A Variant of Anderson Mixing with Minimal Memory Size

Fuchao Wei[1], **Chenglong Bao**[3,4*], **Yang Liu**[1,2], **Guangwen Yang**[1]
[1]Department of Computer Science and Technology, Tsinghua University, China
[2]Institute for AI Industry Research (AIR), Tsinghua University, China
[3]Yau Mathematical Sciences Center, Tsinghua University, China
[4]Yanqi Lake Beijing Institute of Mathematical Sciences and Applications

Abstract

Anderson mixing (AM) is a useful method that can accelerate fixed-point iterations by exploring the information from historical iterations. Despite its numerical success in various applications, the memory requirement in AM remains a bottleneck when solving large-scale optimization problems in a resource-limited machine. To address this problem, we propose a novel variant of AM method, called Min-AM, by storing only one vector pair, that is the minimal memory size requirement in AM. Our method forms a symmetric approximation to the inverse Hessian matrix and is proved to be equivalent to the full-memory Type-I AM for solving strongly convex quadratic optimization. Moreover, for general nonlinear optimization problems, we establish the convergence properties of Min-AM under reasonable assumptions and show that the mixing parameters can be adaptively chosen by estimating the eigenvalues of the Hessian. Finally, we extend Min-AM to solve stochastic programming problems. Experimental results on logistic regression and network training problems validate the effectiveness of the proposed Min-AM.

1 Introduction

Anderson mixing (AM) [2] is a powerful method for accelerating fixed-point iterations, and has been widely used in scientific computing [1, 5, 47] and machine learning [31, 58, 29]. In each iteration, AM extrapolates a new iterate that satisfies certain optimality property by using the historical iterations, and exhibits empirical acceleration over classical fixed-point iterations [60]. Thus, AM can reduce the computation cost when evaluating the fixed-point map is a time-consuming process. Also, the inspiring connections of AM with GMRES [60, 48] and multisecant quasi-Newton methods [22] provide some explanations for the acceleration benefit from AM. In recent years, there have been many works applying AM to solve various optimization problems [56, 62, 41]. Their promising results in different tasks suggest that these AM-based methods are very competitive while being more economical than Newton's method. However, the existing convergence theory of AM is still unsatisfactory [3] and AM can diverge in some case [41]. Consequently, the applications of AM and its variants for machine learning problems and the corresponding theories deserve deep research.

One major concern of AM is the heavier memory overhead compared with the fixed-point iteration. In AM(m), it has to store vector pairs about historical information from the m previous iterations, where m is the memory size. Specifically, AM(m) is called as the limited-memory AM if $m < \infty$, and full-memory AM if $m = \infty$. The AM(0) is the simple fixed-point iteration. In practice, choosing proper memory size m is heuristic and it is observed in [47] that a small m can deteriorate the efficacy. However, for practical applications, a large m can lead to memory issue for solving large-scale and high-dimensional problems when the memory resource is limited.

*Corresponding author. clbao@mail.tsinghua.edu.cn.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

In this paper, we address the memory issue of AM for solving optimization problems by proposing Min-AM, a novel variant of AM(1) that has minimal memory size in AM. Min-AM is built upon the Type-I AM [22, 60], a variant of AM. Let AM-I(m) denote the Type-I AM with memory size m. Using a projection-mixing framework of AM-I, the Min-AM retains a recursively modified vector pair and adds one extra projection step in each step of AM-I(1). Compared to AM(1), Min-AM forms symmetric approximations to the Hessian and is equivalent to AM-I(∞) when the objective function is strongly convex quadratic. By incorporating the restarting strategy, the restarted Min-AM has faster convergence rate than AM(1) for general nonlinear optimization and the spectrum of Hessian can be cheaply estimated that leads to adaptive choices of the mixing parameters. Min-AM is also extended to stochastic optimization. In summary, we highlight the main contributions of this work as follows.

- We propose Min-AM, a variant of AM(1) that achieves the minimal memory size in AM, for solving optimization problems. Compared to AM(1), Min-AM forms symmetric approximations to the Hessian and incorporates more historical information to obtain the update. Thus, Min-AM can significantly reduce the memory requirement and computational cost of AM while maintaining the fast convergence property.
- We show that the local convergence rate of the restarted Min-AM has optimal dependence on the condition number of the objective function. We provide adaptive choices of the mixing parameters by estimating the spectrum of the Hessian economically in each iteration. Also, we give the convergence and complexity analysis of a stochastic extension of Min-AM for solving nonconvex stochastic optimization.
- We verify the properties of Min-AM, and apply the restarted Min-AM to logistic regression and the stochastic Min-AM to train neural networks. The experimental results are consistent with our theoretical analysis and show that the Min-AM is competitive with the limited-memory quasi-Newton methods with large memory size in deterministic optimization and achieves promising results in training neural networks.

Related work. Fixed-point problems can be recast as solving systems of (non)linear equations. AM is a practical alternate for Newton's method when the Jacobian is unavailable or difficult to compute [35]. In the linear case, it is established in [60, 48] that the full-memory AM is equivalent to GMRES [51], a classical Krylov subspace method for linear systems. In the nonlinear case, AM is recognized as a multisecant quasi-Newton method and a variant, called Type-I AM, has been proposed in [22]. From the perspective of convergence analysis, it is proved that the local convergence rate of the limited-memory AM is no worse than that of fixed-point iteration [59, 16] and the potential improvement depends on the quality of extrapolation [21, 46]. However, a counter example [41] shows the divergence behaviour of AM(1) and the recent review [3] indicates the need of further research on the theoretical properties of AM.

The inspiring performance and the easy implementation of AM motivate many new variants of AM for various applications in machine learning. In [55], a regularized variant of AM is proposed to accelerate gradient descent for unconstrained optimization, which is adapted in [23] for nonsmooth convex optimization and in [58] for reinforcement learning. In [62], a stochastic version of AM is developed and has convergence guarantee for stochastic optimization. The Type-I AM is modified in [67] to solve nonsmooth fixed-point problems. A more recent work, the short-term recurrence AM (ST-AM) [63], stores two vector pairs to reduce the memory cost of AM. Our Min-AM method achieves the minimal memory size, i.e. $m = 1$, further reducing the memory size of ST-AM.

The memory overhead is a common issue of quasi-Newton methods. It is desirable to keep memory size as small as possible without sacrificing the efficacy of the algorithm. Some works [37, 9, 8] study the choice of memory size in L-BFGS [39], but the performance with minimal memory size is unsatisfactory. In stochastic programming, the sublinear convergence rate is optimal when only stochastic gradients can be accessed [43]. Thus the heavier memory and computational cost makes quasi-Newton methods [12, 27, 61, 10] less appealing than the more economical first-order methods [49, 36]. In this work, our proposed Min-AM method exhibits the fast convergence as quasi-Newton methods while the memory cost is close to first-order methods.

Notations. Let Δ denote the forward difference operator, e.g. $\Delta x_k = x_{k+1} - x_k$. $\mathrm{range}(X)$ denotes the subspace spanned by the columns of a matrix X. For every matrix $A \in \mathbb{R}^{d\times d}$, the Krylov subspace $\mathcal{K}_m(A, v) \equiv \mathrm{span}\{v, Av, \ldots, A^{m-1}v\}$, and $\|x\|_A = (x^{\mathrm{T}}Ax)^{1/2}$ denotes the A-norm when A is symmetric positive definite (SPD). "$\dagger$" is the Penrose-Moore inverse.

2 The Anderson mixing scheme

In this work, we consider the minimization problem

$$\min f(x),\ x \in \mathbb{R}^d, \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is continously differentiable. The first-order optimality condition of (1) is equivalent to $x = g(x)$, where $g(x) = x - \nabla f(x)$. We describe the Anderson mixing scheme with the projection-mixing framework [62]. Define $r_k = g(x_k) - x_k = -\nabla f(x_k)$ to be the residual. AM(m) maintains m ($m \leq \min\{d, k\}$) vector pairs stored in $X_k, R_k \in \mathbb{R}^{d\times m}$:

$$X_k = [\Delta x_{k-m}, \Delta x_{k-m+1}, \cdots, \Delta x_{k-1}],\ \ R_k = [\Delta r_{k-m}, \Delta r_{k-m+1}, \cdots, \Delta r_{k-1}]. \tag{2}$$

Each update can be decoupled into two steps, namely the *projection step* and the *mixing step*:

$$\bar{x}_k = x_k - X_k\Gamma_k, \quad \text{(Projection step)},\ \ x_{k+1} = \bar{x}_k + \beta_k\bar{r}_k, \quad \text{(Mixing step)}, \tag{3}$$

where $\bar{r}_k := r_k - R_k\Gamma_k$, $\beta_k > 0$ is the mixing parameter, and $\Gamma_k = \arg\min_{\Gamma\in\mathbb{R}^m} \|r_k - R_k\Gamma\|_2$. It is easy to know that the Galerkin's projection condition $\bar{r}_k \perp \mathrm{range}(R_k)$ holds. Assuming $R_k^{\mathrm{T}}R_k$ is nonsingular, the resulting one-step update of AM(m) is $x_{k+1} = x_k + H_k r_k$ where

$$H_k = \beta_k I - (X_k + \beta_k R_k)(R_k^{\mathrm{T}}R_k)^{-1}R_k^{\mathrm{T}}. \tag{4}$$

The AM-I [22, 60] falls into the same projection-mixing framework, and the only difference lies in the choice of Γ_k. The Γ_k of AM-I is determined by another Galerkin's condition:

$$\bar{r}_k^{\mathrm{T}}X_k = (r_k - R_k\Gamma_k)^{\mathrm{T}}X_k = 0. \tag{5}$$

Substituting the solution of (5) into (3), the approximated inverse Hessian matrix H_k of AM-I is

$$H_k = \beta_k I - (X_k + \beta_k R_k)(X_k^{\mathrm{T}}R_k)^{-1}X_k^{\mathrm{T}}, \tag{6}$$

by assuming $X_k^{\mathrm{T}}R_k$ is nonsingular. AM-I has a close relation to Newton's method and an alternative interpretation of AM-I is given in Appendix A.1.

Remark 1. *It is established in [22] that the H_k of AM solves $\min_{H_k} \|H_k - \beta_k I\|_F$ s.t. $H_kR_k = -X_k$ and the H_k of AM-I satisfies $H_k = B_k^{-1}$, where B_k solves $\min_{B_k} \|B_k - \beta_k^{-1}I\|_F$ s.t. $B_kX_k = -R_k$. Thus, both AM and AM-I are multisecant methods. Besides, from* (4) *and* (6)*, the approximated inverse Hessian matrices H_k are generally not symmetric.*

3 The Min-AM methods

In each iteration, AM(m) has to store two matrices $X_k, R_k \in \mathbb{R}^{d\times m}$, which dramatically increases the memory burden in large-scale problems. To reduce the memory requirement, we consider the minimal memory case, i.e. $m = 1$. The proposed Min-AM is a variant of AM(1) and the convergence of Min-AM is established in this section.

3.1 The basic Min-AM

In this section, we consider the simple case of (1): the objective function is $f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x$ where A is SPD. Let $s_k := \Delta x_{k-1}, y_k := \Delta r_{k-1}$, and choose $m = 1$ in (4). The approximated inverse Hessian of AM(1) is $H_k = \beta_k I - (s_k + \beta_k y_k)(y_k^{\mathrm{T}}y_k)^{-1}y_k^{\mathrm{T}}$, which is non-symmetric. However, the Hessian matrix $\nabla^2 f$ is naturally symmetric. To achieve a better approximation of $\nabla^2 f$, we first recursively modify the vector pair s_k, y_k to incorporate more information from the previous iterations. Let $p_1 = \Delta x_0, q_1 = \Delta r_0 \in \mathbb{R}^d$. For $k \geq 2$, we construct p_k and q_k by

$$p_k = \Delta x_{k-1} - p_{k-1}\zeta_k, \quad q_k = \Delta r_{k-1} - q_{k-1}\zeta_k, \tag{7}$$

where $\zeta_k = (p_{k-1}^{\mathrm{T}}q_{k-1})^{-1}\Delta r_{k-1}^{\mathrm{T}}p_{k-1}$, assuming $p_{k-1}^{\mathrm{T}}q_{k-1} \neq 0$. Then, inspired by the Two-Grid Cycle method [53, Algorithm 13.2] in the multigrid techniques [11], we add an extra projection step to (3), and the resulting scheme is

$$x_k^{(1)} = x_k - p_k\Gamma_k^{(1)}, \qquad \text{(Projection step)} \tag{8a}$$

$$x_k^{(2)} = x_k^{(1)} + \beta_k r_k^{(1)}, \qquad \text{(Mixing step)} \tag{8b}$$

$$x_{k+1} = x_k^{(2)} - p_k\Gamma_k^{(2)}, \quad \text{(Projection step)} \tag{8c}$$

where $r_k^{(1)} := r_k - q_k\Gamma_k^{(1)}$ and $\beta_k > 0$. Define $r_k^{(2)} = r_k^{(1)} - \beta_k A r_k^{(1)}$ and $r_k^{(3)} = r_k^{(2)} - q_k\Gamma_k^{(2)}$. It can be verified that $r_k^{(1)}, r_k^{(2)}$, and $r_k^{(3)}$ are equal to the residuals at $x_k^{(1)}, x_k^{(2)}$, and x_{k+1}, respectively. The coefficients $\Gamma_k^{(1)}$ and $\Gamma_k^{(2)}$ are determined by applying the Galerkin's condition of AM-I:

$$r_k^{(1)} = r_k - q_k\Gamma_k^{(1)} \perp p_k, \quad r_k^{(3)} = r_k^{(2)} - q_k\Gamma_k^{(2)} \perp p_k. \tag{9}$$

Assume $p_k^{\mathrm{T}} q_k \neq 0$. The scheme formulated by (8a)-(8c) gives the x_{k+1} by

$$x_{k+1} = x_k + H_k r_k, \quad H_k = -\frac{p_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k} + \beta_k \left(I - \frac{p_k q_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k}\right)\left(I - \frac{q_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k}\right), \tag{10}$$

leading to the basic Min-AM. The derivation of (10) is in Appendix B.1. It is worth mentioning that the approximated inverse Hessian H_k in (10) is symmetric. Due to the space limit, we give the details of the basic Min-AM in Algorithm 2 in Appendix B. Define $P_k = (p_1, p_2, \ldots, p_k), Q_k = (q_1, q_2, \ldots, q_k)$, we summarize the properties of the basic Min-AM in the next theorem.

Theorem 1. *Let $f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x$ where A is SPD and $\{x_k\}$ be the sequence generated by the basic Min-AM for solving* (1)*. x^* is the exact solution. Then, the following properties hold:*
(i) $\|p_k\|_2 > 0, \mathrm{range}(P_k) = \mathcal{K}_k(A, r_0), \mathrm{range}(Q_k) = A\mathcal{K}_k(A, r_0)$;
(ii) $Q_k = -AP_k, p_i \perp q_j (1 \leq i \neq j \leq k)$;
(iii) $r_k^{(1)} \perp \mathrm{range}(P_k)$ *and* $x_k^{(1)} = x_0 + z_k$*, where* $z_k = \arg\min_{z \in \mathcal{K}_k(A, r_0)} \|x_0 + z - x^*\|_A$.
Moreover, if $\|r_k^{(1)}\|_2 = 0$*, then* $x_{k+1} = x^*$.

The proof is deferred to Appendix B.2. The property (i) ensures $p_k^{\mathrm{T}} q_k \neq 0$ during the iterations, so ζ_k and the update (10) are well defined. The property (iii) indicates $x_k^{(1)} = x_k^{\mathrm{CG}}$, where x_k^{CG} is the k-th iterate of the conjugate gradient (CG) method [32]. Also, due to Proposition 1 in Appendix B.2, the intermediate iterate $\bar{x}_k$ of AM-I(∞) satisfies $\bar{x}_k = x_k^{\mathrm{CG}}$, so the basic Min-AM is essentially equivalent to the full-memory AM-I when solving unconstrained strongly convex quadratic problems. In this sense, Min-AM can incorporate more historical information than AM(1).

Remark 2. *Although Min-AM and CG are equivalent for solving SPD linear systems, their numerical performance has a large difference for general nonlinear programming as shown in our experiments.*

Remark 3. *The H_k in* (10) *is reminiscent of the memoryless BFGS method [4], but the key difference lies in the construction of p_k and q_k in* (7)*. In memoryless BFGS, there are no correction terms ($\zeta_k \equiv 0$), and it is generally not equivalent to BFGS [45] and CG. See Appendix A.3 for more details.*

3.2 The restarted Min-AM

For general nonlinear optimization, since global convergence may be unavailable for Min-AM as a counter example exists for AM [41], we introduce the restarting strategy to the basic Min-AM and establish the local convergence analysis. Our purpose is to prove the faster convergence of the restarted Min-AM than that of gradient descent (GD), and sharpen the existing results of AM [59, 21].

Restarting Strategy. Let m_k be the number of iterations from the last restart. Min-AM restarts, i.e., setting $m_k = 0$ and $p_k = q_k = \mathbf{0}$, when any of the following conditions is violated:

$$m_k \leq m, \tag{11a}$$

$$|q_k^{\mathrm{T}} p_k| \geq \tau |q_{k-m_k+1}^{\mathrm{T}} p_{k-m_k+1}|, \tag{11b}$$

$$\|\nabla f(x_k)\|_2 \leq \eta \|\nabla f(x_{k-m_k})\|_2, \tag{11c}$$

where m is a constant positive integer, and $0 < \tau < 1, \eta > 0$ are constants. In the interval between two successive restarts, the iterations are the basic Min-AM iterations.

The check (11a) is to limit the number of high-order terms of errors accumulated during iterations. The check (11b) is for the numerical stability of Min-AM since $p_k^{\mathrm{T}} q_k$ appears in the denominators in (10). The check (11c) is to control divergence. The algorithm is shown in Algorithm 1. Let $\mathcal{B}_\rho(x) := \{y \in \mathbb{R}^d | \|y - x\|_2 \leq \rho\}$ denote the ball centered at x with radius ρ. We impose the following assumptions on f that are standard in the convergence analysis of quasi-Newton methods [45] and the regularized nonlinear acceleration method [56].

Assumption 1. *$f : \mathbb{R}^d \to \mathbb{R}$ is twice Lipschitz continuously differentiable in a local region of a local minimizer x^*, and the Hessian matrix $A := \nabla^2 f(x^*)$ is SPD.*

Algorithm 1 Restarted Min-AM for general nonlinear optimization

Input: $x_0 \in \mathbb{R}^d, \beta_k > 0, m > 0, \tau \in (0,1), \eta > 0$

$p_0, q_0 = \mathbf{0} \in \mathbb{R}^d, m_0 = 0$
for $k = 0, 1, \ldots,$ until convergence **do**
  $r_k = -\nabla f(x_k)$
  if $m_k > m$ **or** $\|r_k\|_2 > \eta \|r_{k-m_k}\|_2$ **then**
    $m_k = 0$
  end if
  if $m_k > 0$ **then**
    $p = x_k - x_{k-1}, q = r_k - r_{k-1}$
    $\zeta_k = (p_{k-1}^{\mathrm{T}} q_{k-1})^{\dagger} p_{k-1}^{\mathrm{T}} q$
    $p_k = p - p_{k-1}\zeta_k, q_k = q - q_{k-1}\zeta_k$
    if $|p_k^{\mathrm{T}} q_k| < \tau |p_{k-m_k+1}^{\mathrm{T}} q_{k-m_k+1}|$ **then**
      $m_k = 0$
    end if
  end if
  if $m_k = 0$ **then**
    $p_k = q_k = \mathbf{0} \in \mathbb{R}^d$
  end if
  $x_{k+1} = x_k - (p_k^{\mathrm{T}} q_k)^{\dagger} p_k p_k^{\mathrm{T}} r_k + \beta_k \left(I - (p_k^{\mathrm{T}} q_k)^{\dagger} p_k q_k^{\mathrm{T}}\right)\left(I - (p_k^{\mathrm{T}} q_k)^{\dagger} q_k p_k^{\mathrm{T}}\right) r_k$
  $m_{k+1} = m_k + 1$
end for
return x_k

From Assumption 1, there exist positive constants $\hat{\rho}, \mu, L$, and $\hat{\kappa}$ such that for all $x \in \mathcal{B}_{\hat{\rho}}(x^*)$,

$$\mu \leq \frac{y^{\mathrm{T}} \nabla^2 f(x) y}{y^{\mathrm{T}} y} \leq L, \text{ for all } y \in \mathbb{R}^d \text{ and } y \neq \mathbf{0}, \tag{12a}$$

$$\|\nabla f(x) - \nabla^2 f(x^*)(x - x^*)\|_2 \leq \frac{1}{2}\hat{\kappa}\|x - x^*\|_2^2. \tag{12b}$$

Here, μ is the strong convexity constant, and $L, \hat{\kappa}$ are Lipschitz constants of ∇f and $\nabla^2 f$ respectively.

With the restarting strategy, we apply a multi-step analysis and has the following convergence theory:

Theorem 2. *Suppose that the Assumption 1 holds. Let $\{x_k\}$ be the sequence generated by the restarted Min-AM for solving* (1) *and $\theta_k = \max\{|1 - \beta_k L|, |1 - \beta_k \mu|\} \leq \theta$ for a constant $\theta \in (0,1)$. Then there exists a positive constant $\rho \leq \hat{\rho}$ such that for any $x_0 \in \mathcal{B}_{\rho}(x^*)$, we have*

$$\|x_{k+1} - x^*\|_A \leq 2\theta_k \left(\frac{\sqrt{L/\mu} - 1}{\sqrt{L/\mu} + 1}\right)^{m_k} \|x_{k-m_k} - x^*\|_A + \hat{\kappa} \cdot \mathcal{O}(\|x_{k-m_k} - x^*\|_2^2). \tag{13}$$

Moreover, if $m_k = d$, then $\|x_{k+1} - x^*\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2)$.

The proof is in Appendix C.1, where a more general theorem (Theorem 6) is given. The result from Theorem 2 suggests that the restarted Min-AM method has a local linear-quadratic convergence rate, and the rate has optimal dependence on the condition number. Compared to the existing results that show the local linear convergence rate of AM is no worse than that of gradient descent (GD) method [59, 16], the bound (13) clearly improves the convergence rate of GD.

Remark 4. *Classical nonlinear CG methods [17] are also memory-efficient with recursively modified searching directions, but the step size relies on line-search or Hessian-vector products which can be time-consuming for solving large-scale optimization problems.*

Remark 5. *The momentum based methods, e.g. Nesterov's accelerated gradient (NAG) method [44], also achieve the optimal convergence rate if the parameters μ and L are known. In restarted Min-AM, we give an economical method that approximates the best step size $2/(L + \mu)$ in the next subsection.*

3.3 Eigenvalue estimates and the choice of mixing parameter

The mixing parameter β_k is critical for AM and AM-I, and an improper choice of β_k can lead to divergence [22]. Fortunately, the spectrum of the Hessian can be economically estimated as a

by-product of Min-AM. To better explain the idea, we consider $\min\ f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x$ where A is SPD. We apply a projection method [54] to estimate the eigenvalues:

$$v \in \mathcal{K}_k(A, r_0), \quad (A - \lambda I)v \perp A\mathcal{K}_k(A, r_0), \tag{14}$$

where $v \in \mathbb{R}^d$ is the approximate eigenvector sought in the Krylov subspace $\mathcal{K}_k(A, r_0)$, and $\lambda \in \mathbb{R}$ is an eigenvalue estimate, which is known as a generalized Ritz value [42]. Our goal is to compute λ. Since $\mathrm{range}(P_k) = \mathcal{K}_k(A, r_0)$ as shown in Theorem 1, we can select $v = P_k y,\ y \in \mathbb{R}^k$. Then the projection condition (14) leads to

$$P_k^{\mathrm{T}} A A P_k y = \lambda P_k^{\mathrm{T}} A P_k y. \tag{15}$$

From the construction of P_k and Q_k, we can derive the three-term recurrence relations (see details in Appendix D.1): there exists a tridiagonal matrix $\bar{T}_k \in \mathbb{R}^{(k+1)\times k}$, such that

$$AP_k = P_{k+1}\bar{T}_k, \quad AQ_k = Q_{k+1}\bar{T}_k. \tag{16}$$

Then, we have $P_k^{\mathrm{T}} A A P_k = P_k^{\mathrm{T}} A P_{k+1}\bar{T}_k$. By Theorem 1, $P_k^{\mathrm{T}} A p_{k+1} = -Q_k^{\mathrm{T}} p_{k+1} = 0$. Thus,

$$P_k^{\mathrm{T}} A P_{k+1}\bar{T}_k = P_k^{\mathrm{T}} A P_k T_k, \tag{17}$$

where $T_k \in \mathbb{R}^{k\times k}$ is obtained from $\bar{T}_k$ by deleting its last row. By (15) and (17), we obtain $T_k y = \lambda y$, which indicates that the generalized eigenvalue problem (15) is reduced to solving the eigenvalues of a tridiagonal matrix T_k. Moreover, the three elements of the k-th column of $\bar{T}_k$ are easy to obtain from the coefficients $\Gamma_k^{(1)}, \Gamma_k^{(2)}, \beta_{k-1}$, and β_k on the fly (see details in Appendix D.1):

$$t_k^{(k-1)} = \frac{\Gamma_{k-1}^{(2)}}{\beta_{k-1}(1-\Gamma_k^{(1)})},\ \ t_k^{(k)} = \frac{1}{1-\Gamma_k^{(1)}}\left(\frac{1}{\beta_{k-1}} - \frac{\Gamma_k^{(2)}}{\beta_k}\right),\ \ t_k^{(k+1)} = -\frac{1}{\beta_k(1-\Gamma_k^{(1)})}, \tag{18}$$

where $\Gamma_0^{(2)} := 0$. For solving general nonlinear optimization with the restarted Min-AM, suppose that $m_k \geq 1$. We can also construct a tridiagonal matrix $T_k \in \mathbb{R}^{m_k\times m_k}$ starting from the $(k-m_k)$-th iteration with the nonzero elements defined using the coefficients during the iterations (see Definition 2 in Appendix D.2). We use the eigenvalues of T_k as the eigenvalue estimates of the Hessian.

