorously provide a case where the DRSOM may result in better local results (in this case, the global one) than the first-order methods; despite that, we only have optimal subspace guarantees in theory.

C.4. Neural Networks

In this section, we implement a vanilla Mini-Batch DRSOM to train neural networks. Our implementation is straightforward: for each mini-batch, we calculate the required gradients and Hessian-vector products, then the computation proceeds just like the "full-batch" version. Finally, we test our Mini-Batch DRSOM optimizer and compare the performance with the SGD and Adam optimizer.

C.4.1. FASHION-MNIST

We train a Neural Network model for classification on Fashion-MNIST, consisting of a training set of 60,000 examples and a test set of 10,000 examples [20]. The dataset is constructed from images with one of 10 labels, including `T-shirt/top`, `bag`, `dress` and so on[4], . In our test,

4. For details, see `https://github.com/zalandoresearch/fashion-mnist`

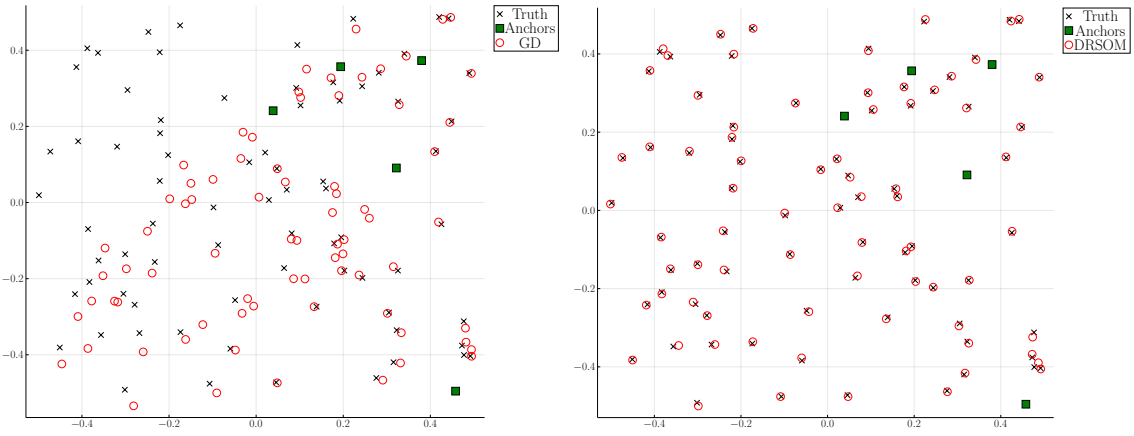

Figure 2: Comparison of localization by GD and DRSOM without SDR initialization. The meaning of each symbol is the same as Figure 1.

we adopt a neural network model with two convolutional and two fully-connected layers[5], which has about 16.84 million parameters. For SGD, we test on different momentum coefficients $\mu \in \{0.85, 0.9, 0.95, 0.99\}$ named after SGD$-\mu$. Also, the Adam and SGD are set with the learning rate at $1e^{-3}$. All optimizers are tested with a batch size of 128.

From Figure 3, we see that in 20 epochs, the Adam, SGD (SGD-μ), and DRSOM are both able to reach over 90% accuracy on the hold-out test set. The DRSOM, even with a naive Mini-batch strategy, is the best in terms of loss and accuracy in the training set. It also has the best test accuracy compared to the competitors. In our preliminary experiments, the DRSOM is two times slower than Adam per-iteration in running this case.

C.4.2. CIFAR10

To further take a look in *deep* neural networks, we also train a ResNet18 model (He et al. [13]) for CIFAR10. In our preliminary experiments, the vanilla DRSOM is five times slower than Adam in running time at the same iteration number; thus, we only run DRSOM in 50 epochs. For Adam, we collect the results in 100 epochs. All optimizers are tested with a batch size of 128. For Adam, we provide a learning rate scheduler to decay the learning rate by a factor of 2 in every k epochs started at an initial rate $1e^{-3}$; we name these variants by Adam-k where $k \in \{30, 40\}$.

To enable fair comparison, for DRSOM, we increase the lower bound of $\underline{\gamma}$ by the rule, $\underline{\gamma} := \underline{\gamma} \cdot \sigma, \sigma \in \{100, 1000\}$, in every 10 epochs corresponding to the update policy (33). We apply the strategy to mimic a mechanism of reducing the learning rate. Using the same fashion, we call them DRSOM-σ. We report the results in Figure 4. As the results show, the DRSOM has a sharp rate of increase in the beginning; with little tuning efforts, DRSOM-$1e^3$ (in 50 epochs) has competitive results to Adam in 100 epochs. These preliminary results, to our belief, motivate future research and better implementation of the DRSOM.