Theorem 3. *Suppose that Assumption 1 holds, and $\{x_k\}$ is the sequence generated by the restarted Min-AM for solving* (1)*. Assume that $\beta_j \in [\beta, \beta']$ ($\forall j \geq 0$) for some positive constants β and β', and there are positive constants η_0, τ_0 such that $\|\nabla f(x_j)\|_2 \leq \eta_0\|\nabla f(x_0)\|_2$ $(0 \leq j \leq k+1)$, $|1-\Gamma_j^{(1)}| \geq \tau_0$ $(1 \leq j \leq k)$; λ is an eigenvalue of T_k constructed by restarted Min-AM. Let $\Theta^{(m_k)}$ denote the set of Ritz values computed by an m_k-step A-norm based Lanczos algorithm (Algorithm 3 in Appendix D.2) for $\nabla f^2(x^*)$ with starting vector $\nabla^2 f(x^*)(x^* - x_{k-m_k})$. Then there is a positive constant $\rho \leq \hat{\rho}$ such that for any $x_0 \in \mathcal{B}_\rho(x^*)$, we have*

$$\min_{\hat{\lambda}\in\Theta^{(m_k)}} |\hat{\lambda} - \lambda| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{19}$$

The proof is in Appendix D.2. Since the Ritz value $\hat{\lambda}$ approximates a true eigenvalue of $\nabla^2 f(x^*)$ [54], Theorem 3 suggests that the restarted Min-AM can give reliable eigenvalue estimates if $\|x_0 - x^*\|_2$ is sufficiently small. At the k-th iteration, where $m_k \geq 2$, let $\tilde{\mu}$ and $\tilde{L}$ be the eigenvalues of T_{k-1} with the smallest absolute value and the largest absolute value, respectively. We use $|\tilde{\mu}|$ and $|\tilde{L}|$ to estimate μ and L. Then we can take the mixing parameter as $\beta_k = 2/(|\tilde{\mu}| + |\tilde{L}|)$ to estimate the optimal value $2/(\mu + L)$. The total computational cost is $\mathcal{O}(m_{k-1}^3)$ flops as it requires solving the eigenvalues of T_{k-1} [24]. Since $m_k \ll d$ in practice, this strategy can be a useful and economical option.

Remark 6. *The spectrum of Hessian is important for determining the step sizes. Once we obtain the estimates of μ and L by Min-AM, we can apply them to many first-order methods.*

3.4 The stochastic Min-AM

For many applications in deep learning, the large data size prohibits the evaluation of the full gradient, and the stochastic optimization stands out as a solution. Specifically, we consider developing a stochastic version of Min-AM to solve the nonconvex optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{T}\sum_{i=1}^{T} f_i(x), \tag{20}$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ may be nonconvex. Since T can be extremely large, we sample a mini-batch $S_k \subseteq [T] := \{1, 2, \ldots, T\}$ and use $f_{S_k}(x_k) := \frac{1}{n_k}\sum_{i\in S_k} f_i(x_k)$ to estimate the full gradient $\nabla f(x_k)$, where $n_k := |S_k|$ is the batch size. Then, $r_k := -\nabla f_{S_k}(x_k)$ is an estimate of $-\nabla f(x_k)$.

Due to the noise in the gradient estimates and the nonconvexity of f, a direct application of the deterministic Min-AM to (20) may be problematic. For example, the searching direction may be not a descent direction since H_k can be indefinite. Hence, we should introduce some fundamental modifications to the basic Min-AM.

Regularization. Inspired by [23], we define the function

$$\Phi(p, q, \delta) = p^{\mathrm{T}} q - \delta(\|p\|_2^2 + \|q\|_2^2). \tag{21}$$

Let $\Phi_k := \Phi(p_k, q_k, \delta_k^{(2)})$ for some $\delta_k^{(2)} > 0$. We substitute Φ_k for the $p_k^{\mathrm{T}} q_k$ appeared in (10) and define $\rho_k = \Phi_k^{\dagger}$, i.e. $\rho_k = 0$ if $\Phi_k = 0$ and $\rho_k = \Phi_k^{-1}$ if $\Phi_k \neq 0$. The resulting regularized Min-AM update is $x_{k+1} = x_k + H_k^A r_k$, where

$$H_k^A = -\rho_k p_k p_k^{\mathrm{T}} + \beta_k (I - \rho_k p_k q_k^{\mathrm{T}})(I - \rho_k q_k p_k^{\mathrm{T}}). \tag{22}$$

Damping. We propose a specialized damping technique for Min-AM, which interpolates the updates of stochastic gradient descent [50] (SGD) and Min-AM, and is described as

$$x_k^G = x_k + \beta_k r_k, \quad x_k^A = x_k + H_k^A r_k, \quad x_{k+1} = (1 - \alpha_k) x_k^G + \alpha_k x_k^A, \tag{23a}$$

where x_k^G is the SGD update, x_k^A is the Min-AM update, and α_k is the damping parameter. It can be seen that SGD and Min-AM serve as the predicted step, and x_{k+1} is a weighted average of them. Moreover, the inverse Hessian approximation

$$H_k = (1 - \alpha_k)\beta_k I + \alpha_k H_k^A \succeq (1 - \alpha_k)\beta_k I \succ 0 \tag{24}$$

provided that $\alpha_k \in [0, 1)$ and $H_k^A \succeq 0$.

By incorporating damping and regularization, we obtain the stochastic Min-AM (sMin-AM), and the algorithm is summarized in Algorithm 4 in Appendix E. To analyze the convergence properties, we first impose the following assumptions on the objective function f, which are the same as [61, 62].

Assumption 2. *$f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable, and $f(x)$ is lower bounded by a real number f^{low} for $\forall x \in \mathbb{R}^d$. ∇f is globally L-Lipschitz continuous.*

Assumption 3. *At each iteration k, the gradient estimate $\nabla f_{\xi_k}(x_k)$ satisfies $\mathbb{E}_{\xi_k}[\nabla f_{\xi_k}(x_k)] = \nabla f(x_k)$, $\mathbb{E}_{\xi_k}[\|\nabla f_{\xi_k}(x_k) - \nabla f(x_k)\|_2^2] \leq \sigma^2$, where $\sigma > 0$, and $\xi_k \in [T], k = 0, 1, \ldots,$ are independent samples, and the random variable ξ_k (for a given k) is independent of $\{x_j\}_{j=0}^k$.*

The diminishing condition about β_k is

$$\sum_{k=0}^{+\infty} \beta_k = +\infty, \qquad \sum_{k=0}^{+\infty} \beta_k^2 < +\infty. \tag{25}$$

With chosen β_k, we choose the α_k and $\delta_k^{(2)}$ of sMin-AM to meet the following conditions:

Assumption 4. *Given constants $\mu \in (0, 1)$, $C_1 > 0$, and $C_2 > 0$, we have $\alpha_k \in [0, \min\{C_1 \beta_k^{1/2}, 1 - \mu\}]$, $\rho_k \leq 0$, and $-(\rho_k/\beta_k)\|p_k\|_2^2 - 2\rho_k\|p_k\|_2\|q_k\|_2 + \rho_k^2\|p_k\|_2^2\|q_k\|_2^2 \leq C_2$.*

We establish the convergence results of sMin-AM for nonconvex stochastic optimization and the proofs can be found in Appendix E.2 and Appendix E.3.

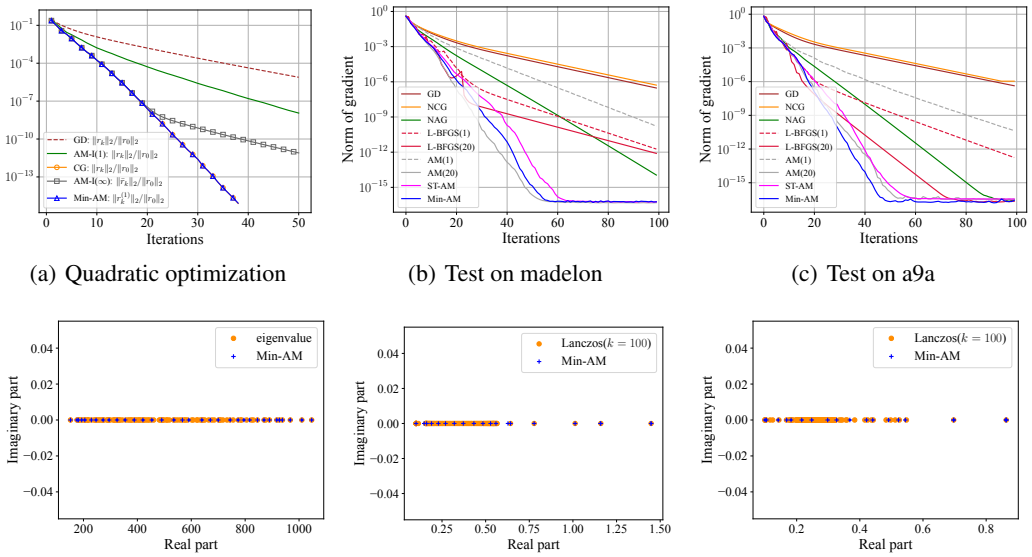
Theorem 4. *Suppose that Assumptions 2-4 hold and $\{x_k\}$ is the sequence generated by sMin-AM with batch size $n_k \equiv n \leq T$. If $\beta_k \in (0, \frac{\mu}{2L(1+C_2)^2}]$ and satisfies (25), then*

$$\liminf_{k\to\infty} \|\nabla f(x_k)\|_2 = 0 \text{ with probability } 1, \tag{26}$$

and for all k, there exists $M_f > 0$ such that $\mathbb{E}[f(x_k)] \leq M_f$.

If $\mathbb{E}_{\xi_k}[\|\nabla f_{\xi_k}(x_k)\|_2^2] \leq M_g$, $\forall k$ where $M_g > 0$ is a constant, we have

$$\lim_{k\to\infty} \|\nabla f(x_k)\|_2 = 0 \text{ with probability } 1. \tag{27}$$



(a) Quadratic optimization (b) Test on madelon (c) Test on a9a

(d) Eigenvalue estimates (e) Eigenvalue estimates (madelon) (f) Eigenvalue estimates (a9a)

Figure 1: Left (strongly convex quadratic optimization): (a) curves about residuals and intermediate residuals; (d) exact eigenvalues, and eigenvalue estimates from Min-AM. Middle and right (regularized logistic regression on madelon and a9a datasets): (b)(c) $\|\nabla f(x_k)\|_2$ of each method; (e)(f) Ritz values of $\nabla^2 f(x^*)$ by k-step Lanczos algorithm, and eigenvalue estimates from Min-AM.

Theorem 5. *Suppose that Assumptions 2-4 hold.* $\{x_k\}_{k=0}^{N-1}$ *are the first* N *iterations generated by sMin-AM with batch size* $n_k \equiv n \leq T$*, and* $\beta_k = \min\{\frac{\mu}{2L(1+C_2)^2}, \frac{\tilde{D}}{\sigma\sqrt{N}}\}$*, where* $\tilde{D}$ *is a problem-independent constant.* R *is a random variable following* $P_R(k) := \text{Prob}\{R = k\} = 1/N$*,* $k = 0, \ldots, N-1$*. Then*

$$\mathbb{E}[\|\nabla f(x_R)\|_2^2] \leq \frac{8D_f L(1+C_2)^2}{N\mu^2} + \frac{\sigma}{\mu\sqrt{N}}\left(\frac{4D_f}{\tilde{D}} + \frac{2(\mu^{-1}C_1^2C_2^2 + L(1+C_2)^2)\tilde{D}}{n}\right), \quad (28)$$

where $D_f := f(x_0) - f^{low}$ *and the expectation is taken with respect to* R *and* $\{S_j\}_{j=0}^{N-1}$*. To ensure* $\mathbb{E}[\|\nabla f(x_R)\|_2^2] \leq \epsilon$*, the number of iterations is* $\mathcal{O}(1/\epsilon^2)$*.*

Remark 7. *The Theorem 4 establishes the global convergence of sMin-AM for solving the problem* (20)*, and Theorem 5 indicates that sMin-AM attains the* $\mathcal{O}(1/\epsilon^2)$ *iteration complexity which is asymptotically optimal for optimization methods based on a stochastic first-order oracle [43].*

4 Experiments

We first tested the basic Min-AM in solving strongly convex quadratic optimization (cf. Theorem 1) and the restarted Min-AM on logistic regression (cf. Theorem 2). Then, we applied the sMin-AM to train deep neural networks on CIFAR [38] and ImageNet [19]. Details about the experimental settings and more numerical results can be found in Appendix F.

Strongly convex quadratic optimization. As shown in Figure 1(a), the first intermediate residual $r_k^{(1)}$ of Min-AM exactly matches the k-th residual of CG, which verifies Theorem 1. Since $X_k^{\mathrm{T}} R_k$ can be ill-conditioned when k is large, it is found that AM-I(∞) fails to coincide with CG and Min-AM in the later iterations. The eigenvalue estimates computed by Min-AM also well approximate the true eigenvalues of the Hessian matrix, which is consistent with our analysis in Section 3.3. In Appendix F.1, we also give a discussion about the cost of the eigenvalue estimation procedure.

Regularized logistic regression. We conducted the regularized logistic regression on the datasets "madelon" and "a9a" from LIBSVM [14]. The compared methods were GD, nonlinear CG (NCG) with line-search, Nesterov's accelerated gradient (NAG), L-BFGS(m), AM(m), and the ST-AM. We

applied the standard k-step Lanczos algorithm [24] to compute 100 Ritz values of $\nabla^2 f(x^*)$, in which the smallest and largest ones were used for the setting of μ, L in NAG. The results in Figure 1 show that Min-AM is competitive with AM(20). It converges much faster than AM(1), L-BFGS(1) (i.e. memoryless BFGS), NCG, and NAG, and outperforms ST-AM. Also, in both datasets, the eigenvalue estimates from Min-AM are accurate enough to approximate the largest and the smallest Ritz values, even though the objective function is non-quadratic. More results including the effect of β_k can be found in Appendix F.2.

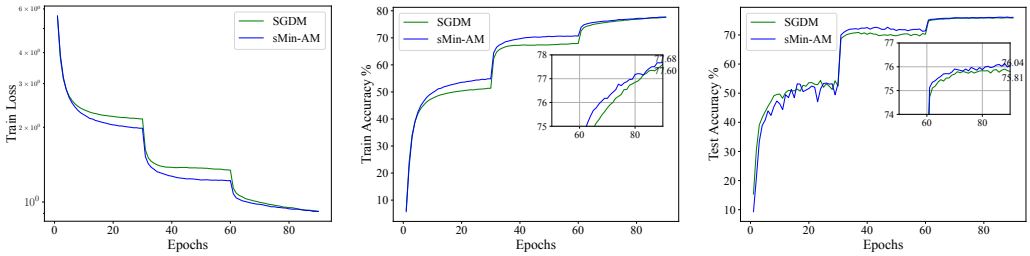
Table 1: Experiments on CIFAR-10/CIFAR-100. "-" means failing to complete the test in our device.

(a) Final TOP1 test accuracy (mean $\pm$ standard deviation) (%) on CIFAR-10/CIFAR-100.

Method	CIFAR-10						CIFAR-100		
	VGG16	ResNet18	ResNet20	ResNet44	ResNet56	WRN16-4	ResNet18	ResNeXt	DenseNet
SGDM	93.52±.15	94.82±.15	92.03±.16	93.10±.23	93.47±.28	94.90±.09	77.27±.09	78.41±.54	78.49±.12
Adam	92.29±.09	93.03±.07	91.17±.13	92.28±.62	92.39±.23	92.45±.11	72.41±.17	73.57±.17	70.80±.23
AdaHessian	93.42±.11	94.36±.09	91.92±.32	92.74±.11	92.40±.06	94.04±.12	76.59±.42	-	-
SAM(1)	93.35±.12	94.95±.10	92.10±.05	93.15±.23	93.44±.19	95.02±.32	77.06±.48	79.12±.40	79.58±.29
SAM(10)	93.59±.11	95.17±.10	**92.43±.19**	93.57±.14	93.77±.12	**95.23±.07**	78.13±.14	79.31±.27	80.09±.52
ST-AM	93.67±.25	95.27±.04	92.39±.11	93.52±.02	93.69±.18	95.21±.09	77.91±.22	79.53±.34	**80.36±.25**
sMin-AM	**94.09±.12**	**95.28±.04**	92.42±.16	**93.72±.27**	**93.82±.10**	95.13±.03	**78.26±.18**	**79.64±.19**	79.88±.09

(b) The memory and computation cost compared with SGDM. The notations "m","t/e", and "t" are abbreviations of memory, per-epoch time, and total training time, respectively.

Cost	CIFAR-10/ResNet18			CIFAR-10/VGG16			CIFAR-100/ResNeXt50			CIFAR-100/DenseNet121		
($\times$ SGDM)	m	t/e	t	m	t/e	t	m	t/e	t	m	t/e	t
SGDM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SAM(10)	1.73	1.78	1.00	2.51	2.59	2.59	1.30	1.16	0.58	1.16	1.19	0.60
ST-AM	1.05	1.46	0.82	1.55	1.91	1.67	1.04	1.07	0.54	1.01	1.11	0.55
sMin-AM	1.01	1.15	0.64	1.35	1.25	0.78	1.03	1.00	0.50	1.01	1.09	0.55



(a) Training loss (b) Train accuracy (c) Test accuracy

Figure 2: Experiment of training ResNet50 on ImageNet.

Experiments on CIFAR. We applied sMin-AM to train VGG16 [57], ResNet18/20/44/56 [30], WideResNet16-4 [66] (abbr. WRN16-4) on CIFAR-10, and train ResNet18, ResNeXt50 [64], DenseNet121 [34] on CIFAR-100. The compared optimizers were SGDM [49], Adam [36], AdaHessian [65], stochastic AM (SAM) [62], and ST-AM [63]. Since the experimental setting was the same as [62, 63], their results were used for reference. Table 1(a) reports the final test accuracy of each optimizer for training 160 epochs. It shows sMin-AM improves SAM(1) and has comparable accuracy to SAM(10). sMin-AM can also use fewer epochs for the training. By setting the accuracy of SGDM as baseline, Table 1(b) shows the memory, per-epoch time, and the total time for an optimizer to achieve a comparable accuracy to SGDM (more details about the number of epochs and final test accuracy can be found in Table 2 in Appendix F.3.2). sMin-AM significantly reduces the memory cost of SAM(10) and the total training time is less than SGDM.

Experiments on ImageNet. We trained ResNet50 on ImageNet with SGDM and sMin-AM. Figure 2 shows that the training process of sMin-AM is faster than SGDM. sMin-AM can achieve both higher train accuracy and higher test accuracy in fewer epochs. In our test, we found the memory cost of sMin-AM was 1.04 times that of SGDM, and the per-epoch training time of sMin-AM was 1.01 times that of SGDM. Hence, for neural network training on a larger dataset, sMin-AM can still be competitive with SGDM.

5 Conclusion

In this paper, we propose Min-AM, a variant of AM(1) with minimal memory size. Min-AM only stores one recursively modified vector pair and is essentially equivalent to the full-memory Type-I AM in strongly convex quadratic optimization. We establish the convergence properties of Min-AM under deterministic and stochastic settings. For deterministic optimization, we also propose an economical method to estimate the spectrum of Hessian, which can be used to adaptively choose the mixing parameters. The experimental results validate the effectiveness of our methods.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key R&D Program of China (No. 2021YFA1001300), National Natural Science Foundation of China (No.61925601), National Key Basic Research Program of China (2020YFB0204800), Tsinghua University Initiative Scientific Research Program, National Natural Science Foundation of China (No.11901338), and Huawei Noah's Ark Lab. We thank all anonymous reviewers for their valuable comments and suggestions on this work.

References

[1] Hengbin An, Xiaowei Jia, and Homer F Walker. Anderson acceleration and application to the three-temperature energy equations. *Journal of Computational Physics*, 347:1–19, 2017.

[2] Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[3] Donald GM Anderson. Comments on "Anderson acceleration, mixing and extrapolation". *Numerical Algorithms*, 80(1):135–234, 2019.

[4] Neculai Andrei. A note on memory-less SR1 and memory-less BFGS methods for large-scale unconstrained optimization. *Numerical Algorithms*, pages 1–18, 2021.

[5] Akash Arora, David C Morse, Frank S Bates, and Kevin D Dorfman. Accelerating self-consistent field theory of block polymers in a variable unit cell. *The Journal of Chemical Physics*, 146(24):244902, 2017.

[6] Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst. *Templates for the solution of algebraic eigenvalue problems: A practical guide*. SIAM, 2000.

[7] Stephen Becker, Jalal Fadili, and Peter Ochs. On quasi-Newton forward-backward splitting: Proximal calculus and convergence. *SIAM Journal on Optimization*, 29(4):2445–2481, 2019.

[8] Albert S Berahas, Frank E Curtis, and Baoyu Zhou. Limited-memory BFGS with displacement aggregation. *Mathematical Programming*, pages 1–37, 2021.

[9] Paul T Boggs and Richard H Byrd. Adaptive, limited-memory BFGS algorithms for unconstrained optimization. *SIAM Journal on Optimization*, 29(2):1282–1299, 2019.

[10] Raghu Bollapragada, Jorge Nocedal, Dheevatsa Mudigere, Hao-Jun Shi, and Ping Tak Peter Tang. A progressive batching L-BFGS method for machine learning. In *International Conference on Machine Learning*, pages 620–629. PMLR, 2018.

[11] William L Briggs, Van Emden Henson, and Steve F McCormick. *A multigrid tutorial*. SIAM, 2000.

[12] 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.

[13] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[14] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

[15] Xiaojun Chen and Masao Fukushima. Proximal quasi-Newton methods for nondifferentiable convex optimization. *Mathematical Programming*, 85(2):313–334, 1999.

[16] Xiaojun Chen and Carl T Kelley. Convergence of the EDIIS algorithm for nonlinear equations. *SIAM Journal on Scientific Computing*, 41(1):A365–A379, 2019.

[17] Arthur I Cohen. Rate of convergence of several conjugate gradient algorithms. *SIAM Journal on Numerical Analysis*, 9(2):248–259, 1972.

[18] Yu-Hong Dai, William W Hager, Klaus Schittkowski, and Hongchao Zhang. The cyclic Barzilai–Borwein method for unconstrained optimization. *IMA Journal of Numerical Analysis*, 26(3): 604–627, 2006.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[20] Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge University Press, 2019.

[21] Claire Evans, Sara Pollock, Leo G Rebholz, and Mengying Xiao. A proof that Anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically). *SIAM Journal on Numerical Analysis*, 58(1):788–810, 2020.

[22] Haw-ren Fang and Yousef Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16(3):197–221, 2009.

[23] Anqi Fu, Junzi Zhang, and Stephen Boyd. Anderson accelerated Douglas–Rachford splitting. *SIAM Journal on Scientific Computing*, 42(6):A3560–A3583, 2020.

[24] Gene H Golub and Charles F Van Loan. Matrix computations, 4th. *Johns Hopkins*, 2013.

[25] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[26] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[27] Robert Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block BFGS: Squeezing more curvature out of data. In *International Conference on Machine Learning*, pages 1869–1878, 2016.

[28] Anne Greenbaum. *Iterative methods for solving linear systems*. SIAM, 1997.

[29] Huan He, Shifan Zhao, Yuanzhe Xi, Joyce Ho, and Yousef Saad. GDA-AM: On the effectiveness of solving min-imax optimization via anderson mixing. In *International Conference on Learning Representations*, 2022.

[30] 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.

[31] Nicholas C Henderson and Ravi Varadhan. Damped Anderson acceleration with restarts and monotonicity control for accelerating EM and EM-like algorithms. *Journal of Computational and Graphical Statistics*, 28(4):834–846, 2019.

[32] Magnus R Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

[33] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge University Press, 2012.

[34] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[35] Carl T Kelley. Numerical methods for nonlinear equations. *Acta Numerica*, 27:207–287, 2018.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[37] Tamara G Kolda, Dianne P O'leary, and Larry Nazareth. BFGS with update skipping and varying memory. *SIAM Journal on Optimization*, 8(4):1060–1083, 1998.

[38] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[39] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[41] Vien Mai and Mikael Johansson. Anderson acceleration of proximal gradient methods. In *International Conference on Machine Learning*, pages 6620–6629. PMLR, 2020.

[42] Thomas A Manteuffel and James S Otto. On the roots of the orthogonal polynomials and residual polynomials associated with a conjugate gradient method. *Numerical Linear Algebra with Applications*, 1(5):449–475, 1994.

[43] Arkadij Semenovič Nemirovski and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[44] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

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

[46] Sara Pollock and Leo G Rebholz. Anderson acceleration for contractive and noncontractive operators. *IMA Journal of Numerical Analysis*, 41(4):2841–2872, 2021.

[47] Sara Pollock, Leo G Rebholz, and Mengying Xiao. Anderson-accelerated convergence of Picard iterations for incompressible Navier–Stokes equations. *SIAM Journal on Numerical Analysis*, 57(2):615–637, 2019.

[48] Florian A Potra and Hans Engler. A characterization of the behavior of the Anderson acceleration on linear problems. *Linear Algebra and Its Applications*, 438(3):1002–1011, 2013.

[49] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

[50] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[51] Youcef Saad and Martin H Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7 (3):856–869, 1986.

[52] Yousef Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37(155):105–126, 1981.

[53] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[54] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011.

[55] Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Regularized nonlinear acceleration. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 712–720, 2016.

[56] Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Regularized nonlinear acceleration. *Mathematical Programming*, 179(1):47–83, 2020.

[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[58] Ke Sun, Yafei Wang, Yi Liu, Bo Pan, Shangling Jui, Bei Jiang, and Linglong Kong. Damped Anderson mixing for deep reinforcement learning: Acceleration, convergence, and stabilization. *Advances in Neural Information Processing Systems*, 34, 2021.

[59] Alex Toth and Carl T Kelley. Convergence analysis for Anderson acceleration. *SIAM Journal on Numerical Analysis*, 53(2):805–819, 2015.

[60] Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

[61] 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.

[62] Fuchao Wei, Chenglong Bao, and Yang Liu. Stochastic Anderson mixing for nonconvex stochastic optimization. *Advances in Neural Information Processing Systems*, 34, 2021.

[63] Fuchao Wei, Chenglong Bao, and Yang Liu. A class of short-term recurrence Anderson mixing methods and their applications. In *International Conference on Learning Representations*, 2022.

[64] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[65] Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10665–10673, 2021.

[66] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

[67] Junzi Zhang, Brendan O'Donoghue, and Stephen Boyd. Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations. *SIAM Journal on Optimization*, 30(4):3170–3197, 2020.

[68] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in Neural Information Processing Systems*, 33, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix G.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B, Appendix C, Appendix D, and Appendix E.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix F and the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We reported standard deviation instead.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix F.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The assets are publicly available.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Contents

1 Introduction **1**

2 The Anderson mixing scheme **3**

3 The Min-AM methods **3**

3.1 The basic Min-AM 3

3.2 The restarted Min-AM 4

3.3 Eigenvalue estimates and the choice of mixing parameter 5

3.4 The stochastic Min-AM 6

4 Experiments **8**

5 Conclusion **10**

A Comparisons with conventional methods **16**

A.1 Relationship between AM-I, Min-AM, and Newton's method 16

A.2 Comparison with the conjugate gradient/residual method 16

A.3 Comparison with the BFGS method 18

A.4 Comparison with the momentum-based method 18

A.5 Comparison with related variants of Anderson mixing 19

B Details of the basic Min-AM **19**

B.1 Derivation of the basic Min-AM 19

B.2 Proof of Theorem 1 21

B.3 Convergence analysis 24

C Details of the restarted Min-AM **25**

C.1 Proof of Theorem 2 25

D Details of the eigenvalue estimation procedure **33**

D.1 Analysis of the quadratic case 33

D.2 Proof of Theorem 3 34

E Details of the stochastic Min-AM **36**

E.1 Some useful results 37

E.2 Proof of Theorem 4 40

E.3 Proof of Theorem 5 41

F Experimental details **42**

F.1 Strongly convex quadratic problem 42

F.2 Regularized logistic regression problem 44

F.3 Deep neural network training problem 46

F.3.1 Hyperparameter setting of the stochastic Min-AM 46
F.3.2 Experiments on CIFAR-10 and CIFAR-100 46
F.3.3 Experiment of training ResNet50 on ImageNet 51
F.3.4 An additional experiment: adversarial training 51

G Limitations **52**

A Comparisons with conventional methods

We compare the proposed Min-AM method with other related methods, including Newton's method, the (nonlinear) conjugate gradient method, BFGS method, momentum-based method, and some other variants of Anderson mixing.

A.1 Relationship between AM-I, Min-AM, and Newton's method

We first give a new interpretation of the AM-I method here, which reveals the relationship between AM-I and Newton's method.

Recall that the historical information is stored in $X_k, R_k \in \mathbb{R}^{d\times m}$. We have the decomposition of the solution space by $\mathbb{R}^d = \mathrm{range}(X_k) \bigoplus \mathrm{range}(X_k)^\perp$. Define $V_k \in \mathbb{R}^{d\times(d-m)}$ whose columns form an orthonormal basis of $\mathrm{range}(X_k)^\perp$. Then for any $x \in \mathbb{R}^d$, we have $x = x_k - X_k\gamma - V_k\eta$, where $\gamma \in \mathbb{R}^m, \eta \in \mathbb{R}^{d-m}$. As a result,

$$\min_{x\in\mathbb{R}^d} f(x) = \min_{\gamma\in\mathbb{R}^m,\eta\in\mathbb{R}^{d-m}} f(x_k - X_k\gamma - V_k\eta). \tag{29}$$

Suppose that f is twice differentiable and $X_k^{\mathrm{T}}\nabla^2 f(x_k)X_k$ is nonsingular. We apply Newton's method in the low dimensional subspace $\mathrm{range}(X_k)$ followed by a gradient descent with stepsize β_k in the complementary subspace $\mathrm{range}(X_k)^\perp$. Then

$$\bar{x}_k = x_k - X_k\gamma_k, \quad x_{k+1} = \bar{x}_k - V_k\eta_k, \tag{30}$$

where $\gamma_k = (X_k^{\mathrm{T}}\nabla^2 f(x_k)X_k)^{-1}X_k^{\mathrm{T}}\nabla f(x_k)$, and $\eta_k = \beta_k V_k^{\mathrm{T}}\nabla f(\bar{x}_k)$. When $m \ll d$, the scheme (30) is more economical than applying the Newton's method in the whole solution space $\mathbb{R}^d$.

Now, consider the simple case that f is a quadratic function: $f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x$. The residual $r_k = -\nabla f(x_k) = b - Ax_k$. Then $r_k - r_{k-1} = -A(x_k - x_{k-1})$, which implies $R_k = -AX_k$. It follows that $\gamma_k = (X_k^{\mathrm{T}}R_k)^{-1}X_k^{\mathrm{T}}r_k$. Also, $\bar{r}_k := r_k - R_k\gamma_k = -\nabla f(\bar{x}_k)$. So $\bar{r}_k \perp \mathrm{range}(X_k)$ due to the choice of γ_k, i.e., a kind of the Galerkin's projection condition [53]. Let U_k be the matrix whose columns form an orthonormal basis of $\mathrm{range}(X_k)$. Then $V_kV_k^{\mathrm{T}} = I - U_kU_k^{\mathrm{T}}$ and $U_k^{\mathrm{T}}\bar{r}_k = 0$. For the gradient descent step, we have $\eta_k = -\beta_k V_k^{\mathrm{T}}\bar{r}_k$, which implies

$$x_{k+1} = \bar{x}_k - V_k\eta_k = \bar{x}_k + \beta_k V_kV_k^{\mathrm{T}}\bar{r}_k = \bar{x}_k + \beta_k\left(I - U_kU_k^{\mathrm{T}}\right)\bar{r}_k = \bar{x}_k + \beta_k\bar{r}_k,$$

namely a mixing step. Therefore, the AM-I method coincides with the scheme (30) for minimizing a quadratic function. Since Min-AM is essentially equivalent to the full-memory AM-I when minimizing a strongly convex quadratic function, we know Min-AM is also closely related to the scheme (30). In the general case, if the objective function f can be well approximated by a quadratic function in a local region around the optima, it is expected that AM-I and Min-AM can behave similarly to the scheme (30) when the iterates enter this local region.

A.2 Comparison with the conjugate gradient/residual method

The conjugate gradient (CG) method [32] and the conjugate residual (CR) method [53, Algorithm 6.20] are classical methods for solving SPD linear systems, where the CG method is extended to solving unconstrained optimization [45]. We first discuss the connection between the proposed Min-AM method and the CG/CR method.

Connection between Min-AM and CG/CR. For solving linear systems, it has been established in [60] that the full-memory AM-I and AM are essentially equivalent to the full orthogonalization

method (FOM) [52] and GMRES [51], respectively. If the linear system is SPD, both FOM and GMRES methods can be simplified to have short-term recurrences: FOM is equivalent to CG, and GMRES is equivalent to CR. Thus, the full-memory AM-I and AM are essentially equivalent to CG and CR respectively, for solving SPD linear systems. However, CG and CR are much more efficient in terms of memory and per-step computational cost, which motivates the development of the proposed Min-AM method that also has short-term recurrences. From Theorem 1, for solving SPD linear systems, Min-AM is essentially equivalent to CG, FOM, and the full-memory AM-I. Compared to CG, Min-AM has the advantage that it does not need explicit matrix-vector products to determine the step size. The update scheme of Min-AM only depends on the historical iterations, which makes it easier to extend to the nonlinear case. Min-AM can also have a similar convergence behaviour to CR for solving SPD linear systems, due to the close relationship between FOM and GMRES [53, Section 6.5.7].

Now, we consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^d} f(x),$$

where $f:\mathbb{R}^d\to\mathbb{R}$ is continuously differentiable and bounded from below. The nonlinear CG (NCG) generates $\{x_k\}$ by the update scheme

$$x_{k+1} = x_k + \alpha_k p_k, \tag{31}$$

where the step size α_k is usually obtained by a line search, and the searching direction p_k is constructed by

$$p_{k+1} = r_k + \beta_k p_k, \quad p_0 = r_0. \tag{32}$$

Here, $r_k := -\nabla f(x_k)$ and the β_k is the momentum term. There are several choices of β_k, for example, the Fletcher-Reeves variant $\beta_k = \frac{\|r_{k+1}\|_2^2}{\|r_k\|_2^2}$ and the Polak-Ribière variant $\beta_k = \frac{r_{k+1}^{\mathrm{T}}(r_{k+1}-r_k)}{\|r_k\|_2^2}$. (See [45].) However, the step size α_k is not easy to obtain. If f is a quadratic function, let A be the Hessian of f. The computation of α_k involves a matrix-vector product, e.g.,

$$\alpha_k = \frac{r_k^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} A p_k},$$

which ensures $r_{k+1} \perp p_k$. If finite difference is used to compute Ap_k, it requires two gradient evaluations. If f is a general nonlinear function, a line search is necessary to ensure the convergence of NCG: α_k is chosen as an approximate solution to the problem

$$\min_{\alpha\geq 0} f(x_k + \alpha p_k), \tag{33}$$

where p_k is supposed to be a descent direction. The d-step quadratic convergence proved in [17], which is similar to our result of the restarted Min-AM, was only established under the assumption of an exact line search, namely, (33) is exactly solved. More practical choices of α_k require some conditions to be satisfied. For example, the strong Wolfe conditions are

$$\begin{aligned} &f(x_k+\alpha_k p_k) \leq f(x_k) + c_1\alpha_k \nabla f(x_k)^{\mathrm{T}} p_k, \\ &|\nabla f(x_k+\alpha_k p_k)^{\mathrm{T}} p_k| \leq c_2 |\nabla f(x_k)^{\mathrm{T}} p_k|, \end{aligned}$$

where $0 < c_1 < c_2 < 1$. Since the line search can incur many times of the backtracking procedure, the total number of gradient evaluations for NCG can be large.

In contrast, our proposed Min-AM method does not need explicit Hessian-vector products or line search to determine the step size, while the convergence results (cf. Theorem 2) match those of NCG with exact line search. Therefore, Min-AM is more economical than NCG, and can have better convergence than the NCG with inexact line search. Also, it should be pointed out that even in strongly convex quadratic optimization, Min-AM generally does not generate the same sequence of iterates as CG: They are equivalent in the sense that $x_k^{(1)} = x_k^{\mathrm{CG}}$, where $x_k^{(1)}$ is the intermediate step in (8a).

A.3 Comparison with the BFGS method

The BFGS constructs the approximate inverse Hessian by solving

$$H_k = \arg\min_H \|H - H_{k-1}\|_{F(W)}, \text{ s.t. } Hy_k = s_k,\ H = H^{\mathrm{T}}, \tag{34}$$

where $s_k := x_k - x_{k-1}, y_k := \nabla f(x_k) - \nabla f(x_{k-1})$. The norm $\|\cdot\|_{F(W)}$ is the weighted Frobenius norm (i.e., $\|X\|^2_{F(W)} := \|W^{1/2}XW^{1/2}\|_F$ for a matrix $X \in \mathbb{R}^{d\times d}$) and the weight matrix W satisfies $Ws_k = y_k$. The solution is

$$H_k = \left(I - \frac{s_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}\right) H_{k-1} \left(I - \frac{y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}\right) + \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}. \tag{35}$$

Note that the $H_k \in \mathbb{R}^{d\times d}$ is recursively constructed and can be dense during the later iterations. To reduce memory overhead, the limited-memory BFGS [39] is often used. In memoryless BFGS, the previous approximate Hessian is always reset to I_d, i.e.,

$$H_k = \left(I - \frac{s_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}\right) \left(I - \frac{y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}\right) + \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}. \tag{36}$$

Memoryless BFGS has minimal memory size in L-BFGS, but has no equivalence to BFGS. It is pointed in [45] that for the update $x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k)$ of memoryless BFGS, if α_k is chosen by an exact line search, then memoryless BFGS is equivalent to CG for quadratic function. However, an exact line search can incur prohibitive cost in practice.

The Min-AM method has a similar form to that of memoryless BFGS, but uses a recursively constructed vector pair. Min-AM is essentially equivalent to the full-memory AM-I and CG for minimizing strongly convex quadratic functions. In fact, let $P_k = (p_1, \dots, p_k), Q_k = (q_1, \dots, q_k)$, and define

$$H_k^A = -P_k(P_k^{\mathrm{T}} Q_k)^{-1} P_k^{\mathrm{T}} + \beta_k (I - P_k(P_k^{\mathrm{T}} Q_k)^{-1} Q_k^{\mathrm{T}})(I - Q_k(P_k^{\mathrm{T}} Q_k)^{-1} P_k^{\mathrm{T}}), \tag{37}$$

assuming $P_k^{\mathrm{T}} Q_k$ is nonsingular. We call the iterations defined by $x_{k+1} = x_k + H_k^A r_k$ as Scheme A. Clearly, $H_k^A Q_k = -P_k$ holds. Moreover, it can be proved by direct computation (using the properties in Theorem 1) that for strongly convex quadratic optimization, if fixed β_k is used, i.e., $\beta_k \equiv \beta$, where β is a constant, we have

$$H_k^A = -\frac{p_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k} + \left(I - \frac{p_k q_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k}\right) H_{k-1}^A \left(I - \frac{q_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k}\right), \tag{38}$$

starting from $H_0^A := \beta I$. Hence, the H_k^A in (37) solves

$$H_k^A = \arg\min_H \|H - H_{k-1}^A\|_{F(W)}, \text{ s.t. } Hq_k = -p_k,\ H = H^{\mathrm{T}}, \tag{39}$$

where the weight matrix satisfies $Wp_k = -q_k$. The (39) is similar to (34), but based on modified vector pairs. (Note that we use $r_k - r_{k-1} = -(\nabla f(x_k) - \nabla f(x_{k-1}))$ to construct q_k, so there is a difference of sign.) It can also be verified that for strongly convex quadratic optimization, $H_k r_k = H_k^A r_k$, where H_k is defined in (10). So the iterations of Min-AM are identical to those of Scheme A in this case. In this sense, though only using one vector pair, Min-AM implicitly constructs H_k^A which can well approximate the inverse Hessian.

A.4 Comparison with the momentum-based method

The momentum-based method is similar to the scheme defined in (31) and (32), but the momentum β_k is different from that in CG. We take the Nesterov's accelerated gradient (NAG) method as an example. The update scheme of NAG is defined as

$$y_{k+1} = x_k - \frac{1}{L}\nabla f(x_k), \tag{40a}$$

$$x_{k+1} = y_{k+1} + \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}(y_{k+1} - y_k), \tag{40b}$$

where L and μ are the Lipschitz constant and the strong convexity constant of ∇f, respectively. Unlike Min-AM, NAG generally does not form a symmetric approximation to the Hessian. In Figure 3, we show the convergence behaviours of NAG with different settings of μ and L. It is found that the setting of L and μ can have a large effect on the convergence. On the contrary, Min-AM forms symmetric Hessian approximations and can give useful information of L and μ via an economical eigenvalue estimation procedure.

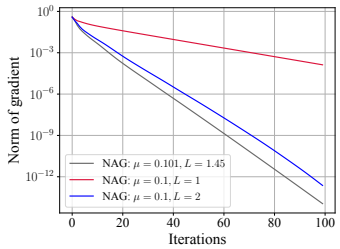


Figure 3: Convergence behaviours of NAG with different settings of L and μ, for the regularized logistic regression on the madelon dataset.

A.5 Comparison with related variants of Anderson mixing

Compared with the original AM, AM-I, the regularized nonlinear acceleration scheme [55], and the stochastic AM [62], the proposed Min-AM has the minimal memory size, the same as that of AM(1). Nonetheless, Min-AM incorporates more historical information than AM(1) and AM-I(1), and is equivalent to the full-memory AM-I in solving strongly convex quadratic optimization. The convergence analysis in Section 3.4 also shows that the stochastic Min-AM has similar convergence to that of the stochastic AM in theory.

The short-term recurrence AM (ST-AM) [63] stores two vector pairs and is equivalent to the full-memory AM in solving strongly convex quadratic optimization, but the approximated Hessian from ST-AM is generally not symmetric. It is also worth pointing out that for AM and ST-AM, their existing theoretical results do not show the significantly better convergence rate than the simple gradient descent method for unconstrained optimization.

Our proposed Min-AM further reduces the memory size of ST-AM and the faster convergence than gradient descent for general nonlinear optimization is rigorously justified in theory.

B Details of the basic Min-AM

The procedure of the basic Min-AM is described in Algorithm 2. Next, we give the derivation of the basic Min-AM, and prove the related theoretical properties.

B.1 Derivation of the basic Min-AM

We give the derivation of the update scheme (10) for solving the quadratic optimization $\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x$ here. Recall that the steps in one iteration of the basic Min-AM are:

$$x_k^{(1)} = x_k - p_k\Gamma_k^{(1)}, \quad x_k^{(2)} = x_k^{(1)} + \beta_k r_k^{(1)}, \quad x_{k+1} = x_k^{(2)} - p_k\Gamma_k^{(2)}, \tag{41}$$

where $r_k^{(1)} = r_k - q_k\Gamma_k^{(1)}$ and $\beta_k > 0$. Define $r_k^{(2)} = r_k^{(1)} - \beta_k A r_k^{(1)}$ and $r_k^{(3)} = r_k^{(2)} - q_k\Gamma_k^{(2)}$. The $\Gamma_k^{(1)}$ and $\Gamma_k^{(2)}$ are determined by imposing the projection conditions:

$$r_k^{(1)} \perp p_k, \quad r_k^{(3)} \perp p_k.$$

Here, we define $p_1 = \Delta x_0, q_1 = \Delta r_0$, and for $k \geq 2$, the construction of p_k, q_k at the beginning of the k-th iteration is

$$p_k = \Delta x_{k-1} - p_{k-1}\zeta_k, \quad q_k = \Delta r_{k-1} - q_{k-1}\zeta_k,$$

where $\zeta_k = (p_{k-1}^{\mathrm{T}} q_{k-1})^{-1}\Delta r_{k-1}^{\mathrm{T}} p_{k-1}$, assuming $p_{k-1}^{\mathrm{T}} q_{k-1} \neq 0$.

Algorithm 2 Min-AM for strongly convex quadratic optimization

Input: $x_0 \in \mathbb{R}^d, \beta_k > 0, 0 < max_iter \leq d$

$p_0, q_0 = \mathbf{0} \in \mathbb{R}^d$
for $k = 0, 1, \ldots, max_iter$ **do**
  $r_k = -\nabla f(x_k)$
  if $k > 0$ **then**
    $p = x_k - x_{k-1}, q = r_k - r_{k-1}$ (Compute $\Delta x_{k-1}, \Delta r_{k-1}$)
    $\zeta_k = (p_{k-1}^{\mathrm{T}} q_{k-1})^{\dagger} p_{k-1}^{\mathrm{T}} q$
    $p_k = p - p_{k-1}\zeta_k,\ q_k = q - q_{k-1}\zeta_k$ ($p_k \perp q_{k-1}, q_k \perp p_{k-1}$)
  end if
  $\Gamma_k^{(1)} = (p_k^{\mathrm{T}} q_k)^{\dagger} p_k^{\mathrm{T}} r_k$
  $x_k^{(1)} = x_k - p_k \Gamma_k^{(1)},\ r_k^{(1)} = r_k - q_k \Gamma_k^{(1)}$ (Projection step: $r_k^{(1)} \perp p_k$)
  $x_k^{(2)} = x_k^{(1)} + \beta_k r_k^{(1)}$ (Mixing step)
  $\Gamma_k^{(2)} = \beta_k (p_k^{\mathrm{T}} q_k)^{\dagger} q_k^{\mathrm{T}} r_k^{(1)}$
  $x_{k+1} = x_k^{(2)} - p_k \Gamma_k^{(2)}$ (Projection step: $r_{k+1} \perp p_k$)
  if $\|r_k^{(1)}\|_2 = 0$ **then**
    break
  end if
end for
return x_k

We first prove that

$$q_k = -Ap_k \tag{42}$$

by induction. By the definition of f and $r_k = -\nabla f(x_k)$, it is clear that $r_k - r_{k-1} = -A(x_k - x_{k-1})$. So (42) holds for $k = 1$. For $k \geq 2$, suppose that (42) holds for $k-1$. Then $q_k = \Delta r_{k-1} - q_{k-1}\zeta_k = -A\Delta x_{k-1} + Ap_{k-1}\zeta_k = -Ap_k$, which completes the induction.

Suppose that $p_k^{\mathrm{T}} q_k \neq 0$. We have $\Gamma_k^{(1)} = (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}} r_k$. For $\Gamma_k^{(2)}$, the exact solution is

$$\begin{aligned}\Gamma_k^{(2)} &= (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}} r_k^{(2)} = (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}} (r_k^{(1)} - \beta_k A r_k^{(1)}) \\ &= -\beta_k (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}} A r_k^{(1)} = -\beta_k (p_k^{\mathrm{T}} q_k)^{-1} (Ap_k)^{\mathrm{T}} r_k^{(1)} = \beta_k (p_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}} r_k^{(1)}\end{aligned} \tag{43}$$

due to $p_k^{\mathrm{T}} r_k^{(1)} = 0$, the symmetry of A, and (42). As a result,

$$\begin{aligned}x_{k+1} &= x_k^{(2)} - p_k \Gamma_k^{(2)} = x_k^{(1)} + \beta_k r_k^{(1)} - p_k \Gamma_k^{(2)} \\ &= x_k - p_k \Gamma_k^{(1)} + \beta_k r_k^{(1)} - p_k \Gamma_k^{(2)} \\ &= x_k - p_k \Gamma_k^{(1)} + \beta_k r_k^{(1)} - p_k \cdot \beta_k (p_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}} r_k^{(1)} \\ &= x_k - p_k (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}} r_k + \beta_k \left(I - p_k (p_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}}\right) r_k^{(1)} \\ &= x_k - p_k (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}} r_k + \beta_k (I - p_k (p_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}})(I - q_k (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}}) r_k.\end{aligned} \tag{44}$$

The corresponding inverse Hessian approximation is

$$H_k = -p_k (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}} + \beta_k (I - p_k (p_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}})(I - q_k (p_k^{\mathrm{T}} q_k)^{-1} p_k^{\mathrm{T}}). \tag{45}$$

The reason to use the projection condition of AM-I. Note that the Galerkin's projection condition of AM-I is used to determine $\Gamma_k^{(1)}$ and $\Gamma_k^{(2)}$. On the other side, if the Galerkin's condition of the original AM is used to determine $\Gamma_k^{(1)}$ and $\Gamma_k^{(2)}$, the scheme (41) leads to

$$x_{k+1} = x_k - p_k (q_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}} r_k + \beta_k (I + p_k (q_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}} A)(I - q_k (q_k^{\mathrm{T}} q_k)^{-1} q_k^{\mathrm{T}}) r_k, \tag{46}$$

which is not a practical method since Aq_k needs to be explicitly computed.

B.2 Proof of Theorem 1

Recall that the strongly convex quadratic optimization is formulated as

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x, \tag{47}$$

where $A \in \mathbb{R}^{d\times d}$ is SPD, $b \in \mathbb{R}^d$. Solving (47) is equivalent to solving the SPD linear system

$$Ax = b. \tag{48}$$

We first state the relationship of the AM-I with the full orthogonalization method [52] (FOM) in the following propositions. Let x_k^{FOM} and $r_k^{\text{FOM}} := b - Ax_k^{\text{FOM}}$ denote the k-th FOM iterate and residual, respectively. Define $e^j := (1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^j$ for $j \geq 1$. The main results of the full-memory AM-I are stated in Proposition 1 and Proposition 2. (Proposition 1 is a known result in [60].)

Proposition 1 (General linear system)**.** *For solving a general linear system $Ax = b$ with the full-memory AM-I, suppose that $\beta_k > 0$ and the fixed-point map is $g(x) = (I - A)x + b$. If the initial point of AM-I is $x_0 = x_0^{\text{FOM}}$, and $X_j^{\mathrm{T}}R_j$ is nonsingular for $j = 1, \ldots, k$, then the intermediate iterate $\bar{x}_k$ satisfies $\bar{x}_k = x_k^{\text{FOM}}$.*

We give the proof here, which is similar to [63, Proof of Proposition 1], but applies to the Type-I AM.

Proof. The definition of the fixed-point map suggests that the residual $r_k = g(x_k) - x_k = b - Ax_k$. From (5), for $j = 1, \ldots, k$, the nonsingularity of $X_j^{\mathrm{T}}R_j$ ensures that each Γ_j is uniquely determined. Thus the updates of AM-I are well defined.

Since $X_k^{\mathrm{T}}R_k$ is nonsingular, we have $\mathrm{rank}(X_k) = \mathrm{rank}(R_k) = k$. We first show

$$\mathrm{range}(X_k) = \mathcal{K}_k(A, r_0^{\text{FOM}}) \tag{49}$$

by induction. We abbreviate $\mathcal{K}_k(A, r_0^{\text{FOM}})$ as $\mathcal{K}_k$ in this proof.