5. For details, see `https://github.com/ashmeet13/FashionMNIST-CNN/blob/master/Fashion.py`

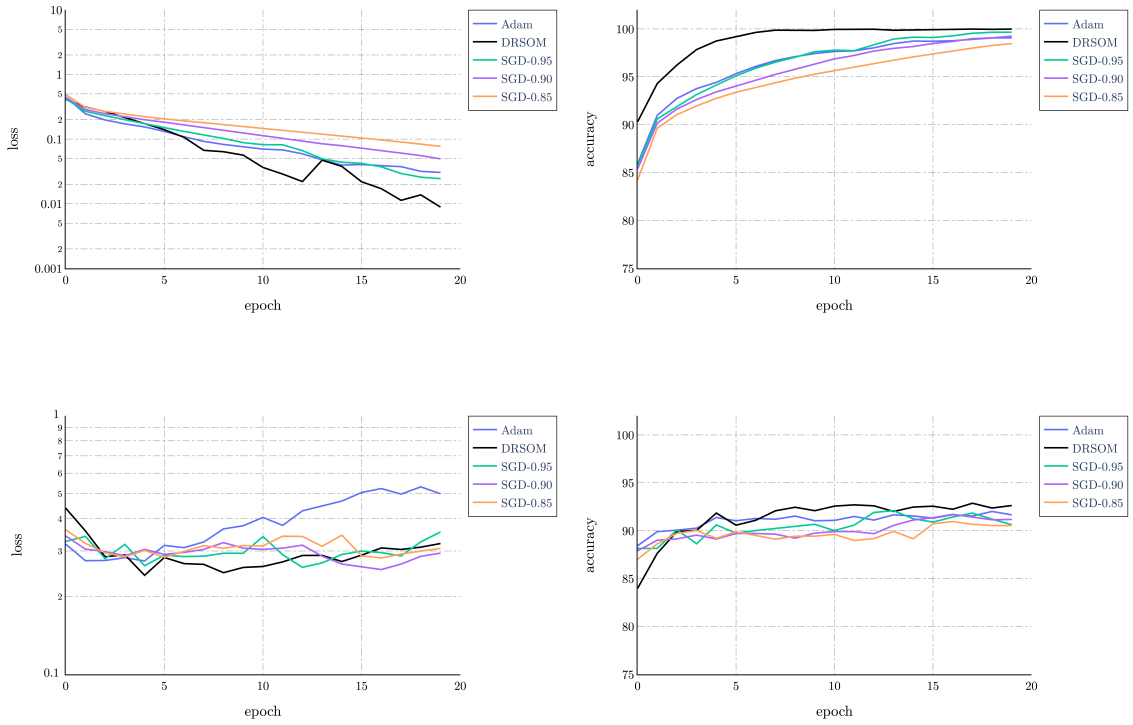

Figure 3: Training and test results of Adam, SGD, and DRSOM for a Neural network on Fashion-MNIST dataset

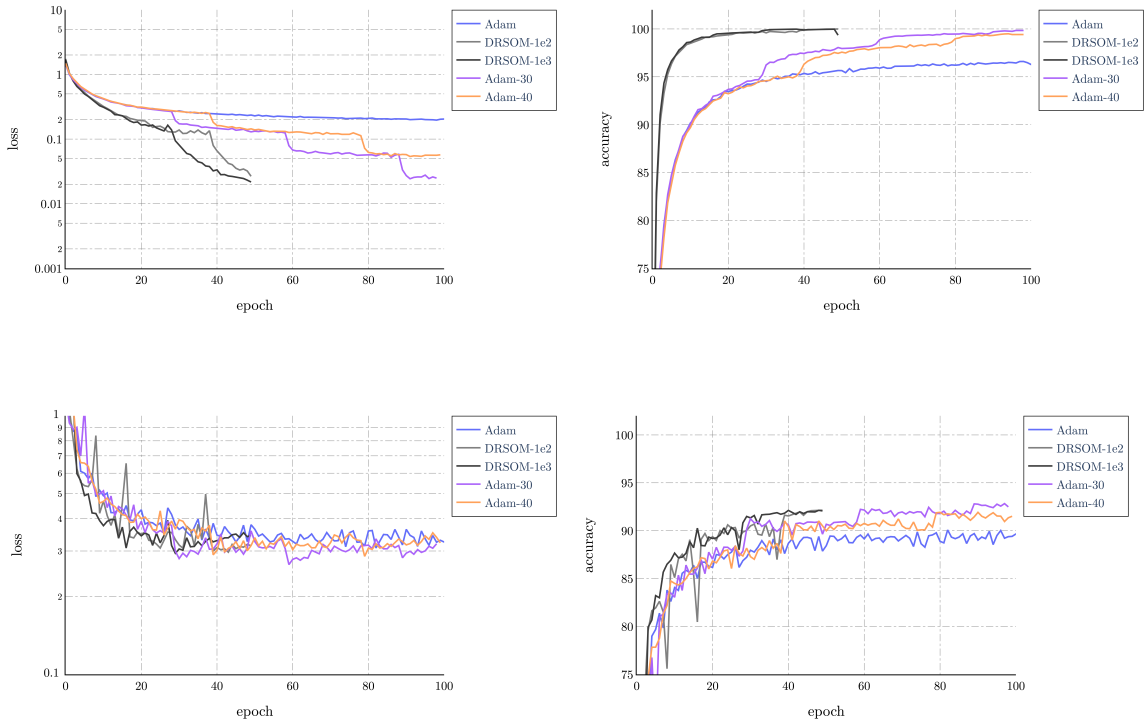

Figure 4: Training and test results of Adam and DRSOM for ResNet18 on CIFAR10