First, $\Delta x_0 = \beta_0 r_0 = \beta_0 r_0^{\text{FOM}}$ since $x_1 = x_0 + \beta_0 r_0$. If $k = 1$, then the proof is complete. Then, suppose that $k > 1$ and, as an inductive hypothesis, that $\mathrm{range}(X_{k-1}) = \mathcal{K}_{k-1}$. From (3), we have $x_{k+1} = x_k + \beta_k r_k - (X_k + \beta_k R_k)\Gamma_k$. Noting that $R_k = -AX_k$, it follows that

$$\begin{aligned}
\Delta x_{k-1} &= x_k - x_{k-1} \\
&= \beta_{k-1} r_{k-1} - (X_{k-1} + \beta_{k-1} R_{k-1})\Gamma_{k-1} \\
&= \beta_{k-1}(b - Ax_{k-1}) - (X_{k-1} - \beta_{k-1} AX_{k-1})\Gamma_{k-1} \\
&= \beta_{k-1} b - \beta_{k-1} A(x_0 + \Delta x_0 + \cdots + \Delta x_{k-2}) - (X_{k-1} - \beta_{k-1} AX_{k-1})\Gamma_{k-1} \\
&= \beta_{k-1} r_0 - \beta_{k-1} AX_{k-1} e^{k-1} - (X_{k-1} - \beta_{k-1} AX_{k-1})\Gamma_{k-1}.
\end{aligned} \tag{50}$$

Since $r_0 \in \mathcal{K}_{k-1}$, and by the inductive hypothesis $\mathrm{range}(X_{k-1}) = \mathcal{K}_{k-1}$ which also implies $\mathrm{range}(AX_{k-1}) \subseteq \mathcal{K}_k$, we know $\Delta x_{k-1} \in \mathcal{K}_k$. Thus, $\mathrm{range}(X_k) \subseteq \mathcal{K}_k$. Since $\mathrm{rank}(X_k) = k$, namely $\dim(\mathrm{range}(X_k)) = \dim(\mathcal{K}_k)$, we have $\mathrm{range}(X_k) = \mathcal{K}_k$, thus completing the induction. As a result, we also have

$$\mathrm{range}(R_k) = \mathrm{range}(AX_k) = A\mathcal{K}_k(A, r_0). \tag{51}$$

Recalling that to determine Γ_k, we solve the projection condition:

$$\bar{r}_k = r_k - R_k\Gamma_k \perp \mathrm{range}(X_k). \tag{52}$$

The nonsingluarity of $X_k^{\mathrm{T}}R_k$ ensures that Γ_k is uniquely determined by

$$\Gamma_k = (X_k^{\mathrm{T}}R_k)^{-1}X_k^{\mathrm{T}}r_k. \tag{53}$$

Also, since $r_k = b - Ax_k = b - A(x_0 + X_k e^k) = r_0 - AX_k e^k$, we have $r_k - R_k\Gamma = r_k + AX_k\Gamma = r_0 - AX_k e^k + AX_k\Gamma = r_0 - AX_k\tilde{\Gamma}$, where $\tilde{\Gamma} = e^k - \Gamma$, for $\forall\, \Gamma \in \mathbb{R}^k$. So Γ_k solves (52) if and only if $\tilde{\Gamma}_k = e^k - \Gamma_k$ solves

$$r_0 - AX_k\tilde{\Gamma}_k \perp \mathrm{range}(X_k). \tag{54}$$

According to (49), the condition (54) is equivalent to

$$r_0 - Az \perp \mathrm{range}(X_k) \ \text{ s.t. } \ z \in \mathcal{K}_k, \tag{55}$$

where $\mathrm{range}(X_k) = \mathcal{K}_k$. Since the initializations are identical, the condition (55) for AM-I is the Petrov-Galerkin condition for FOM. Due to the nonsingularity of $X_k^{\mathrm{T}} R_k$, the solution of (54) is also unique. Therefore, we have

$$\bar{x}_k = x_k - X_k \Gamma_k = x_k - X_k(e^k - \tilde{\Gamma}_k) = x_0 + X_k \tilde{\Gamma}_k = x_k^{\mathrm{FOM}}.$$

□

In Proposition 1, the assumption that $X_k^{\mathrm{T}} R_k$ is nonsingular is critical to ensure no stagnation occurs at the k-th iteration for solving a general linear system. In fact, for SPD linear systems (48) or strongly convex quadratic optimization (47), when AM-I breaks down, i.e. $X_k^{\mathrm{T}} R_k$ is singular, AM-I obtains the exact solution, as shown in the next proposition.

Proposition 2 (SPD)**.** *For applying the AM-I to minimize a strongly convex quadratic problem (47), or equivalently, solve an SPD linear system (48), suppose that $\beta_k > 0$ and the fixed-point map is $g(x) = (I - A)x + b$. If the condition $X_k^{\mathrm{T}} R_k$ is nonsingular holds for $1 \le k < s$ while failing to hold for $k = s$, where $s \ge 1$, then the residual of AM-I satisfies $r_s = \bar{r}_{s-1} = 0$.*

We give the proof here, which is similar to [63, Proof of Proposition 2], but applies to the Type-I AM.

Proof. The definition of g suggests that the residual $r_k = g(x_k) - x_k = b - Ax_k$. The relation $R_k = -AX_k$ holds during the iterations and the nonsingularity of A implies $\mathrm{rank}(X_k) = \mathrm{rank}(R_k)$. Since A is SPD and $X_k^{\mathrm{T}} R_k = -X_k^{\mathrm{T}} A X_k$, it follows that $X_k^{\mathrm{T}} R_k$ being nonsingular $\Leftrightarrow \mathrm{rank}(X_k) = \mathrm{rank}(R_k) = k$. Hence $\mathrm{rank}(X_k) = k$ holds for $1 \le k < s$ while failing to hold for $k = s$.

For $s = 1$, since the first step of AM is $x_1 = x_0 + \beta_0 r_0$, the assumption $\mathrm{rank}(X_1) = 0$ implies that $\mathrm{rank}(r_0) = \mathrm{rank}(X_1) = 0$, then $r_1 = \bar{r}_0 := 0$.

For $s > 1$, $\Delta x_{s-1} = x_s - x_{s-1} = -X_{s-1}\Gamma_{s-1} + \beta_{s-1}\bar{r}_{s-1}$. The rank deficiency of X_s implies $\Delta x_{s-1} \in \mathrm{range}(X_{s-1})$, which further implies $\bar{r}_{s-1} \in \mathrm{range}(X_{s-1})$. So there exists $\zeta \in \mathbb{R}^{s-1}$, such that $\bar{r}_{s-1} = X_{s-1}\zeta$. Due to $\bar{r}_{s-1} \perp X_{s-1}$, we have

$$0 = (\bar{r}_{s-1})^{\mathrm{T}} X_{s-1} = (X_{s-1}\zeta)^{\mathrm{T}} X_{s-1} = \zeta^{\mathrm{T}} X_{s-1}^{\mathrm{T}} X_{s-1}. \tag{56}$$

Because $\mathrm{rank}(X_{s-1}) = s - 1$, we have $\zeta = 0$, which implies $\bar{r}_{s-1} = 0$. Hence $x_s = \bar{x}_{s-1}$ and $r_s = \bar{r}_{s-1} = 0$. □

It is also known that FOM method can be simplified to the conjugate gradient (CG) method [53] in this case. Now we prove Theorem 1, following a similar procedure of [63, Proof of Theorem 1].

Proof of Theorem 1. The A-norm minimization problem in the property (iii) is equivalent to the Galerkin condition [53], i.e. $z_k = \arg\min_{z \in \mathcal{K}_k(A, r_0)} \|x_0 + z - x^*\|_A \Leftrightarrow r_0 - Az_k \perp \mathcal{K}_k(A, r_0)$. Also, $r_k^{(3)} = r_{k+1} = r_k^{(2)} - q_k \Gamma_k^{(2)}$.

Besides relations (i)-(iii), we add an auxiliary relation here:
(iv) $r_k = r_0 + Q_k \bar{\Gamma}_k \in \mathcal{K}_{k+1}(A, r_0)$, where $\bar{\Gamma}_k \in \mathbb{R}^k$.
We prove the relations (i)-(iv) by induction.

For $k = 1$, since $r_0 = r_0^{(1)} \neq 0$, we have $\mathrm{rank}(\Delta x_0) = \mathrm{rank}(X_1) = 1$ and $\mathrm{rank}(\Delta r_0) = \mathrm{rank}(R_1) = 1$. So $p_k \neq 0$ and $p_k^{\mathrm{T}} q_k = -p_k^{\mathrm{T}} A p_k < 0$. The relation (i) holds. Since $p_1 = \Delta x_0, q_1 = \Delta r_0$, and $\Delta r_0 = -A\Delta x_0$, the equality $Q_1 = -AP_1$ also holds. Since $r_1 = r_0 - \beta_0 A r_0$ and $\mathrm{range}(Q_1) = \mathrm{span}\{Ar_0\}$, it is clear that $r_1 = r_0 - Q_1 \bar{\Gamma}_1 \in \mathcal{K}_2(A, r_0)$, namely relation (iv). Due to the projection step (Line 10 in Algorithm 2), $r_1^{(1)} \perp \mathrm{range}(P_1)$ and $r_1^{(1)}$ is unique, which is guaranteed by $p_1^{\mathrm{T}} q_1 \neq 0$. Also, $r_1^{(1)} = r_1 - q_1 \Gamma_1^{(1)} = r_0 - \beta_0 A r_0 - q_1 \Gamma_1^{(1)} = r_0 - Q_1 \eta_1$, where the last equality is due to $\mathrm{span}\{Ar_0\} = \mathrm{range}(Q_1)$. For $r_1^{\mathrm{FOM}} = r_0 - Az_1$, where $z_1 \in \mathcal{K}_1(A, r_0)$, it holds $r_1^{\mathrm{FOM}} \perp \mathcal{K}_1(A, r_0) = \mathrm{range}(P_1)$. As a result, both $r_1^{(1)}$ and r_1^{FOM} are the oblique projections of r_0 onto the subspace $\mathrm{range}(P_1)^{\perp}$ along $\mathrm{range}(Q_1)$. Since $p_1^{\mathrm{T}} q_1 \neq 0$, the projection exists and is

unique, which implies $r_1^{(1)} = r_1^{\text{FOM}}$. So $x_1^{(1)} = x_1^{\text{FOM}} = x_0 + z_1$ because their residuals are equal and A is nonsingular. Hence the relation (iii) holds.

Suppose that $k > 1$, and as an inductive hypothesis, the relations (i)-(iv) hold for $j = 1, \ldots, k-1$. Consider the k-th iteration. From Line 7 in Algorithm 2, $q_k \in \text{span}\{\Delta r_{k-1}, q_{k-1}\}$, and $p_k \in \text{span}\{\Delta x_{k-1}, p_{k-1}\}$. Due to $p_{k-1}^{\text{T}} q_{k-1} = -p_{k-1}^{\text{T}} A p_{k-1} \neq 0$, we have $q_k^{\text{T}} p_{k-1} = 0$. We first prove that $p_k \neq 0$ by contradiction.

If $p_k = 0$, then from Line 7 in Algorithm 2, $\Delta x_{k-1} \in \text{span}\{p_{k-1}\}$. From Line 10, Line 11, and Line 13, we have

$$\Delta x_{k-1} = x_k - x_{k-1} = \beta_{k-1} r_{k-1}^{(1)} - p_{k-1}(\Gamma_{k-1}^{(1)} + \Gamma_{k-1}^{(2)}). \tag{57}$$

So $r_{k-1}^{(1)} \in \text{span}\{p_{k-1}\}$ since $\Delta x_{k-1} \in \text{span}\{p_{k-1}\}$. Hence there exists $\zeta \in \mathbb{R}$, such that $r_{k-1}^{(1)} = p_{k-1}\zeta$. From the Line 10, we know $r_{k-1}^{(1)} \perp \text{span}\{p_{k-1}\}$, so we have

$$0 = (r_{k-1}^{(1)})^{\text{T}} p_{k-1} = (p_{k-1}\zeta)^{\text{T}} p_{k-1} = \zeta^{\text{T}} p_{k-1}^{\text{T}} p_{k-1}.$$

Since $p_{k-1} \neq 0$, it follows that $\zeta = 0$ which implies $r_{k-1}^{(1)} = 0$. It is impossible otherwise Algorithm 2 has terminated in the $(k-1)$-th iteration. So $p_k \neq 0$. Moreover $p_k^{\text{T}} q_k = -p_k^{\text{T}} A p_k \neq 0$ since A is SPD.

Since $r_{k-1}^{(1)} = r_{k-1} - q_{k-1}\Gamma_{k-1}^{(1)}$, and $r_{k-1} \in \mathcal{K}_k(A, r_0), q_{k-1} \in \text{range}(Q_{k-1}) = A\mathcal{K}_{k-1}(A, r_0)$ due to the inductive hypothesis, we have $r_{k-1}^{(1)} \in \mathcal{K}_k(A, r_0)$, which together with (57) and $p_{k-1} \in \text{range}(P_{k-1}) = \mathcal{K}_{k-1}(A, r_0)$ infers $\Delta x_{k-1} \in \mathcal{K}_k(A, r_0)$. As a result, $p_k \in \text{span}\{\Delta x_{k-1}, p_{k-1}\} \subseteq \mathcal{K}_k(A, r_0)$. So $\text{range}(P_k) = \text{range}(P_{k-1}, p_k) \subseteq \mathcal{K}_k(A, r_0)$. Moreover, $p_k \notin \text{range}(P_{k-1})$. We prove it by contradiction.

If $p_k \in \text{range}(P_{k-1})$, then from Line 7 in Algorithm 2, $\Delta x_{k-1} \in \text{span}\{p_{k-1}, p_k\} \subseteq \text{range}(P_{k-1})$, which together with (57) leads to $r_{k-1}^{(1)} \in \text{range}(P_{k-1})$. Hence there exists $\xi \in \mathbb{R}^{k-1}$ such that $r_{k-1}^{(1)} = P_{k-1}\xi$. From the inductive hypothesis, we know $r_{k-1}^{(1)} \perp \text{range}(P_{k-1})$. So we have

$$0 = (r_{k-1}^{(1)})^{\text{T}} P_{k-1} = (P_{k-1}\xi)^{\text{T}} P_{k-1} = \xi^{\text{T}} P_{k-1}^{\text{T}} P_{k-1}.$$

Since $-P_{k-1}^{\text{T}} A P_{k-1} = P_{k-1}^{\text{T}} Q_{k-1}$ is nonsingular due to the inductive hypothesis $p_i \perp q_j (1 \leq i \neq j \leq k-1)$ and $p_i^{\text{T}} q_i = -p_i^{\text{T}} A p_i \neq 0 (1 \leq i \leq k-1)$ as $p_i \neq 0$, it follows that $\text{rank}(P_{k-1}) = k-1$. So $P_{k-1}^{\text{T}} P_{k-1}$ is nonsingular which implies that $\xi = 0$. Then $r_{k-1}^{(1)} = P_{k-1}\xi = 0$. But it is impossible otherwise Algorithm 2 has terminated in the $(k-1)$-th iteration. Hence $p_k \notin \text{range}(P_{k-1})$. As a result, $\text{range}(P_k) = \mathcal{K}_k(A, r_0)$.

Because $\Delta r_{k-1} = -A\Delta x_{k-1}$ and $q_{k-1} = -Ap_{k-1}$, Line 7 in Algorithm 2 infers $q_k = -Ap_k$. So $Q_k = -AP_k$. Hence $\text{range}(Q_k) = A\mathcal{K}_k(A, r_0)$.

To prove $p_i \perp q_j$ for $1 \leq i \neq j \leq k$, it suffices to show $q_k \perp P_{k-1}$. From the construction of q_k in Line 7 in Algorithm 2 and $p_{k-1}^{\text{T}} q_{k-1} \neq 0$, we know $q_k^{\text{T}} p_{k-1} = 0$. To further prove $q_k \perp \text{range}(P_{k-2})$ $(k \geq 3)$, note that

$$\begin{aligned} \Delta r_{k-1} &= -A\Delta x_{k-1} = Ap_{k-1}(\Gamma_{k-1}^{(1)} + \Gamma_{k-1}^{(2)}) - \beta_{k-1} A r_{k-1}^{(1)} \\ &= -q_{k-1}(\Gamma_{k-1}^{(1)} + \Gamma_{k-1}^{(2)}) - \beta_{k-1} A r_{k-1}^{(1)}, \end{aligned}$$

where the second equality is a direct substitution with (57). Therefore,

$$P_{k-2}^{\text{T}} \Delta r_{k-1} = -P_{k-2}^{\text{T}} q_{k-1}(\Gamma_{k-1}^{(1)} + \Gamma_{k-1}^{(2)}) - \beta_{k-1} P_{k-2}^{\text{T}} A r_{k-1}^{(1)} = 0 - \beta_{k-1}(AP_{k-2})^{\text{T}} r_{k-1}^{(1)} = 0, \tag{58}$$

where the second equality is due to $q_{k-1} \perp \text{range}(P_{k-2})$ and A is SPD, and the third equality is due to $r_{k-1}^{(1)} \perp \text{range}(P_{k-1}) = \mathcal{K}_{k-1}(A, r_0)$ and $\text{range}(AP_{k-2}) = A\mathcal{K}_{k-2}(A, r_0) \subseteq \mathcal{K}_{k-1}(A, r_0)$. As a result, we obtain

$$P_{k-2}^{\text{T}} q_k = P_{k-2}^{\text{T}}(\Delta r_{k-1} - q_{k-1}\zeta_k) = 0,$$

which is due to (58) and $q_{k-1} \perp \mathrm{range}(P_{k-2})$. Therefore, $q_k \perp \mathrm{range}(P_{k-1})$. Since A is SPD, we also have $p_k^{\mathrm{T}} Q_{k-1} = -p_k^{\mathrm{T}} A P_{k-1} = -(Ap_k)^{\mathrm{T}} P_{k-1} = q_k^{\mathrm{T}} P_{k-1} = 0$. Hence relation (ii) holds in the k-th iteration.

Next, we prove the relation (iv). We have

$$\begin{aligned} r_k &= r_{k-1}^{(2)} - q_{k-1}\Gamma_{k-1}^{(2)} \\ &= r_{k-1}^{(1)} - \beta_{k-1} A r_{k-1}^{(1)} - q_{k-1}\Gamma_{k-1}^{(2)} \\ &= r_{k-1} - q_{k-1}\Gamma_{k-1}^{(1)} - \beta_{k-1} A(r_{k-1} - q_{k-1}\Gamma_{k-1}^{(1)}) - q_{k-1}\Gamma_{k-1}^{(2)} \\ &= r_0 + Q_{k-1}\bar{\Gamma}_{k-1} - q_{k-1}\Gamma_{k-1}^{(1)} - \beta_{k-1} A(r_0 + Q_{k-1}\bar{\Gamma}_{k-1}) + \beta_{k-1} A q_{k-1}\Gamma_{k-1}^{(1)} - q_{k-1}\Gamma_{k-1}^{(2)}, \end{aligned}$$

where the last equality is due to $r_{k-1} = r_0 + Q_{k-1}\bar{\Gamma}_{k-1}$ by the inductive hypothesis. Since $\mathrm{range}(Q_{k-1}) = A\mathcal{K}_{k-1}(A, r_0) \subseteq A\mathcal{K}_k(A, r_0)$, $q_{k-1} \in \mathrm{range}(Q_{k-1})$, $Ar_0 \in A\mathcal{K}_k(A, r_0)$, $\mathrm{range}(AQ_{k-1}) \subseteq A^2\mathcal{K}_{k-1}(A, r_0) \subseteq A\mathcal{K}_k(A, r_0)$, and $Aq_{k-1} \in \mathrm{range}(AQ_{k-1})$, it is clear that $r_k = r_0 + Q_k\bar{\Gamma}_k \in \mathcal{K}_{k+1}(A, r_0)$ for some $\bar{\Gamma}_k \in \mathbb{R}^k$. The relation (iv) is proved.

Finally, we prove the relation (iii). For proving $r_k^{(1)} \perp \mathrm{range}(P_k)$, note that $r_k^{(1)} \perp \mathrm{span}\{p_k\}$ already holds due to the projection step (Line 9 and Line 10 in Algorithm 2) and $p_k^{\mathrm{T}} q_k \neq 0$. It suffices to prove $r_k^{(1)} \perp \mathrm{range}(P_{k-1})$. We first prove $r_k \perp \mathrm{range}(P_{k-1})$. The projection step (Line 12 and Line 13 in Algorithm 2) at the $(k-1)$-iteration ensures $r_k \perp p_{k-1}$. It remains to prove $r_k \perp \mathrm{range}(P_{k-2})$. Because $r_k = r_{k-1}^{(2)} - q_{k-1}\Gamma_{k-1}^{(2)} = r_{k-1}^{(1)} - \beta_{k-1} A r_{k-1}^{(1)} - q_{k-1}\Gamma_{k-1}^{(2)}$, and $r_{k-1}^{(1)} \perp \mathrm{range}(P_{k-2}), q_{k-1} \perp \mathrm{range}(P_{k-2}), (Ar_{k-1}^{(1)})^{\mathrm{T}} P_{k-2} = (r_{k-1}^{(1)})^{\mathrm{T}}(AP_{k-2}) = 0$ as $\mathrm{range}(AP_{k-2}) \subseteq \mathcal{K}_{k-1}(A, r_0) = \mathrm{range}(P_{k-1})$, it follows that $r_k \perp \mathrm{range}(P_{k-2})$. Therefore, $r_k \perp \mathrm{range}(P_{k-1})$. Also noting that $q_k \perp \mathrm{range}(P_{k-1})$, we have $r_k^{(1)} = r_k - q_k\Gamma_k^{(1)} \perp \mathrm{range}(P_{k-1})$. So $r_k^{(1)} \perp \mathrm{range}(P_k)$.

To prove $x_k^{(1)} = x_k^{\mathrm{FOM}} := x_0 + z_k$, where $z_k \in \mathcal{K}_k(A, r_0)$ should satisfy $r_0 - Az_k \perp \mathcal{K}_k(A, r_0)$, first we have $r_k = r_0 + Q_k\bar{\Gamma}_k$, where $\bar{\Gamma}_k \in \mathbb{R}^k$. Hence $r_k^{(1)} = r_k - q_k\Gamma_k^{(1)} = r_0 + Q_k\bar{\Gamma}_k - q_k\Gamma_k^{(1)} = r_0 - Q_k\eta_k$, where $\eta_k \in \mathbb{R}^k$. Since $P_k^{\mathrm{T}} Q_k$ is nonsingular, there exists a unique $\eta_k \in \mathbb{R}^k$ such that $r_k^{(1)} \perp \mathrm{range}(P_k)$, i.e., $r_k^{(1)}$ is the oblique projection of r_0 onto the subspace $\mathrm{range}(P_k)^{\perp}$ along $\mathrm{range}(Q_k)$. On the other side, for FOM, $r_k^{\mathrm{FOM}} = r_0 - Az_k \perp \mathcal{K}_k(A, r_0) = \mathrm{range}(P_k)$, so r_k^{FOM} is also the oblique projection of r_0 onto the subspace $\mathrm{range}(P_k)^{\perp}$ along $\mathrm{range}(Q_k)$. So $r_k^{(1)} = r_k^{\mathrm{FOM}}$, which further indicates $x_k^{(1)} = x_k^{\mathrm{FOM}}$. Hence, the relation (iii) holds.

With relations (i)-(iv) being proved in the k-th iteration, we complete the induction. □

B.3 Convergence analysis

Since the basic Min-AM is equivalent to FOM (or CG) method in the sense that $\bar{x}_k^{(1)} = x_k^{\mathrm{FOM}} = x_k^{\mathrm{CG}}$ for strongly convex quadratic optimization, where x_k^{CG} is the k-th iterate of CG, we can obtain the convergence result as a corollary of Theorem 1.

Corollary 1. *For solving the strongly convex quadratic problem* (47)*, let x^* be the exact solution, and $\{x_k\}$ is the sequence generated by the basic Min-AM described in Algorithm 2. Define $\theta_k = \|I - \beta_k A\|_2$. Then*

$$\|x_{k+1} - x^*\|_A \leq \theta_k \min_{p \in \mathcal{P}_k, p(0)=1} \|p(A)(x_0 - x^*)\|_A, \tag{59}$$

where $\mathcal{P}_k$ denotes the space of polynomials of degree not exceeding k. Moreover, the algorithm finds the exact solution in at most $(d+1)$ iterations.

Proof. From Theorem 1, the classical convergence analysis of the CG method [28, 53], and assuming that CG and Min-AM are initialized from the same starting point, i.e., $x_0^{\mathrm{CG}} = x_0$, we have

$$\|x_k^{(1)} - x^*\|_A = \|x_k^{\mathrm{CG}} - x^*\|_A = \min_{p \in \mathcal{P}_k, p(0)=1} \|p(A)(x_0 - x^*)\|_A. \tag{60}$$

Since $x_k^{(2)} = x_k^{(1)} + \beta_k r_k^{(1)}$, it follows that

$$x_k^{(2)} - x^* = x_k^{(1)} - x^* + \beta_k(b - Ax_k^{(1)}) = x_k^{(1)} - x^* - \beta_k A(x_k^{(1)} - x^*) = (I - \beta_k A)(x_k^{(1)} - x^*).$$

Hence,

$$\|x_k^{(2)} - x^*\|_A \le \|I - \beta_k A\|_A \|x_k^{(1)} - x^*\|_A = \theta_k \|x_k^{(1)} - x^*\|_A, \tag{61}$$

where we use the fact that

$$\|I - \beta_k A\|_A = \max_{\|x\|_A = 1} \|(I - \beta_k A)x\|_A = \max_{\|A^{1/2}x\|_2 = 1} \|A^{1/2}(I - \beta_k A)x\|_2 \tag{62}$$

$$= \max_{\|y\|_2 = 1} \|A^{1/2}(I - \beta_k A)A^{-1/2}y\|_2 = \|A^{1/2}(I - \beta_k A)A^{-1/2}\|_2 = \|I - \beta_k A\|_2. \tag{63}$$

Noting that $x_{k+1} = x_k^{(2)} - p_k \Gamma_k^{(2)}$, $r_{k+1} = r_k^{(2)} + Ap_k \Gamma_k^{(2)} \perp p_k$, it follows that

$$\Gamma_k^{(2)} = \arg\min_{\Gamma \in \mathbb{R}} \|x_k^{(2)} - p_k \Gamma - x^*\|_A.$$

So

$$\|x_{k+1} - x^*\|_A \le \|x_k^{(2)} - x^*\|_A. \tag{64}$$

Combining (60), (61), and (64) yields (59).

Since CG iterates at most d steps to obtain x^*, the Min-AM will obtain the same solution in at most $(d+1)$ steps. □

C Details of the restarted Min-AM

C.1 Proof of Theorem 2

For the proof of Theorem 2 about the restarted Min-AM, we use a local quadratic model inspired by the techniques in [17, 18]. The second-order Taylor expansion $\hat{f}$ of f around x^* is

$$\hat{f}(x) = f(x^*) + \frac{1}{2}(x - x^*)^{\mathrm{T}} A (x - x^*), \tag{65}$$

where $A := \nabla^2 f(x^*)$.

We compare the iterates $\{x_k\}$ generated by Min-AM to the iterates $\{\hat{x}_k\}$ generated by applying Min-AM to minimizing the quadratic model (65). More precisely, we give the following definition.

Definition 1. *Let the mixing parameter $\{\beta_k\}$ be chosen to satisfy $\beta \le |\beta_k| \le \beta'$, where β and β' are two positive constants. The sequences $\{x_k\}$ and $\{\hat{x}\}$ are generated by the following two processes:*

(1) Process I: Solve the optimization problem (1) *with the restarted Min-AM (see Algorithm 1), and the resulting sequence is $\{x_k\}$.*

(2) Process II: Apply the basic Min-AM to minimize $\hat{f}(x)$ in each interval between two successive restarts in Process I. Specifically, let m_k, β_k be the same as those in Process I. Define $\hat{r}_k = -\nabla \hat{f}(\hat{x}_k)$. Then, $\hat{x}_k = x_k$, and $\hat{p}_k = \hat{q}_k = \mathbf{0} \in \mathbb{R}^d$, if $m_k = 0$. For $m_k \ge 1$,

$$\Delta \hat{x}_{k-1} = \hat{x}_k - \hat{x}_{k-1}, \quad \Delta \hat{r}_{k-1} = \hat{r}_k - \hat{r}_{k-1},$$

$$\hat{p}_k = \Delta \hat{x}_{k-1} - \hat{p}_{k-1}\hat{\zeta}_k, \quad \hat{q}_k = \Delta \hat{r}_{k-1} - \hat{q}_{k-1}\hat{\zeta}_k, \quad \textit{where } \hat{\zeta}_k = (\hat{p}_{k-1}^{\mathrm{T}} \hat{q}_{k-1})^{\dagger} \hat{p}_{k-1}^{\mathrm{T}} \Delta \hat{r}_{k-1};$$

defining $\hat{r}_k^{(1)} = \hat{r}_k - \hat{q}_k \hat{\Gamma}_k^{(1)}$, the update of $\hat{x}_k$ is

$$\begin{aligned} &\hat{x}_k^{(1)} = \hat{x}_k - \hat{p}_k \hat{\Gamma}_k^{(1)}, \quad \textit{where } \hat{\Gamma}_k^{(1)} = (\hat{p}_k^{\mathrm{T}} \hat{q}_k)^{\dagger} \hat{p}_k^{\mathrm{T}} \hat{r}_k, \\ &\hat{x}_k^{(2)} = \hat{x}_k^{(1)} + \beta_k \hat{r}_k^{(1)}, \\ &\hat{x}_{k+1} = \hat{x}_k^{(2)} - \hat{p}_k \hat{\Gamma}_k^{(2)}, \quad \textit{where } \hat{\Gamma}_k^{(2)} = \beta_k (\hat{p}_k^{\mathrm{T}} \hat{q}_k)^{\dagger} \hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}. \end{aligned} \tag{66}$$

By comparing $\{x_k\}$ and $\{\hat{x}\}$, we first have the following lemma.

Lemma 1. *Suppose that Assumption 1 holds for the optimization problem* (1). *For the sequences* $\{x_k\}$ *and* $\{\hat{x}_k\}$ *defined in Definition 1, if* $\|x_0 - x^*\|_2$ *is sufficiently small and there exists a positive constant* η_0 *such that* $\|\nabla f(x_j)\|_2 \le \eta_0 \|\nabla f(x_0)\|_2$ *for* $j = 0, \dots, k$, *then*

$$\|r_k - \hat{r}_k\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2), \tag{67}$$

$$\|x_{k+1} - \hat{x}_{k+1}\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2). \tag{68}$$

Proof. Besides (67) and (68), we will also prove the following relations.

$$x_k \in \mathcal{B}_{\hat{\rho}}(x^*), \tag{69}$$

$$|\zeta_k| = \mathcal{O}(1), \qquad |\hat{\zeta}_k - \zeta_k| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2), \tag{70}$$

$$\|p_k\|_2 = \mathcal{O}(\|x_{k-m_k} - x^*\|_2), \qquad \|q_k\|_2 = \mathcal{O}(\|x_{k-m_k} - x^*\|_2) \tag{71}$$

$$\|p_k - \hat{p}_k\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2), \qquad \|q_k - \hat{q}_k\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2), \tag{72}$$

$$|\Gamma_k^{(1)}| = \mathcal{O}(1), \qquad |\hat{\Gamma}_k^{(1)} - \Gamma_k^{(1)}| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2), \tag{73}$$

$$\|x_k^{(1)} - \hat{x}_k^{(1)}\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2), \qquad \|r_k^{(1)} - \hat{r}_k^{(1)}\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2), \tag{74}$$

$$\|x_k^{(2)} - \hat{x}_k^{(2)}\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2). \tag{75}$$

$$|\Gamma_k^{(2)}| = \mathcal{O}(1), \qquad |\hat{\Gamma}_k^{(2)} - \Gamma_k^{(2)}| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{76}$$

Here, for convenience, we define $\zeta_k = \hat{\zeta}_k = 0$ if $m_k = 0$.

We first prove (69). Recall that $r_k = -\nabla f(x_k)$. Due to (12a), we have

$$\mu\|x_k - x^*\|_2 \le \|r_k\|_2 = \|\nabla f(x_k) - \nabla f(x^*)\|_2 \le L\|x_k - x^*\|_2. \tag{77}$$

Note that $\|r_k\|_2 \le \eta_0\|r_0\|_2$. By choosing $\|x_0 - x^*\|_2 \le \frac{\mu\hat{\rho}}{\eta_0 L}$, we can ensure

$$\|x_k - x^*\|_2 \le \frac{1}{\mu}\|r_k\|_2 \le \frac{\eta_0}{\mu}\|r_0\|_2 \le \frac{\eta_0 L}{\mu}\|x_0 - x^*\|_2 \le \frac{\eta_0 L}{\mu} \cdot \frac{\mu\hat{\rho}}{\eta_0 L} = \hat{\rho}. \tag{78}$$

Then (69) holds. Since $\|r_j\|_2 \le \eta_0\|r_0\|_2$ holds for $j = 0, \dots, k$, (78) implies that a sufficiently small $\|x_0 - x^*\|_2$ ensures $\|x_{k-m_k} - x^*\|_2 \le \frac{\eta_0 L}{\mu}\|x_0 - x^*\|_2$ is sufficiently small. Next, we prove (67), (68), and (70)-(76) by induction.

For $k = 0$, we have $\Gamma_k^{(1)} = \hat{\Gamma}_k^{(1)} = 0, \Gamma_k^{(2)} = \hat{\Gamma}_k^{(2)} = 0$. The relations (70)-(74), and (76) clearly hold. Also, due to (12b),

$$\|r_0 - \hat{r}_0\|_2 = \|\nabla f(x_0) - \nabla \hat{f}(\hat{x}_0)\|_2 = \|\nabla f(x_0) - \nabla^2 f(x^*)(x_0 - x^*)\|_2 \le \frac{1}{2}\hat{\kappa}\|x_0 - x^*\|_2^2,$$

namely (67). Note that $x_{k+1} = x_k^{(2)} = x_k + \beta_k r_k$ and $\hat{x}_{k+1} = \hat{x}_k^{(2)} = \hat{x}_k + \beta_k \hat{r}_k$ in this case. Then (68) and (75) follow from

$$\|x_1 - \hat{x}_1\|_2 = \|x_0 + \beta_0 r_0 - (\hat{x}_0 + \beta_0 \hat{r}_0)\|_2 = \beta_0\|r_0 - \hat{r}_0\|_2 \le \frac{\beta_0\hat{\kappa}}{2}\|x_0 - x^*\|_2^2.$$

Suppose that $k \ge 1$, and as an inductive hypothesis, the relations (67), (68), and (70)-(76) hold for $j = 0, \dots, k-1$. Consider the k-th iteration.

If $m_k = 0$, i.e., a condition in (11a)-(11c) is violated at the beginning of the k-th iteration, then $\hat{x}_k = x_k$. The same as the case that $k = 0$, (67), (68), and (70)-(76) hold.

Consider the nontrivial case that $m_k \ge 1$. From (11c), it follows that

$$\|x_j - x^*\|_2 \le \frac{1}{\mu}\|r_j\|_2 \le \frac{\eta}{\mu}\|r_{k-m_k}\|_2 \le \frac{\eta L}{\mu}\|x_{k-m_k} - x^*\|_2, \;\; j = k - m_k + 1, \dots, k.$$

Therefore,

$$\|x_j - x^*\|_2 = \mathcal{O}(\|x_{k-m_k} - x^*\|_2), \;\; j = k - m_k, \dots, k. \tag{79}$$

Since $x_k \in \mathcal{B}_{\hat{\rho}}(x^*)$, we have

$$\begin{aligned}
\|r_k - \hat{r}_k\|_2 &= \|\nabla f(x_k) - \nabla \hat{f}(\hat{x}_k)\|_2 \\
&\le \|\nabla f(x_k) - \nabla \hat{f}(x_k)\|_2 + \|\nabla \hat{f}(x_k) - \nabla \hat{f}(\hat{x}_k)\|_2 \\
&= \|\nabla f(x_k) - \nabla^2 f(x^*)(x_k - x^*)\|_2 + \|\nabla^2 f(x^*)(x_k - \hat{x}_k)\|_2 \\
&\le \frac{1}{2}\hat{\kappa}\|x_k - x^*\|_2^2 + L\|x_k - \hat{x}_k\|_2 \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2),
\end{aligned}$$

where the second inequality is due to (12b) and (12a), and the last inequality is due to (79) and the inductive hypothesis (68). Hence, the relation (67) holds. Next, we prove the relation (70).

Due to the check (11b), with $\underline{\kappa} := \tau\mu^3\beta^2$, we have

$$\begin{aligned}
|p_j^{\mathrm{T}} q_j| &\ge \tau |p_{k-m_k+1}^{\mathrm{T}} q_{k-m_k+1}| = \tau |\Delta x_{k-m_k}^{\mathrm{T}} \Delta r_{k-m_k}| \\
&\ge \tau \left| \Delta x_{k-m_k}^{\mathrm{T}} \int_0^1 \nabla^2 f(x_{k-m_k} + t\Delta x_{k-m_k}) \Delta x_{k-m_k} dt \right| \\
&\ge \tau\mu \|\Delta x_{k-m_k}\|_2^2 = \tau\mu\beta_{k-m_k}^2 \|r_{k-m_k}\|_2^2 \\
&\ge \tau\mu^3\beta^2 \|x_{k-m_k} - x^*\|_2^2 = \underline{\kappa}\|x_{k-m_k} - x^*\|_2^2,
\end{aligned} \tag{80}$$

for $j = k - m_k + 1, \ldots, k$.

If $m_k = 1$, then $\zeta_k = \hat{\zeta}_k = 0$, and (70) holds. For $m_k > 1$, we have

$$|\zeta_k| = \left| \frac{p_{k-1}^{\mathrm{T}} \Delta r_{k-1}}{p_{k-1}^{\mathrm{T}} q_{k-1}} \right| \le \frac{\|p_{k-1}\|_2 \cdot \|\Delta r_{k-1}\|_2}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \frac{\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2)}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \mathcal{O}(1), \tag{81}$$

due to (71) and $\|\Delta r_{k-1}\|_2 \le \|r_k\|_2 + \|r_{k-1}\|_2 \le 2\eta\|r_{k-m_k}\|_2 = \mathcal{O}(\|x_{k-m_k} - x^*\|_2)$.

Next, if $p_{k-1}^{\mathrm{T}} \Delta r_{k-1} \neq 0$, then

$$\begin{aligned}
|\zeta_k - \hat{\zeta}_k| &= |\zeta_k| \cdot \left| 1 - \frac{\hat{\zeta}_k}{\zeta_k} \right| = |\zeta_k| \cdot \left| 1 - \frac{\hat{p}_{k-1}^{\mathrm{T}} \Delta \hat{r}_{k-1}}{\hat{p}_{k-1}^{\mathrm{T}} \hat{q}_{k-1}} \cdot \frac{p_{k-1}^{\mathrm{T}} q_{k-1}}{p_{k-1}^{\mathrm{T}} \Delta r_{k-1}} \right| \\
&= |\zeta_k| \cdot \left| 1 - \frac{\hat{p}_{k-1}^{\mathrm{T}} \Delta \hat{r}_{k-1}}{p_{k-1}^{\mathrm{T}} \Delta r_{k-1}} \cdot \frac{p_{k-1}^{\mathrm{T}} q_{k-1}}{\hat{p}_{k-1}^{\mathrm{T}} \hat{q}_{k-1}} \right| \\
&= |\zeta_k| \cdot |a(1-b) + b| \le |\zeta_k| \cdot (|a| + |b| + |ab|),
\end{aligned} \tag{82}$$

where $a := 1 - \frac{\hat{p}_{k-1}^{\mathrm{T}} \Delta \hat{r}_{k-1}}{p_{k-1}^{\mathrm{T}} \Delta r_{k-1}}$, $b := 1 - \frac{p_{k-1}^{\mathrm{T}} q_{k-1}}{\hat{p}_{k-1}^{\mathrm{T}} \hat{q}_{k-1}}$. We have

$$|\zeta_k| \cdot |a| = \left| \frac{p_{k-1}^{\mathrm{T}} \Delta r_{k-1}}{p_{k-1}^{\mathrm{T}} q_{k-1}} \right| \cdot \left| \frac{p_{k-1}^{\mathrm{T}} \Delta r_{k-1} - \hat{p}_{k-1}^{\mathrm{T}} \Delta \hat{r}_{k-1}}{p_{k-1}^{\mathrm{T}} \Delta r_{k-1}} \right| = \left| \frac{p_{k-1}^{\mathrm{T}} \Delta r_{k-1} - \hat{p}_{k-1}^{\mathrm{T}} \Delta \hat{r}_{k-1}}{p_{k-1}^{\mathrm{T}} q_{k-1}} \right|. \tag{83}$$

From (67) and (71),

$$|p_{k-1}^{\mathrm{T}} (\Delta r_{k-1} - \Delta \hat{r}_{k-1})| \le \|p_{k-1}\|_2 \cdot \|r_k - r_{k-1} - \hat{r}_k + \hat{r}_{k-1}\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).$$

We also have

$$\begin{aligned}
|(p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}} \Delta \hat{r}_{k-1}| &= |(p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}} A \Delta \hat{x}_{k-1}| \\
&= |(p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}} A \Delta x_{k-1} - (p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}} A (\Delta x_{k-1} - \Delta \hat{x}_{k-1})| \\
&\le |(p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}} A \Delta x_{k-1}| + |(p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}} A (\Delta x_{k-1} - \Delta \hat{x}_{k-1})| \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3) + \hat{\kappa}^2\mathcal{O}(\|x_{k-m_k} - x^*\|_2^4) \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).
\end{aligned}$$

Hence,

$$\begin{aligned}|p_{k-1}^{\mathrm{T}}\Delta r_{k-1} - \hat{p}_{k-1}^{\mathrm{T}}\Delta\hat{r}_{k-1}| &= |p_{k-1}^{\mathrm{T}}(\Delta r_{k-1} - \Delta\hat{r}_{k-1}) + (p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}}\Delta\hat{r}_{k-1}| \\ &\le |p_{k-1}^{\mathrm{T}}(\Delta r_{k-1} - \Delta\hat{r}_{k-1})| + |(p_{k-1} - \hat{p}_{k-1})^{\mathrm{T}}\Delta\hat{r}_{k-1}| \\ &= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3). \end{aligned} \tag{84}$$

Combining (83) and (84), we have

$$|\zeta_k| \cdot |a| \le \frac{\hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3)}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{85}$$

Note that

$$\begin{aligned}&|\hat{p}_{k-1}^{\mathrm{T}}\hat{q}_{k-1} - p_{k-1}^{\mathrm{T}}q_{k-1}| = |\hat{p}_{k-1}^{\mathrm{T}}(\hat{q}_{k-1} - q_{k-1}) + (\hat{p}_{k-1} - p_{k-1})^{\mathrm{T}}q_{k-1}| \\ &= |(\hat{p}_{k-1} - p_{k-1})^{\mathrm{T}}(\hat{q}_{k-1} - q_{k-1}) + p_{k-1}^{\mathrm{T}}(\hat{q}_{k-1} - q_{k-1}) + (\hat{p}_{k-1} - p_{k-1})^{\mathrm{T}}q_{k-1}| \\ &\le |(\hat{p}_{k-1} - p_{k-1})^{\mathrm{T}}(\hat{q}_{k-1} - q_{k-1})| + |p_{k-1}^{\mathrm{T}}(\hat{q}_{k-1} - q_{k-1})| + |(\hat{p}_{k-1} - p_{k-1})^{\mathrm{T}}q_{k-1}| \\ &= \hat{\kappa}^2\mathcal{O}(\|x_{k-m_k} - x^*\|_2^4) + \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3) + \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3) \\ &= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3), \end{aligned} \tag{86}$$

and

$$\begin{aligned}|\hat{p}_{k-1}^{\mathrm{T}}\hat{q}_{k-1}| &\ge |p_{k-1}^{\mathrm{T}}q_{k-1}| - |\hat{p}_{k-1}^{\mathrm{T}}\hat{q}_{k-1} - p_{k-1}^{\mathrm{T}}q_{k-1}| \\ &\ge \underline{\kappa}\|x_{k-m_k} - x^*\|_2^2 - \hat{\kappa}c_1\|x_{k-m_k} - x^*\|_2^3 \ge \frac{1}{2}\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2, \end{aligned} \tag{87}$$

where the existence of the constant c_1 is guaranteed by (86), and the last inequality holds by choosing $\|x_0 - x^*\|_2 \le \frac{\mu\underline{\kappa}}{2\eta_0 L\hat{\kappa}c_1}$, which ensures $\hat{\kappa}c_1\|x_{k-m_k} - x^*\|_2 \le \hat{\kappa}c_1\frac{\eta_0 L}{\mu}\|x_0 - x^*\|_2 \le \frac{1}{2}\underline{\kappa}$.

Then, we have

$$|b| = \left|1 - \frac{p_{k-1}^{\mathrm{T}}q_{k-1}}{\hat{p}_{k-1}^{\mathrm{T}}\hat{q}_{k-1}}\right| = \left|\frac{\hat{p}_{k-1}^{\mathrm{T}}\hat{q}_{k-1} - p_{k-1}^{\mathrm{T}}q_{k-1}}{\hat{p}_{k-1}^{\mathrm{T}}\hat{q}_{k-1}}\right| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{88}$$

As a result, by (85), (88), (81), and (82), we have

$$|\zeta_k - \hat{\zeta}_k| \le |\zeta_k||a| + |\zeta_k||b| + |\zeta_k||a||b| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{89}$$

Now consider the case that $p_{k-1}^{\mathrm{T}}\Delta r_{k-1} = 0$. It is clear that $\zeta_k = 0$. Then

$$|\zeta_k - \hat{\zeta}_k| = \left|\frac{\hat{p}_{k-1}^{\mathrm{T}}\Delta\hat{r}_{k-1}}{\hat{p}_{k-1}^{\mathrm{T}}\hat{q}_{k-1}}\right| \le \frac{\hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3)}{\frac{1}{2}\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2),$$

where the inequality is due to (84) and (87).

Hence the (70) holds.

For the relation (71), it holds since

$$\begin{aligned}\|p_k\|_2 = \|\Delta x_{k-1} - p_{k-1}\zeta_k\|_2 &\le \|x_k - x^*\|_2 + \|x_{k-1} - x^*\|_2 + \|p_{k-1}\|_2|\zeta_k| \\ &= \mathcal{O}(\|x_{k-m_k} - x^*\|_2), \\ \|q_k\|_2 = \|\Delta r_{k-1} - q_{k-1}\zeta_k\|_2 &\le L\|x_k - x_{k-1}\|_2 + \|q_{k-1}\|_2|\zeta_k| \\ &\le L\|x_k - x^*\|_2 + L\|x_{k-1} - x^*\|_2 + \|q_{k-1}\|_2|\zeta_k| \\ &= \mathcal{O}(\|x_{k-m_k} - x^*\|_2). \end{aligned}$$

Next, we prove (72). We have

$$\begin{aligned}&|p_{k-1}\zeta_k - \hat{p}_{k-1}\hat{\zeta}_k\|_2 = \|p_{k-1}\zeta_k - \hat{p}_{k-1}(\hat{\zeta}_k - \zeta_k) - \hat{p}_{k-1}\zeta_k\|_2 \\ &\le \|(p_{k-1} - \hat{p}_{k-1})\zeta_k\|_2 + \|\hat{p}_{k-1}(\hat{\zeta}_k - \zeta_k)\|_2 \\ &\le \|(p_{k-1} - \hat{p}_{k-1})\zeta_k\|_2 + \|(\hat{p}_{k-1} - p_{k-1})(\hat{\zeta}_k - \zeta_k)\|_2 + \|p_{k-1}(\hat{\zeta}_k - \zeta_k)\|_2 \\ &= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2), \end{aligned} \tag{90}$$

$$
\begin{aligned}
&|q_{k-1}\zeta_k - \hat{q}_{k-1}\hat{\zeta}_k\|_2 = \|q_{k-1}\zeta_k - \hat{q}_{k-1}(\hat{\zeta}_k - \zeta_k) - \hat{q}_{k-1}\zeta_k\|_2 \\
&\le \|(q_{k-1} - \hat{q}_{k-1})\zeta_k\|_2 + \|(\hat{q}_{k-1} - q_{k-1})(\hat{\zeta}_k - \zeta_k)\|_2 + \|q_{k-1}(\hat{\zeta}_k - \zeta_k)\|_2 \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2).
\end{aligned} \tag{91}
$$

Therefore,

$$
\begin{aligned}
\|p_k - \hat{p}_k\|_2 &= \|\Delta x_{k-1} - p_{k-1}\zeta_k - (\Delta\hat{x}_{k-1} - \hat{p}_{k-1}\hat{\zeta}_k)\|_2 \\
&\le \|\Delta x_{k-1} - \Delta\hat{x}_{k-1}\|_2 + \|p_{k-1}\zeta_k - \hat{p}_{k-1}\hat{\zeta}_k\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2),
\end{aligned} \tag{92}
$$

$$
\begin{aligned}
\|q_k - \hat{q}_k\|_2 &= \|\Delta r_{k-1} - q_{k-1}\zeta_k - (\Delta\hat{r}_{k-1} - \hat{q}_{k-1}\hat{\zeta}_k)\|_2 \\
&\le \|\Delta r_{k-1} - \Delta\hat{r}_{k-1}\|_2 + \|q_{k-1}\zeta_k - \hat{q}_{k-1}\hat{\zeta}_k\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2).
\end{aligned} \tag{93}
$$

The (72) is proved.

Now, we prove (73), which follows a very similar procedure in proving (70). For the completeness, we still give the proof.

We have

$$
|\Gamma_k^{(1)}| = \left|\frac{p_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} q_k}\right| \le \frac{\|p_k\|_2 \cdot \|r_k\|_2}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \frac{\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2)}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \mathcal{O}(1), \tag{94}
$$

due to (71) and $\|r_k\|_2 \le \eta\|r_{k-m_k}\|_2 = \mathcal{O}(\|x_{k-m_k} - x^*\|_2)$.

Next, if $p_k^{\mathrm{T}} r_k \neq 0$, then

$$
\begin{aligned}
|\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)}| &= |\Gamma_k^{(1)}| \cdot \left|1 - \frac{\hat{\Gamma}_k^{(1)}}{\Gamma_k^{(1)}}\right| = |\Gamma_k^{(1)}| \cdot \left|1 - \frac{\hat{p}_k^{\mathrm{T}}\hat{r}_k}{\hat{p}_k^{\mathrm{T}}\hat{q}_k} \cdot \frac{p_k^{\mathrm{T}} q_k}{p_k^{\mathrm{T}} r_k}\right| = |\Gamma_k^{(1)}| \cdot \left|1 - \frac{\hat{p}_k^{\mathrm{T}}\hat{r}_k}{p_k^{\mathrm{T}} r_k} \cdot \frac{p_k^{\mathrm{T}} q_k}{\hat{p}_k^{\mathrm{T}}\hat{q}_k}\right| \\
&= |\Gamma_k^{(1)}| \cdot |a_1(1 - b_1) + b_1| \le |\Gamma_k^{(1)}| \cdot (|a_1| + |b_1| + |a_1 b_1|),
\end{aligned} \tag{95}
$$

where $a_1 := 1 - \frac{\hat{p}_k^{\mathrm{T}}\hat{r}_k}{p_k^{\mathrm{T}} r_k}, b_1 := 1 - \frac{p_k^{\mathrm{T}} q_k}{\hat{p}_k^{\mathrm{T}}\hat{q}_k}$. We have

$$
|\Gamma_k^{(1)}| \cdot |a_1| = \left|\frac{p_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} q_k}\right| \cdot \left|\frac{p_k^{\mathrm{T}} r_k - \hat{p}_k^{\mathrm{T}}\hat{r}_k}{p_k^{\mathrm{T}} r_k}\right| = \left|\frac{p_k^{\mathrm{T}} r_k - \hat{p}_k^{\mathrm{T}}\hat{r}_k}{p_k^{\mathrm{T}} q_k}\right|. \tag{96}
$$

From (67) and (71) which have been proved,

$$
|p_k^{\mathrm{T}}(r_k - \hat{r}_k)| \le \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).
$$

We also have

$$
\begin{aligned}
|(p_k - \hat{p}_k)^{\mathrm{T}}\hat{r}_k| &= |(p_k - \hat{p}_k)^{\mathrm{T}} A(\hat{x}_k - x^*)| \\
&= |(p_k - \hat{p}_k)^{\mathrm{T}} A(x_k - x^*) - (p_k - \hat{p}_k)^{\mathrm{T}} A(x_k - \hat{x}_k)| \\
&\le |(p_k - \hat{p}_k)^{\mathrm{T}} A(x_k - x^*)| + |(p_k - \hat{p}_k)^{\mathrm{T}} A(x_k - \hat{x}_k)| \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3) + \hat{\kappa}^2\mathcal{O}(\|x_{k-m_k} - x^*\|_2^4) = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).
\end{aligned}
$$

Hence,

$$
\begin{aligned}
|p_k^{\mathrm{T}} r_k - \hat{p}_k^{\mathrm{T}}\hat{r}_k| &= |p_k^{\mathrm{T}}(r_k - \hat{r}_k) + (p_k - \hat{p}_k)^{\mathrm{T}}\hat{r}_k| \\
&\le |p_k^{\mathrm{T}}(r_k - \hat{r}_k)| + |(p_k - \hat{p}_k)^{\mathrm{T}}\hat{r}_k| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).
\end{aligned} \tag{97}
$$

Combining (96) and (97), we have

$$
|\Gamma_k^{(1)}| \cdot |a_1| \le \frac{\hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3)}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{98}
$$

Note that

$$
\begin{aligned}
&|\hat{p}_k^{\mathrm{T}}\hat{q}_k - p_k^{\mathrm{T}} q_k| = |\hat{p}_k^{\mathrm{T}}(\hat{q}_k - q_k) + (\hat{p}_k - p_k)^{\mathrm{T}} q_k| \\
&= |(\hat{p}_k - p_k)^{\mathrm{T}}(\hat{q}_k - q_k) + p_k^{\mathrm{T}}(\hat{q}_k - q_k) + (\hat{p}_k - p_k)^{\mathrm{T}} q_k| \\
&\le |(\hat{p}_k - p_k)^{\mathrm{T}}(\hat{q}_k - q_k)| + |p_k^{\mathrm{T}}(\hat{q}_k - q_k)| + |(\hat{p}_k - p_k)^{\mathrm{T}} q_k| \\
&= \hat{\kappa}^2\mathcal{O}(\|x_{k-m_k} - x^*\|_2^4) + \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3) + \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3) \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3),
\end{aligned} \tag{99}
$$

and

$$
\begin{aligned}
|\hat{p}_k^{\mathrm{T}}\hat{q}_k| &\geq |p_k^{\mathrm{T}}q_k| - |\hat{p}_k^{\mathrm{T}}\hat{q}_k - p_k^{\mathrm{T}}q_k| \\
&\geq \underline{\kappa}\|x_{k-m_k} - x^*\|_2^2 - \hat{\kappa}c_2\|x_{k-m_k} - x^*\|_2^3 \geq \frac{1}{2}\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2,
\end{aligned} \tag{100}
$$

where the existence of the constant c_2 is guaranteed by (99), and the last inequality holds by choosing $\|x_0 - x^*\|_2 \leq \frac{\mu\underline{\kappa}}{2\eta_0 L\hat{\kappa}c_2}$ which ensures $\hat{\kappa}c_2\|x_{k-m_k} - x^*\|_2 \leq \hat{\kappa}c_2\frac{\eta_0 L}{\mu}\|x_0 - x^*\|_2 \leq \frac{1}{2}\underline{\kappa}$.

Then, we have

$$
|b_1| = \left|1 - \frac{p_k^{\mathrm{T}}q_k}{\hat{p}_k^{\mathrm{T}}\hat{q}_k}\right| = \left|\frac{\hat{p}_k^{\mathrm{T}}\hat{q}_k - p_k^{\mathrm{T}}q_k}{\hat{p}_k^{\mathrm{T}}\hat{q}_k}\right| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{101}
$$

As a result, by (98), (101), (94), and (95), we have

$$
|\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)}| \leq |\Gamma_k^{(1)}||a_1| + |\Gamma_k^{(1)}||b_1| + |\Gamma_k^{(1)}|||a_1||b_1| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{102}
$$

Now consider the case that $p_k^{\mathrm{T}}r_k = 0$. It is clear that $\Gamma_k^{(1)} = 0$. Then

$$
|\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)}| = \left|\frac{\hat{p}_k^{\mathrm{T}}\hat{r}_k}{\hat{p}_k^{\mathrm{T}}\hat{q}_k}\right| \leq \frac{\hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^3)}{\frac{1}{2}\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2),
$$

where the inequality is due to (97) and (100).

Hence the (73) holds.

Now, we prove (74). We have

$$
\begin{aligned}
\|x_k^{(1)} - \hat{x}_k^{(1)}\|_2 &= \|x_k - p_k\Gamma_k^{(1)} - (\hat{x}_k - \hat{p}_k\hat{\Gamma}_k^{(1)})\|_2 \leq \|x_k - \hat{x}_k\|_2 + \|p_k\Gamma_k^{(1)} - \hat{p}_k\hat{\Gamma}_k^{(1)}\|_2 \\
&\leq \|x_k - \hat{x}_k\|_2 + \|p_k(\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)})\|_2 + \|(p_k - \hat{p}_k)\hat{\Gamma}_k^{(1)}\|_2 \\
&\leq \|x_k - \hat{x}_k\|_2 + \|p_k(\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)})\|_2 + \|(p_k - \hat{p}_k)\Gamma_k^{(1)}\|_2 \\
&\quad + \|(p_k - \hat{p}_k)(\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)})\|_2 \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2),
\end{aligned} \tag{103}
$$

$$
\begin{aligned}
\|r_k^{(1)} - \hat{r}_k^{(1)}\|_2 &= \|r_k - q_k\Gamma_k^{(1)} - (\hat{r}_k - \hat{q}_k\hat{\Gamma}_k^{(1)})\|_2 \leq \|r_k - \hat{r}_k\|_2 + \|q_k\Gamma_k^{(1)} - \hat{q}_k\hat{\Gamma}_k^{(1)}\|_2 \\
&\leq \|r_k - \hat{r}_k\|_2 + \|q_k(\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)})\|_2 + \|(q_k - \hat{q}_k)\hat{\Gamma}_k^{(1)}\|_2 \\
&\leq \|r_k - \hat{r}_k\|_2 + \|q_k(\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)})\|_2 + \|(q_k - \hat{q}_k)\Gamma_k^{(1)}\|_2 \\
&\quad + \|(q_k - \hat{q}_k)(\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)})\|_2 \\
&= \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2).
\end{aligned} \tag{104}
$$

Then, (75) holds as

$$
\begin{aligned}
\|x_k^{(2)} - \hat{x}_k^{(2)}\|_2 &= \|x_k^{(1)} + \beta_k r_k^{(1)} - (\hat{x}_k^{(1)} + \beta_k\hat{r}_k^{(1)})\|_2 \\
&\leq \|x_k^{(1)} - \hat{x}_k^{(1)}\|_2 + \beta_k\|r_k^{(1)} - \hat{r}_k^{(1)}\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2)
\end{aligned}
$$

is a consequence of (103) and (104).

Next, we prove (76), which follows a very similar procedure in proving (73). For the completeness, we still give the proof.

We have

$$
|\Gamma_k^{(2)}| = \left|\beta_k\frac{q_k^{\mathrm{T}}r_k^{(1)}}{p_k^{\mathrm{T}}q_k}\right| \leq \frac{\beta_k\|q_k\|_2 \cdot \|r_k^{(1)}\|_2}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \frac{\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2)}{\underline{\kappa}\|x_{k-m_k} - x^*\|_2^2} = \mathcal{O}(1), \tag{105}
$$

due to (71) and $\|r_k^{(1)}\|_2 = \|r_k - q_k\Gamma_k^{(1)}\|_2 \leq \|r_k\|_2 + \|q_k\|_2|\Gamma_k^{(1)}| \leq \eta\|r_{k-m_k}\|_2 + \mathcal{O}(\|x_{k-m_k} - x^*\|_2) = \mathcal{O}(\|x_{k-m_k} - x^*\|_2)$.

Next, if $q_k^{\mathrm{T}} r_k^{(1)} \neq 0$, then

$$|\Gamma_k^{(2)} - \hat{\Gamma}_k^{(2)}| = |\Gamma_k^{(2)}| \cdot \left|1 - \frac{\hat{\Gamma}_k^{(2)}}{\Gamma_k^{(2)}}\right| = |\Gamma_k^{(2)}| \cdot \left|1 - \frac{\hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}}{\hat{p}_k^{\mathrm{T}} \hat{q}_k} \cdot \frac{p_k^{\mathrm{T}} q_k}{q_k^{\mathrm{T}} r_k^{(1)}}\right| = |\Gamma_k^{(2)}| \cdot \left|1 - \frac{\hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}}{q_k^{\mathrm{T}} r_k^{(1)}} \cdot \frac{p_k^{\mathrm{T}} q_k}{\hat{p}_k^{\mathrm{T}} \hat{q}_k}\right|$$
$$= |\Gamma_k^{(2)}| \cdot |a_2(1 - b_2) + b_2| \leq |\Gamma_k^{(2)}| \cdot (|a_2| + |b_2| + |a_2 b_2|), \tag{106}$$

where $a_2 := 1 - \frac{\hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}}{q_k^{\mathrm{T}} r_k^{(1)}}, b_2 := 1 - \frac{p_k^{\mathrm{T}} q_k}{\hat{p}_k^{\mathrm{T}} \hat{q}_k}$. We have

$$|\Gamma_k^{(2)}| \cdot |a_2| = \left|\beta_k \frac{q_k^{\mathrm{T}} r_k^{(1)}}{p_k^{\mathrm{T}} q_k}\right| \cdot \left|\frac{q_k^{\mathrm{T}} r_k^{(1)} - \hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}}{q_k^{\mathrm{T}} r_k^{(1)}}\right| = \left|\beta_k \frac{q_k^{\mathrm{T}} r_k^{(1)} - \hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}}{p_k^{\mathrm{T}} q_k}\right|. \tag{107}$$

From (71) and (74) which have been proved,

$$|q_k^{\mathrm{T}} (r_k^{(1)} - \hat{r}_k^{(1)})| \leq \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).$$

We also have

$$\begin{aligned}
|(q_k - \hat{q}_k)^{\mathrm{T}} \hat{r}_k^{(1)}| &= |(q_k - \hat{q}_k)^{\mathrm{T}} A(\hat{x}_k^{(1)} - x^*)| \\
&= |(q_k - \hat{q}_k)^{\mathrm{T}} A(x_k^{(1)} - x^*) - (q_k - \hat{q}_k)^{\mathrm{T}} A(x_k^{(1)} - \hat{x}_k^{(1)})| \\
&\leq |(q_k - \hat{q}_k)^{\mathrm{T}} A(x_k - p_k \Gamma_k^{(1)} - x^*)| + |(q_k - \hat{q}_k)^{\mathrm{T}} A(x_k^{(1)} - \hat{x}_k^{(1)})| \\
&= \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^3) + \hat{\kappa}^2 \mathcal{O}(\|x_{k-m_k} - x^*\|_2^4) = \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).
\end{aligned}$$

Hence,

$$\begin{aligned}
|q_k^{\mathrm{T}} r_k^{(1)} - \hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}| &= |q_k^{\mathrm{T}} (r_k^{(1)} - \hat{r}_k^{(1)}) + (q_k - \hat{q}_k)^{\mathrm{T}} \hat{r}_k^{(1)}| \\
&\leq |q_k^{\mathrm{T}} (r_k^{(1)} - \hat{r}_k^{(1)})| + |(q_k - \hat{q}_k)^{\mathrm{T}} r_k^{(1)}| + |(q_k - \hat{q}_k)^{\mathrm{T}} (r_k^{(1)} - \hat{r}_k^{(1)})| \\
&= \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^3).
\end{aligned} \tag{108}$$

Combining (107) and (108), we have

$$|\Gamma_k^{(2)}| \cdot |a_2| \leq \frac{\hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^3)}{\underline{\kappa} \|x_{k-m_k} - x^*\|_2^2} = \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{109}$$

From (101),

$$|b_2| = |b_1| = \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{110}$$

As a result, by (109), (110), (105), and (106), we have

$$|\Gamma_k^{(2)} - \hat{\Gamma}_k^{(2)}| \leq |\Gamma_k^{(2)}||a_2| + |\Gamma_k^{(2)}||b_2| + |\Gamma_k^{(2)}|||a_2||b_2| = \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2). \tag{111}$$

Now consider the case that $q_k^{\mathrm{T}} r_k^{(1)} = 0$. It is clear that $\Gamma_k^{(2)} = 0$. Then

$$|\Gamma_k^{(2)} - \hat{\Gamma}_k^{(2)}| = \left|\frac{\hat{q}_k^{\mathrm{T}} \hat{r}_k^{(1)}}{\hat{p}_k^{\mathrm{T}} \hat{q}_k}\right| \leq \frac{\hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^3)}{\frac{1}{2} \underline{\kappa} \|x_{k-m_k} - x^*\|_2^2} = \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2),$$

where the inequality is due to (108) and (100).

Hence the (76) holds.

Finally, we prove $\|x_{k+1} - \hat{x}_{k+1}\|_2 = \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^2)$. It follows from (8c) that

$$\begin{aligned}
\|x_{k+1} - \hat{x}_{k+1}\|_2 &= \|x_k^{(2)} - p_k \Gamma_k^{(2)} - (\hat{x}_k^{(2)} - \hat{p}_k \hat{\Gamma}_k^{(2)})\|_2 \\
&\leq \|x_k^{(2)} - \hat{x}_k^{(2)}\|_2 + \|p_k \Gamma_k^{(2)} - \hat{p}_k \hat{\Gamma}_k^{(2)}\|_2 \\
&\leq \|x_k^{(2)} - \hat{x}_k^{(2)}\|_2 + \|(p_k - \hat{p}_k) \Gamma_k^{(2)}\|_2 + \|p_k (\Gamma_k^{(2)} - \hat{\Gamma}_k^{(2)})\|_2 \\
&\quad + \|(\hat{p}_k - p_k)(\Gamma_k^{(2)} - \hat{\Gamma}_k^{(2)})\|_2 \\
&= \hat{\kappa} \mathcal{O}(\|x_{k-m_k} - x^*\|_2^2),
\end{aligned}$$

where the last equality is due to (75), (71), (72), and (76) that have been proved. Therefore, the relation (68) holds.

As a result, we complete the induction. The relations (67) and (68) are proved. □

Based on Lemma 1 and Corollary 1, we can obtain the convergence theorem of the restarted Min-AM.

Theorem 6. *Suppose that Assumption 1 holds for the optimization problem* (1)*. Let* $\{x_k\}$ *denote the sequence of iterates generated by the restarted Min-AM, and define* $\theta_k = \|I - \beta_k A\|_2$*. Assume* $\beta_j \in [\beta, \beta']$ $(j \geq 0)$ *for some positive constants* β *and* β'*. If* $\|\nabla f(x_j)\|_2 \leq \eta_0 \|\nabla f(x_0)\|_2$ $(0 \leq j \leq k)$ *for a constant* $\eta_0 > 0$*, and* x_0 *is sufficiently close to* x^**, then*

$$\|x_{k+1} - x^*\|_A \leq \theta_k \min_{p \in \mathcal{P}_{m_k}, p(0)=1} \|p(A)(x_{k-m_k} - x^*)\|_A + \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2), \tag{112}$$

where $\mathcal{P}_{m_k}$ *is the space of polynomials of degree not exceeding* m_k*. As a result,*

$$\|x_{k+1} - x^*\|_A \leq 2\theta_k \left(\frac{\sqrt{L/\mu} - 1}{\sqrt{L/\mu} + 1}\right)^{m_k} \|x_{k-m_k} - x^*\|_A + \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2). \tag{113}$$

Proof. From Corollary 1, the auxiliary sequence $\{\hat{x}_k\}$ in Definition 1 satisfies

$$\|\hat{x}_{k+1} - x^*\|_A \leq \theta_k \min_{p \in \mathcal{P}_{m_k}, p(0)=1} \|p(A)(x_{k-m_k} - x^*)\|_A. \tag{114}$$

Then (112) follows from (114), (68), and the fact that $\|x_{k+1} - \hat{x}_{k+1}\|_A \leq \sqrt{L}\|x_{k+1} - \hat{x}_{k+1}\|_2$.

Since A is SPD, we can choose the Chebyshev polynomial for $p \in \mathcal{P}_{m_k}$ to obtain (e.g., see Section 6.11.3 in [53])

$$\min_{p \in \mathcal{P}_{m_k}, p(0)=1} \|p(A)\|_2 \leq \min_{p \in \mathcal{P}_{m_k}, p(0)=1} \max_{\lambda \in [\mu, L]} |p(\lambda)| \leq 2\left(\frac{\sqrt{L/\mu} - 1}{\sqrt{L/\mu} + 1}\right)^{m_k}. \tag{115}$$

Like (63) in the proof of Corollary 1, we also have $\|p(A)\|_A = \|p(A)\|_2$. Then $\|p(A)(x_{k-m_k} - x^*)\|_A \leq \|p(A)\|_A \|x_{k-m_k} - x^*\|_A = \|p(A)\|_2 \|x_{k-m_k} - x^*\|_A$. With (115) and (112), the bound (113) holds. □

Remark 8. *In Theorem 6, we do not assume each* β_k *is chosen such that* $\theta_k = \|I - \beta_k A\|_2 < 1$*. It is expected that when* θ_k *is not too large, and* m_k *is sufficiently large, the minimization problem on the right-hand side of* (112) *can dominate the convergence rate, thus leading to fast convergence. However, in the case that an improper choice of* β_k *causes* $\theta_k \gg 1$*, the restarted Min-AM may have an erratic convergence behaviour or even diverge if* m_k *is too small.*

Remark 9. *From* (112)*, a large* m_k *can make the first-order term diminish significantly. However, the high-order terms of errors* $\hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2^2)$ *can be large since they are accumulated from the last restart. In practice, we can first choose large* m*, small* τ*, and large* η *in* (11a)-(11c) *so that the restarts do not occur too frequently. When the restarted Min-AM has a problematic convergence behaviour due to the high nonlinearity of* ∇f*, more frequent restarts are necessary.*

Now, we give the proof of Theorem 2, which can be obtained from Theorem 6.

Proof of Theorem 2. Under the assumptions of Theorem 2, we prove that

$$\|x_j - x^*\|_A \leq \|x_0 - x^*\|_A, \;\; j = 0, \ldots, k, \tag{116}$$

and there exists a constant $\eta_0 > 0$ such that

$$\|\nabla f(x_j)\|_2 \leq \eta_0 \|\nabla f(x_0)\|_2, \;\; j = 0, \ldots, k. \tag{117}$$

We prove (116) and (117) by induction. For $k = 0$, (116) and (117) hold. Suppose that $k \geq 0$, and the results hold for k. We establish the results for $k + 1$. From (114) in the proof of Theorem 6, it follows that

$$\|\hat{x}_{k+1} - x^*\|_A \leq \theta_k \|x_{k-m_k} - x^*\|_A \leq \theta \|x_{k-m_k} - x^*\|_A. \tag{118}$$

Since $\theta_k \leq \theta$, we know $\beta_k \in [\frac{1-\theta}{\mu}, \frac{1+\theta}{L}]$. With the inductive hypothesis (117), for sufficiently small $\|x_0 - x^*\|_2$, the relations (67) and (68) hold. From (68), it follows that $\|x_{k+1} - \hat{x}_{k+1}\|_A = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_A^2)$. Then, there exists a constant c_1 such that $\|x_{k+1} - \hat{x}_{k+1}\|_A \leq \hat{\kappa} c_1 \|x_{k-m_k} - x^*\|_A^2$. With (118), we have

$$\|x_{k+1} - x^*\|_A \leq \theta \|x_{k-m_k} - x^*\|_A + \hat{\kappa} c_1 \|x_{k-m_k} - x^*\|_A^2.$$

Hence, by choosing $\|x_0 - x^*\|_2 \le \frac{1-\theta}{2\sqrt{L}\hat{\kappa}c_1}$, which ensures $\|x_{k-m_k} - x^*\|_A \le \|x_0 - x^*\|_A \le \sqrt{L}\|x_0 - x^*\|_2 \le \frac{1-\theta}{2\hat{\kappa}c_1}$ due to the inductive hypothesis (116), it follows that

$$\|x_{k+1} - x^*\|_A \le \frac{1+\theta}{2}\|x_{k-m_k} - x^*\|_A < \|x_{k-m_k} - x^*\|_A \le \|x_0 - x^*\|_A,$$

namely (116) for $k+1$. Also, $\|x_{k+1}-x^*\|_2 \le \frac{1}{\sqrt{\mu}}\|x_{k+1}-x^*\|_A \le \frac{1}{\sqrt{\mu}}\|x_0-x^*\|_A \le \frac{\sqrt{L}}{\sqrt{\mu}}\|x_0-x^*\|_2$. So we can impose $\|x_0-x^*\|_2 \le \frac{\sqrt{\mu}\hat{\rho}}{\sqrt{L}}$ to ensure $x_{k+1} \in \mathcal{B}_{\hat{\rho}}(x^*)$, which further yields that $\|r_{k+1}\|_2 \le L\|x_{k+1} - x^*\|_2 \le \frac{L\sqrt{L}}{\sqrt{\mu}}\|x_0 - x^*\|_2 \le \frac{L\sqrt{L}}{\mu\sqrt{\mu}}\|r_0\|_2$, namely (117) for $k+1$, and $\eta_0 = \frac{L\sqrt{L}}{\mu\sqrt{\mu}}$. Hence, we complete the induction. The convergence result (13) follows from (113) in Theorem 6.

If $m_k = d$, then $\hat{x}_{k+1} = x^*$ from Corollary 1. Therefore, $\|x_{k+1}-x^*\|_2 = \hat{\kappa}\mathcal{O}(\|x_{k-m_k}-x^*\|_2^2)$. □

D Details of the eigenvalue estimation procedure

D.1 Analysis of the quadratic case

The procedure of the basic Min-AM leads to three-term recurrences for P_k and Q_k.

Proposition 3. *Under the same assumptions of Theorem 1 for solving strongly convex quadratic optimization* (47)*, we have*

$$Ap_k = t_k^{(k-1)}p_{k-1} + t_k^{(k)}p_k + t_k^{(k+1)}p_{k+1}, \tag{119}$$

$$Aq_k = t_k^{(k-1)}q_{k-1} + t_k^{(k)}q_k + t_k^{(k+1)}q_{k+1}, \tag{120}$$

where the coefficients are given by

$$t_k^{(k-1)} = \frac{\phi_{k-1}}{\beta_{k-1}(1-\Gamma_k^{(1)})}, \quad t_k^{(k)} = \frac{1}{1-\Gamma_k^{(1)}}\left(\frac{1}{\beta_{k-1}} - \frac{\phi_k}{\beta_k}\right), \quad t_k^{(k+1)} = -\frac{1}{\beta_k(1-\Gamma_k^{(1)})}, \tag{121}$$

and $\phi_k := \Gamma_k^{(1)} + \Gamma_k^{(2)} + \zeta_{k+1}$. *Then* $\phi_k = \Gamma_k^{(2)}$ *and there exists a tridiagonal matrix* $\bar{T}_k \in \mathbb{R}^{(k+1)\times k}$, *such that*

$$AP_k = P_{k+1}\bar{T}_k, \quad AQ_k = Q_{k+1}\bar{T}_k. \tag{122}$$

Proof. The relations can be verified by direct computation. In the $(k+1)$-th iteration, since

$$x_{k+1} = x_k^{(2)} - p_k\Gamma_k^{(2)} = x_k^{(1)} + \beta_k r_k^{(1)} - p_k\Gamma_k^{(2)} = x_k + \beta_k r_k^{(1)} - p_k(\Gamma_k^{(1)} + \Gamma_k^{(2)}), \tag{123}$$

we have

$$\begin{aligned}
p_{k+1} &= \Delta x_k - p_k\zeta_{k+1} = \beta_k r_k^{(1)} - p_k(\Gamma_k^{(1)} + \Gamma_k^{(2)} + \zeta_{k+1}) \\
&= \beta_k(r_k - q_k\Gamma_k^{(1)}) - p_k\phi_k = \beta_k(r_{k-1}^{(2)} - q_{k-1}\Gamma_{k-1}^{(2)} - q_k\Gamma_k^{(1)}) - p_k\phi_k \\
&= \beta_k(I - \beta_{k-1}A)r_{k-1}^{(1)} - \beta_k q_{k-1}\Gamma_{k-1}^{(2)} - \beta_k q_k\Gamma_k^{(1)} - p_k\phi_k \\
&= \beta_k(I - \beta_{k-1}A)\frac{p_k + p_{k-1}\phi_{k-1}}{\beta_{k-1}} - \beta_k q_{k-1}\Gamma_{k-1}^{(2)} - \beta_k q_k\Gamma_k^{(1)} - p_k\phi_k \\
&= \frac{\beta_k}{\beta_{k-1}}(p_k + p_{k-1}\phi_{k-1}) - \beta_k A(p_k + p_{k-1}\phi_{k-1}) + \beta_k Ap_{k-1}\Gamma_{k-1}^{(2)} + \beta_k Ap_k\Gamma_k^{(1)} - p_k\phi_k \\
&= \left(\frac{\beta_k}{\beta_{k-1}} - \phi_k\right)p_k + \left(\frac{\beta_k}{\beta_{k-1}}\phi_{k-1}\right)p_{k-1} + (\beta_k\Gamma_k^{(1)} - \beta_k)Ap_k \\
&\quad + (\beta_k\Gamma_{k-1}^{(2)} - \beta_k\phi_{k-1})Ap_{k-1} \\
&= \left(\frac{\beta_k}{\beta_{k-1}} - \phi_k\right)p_k + \left(\frac{\beta_k}{\beta_{k-1}}\phi_{k-1}\right)p_{k-1} + (\beta_k\Gamma_k^{(1)} - \beta_k)Ap_k,
\end{aligned} \tag{124}$$

where the last term vanishes due to

$$\begin{aligned}
\Gamma_{k-1}^{(2)} - \phi_{k-1} &= \Gamma_{k-1}^{(2)} - \Gamma_{k-1}^{(1)} - \Gamma_{k-1}^{(2)} - \zeta_k \\
&= -\frac{p_{k-1}^{\mathrm{T}}r_{k-1}}{p_{k-1}^{\mathrm{T}}q_{k-1}} - \frac{p_{k-1}^{\mathrm{T}}\Delta r_{k-1}}{p_{k-1}^{\mathrm{T}}q_{k-1}} = -\frac{p_{k-1}^{\mathrm{T}}r_k}{p_{k-1}^{\mathrm{T}}q_{k-1}} = 0,
\end{aligned} \tag{125}$$

since $r_k \perp p_{k-1}$. It also indicates that

$$\phi_k = \Gamma_k^{(1)} + \Gamma_k^{(2)} + \zeta_{k+1} = \Gamma_k^{(2)} + \frac{p_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} q_k} + \frac{p_k^{\mathrm{T}} \Delta r_k}{p_k^{\mathrm{T}} q_k} = \Gamma_k^{(2)} + \frac{p_k^{\mathrm{T}} r_{k+1}}{p_k^{\mathrm{T}} q_k} = \Gamma_k^{(2)}. \tag{126}$$

Now, it follows from Equation (124) that

$$Ap_k = \frac{1}{1-\Gamma_k^{(1)}} \frac{\phi_{k-1}}{\beta_{k-1}} p_{k-1} + \frac{1}{1-\Gamma_k^{(1)}} \left(\frac{1}{\beta_{k-1}} - \frac{\phi_k}{\beta_k} \right) p_k + \frac{-1}{\beta_k (1-\Gamma_k^{(1)})} p_{k+1}. \tag{127}$$

The second relation about Aq_k can be obtained by noticing that $q_k = -Ap_k$. The tridiagonal matrix $\bar{T}_k$ is

$$\bar{T}_k = \begin{pmatrix} t_1^{(1)} & t_2^{(1)} & & & & \\ t_1^{(2)} & t_2^{(2)} & t_3^{(2)} & & & \\ & t_2^{(3)} & t_3^{(3)} & t_4^{(3)} & & \\ & & \cdots & & & \\ & & & t_{k-2}^{(k-1)} & t_{k-1}^{(k-1)} & t_k^{(k-1)} \\ & & & & t_{k-1}^{(k)} & t_k^{(k)} \\ & & & & & t_k^{(k+1)} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}. \tag{128}$$

□

Algorithm 3 k-step A-norm based Lanczos Algorithm for $Ax = \lambda x$, where A is SPD.

Input: Starting vector $v \in \mathbb{R}^d$
Output: Approximate eigenvalues $\{\theta_i\}_{i=1}^k$

```
r = v, v_0 = 0
β_0 = ||r||_A
for j = 1, 2, ..., until convergence do
    v_j = r / β_{j-1}
    r = A v_j
    r = r − v_{j-1} β_{j-1}
    α_j = v_j^T A r
    r = r − v_j α_j
    β_j = ||r||_A
    if j = k then
        break
    end if
end for
Construct tridiagonal T_k ∈ R^{k×k}: The main diagonal is {α_i}_{i=1}^k; the subdiagonals are {β_i}_{i=1}^{k-1}
Compute the eigenvalue decomposition T_k = S_k Θ^{(k)} S_k^T
return Θ^{(k)}
```

D.2 Proof of Theorem 3

For solving general nonlinear optimization, the iterations between two successive restarts in the restarted Min-AM are the basic Min-AM iterations. Hence, at the $(k+1)$-th iteration, a tridiagonal matrix $T_k \in \mathbb{R}^{m_k \times m_k}$ can be constructed based on the coefficients starting from the $(k-m_k)$-th iteration. We formulate it in the following definition.

Definition 2. *For solving optimization problem* (1) *with the restarted Min-AM (see Algorithm 1), suppose that* $m_k \geq 1$*, and* $t_k^{(k-1)}, t_k^{(k)}, t_k^{(k+1)}$ *are obtained by* (121)*. Then* $T_k \in \mathbb{R}^{m_k \times m_k}$ *is defined as* $T_k = t_k$ *if* $m_k = 1$*, and* $T_k = (\bar{T}_{k-1}, t_k)$ *if* $m_k \geq 2$*. Here,* $\bar{T}_k = (T_k^{\mathrm{T}}, t_k^{(k+1)} e_{m_k})^{\mathrm{T}} \in \mathbb{R}^{(m_k+1)\times m_k}$*, where* e_{m_k} *is the last column of* I_{m_k}*;* $t_k = t_k^{(k)}$ *if* $m_k = 1$*, and* $t_k = (0, \ldots, 0, t_k^{(k-1)}, t_k^{(k)})^{\mathrm{T}} \in \mathbb{R}^{m_k}$ *if* $m_k \geq 2$*.*

Now, we prove Theorem 3.

Proof of Theorem 3. Without loss of generality, we consider the case that no restart happens during the iterations, i.e. $m_k = k$. Let $A := \nabla^2 f(x^*)$.

The A-norm based Lanczos algorithm with a starting vector v for computing eigenvalues of an SPD matrix A is described in Algorithm 3, which is a modification of the Algorithm 4.6 in [6] with A-norm. It forms an A-orthonormal basis V_k of the Krylov subspace $\mathcal{K}_k(A, \hat{r}_0)$ by the Lanczos A-orthogonalization, i.e. $V_k^{\mathrm{T}} A V_k = I$, where $\hat{r}_0 = A(x^* - x_0)$. Then the A-norm based Lanczos algorithm seeks to find $\tilde{\lambda} \in \mathbb{R}$ and $z \in \mathbb{R}^k$ such that

$$(A - \tilde{\lambda} I) V_k z \perp A V_k.$$

Then we have

$$V_k^{\mathrm{T}} A A V_k z = (A V_k)^{\mathrm{T}} A V_k z = \tilde{\lambda} (A V_k)^{\mathrm{T}} V_k z = \tilde{\lambda} z. \tag{129}$$

The eigenvalues of $(AV_k)^{\mathrm{T}} A V_k$ are the approximations to the true eigenvalues of A.

For Min-AM, we still use an auxiliary solving procedure of Min-AM on the local quadratic approximation (65), i.e. Process II in Definition 1. The same as $P_k, T_k, \bar{T}_k, t_k^{(k-1)}, t_k^{(k)}, t_k^{(k+1)}$ in Process I, the notations $\hat{P}_k, \hat{T}_k, \bar{\hat{T}}_k, \hat{t}_k^{(k-1)}, \hat{t}_k^{(k)}, \hat{t}_k^{(k+1)}$ are defined for Process II, correspondingly. Then the Min-AM in Process II seeks $\hat{\lambda} \in \mathbb{R}$ and $y \in \mathbb{R}^k$, such that

$$(A - \hat{\lambda} I) \hat{P}_k y \perp A \hat{P}_k,$$

namely,

$$(A \hat{P}_k)^{\mathrm{T}} A \hat{P}_k y = \hat{\lambda} (A \hat{P}_k)^{\mathrm{T}} \hat{P}_k y.$$

According to Proposition 3, we have $A \hat{P}_k = \hat{P}_{k+1} \bar{\hat{T}}_k$. Then

$$(A \hat{P}_k)^{\mathrm{T}} A \hat{P}_k = \hat{P}_k^{\mathrm{T}} A \hat{P}_{k+1} \bar{\hat{T}}_k = \hat{P}_k^{\mathrm{T}} A \hat{P}_k \hat{T}_k.$$

Then the eigenvalues of

$$\hat{T}_k = (\hat{P}_k^{\mathrm{T}} A \hat{P}_k)^{-1} (A \hat{P}_k)^{\mathrm{T}} A \hat{P}_k$$

are the approximations of the true eigenvalues of A. $\hat{T}_k$ is tridiagonal but generally not symmetric. However, it is clear that $\hat{T}_k$ is similar to

$$(\hat{P}_k^{\mathrm{T}} A \hat{P}_k)^{1/2} \hat{T}_k (\hat{P}_k^{\mathrm{T}} A \hat{P}_k)^{-1/2} = (\hat{P}_k^{\mathrm{T}} A \hat{P}_k)^{-1/2} \hat{P}_k^{\mathrm{T}} A A \hat{P}_k (\hat{P}_k^{\mathrm{T}} A \hat{P}_k)^{-1/2}, \tag{130}$$

which is symmetric. The columns of $U_k := \hat{P}_k (\hat{P}_k^{\mathrm{T}} A \hat{P}_k)^{-1/2}$ are A-orthonormal and also span the same Krylov subspace $\mathcal{K}_k(A, \hat{r}_0)$, which implies that there exists an orthonormal matrix $S \in \mathbb{R}^{k \times k}$, such that $V_k = U_k S$. Therefore,

$$V_k^{\mathrm{T}} A A V_k = S^{\mathrm{T}} U_k^{\mathrm{T}} A A U_k S. \tag{131}$$

Since $S^{\mathrm{T}} S = I$, the eigenvalues of (129) and (130) are identical.

Recall that we have the explicit forms of $T_k, \hat{T}_k$ by (121), and we restate it here for convenience.

$$t_k^{(k-1)} = \frac{\phi_{k-1}}{\beta_{k-1}(1 - \Gamma_k^{(1)})}, \quad t_k^{(k)} = \frac{1}{1 - \Gamma_k^{(1)}} \left(\frac{1}{\beta_{k-1}} - \frac{\phi_k}{\beta_k} \right), \quad t_k^{(k+1)} = -\frac{1}{\beta_k (1 - \Gamma_k^{(1)})}, \tag{132a}$$

$$\hat{t}_k^{(k-1)} = \frac{\hat{\phi}_{k-1}}{\beta_{k-1}(1 - \hat{\Gamma}_k^{(1)})}, \quad \hat{t}_k^{(k)} = \frac{1}{1 - \hat{\Gamma}_k^{(1)}} \left(\frac{1}{\beta_{k-1}} - \frac{\hat{\phi}_k}{\beta_k} \right), \quad \hat{t}_k^{(k+1)} = -\frac{1}{\beta_k (1 - \hat{\Gamma}_k^{(1)})}, \tag{132b}$$

and $\phi_k := \Gamma_k^{(1)} + \Gamma_k^{(2)} + \zeta_{k+1}$, $\phi_k := \hat{\Gamma}_k^{(1)} + \hat{\Gamma}_k^{(2)} + \hat{\zeta}_{k+1}$. With the already proved relations (69)-(76), we can compute the bounds of $t_k^{(j)} - \hat{t}_k^{(j)}$, $j = k-1, k, k+1$. Let $\epsilon := \hat{\kappa} \mathcal{O}(\|x_0 - x^*\|_2)$ to simplify the notation. Then

$$\frac{1}{1 - \Gamma_k^{(1)}} - \frac{1}{1 - \hat{\Gamma}_k^{(1)}} = \frac{1}{1 - \Gamma_k^{(1)}} - \frac{1}{1 - \Gamma_k^{(1)} + \epsilon} = \frac{\epsilon}{(1 - \Gamma_k^{(1)})(1 - \Gamma_k^{(1)} + \epsilon)} = \hat{\kappa} \mathcal{O}(\|x_0 - x^*\|_2), \tag{133}$$

where we use the fact that $|\Gamma_k^{(1)} - \hat{\Gamma}_k^{(1)}| = \hat{\kappa}\mathcal{O}(\|x_0 - x^*\|_2)$, and the assumption that $|1 - \Gamma_k^{(1)}| \geq \tau_0$. We also have $|\phi_k| = \mathcal{O}(1)$ and $|\hat{\phi}_k - \phi_k| = \hat{\kappa}\mathcal{O}(\|x_{k-m_k} - x^*\|_2)$. Hence,

$$|t_k^{(k-1)} - \hat{t}_k^{(k-1)}| \leq \left|\frac{\phi_{k-1} - \hat{\phi}_{k-1}}{\beta_{k-1}(1 - \Gamma_k^{(1)})}\right| + \left|\frac{\hat{\phi}_{k-1}}{\beta_{k-1}}\left(\frac{1}{1 - \Gamma_k^{(1)}} - \frac{1}{1 - \hat{\Gamma}_k^{(1)}}\right)\right| = \hat{\kappa}\mathcal{O}(\|x_0 - x^*\|_2), \tag{134}$$

$$\begin{aligned} |t_k^{(k)} - \hat{t}_k^{(k)}| &\leq \left|\frac{1}{\beta_{k-1}}\left(\frac{1}{1 - \Gamma_k^{(1)}} - \frac{1}{1 - \hat{\Gamma}_k^{(1)}}\right)\right| + \left|\frac{\phi_k - \hat{\phi}_k}{\beta_k(1 - \Gamma_k^{(1)})}\right| \\ &\quad + \left|\frac{\hat{\phi}_k}{\beta_k}\left(\frac{1}{1 - \Gamma_k^{(1)}} - \frac{1}{1 - \hat{\Gamma}_k^{(1)}}\right)\right| \\ &= \hat{\kappa}\mathcal{O}(\|x_0 - x^*\|_2), \end{aligned} \tag{135}$$

$$|t_k^{(k+1)} - \hat{t}_k^{(k+1)}| = \hat{\kappa}\mathcal{O}(\|x_0 - x^*\|_2). \tag{136}$$

With $m_k \leq m$, we have that

$$\|T_k - \hat{T}_k\|_2 = \hat{\kappa}\mathcal{O}(\|x_0 - x^*\|_2). \tag{137}$$

Define $\hat{D}_k = \hat{P}_k^{\mathrm{T}} A \hat{P}_k$. It is clear that $\hat{D}_k$ is a diagonal matrix, and the i-th diagonal element is $-\hat{p}_i^{\mathrm{T}}\hat{q}_i$. From (100), (71), and (72), it follows that

$$\|\hat{D}_k\|_2 \|\hat{D}_k^{-1}\|_2 = \frac{\max_i\{|\hat{p}_i^{\mathrm{T}}\hat{q}_i|\}}{\min_j\{|\hat{p}_j^{\mathrm{T}}\hat{q}_j|\}} = \mathcal{O}(1). \tag{138}$$

By (130), we know that $\hat{T}_k$ can be diagonalized. There exists orthonormal matrix $W_k \in \mathbb{R}^{k\times k}$ and diagonal matrix $\wedge_k \in \mathbb{R}^{k\times k}$, such that

$$\hat{D}_k^{1/2}\hat{T}_k\hat{D}_k^{-1/2} = W_k \wedge_k W_k^{\mathrm{T}}. \tag{139}$$

Therefore, $\hat{T}_k = \hat{D}_k^{-1/2} W_k \wedge_k W_k^{\mathrm{T}} \hat{D}_k^{1/2}$ can be diagonalized. Then with (137), (138), and applying the Bauer-Fike Theorem [33], we know that for an eigenvalue λ of T_k, the following result holds:

$$\begin{aligned} \min_{\hat{\lambda}\in\sigma(\hat{T}_k)} |\hat{\lambda} - \lambda| &\leq \|\hat{D}_k^{-1/2} W_k\|_2 \|W_k^{\mathrm{T}} \hat{D}_k^{1/2}\|_2 \|T_k - \hat{T}_k\|_2 \\ &\leq \|\hat{D}_k^{-1/2}\|_2 \|\hat{D}_k^{1/2}\|_2 \|T_k - \hat{T}_k\|_2 = \hat{\kappa}\mathcal{O}(\|x_0 - x^*\|_2), \end{aligned} \tag{140}$$

where $\sigma(\hat{T}_k)$ denotes the spectrum of $\hat{T}_k$. □

E Details of the stochastic Min-AM

Now, we prove the theorems about the stochastic Min-AM (sMin-AM) in Section 3.4. The algorithm is given in Algorithm 4. For brevity, we use δ_k to denote $\delta_k^{(2)}$, i.e. $\delta_k \equiv \delta_k^{(2)}$. Our proofs follow those of SAM [62] and ST-AM [63].

From Assumption 3, for the mini-batch gradient $f_{S_k}(x_k) = \frac{1}{n_k}\sum_{i\in S_k} f_i(x_k)$, where $n_k = |S_k|$, we have

$$\mathbb{E}[\nabla f_{S_k}(x)|x_k] = \nabla f(x_k), \tag{141a}$$

$$\mathbb{E}[\|\nabla f_{S_k}(x_k) - \nabla f(x_k)\|_2^2 | x_k] \leq \frac{\sigma^2}{n_k}. \tag{141b}$$

Note that the update of sMin-AM (Lines 10-12 in Algorithm 4) can be written as $x_{k+1} = x_k + H_k r_k$, where $r_k = -\nabla f_{S_k}(x_k)$, and for $k \geq 0$,

$$H_k = (1 - \alpha_k)\beta_k I + \alpha_k H_k^A = \beta_k I + \alpha_k(H_k^A - \beta_k I). \tag{142}$$

To prove the theorems, we need H_k to be positive definite, and show that the noise in gradient estimates is suppressed during iterations.

We first state some useful results about sMin-AM, then we prove Theorem 4 and Theorem 5.

Algorithm 4 Stochastic Min-AM

Input: $x_0 \in \mathbb{R}^d, \beta_k > 0, \alpha_k \in [0,1], \delta_k^{(1)} \geq 0, \delta_k^{(2)} \geq 0$

$p_0, q_0 = \mathbf{0} \in \mathbb{R}^d$
for $k = 0, 1, \ldots,$ until convergence, **do**
  $r_k = -\nabla f_{S_k}(x_k)$
  if $k > 0$ **then**
    $p = x_k - x_{k-1},\ q = r_k - r_{k-1}$
    $\zeta_k = \Phi(p_{k-1}, q_{k-1}, \delta_k^{(1)})^\dagger p_{k-1}^{\mathrm{T}} q$
    $q_k = q - q_{k-1}\zeta_k,\ p_k = p - p_{k-1}\zeta_k$
  end if
  $\rho_k = \Phi(p_k, q_k, \delta_k^{(2)})^\dagger$
  $x_k^A = x_k + \big(-\rho_k p_k p_k^{\mathrm{T}} + \beta_k (I - \rho_k p_k q_k^{\mathrm{T}})(I - \rho_k q_k p_k^{\mathrm{T}})\big) r_k$
  $x_k^G = x_k + \beta_k r_k$
  $x_{k+1} = (1-\alpha_k) x_k^G + \alpha_k x_k^A$
  Apply learning rate schedule of α_k, β_k
end for
return x_k

E.1 Some useful results

Lemma 2. *Suppose that the sequence $\{x_k\}$ is generated by sMin-AM. If there are constants $\mu \in (0,1)$, $C_2 > 0$ such that $\alpha_k \in [0, 1-\mu]$, $\rho_k \leq 0$ and $-\rho_k \|p_k\|_2^2 - 2\beta_k \rho_k \|p_k\|_2 \|q_k\|_2 + \beta_k \rho_k^2 \|p_k\|_2^2 \|q_k\|_2^2 \leq \beta_k C_2$, then we have that*

$$\|H_k^A - \beta_k I\|_2 \leq C_2 \beta_k, \tag{143}$$

$$\|H_k\|_2 \leq \beta_k (1 + C_2), \tag{144}$$

$$H_k \succeq \beta_k \mu I. \tag{145}$$

Proof. Since

$$\begin{aligned} H_k^A &= -\rho_k p_k p_k^{\mathrm{T}} + \beta_k (1 - \rho_k p_k q_k^{\mathrm{T}})(1 - \rho_k q_k p_k^{\mathrm{T}}) \\ &= -\rho_k p_k p_k^{\mathrm{T}} + \beta_k I - \beta_k \rho_k q_k p_k^{\mathrm{T}} - \beta_k \rho_k p_k q_k^{\mathrm{T}} + \beta_k \rho_k^2 p_k q_k^{\mathrm{T}} q_k p_k^{\mathrm{T}}, \end{aligned}$$

we have

$$H_k^A - \beta_k I = -\rho_k p_k p_k^{\mathrm{T}} - \beta_k \rho_k q_k p_k^{\mathrm{T}} - \beta_k \rho_k p_k q_k^{\mathrm{T}} + \beta_k \rho_k^2 p_k q_k^{\mathrm{T}} q_k p_k^{\mathrm{T}}. \tag{146}$$

Hence,

$$\begin{aligned} \|H_k^A - \beta_k I\|_2 &\leq |\rho_k| \|p_k p_k^{\mathrm{T}}\|_2 + \beta_k |\rho_k| \|q_k p_k^{\mathrm{T}}\|_2 + \beta_k |\rho_k| \|p_k q_k^{\mathrm{T}}\|_2 + \beta_k \rho_k^2 \|p_k q_k^{\mathrm{T}} q_k p_k^{\mathrm{T}}\|_2 \\ &\leq |\rho_k| \|p_k\|_2^2 + \beta_k |\rho_k| \|q_k\|_2 \|p_k\|_2 + \beta_k |\rho_k| \|p_k\|_2 \|q_k\|_2 + \beta_k \rho_k^2 \|q_k\|_2^2 \|p_k\|_2^2 \\ &\leq \beta_k C_2. \end{aligned} \tag{147}$$

It follows that

$$\|H_k\|_2 \leq \|\beta_k I\|_2 + \alpha_k \|H_k^A - \beta_k I\|_2 \leq \beta_k + C_2 \beta_k = (1 + C_2)\beta_k. \tag{148}$$

Because $\rho_k \leq 0$, it follows that $H_k^A \succeq 0$. Hence $H_k \succeq (1-\alpha_k)\beta_k I \succeq \mu \beta_k I$. □

Remark 10. *The conditions that $\rho_k \leq 0$ and $-\rho_k \|p_k\|_2^2 - 2\beta_k \rho_k \|p_k\|_2 \|q_k\|_2 + \beta_k \rho_k^2 \|p_k\|_2^2 \|q_k\|_2^2 \leq \beta_k C_2$ for a positive constant C_2 can be fulfilled with some proper choice of δ_k. For example, if C_2 satisfies $(1 + \frac{1}{2}\beta_k)^2 \leq C_2$ and we choose $\delta_k \geq \frac{1}{2} + \beta_k^{-1}$, then it can be proved that the conditions hold. Moreover, in practice, δ_k can be chosen as a constant, as shown in our experiments.*

With Lemma 2, we can prove the convergence of sMin-AM for full-batch training.

Theorem 7. *Suppose that Assumption 2 holds and $\{x_k\}$ is the sequence generated by full-batch sMin-AM, i.e. $n_k = T$. Given constants $\mu \in (0,1)$, $C_2 > 0$, let $\beta_k = \beta \in (0, \frac{\mu}{L(1+C_2)^2}]$ be a*

constant, $\alpha_k \in [0, 1-\mu]$. The $\delta_k^{(2)}$ is chosen such that $\rho_k \leq 0$ and $-\rho_k\|p_k\|_2^2 - 2\beta_k\rho_k\|p_k\|_2\|q_k\|_2 + \beta_k\rho_k^2\|p_k\|_2^2\|q_k\|_2^2 \leq \beta_k C_2$. Then

$$\frac{1}{N}\sum_{k=0}^{N-1}\|\nabla f(x_k)\|_2^2 \leq \frac{2(f(x_0) - f^{low})}{N\mu\beta}, \tag{149}$$

in the N-th iterations. To ensure $\frac{1}{N}\sum_{k=0}^{N-1}\|\nabla f(x_k)\|_2^2 < \epsilon$, the number of iterations is $\mathcal{O}(1/\epsilon)$.

Proof of Theorem 7. With Lemma 2, we have

$$\|H_k r_k\|_2^2 \leq \beta^2(1+C_2)^2\|r_k\|_2^2,$$

and

$$r_k^{\mathrm{T}} H_k r_k \geq \beta\mu\|r_k\|_2^2.$$

Then, under Assumption 2, we have

$$\begin{aligned}
f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)^{\mathrm{T}}(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|_2^2 \\
&= f(x_k) - r_k^{\mathrm{T}} H_k r_k + \frac{L}{2}\|H_k r_k\|_2^2 \\
&\leq f(x_k) - \beta\mu\|r_k\|_2^2 + \frac{L}{2}\beta^2(1+C_2)^2\|r_k\|_2^2 \\
&= f(x_k) - \beta\left(\mu - \frac{L}{2}\beta(1+C_2)^2\right)\|r_k\|_2^2 \\
&\leq f(x_k) - \frac{1}{2}\beta\mu\cdot\|\nabla f(x_k)\|_2^2,
\end{aligned} \tag{150}$$

where the last inequality is due to $0 < \beta \leq \frac{\mu}{L(1+C_2)^2}$. Thus, $f(x_{k+1}) - f(x_k) \leq -\frac{1}{2}\beta\mu\|\nabla f(x_k)\|_2^2$.

Summing both sides of this inequality for $k \in \{0, \ldots, N-1\}$ and recalling $f(x) > f^{low}$ in Assumption 2 gives

$$f^{low} - f(x_0) \leq f(x_N) - f(x_0) \leq -\frac{1}{2}\beta\mu\sum_{k=0}^{N-1}\|\nabla f(x_k)\|_2^2.$$

Rearranging and dividing further by N yields (149). □

The next lemmas are about the stochastic case.

Lemma 3. *Suppose that Assumption 3 and Assumption 4 hold and $\{x_k\}$ is the sequence generated by sMin-AM with $\beta_k > 0$. Then*

$$\mathbb{E}_{S_k}[\|H_k r_k\|_2^2] \leq \beta_k^2(1+C_2)^2\cdot\left(\|\nabla f(x_k)\|_2^2 + \frac{\sigma^2}{n_k}\right), \tag{151a}$$

$$\nabla f(x_k)^{\mathrm{T}}\mathbb{E}_{S_k}[H_k r_k] \leq -\frac{1}{2}\beta_k\mu\|\nabla f(x_k)\|_2^2 + \frac{1}{2}\beta_k^2\cdot\mu^{-1}C_1^2C_2^2\frac{\sigma^2}{n_k}. \tag{151b}$$

Proof. (i) From Lemma 2, we have

$$\mathbb{E}_{S_k}[\|H_k r_k\|_2^2] \leq \beta_k^2(1+C_2)^2\mathbb{E}_{S_k}[\|r_k\|_2^2]. \tag{152}$$

From Assumption 3, we have

$$\mathbb{E}_{S_k}[\|r_k\|_2^2] = \mathbb{E}_{S_k}[\|r_k - \mathbb{E}_{S_k}[r_k]\|_2^2] + \|\mathbb{E}_{S_k}[r_k]\|_2^2 \leq \|\nabla f(x_k)\|_2^2 + \sigma^2/n_k. \tag{153}$$

With (152) and (153), we obtain (151a).

(ii) Recalling that $H_0 = \beta_0 I$, the result holds for $k = 0$. Define $\epsilon_k = \nabla f_{S_k}(x_k) - \nabla f(x_k) = -r_k - \nabla f(x_k)$, then $H_k r_k = H_k\left(-\epsilon_k - \nabla f(x_k)\right)$. First, we have

$$\nabla f(x_k)^{\mathrm{T}} H_k \nabla f(x_k) \geq \beta_k\mu\|\nabla f(x_k)\|_2^2,$$

which implies

$$\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \nabla f(x_k)] \geq \beta_k \mu \|\nabla f(x_k)\|_2^2. \tag{154}$$

Let $M_k := \alpha_k(H_k^A - \beta I)$, then $H_k = \beta_k I + M_k$. With the assumption (141a), i.e. $\mathbb{E}_{S_k}[\epsilon_k] = 0$, we have

$$\begin{aligned}\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \epsilon_k] &= \mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} (\beta_k \epsilon_k + M_k \epsilon_k)] \\ &= \beta_k \nabla f(x_k)^{\mathrm{T}} \mathbb{E}_{S_k}[\epsilon_k] + \mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} M_k \epsilon_k] = \mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} M_k \epsilon_k].\end{aligned}$$

Using the *Cauchy-Schwarz inequality with expectations*, we obtain

$$\begin{aligned}|\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \epsilon_k]| = |\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} M_k \epsilon_k]| &\leq \sqrt{\mathbb{E}_{S_k}[\|\nabla f(x_k)\|_2^2]}\sqrt{\mathbb{E}_{S_k}[\|M_k \epsilon_k\|_2^2]} \\ &= \|\nabla f(x_k)\|_2 \sqrt{\mathbb{E}_{S_k}[\|M_k \epsilon_k\|_2^2]}.\end{aligned} \tag{155}$$

We now bound $\|M_k \epsilon_k\|_2^2$.

$$\|M_k\|_2 = \alpha_k \|H_k^A - \beta_k I\|_2 \leq C_2 \alpha_k \beta_k. \tag{156}$$

With (156), we have $\|M_k \epsilon_k\|_2 \leq C_2 \alpha_k \beta_k \|\epsilon_k\|_2$, which implies

$$\mathbb{E}_{S_k}[\|M_k \epsilon_k\|_2^2] \leq C_2^2 \alpha_k^2 \beta_k^2 \mathbb{E}_{S_k}[\|\epsilon_k\|_2^2] \leq C_2^2 \alpha_k^2 \beta_k^2 \frac{\sigma^2}{n_k}, \tag{157}$$

where the last inequality is due to (141b). Now we can obtain the bound of $|\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \epsilon_k]|$ as follows (cf. (155)):

$$\begin{aligned}|\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \epsilon_k]| &\leq \|\nabla f(x_k)\|_2 \sqrt{\mathbb{E}_{S_k}[\|M_k \epsilon_k\|_2^2]} \\ &\leq C_2 \alpha_k \beta_k \|\nabla f(x_k)\|_2 \sqrt{\mathbb{E}_{S_k}[\|\epsilon_k\|_2^2]} \\ &\leq C_2 \alpha_k \beta_k \frac{\sigma}{\sqrt{n_k}} \|\nabla f(x_k)\|_2 \\ &= \sqrt{\beta_k \mu} \|\nabla f(x_k)\|_2 \cdot \frac{C_2 \alpha_k \beta_k}{\sqrt{\beta_k \mu}} \frac{\sigma}{\sqrt{n_k}} \\ &\leq \frac{1}{2} \beta_k \mu \|\nabla f(x_k)\|_2^2 + \frac{1}{2} \frac{C_2^2 \alpha_k^2 \beta_k^2}{\beta_k \mu} \cdot \frac{\sigma^2}{n_k}.\end{aligned} \tag{158}$$

With the inequality (154) and (158), we obtain

$$\begin{aligned}&\nabla f(x_k)^{\mathrm{T}} \mathbb{E}_{S_k}[H_k r_k] \\ &= -\nabla f(x_k)^{\mathrm{T}} \mathbb{E}_{S_k}[H_k (\epsilon_k + \nabla f(x_k))] \\ &= -\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \nabla f(x_k)] - \mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \epsilon_k] \\ &\leq -\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \nabla f(x_k)] + |\mathbb{E}_{S_k}[\nabla f(x_k)^{\mathrm{T}} H_k \epsilon_k]| \\ &\leq -\beta_k \mu \|\nabla f(x_k)\|_2^2 + \frac{1}{2} \beta_k \mu \|\nabla f(x_k)\|_2^2 + \frac{1}{2} \frac{C_2^2 \alpha_k^2 \beta_k^2}{\beta_k \mu} \cdot \frac{\sigma^2}{n_k} \\ &= -\frac{1}{2} \beta_k \mu \|\nabla f(x_k)\|_2^2 + \frac{1}{2} \frac{C_2^2 \alpha_k^2 \beta_k^2}{\beta_k \mu} \cdot \frac{\sigma^2}{n_k} \leq -\frac{1}{2} \beta_k \mu \|\nabla f(x_k)\|_2^2 + \frac{1}{2} \beta_k^2 \mu^{-1} C_1^2 C_2^2 \cdot \frac{\sigma^2}{n_k},\end{aligned} \tag{159}$$

where the last inequality is due to $\alpha_k \leq C_1 \beta_k^{1/2}$. □

Using Lemma 3 we obtain the descent property of sMin-AM:

Lemma 4. *Suppose that Assumptions 2-4 hold, $\beta_k \in (0, \frac{\mu}{2L(1+C_2)^2}]$, and $\{x_k\}$ is the sequence generated by sMin-AM. Then*

$$\mathbb{E}_{S_k}[f(x_{k+1})] \leq f(x_k) - \frac{1}{4} \beta_k \mu \|\nabla f(x_k)\|_2^2 + \frac{\beta_k^2}{2} (\mu^{-1} C_1^2 C_2^2 + L(1 + C_2)^2) \frac{\sigma^2}{n_k}. \tag{160}$$

Proof. Due to Assumption 2, we have

$$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^{\mathrm{T}}(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|_2^2$$
$$= f(x_k) + \nabla f(x_k)^{\mathrm{T}} H_k r_k + \frac{L}{2}\|H_k r_k\|_2^2. \tag{161}$$

Taking expectation with respect to the mini-batch S_k on both sides of (161) and using Lemma 3, we obtain

$$\begin{aligned}
&\mathbb{E}_{S_k}[f(x_{k+1})] \\
&\leq f(x_k) + \nabla f(x_k)^{\mathrm{T}} \mathbb{E}_{S_k}[H_k r_k] + \frac{L}{2}\mathbb{E}_{S_k}\|H_k r_k\|_2^2 \\
&\leq f(x_k) - \frac{1}{2}\beta_k \mu \|\nabla f(x_k)\|_2^2 + \frac{1}{2}\beta_k^2 \cdot \mu^{-1} C_1^2 C_2^2 \frac{\sigma^2}{n_k} + \frac{L}{2}\beta_k^2 (1 + C_2)^2 \left(\|\nabla f(x_k)\|_2^2 + \frac{\sigma^2}{n_k} \right) \\
&= f(x_k) - \beta_k \left(\frac{1}{2}\mu - \frac{L}{2}\beta_k (1 + C_2)^2 \right) \|\nabla f(x_k)\|_2^2 + \frac{\beta_k^2}{2}(\mu^{-1} C_1^2 C_2^2 + L(1 + C_2)^2) \frac{\sigma^2}{n_k}.
\end{aligned} \tag{162}$$

Then (162) combined with the assumption $\beta_k \leq \frac{\mu}{2L(1+C_2)^2}$ implies (160). □

E.2 Proof of Theorem 4

Following the proofs in [61, 62, 63], we introduce the definition of a *supermartingale*.

Definition 3. *Let $\{\mathcal{F}_k\}$ be an increasing sequence of σ-algebras. If $\{X_k\}$ is a stochastic process satisfying (i) $\mathbb{E}[|X_k|] < \infty$, (ii) $X_k \in \mathcal{F}_k$ for all k, and (iii) $\mathbb{E}[X_{k+1}|\mathcal{F}_k] \leq X_k$ for all k, then $\{X_k\}$ is called a supermartingale.*

Proposition 4 (Supermartingale convergence theorem, see, e.g., Theorem 4.2.12 in [20])**.** *If $\{X_k\}$ is a nonnegative supermartingale, then $\lim_{k\to\infty} X_k \to X$ almost surely and $\mathbb{E}[X] \leq \mathbb{E}[X_0]$.*

Now, we prove Theorem 4 of sMin-AM.

Proof of Theorem 4. (i) Define $\phi_k = \frac{\beta_k \mu}{4}\|\nabla f(x_k)\|_2^2$ and $\tilde{L} = \frac{1}{2}\left(\mu^{-1} C_1^2 C_2^2 + L(1 + C_2)^2\right)$, $\gamma_k = f(x_k) + \tilde{L}\frac{\sigma^2}{n}\sum_{i=k}^{\infty}\beta_i^2$. Let $\mathcal{F}_k$ be the σ-algebra measuring ϕ_k, γ_k, and x_k. From (160) we know that for any k,

$$\begin{aligned}
\mathbb{E}[\gamma_{k+1}|\mathcal{F}_k] &= \mathbb{E}[f(x_{k+1})|\mathcal{F}_k] + \tilde{L}\frac{\sigma^2}{n}\sum_{i=k+1}^{\infty}\beta_i^2 \\
&\leq f(x_k) - \frac{1}{4}\beta_k \mu \|\nabla f(x_k)\|_2^2 + \tilde{L}\frac{\sigma^2}{n}\sum_{i=k}^{\infty}\beta_i^2 = \gamma_k - \phi_k,
\end{aligned} \tag{163}$$

which implies that $\mathbb{E}[\gamma_{k+1} - f^{low}|\mathcal{F}_k] \leq \gamma_k - f^{low} - \phi_k$. Since $\phi_k \geq 0$, we have $0 \leq \mathbb{E}[\gamma_k - f^{low}] \leq \gamma_0 - f^{low} < +\infty$. As the diminishing condition (25) holds, we obtain $\mathbb{E}[f(x_k)] \leq M_f$ for some constant $M_f > 0$. According to Definition 3, $\{\gamma_k - f^{low}\}$ is a supermartingale. Therefore, Proposition 4 indicates that there exists a constant γ such that $\lim_{k\to\infty}\gamma_k = \gamma$ with probability 1, and $\mathbb{E}[\gamma] \leq \mathbb{E}[\gamma_0]$. Note that from (163) we have $\mathbb{E}[\phi_k] \leq \mathbb{E}[\gamma_k] - \mathbb{E}[\gamma_{k+1}]$. Thus,

$$\mathbb{E}\left[\sum_{k=0}^{\infty}\phi_k\right] \leq \sum_{k=0}^{\infty}(\mathbb{E}[\gamma_k] - \mathbb{E}[\gamma_{k+1}]) < +\infty,$$

which further yields that

$$\sum_{k=0}^{\infty}\phi_k = \frac{\mu}{4}\sum_{k=0}^{\infty}\beta_k\|\nabla f(x_k)\|_2^2 < +\infty \text{ with probability 1.} \tag{164}$$

Since $\sum_{k=0}^{\infty}\beta_k = +\infty$, it follows that (26) holds.

(ii) If the noisy gradient is bounded, i.e.,

$$\mathbb{E}_{\xi_k}[\|\nabla f_{\xi_k}(x_k)\|_2^2] \leq M_g, \tag{165}$$

where $M_g > 0$ is a constant, then a stronger result can be obtained.

For any given $\epsilon > 0$, according to (26), there exist infinitely many iterates x_k such that $\|\nabla f(x_k)\|_2 \leq \epsilon$. Then if (27) does not hold, there must exist two infinite sequences of indices $\{s_i\}$, $\{t_i\}$ with $t_i > s_i$, such that for $i = 0, 1, \ldots$, $k = s_i + 1, \ldots, t_i - 1$,

$$\|\nabla f(x_{s_i})\|_2 \geq 2\epsilon, \|\nabla f(x_{t_i})\|_2 < \epsilon, \|\nabla f(x_k)\|_2 \geq \epsilon. \tag{166}$$

Then from (164) it follows that

$$+\infty > \sum_{k=0}^{\infty} \beta_k \|\nabla f(x_k)\|_2^2 \geq \sum_{i=0}^{+\infty} \sum_{k=s_i}^{t_i-1} \beta_k \|\nabla f(x_k)\|_2^2 \geq \epsilon^2 \sum_{i=0}^{+\infty} \sum_{k=s_i}^{t_i-1} \beta_k \text{ with probability } 1,$$

which implies that

$$\sum_{k=s_i}^{t_i-1} \beta_k \to 0 \text{ with probability } 1, \text{as } i \to +\infty. \tag{167}$$

According to (144), we have

$$\begin{aligned} &\mathbb{E}[\|x_{k+1} - x_k\|_2 | x_k] \\ &= \mathbb{E}[\|H_k r_k\|_2 | x_k] \leq \beta_k (1 + C_2) \mathbb{E}[\|r_k\|_2 | x_k] \leq \beta_k (1 + C_2) (\mathbb{E}[\|r_k\|_2^2 | x_k])^{\frac{1}{2}} \leq \beta_k (1 + C_2) M_g^{\frac{1}{2}}, \end{aligned} \tag{168}$$

where the last inequalities are due to *Cauchy-Schwarz inequality* and (165). Then it follows from (168) that

$$\mathbb{E}[\|x_{t_i} - x_{s_i}\|_2] \leq (1 + C_2) M_g^{\frac{1}{2}} \sum_{k=s_i}^{t_i-1} \beta_k,$$

which together with (167) implies that $\|x_{t_i} - x_{s_i}\|_2 \to 0$ with probability 1, as $i \to +\infty$. Hence, from the Lipschitz continuity of ∇f, it follows that $\|\nabla f(x_{t_i}) - \nabla f(x_{s_i})\|_2 \to 0$ with probability 1 as $i \to +\infty$. However, this contradicts (166). Therefore, the assumption that (27) does not hold is not true. □

E.3 Proof of Theorem 5

Proof of Theorem 5. According to (162) in Lemma 4, we have

$$\begin{aligned} &\sum_{k=0}^{N-1} \beta_k \left(\frac{1}{2}\mu - \frac{L}{2}\beta_k (1 + C_2)^2 \right) \mathbb{E}\|\nabla f(x_k)\|_2^2 \\ &\leq f(x_0) - f^{low} + \sum_{k=0}^{N-1} \frac{\beta_k^2}{2} (\mu^{-1} C_1^2 C_2^2 + L(1 + C_2)^2) \frac{\sigma^2}{n_k}, \end{aligned} \tag{169}$$

where the expectation is taken with respect to $\{S_j\}_{j=0}^{N-1}$. Define

$$P_R(k) = \text{Prob}\{R = k\} = \frac{\beta_k \left(\frac{1}{2}\mu - \frac{L}{2}\beta_k (1 + C_2)^2 \right)}{\sum_{j=0}^{N-1} \beta_j \left(\frac{1}{2}\mu - \frac{L}{2}\beta_j (1 + C_2)^2 \right)}, \ k = 0, \ldots, N-1, \tag{170}$$

then

$$\begin{aligned} \mathbb{E}\left[\|\nabla f(x_R)\|_2^2\right] &= \frac{\sum_{k=0}^{N-1} \beta_k \left(\frac{1}{2}\mu - \frac{L}{2}\beta_k (1 + C_2)^2 \right) \mathbb{E}\left[\|\nabla f(x_k)\|_2^2\right]}{\sum_{j=0}^{N-1} \beta_j \left(\frac{1}{2}\mu - \frac{L}{2}\beta_j (1 + C_2)^2 \right)} \\ &\leq \frac{D_f + \frac{\sigma^2}{2} (\mu^{-1} C_1^2 C_2^2 + L(1 + C_2)^2) \sum_{k=0}^{N-1} \beta_k^2 / n_k}{\sum_{j=0}^{N-1} \beta_j \left(\frac{1}{2}\mu - \frac{L}{2}\beta_j (1 + C_2)^2 \right)}. \end{aligned} \tag{171}$$

Let $\tilde{D}$ be a problem-independent constant. If we choose $\beta_k = \beta := \min\{\frac{\mu}{2L(1+C_2)^2}, \frac{\tilde{D}}{\sigma\sqrt{N}}\}$, and $n_k \equiv n$, then the definition of P_R simplifies to $P_R(k) = 1/N$. From (171) we have

$$\begin{aligned}
\mathbb{E}[\|\nabla f(x_R)\|_2^2] &\leq \frac{D_f + \frac{\sigma^2}{2}(\mu^{-1}C_1^2C_2^2 + L(1+C_2)^2)\frac{N\beta^2}{n}}{\sum_{j=0}^{N-1}\beta(\frac{1}{2}\mu - \frac{\mu}{4})} \\
&= \frac{D_f + \frac{\sigma^2}{2}(\mu^{-1}C_1^2C_2^2 + L(1+C_2)^2)\frac{N\beta^2}{n}}{N\beta\cdot\frac{1}{4}\mu} \\
&= \frac{4D_f}{N\beta\mu} + \frac{\frac{\sigma^2}{2}(\mu^{-1}C_1^2C_2^2 + L(1+C_2)^2)\cdot\beta}{\frac{1}{4}n\mu} \\
&\leq \frac{4D_f}{N\mu}\max\left\{\frac{2L(1+C_2)^2}{\mu}, \frac{\sigma\sqrt{N}}{\tilde{D}}\right\} + \frac{2\sigma^2(\mu^{-1}C_1^2C_2^2 + L(1+C_2)^2)}{n\mu}\cdot\frac{\tilde{D}}{\sigma\sqrt{N}} \\
&\leq \frac{4D_f}{N\mu}\left(\frac{2L(1+C_2)^2}{\mu} + \frac{\sigma\sqrt{N}}{\tilde{D}}\right) + \frac{2\sigma(\mu^{-1}C_1^2C_2^2 + L(1+C_2)^2)\tilde{D}}{n\mu\sqrt{N}} \\
&= \frac{8D_fL(1+C_2)^2}{N\mu^2} + \frac{\sigma}{\mu\sqrt{N}}\left(\frac{4D_f}{\tilde{D}} + \frac{2(\mu^{-1}C_1^2C_2^2 + L(1+C_2)^2)\tilde{D}}{n}\right).
\end{aligned}$$

Therefore, to ensure $\mathbb{E}[\|\nabla f(x_R)\|_2^2] \leq \epsilon$, the number of iterations is $\mathcal{O}(1/\epsilon^2)$. □

F Experimental details

We implemented the algorithms based on PyTorch [2] and used one GeForce RTX 2080 Ti GPU for training neural networks (except that four GPUs were used for training ResNet50 on ImageNet). We tested the basic Min-AM on solving strongly convex quadratic optimization, the restarted Min-AM on solving regularized logistic regression, and the stochastic Min-AM on training neural networks.

F.1 Strongly convex quadratic problem

The experiments on solving strongly convex quadratic optimization were conducted to verify Theorem 1 about the basic Min-AM. The problem is

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{2}\|Ax - b\|_2^2, \tag{172}$$

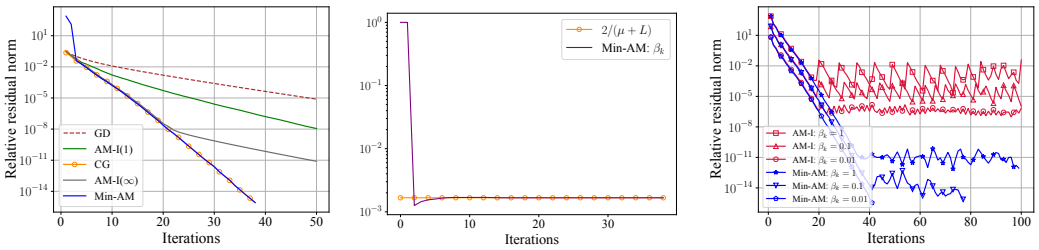
where $A \in \mathbb{R}^{\ell\times d}$, $b \in \mathbb{R}^\ell$. For the test, we first generated a random matrix $A \in \mathbb{R}^{500\times 100}$ and a random vector $v \in \mathbb{R}^{100}$ following Gaussian distribution, then $b \in \mathbb{R}^{500}$ was obtained as $b = Av$.

The fixed step size β for the gradient descent (GD) was chosen by a grid search in $\{0.001, 0.002, \ldots, 0.01\}$. We set $\beta = 0.001$ that guarantees the convergence of GD. The AM-I(1) and the full-memory AM-I (i.e. AM-I(∞)) also used the same mixing parameter setting $\beta_k = 0.001$. For Min-AM, we set the initial mixing parameter $\beta_0 = 1$, and the later mixing parameters $\{\beta_k\}$ were adaptively determined based on the eigenvalue estimates (see Section 3.3).

Figure 4(a) compares the convergence behaviours of different methods in terms of relative residual norm. It can be observed that due to the improper initial setting of β_0 for Min-AM, Min-AM does not perform well in the beginning. Nonetheless, as shown in Figure 4(b), the β_k of Min-AM can be quickly adapted to the optimal value $2/(\mu + L) = 1.67 \times 10^{-3}$ based on the eigenvalue estimates.

In Figure 4(c), we show the effects of β_k on the full-memory AM-I and Min-AM, where both methods used fixed β_k chosen from $\{0.01, 0.1, 1\}$. (For Min-AM, we disable the adaptive choice of β_k.) It should be noted that the GD method diverges when choosing the step size in $\{0.01, 0.1, 1\}$, which suggests that $\theta_k > 1$ for the Min-AM in these settings. Nevertheless, we find that Min-AM still converges to the tolerance of $\|r_k\|_2/\|r_0\|_2 \leq 10^{-11}$, which validates the convergence property (59) in Corollary 1, i.e., the minimization problem in (59) dominates the convergence when k is

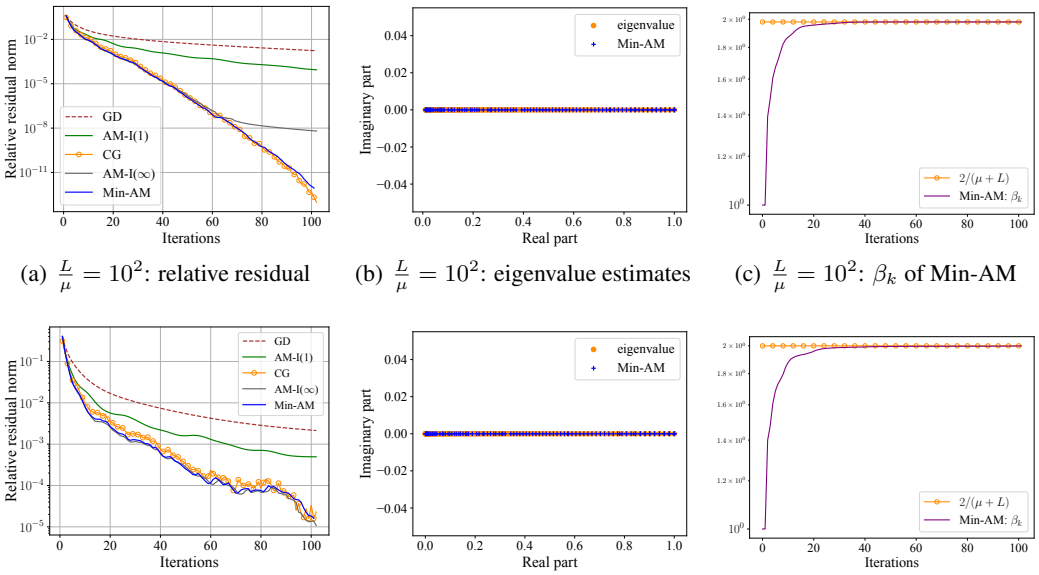
[2] https://pytorch.org.



(a) Relative residual (b) β_k of Min-AM (c) AM-I/Min-AM with different β_k

Figure 4: (a) $\|r_k\|_2/\|r_0\|_2$ of each method; (b) the mixing parameter β_k of Min-AM and the optimal choice $2/(\mu + L)$; (c) $\|r_k\|_2/\|r_0\|_2$ of AM-I and Min-AM with different β_k.

large. Note that AM-I fails to coincide with Min-AM in the later iterations. It is due to the fact that AM-I needs to solve $(X_k^{\mathrm{T}} R_k)^{-1} X_k^{\mathrm{T}} r_k$ to determine Γ_k, where the matrix inverse operation can have numerical weakness when k is large.



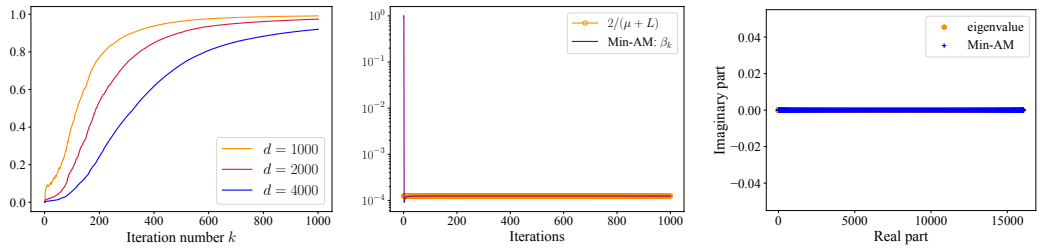
(a) $\frac{L}{\mu} = 10^2$: relative residual (b) $\frac{L}{\mu} = 10^2$: eigenvalue estimates (c) $\frac{L}{\mu} = 10^2$: β_k of Min-AM

(d) $\frac{L}{\mu} = 10^8$: relative residual (e) $\frac{L}{\mu} = 10^8$: eigenvalue estimates (f) $\frac{L}{\mu} = 10^8$: β_k of Min-AM

Figure 5: (a)(b)(c): relative residual norm, eigenvalue estimates, and β_k of Min-AM, when $L/\mu = 10^2$. (d)(e)(f): relative residual norm, eigenvalue estimates, and β_k of Min-AM, when $L/\mu = 10^8$. In (b) and (c), "eigenvalue" denotes the exact eigenvalues, and "Min-AM" denotes the eigenvalue estimates computed by Min-AM.

Results about the problem with different condition numbers. In Figure 5, we show the convergence of each method, the eigenvalue estimates from Min-AM, and the β_k of Min-AM, in the tests with different condition numbers characterized by L/μ. The eigenvalues of $A \in \mathbb{R}^{100\times 100}$ were chosen to be in $(0, 1]$ with equal interval. The results show that Min-AM is competitive with CG and the full-memory AM-I. Min-AM also gives accurate enough estimates of the largest and the smallest eigenvalues, and the β_k is quickly adapted to approximate the optimal value $2/(\mu + L)$.

Cost of the eigenvalue estimation procedure. In our implementation of the eigenvalue estimation procedure in the Min-AM, we used the function "np.linalg.eigvals" in NumPy to compute the eigenvalues of T_k constructed by (18), which needs $\mathcal{O}(k^3)$ flops. Hence, the cost of the eigenvalue estimation increases with increasing k, and at the k-th iteration, the ratio of this cost to the total cost is $\mathcal{O}(k^3)/(\mathcal{O}(k^3) + \mathcal{O}(d^2))$, where $\mathcal{O}(d^2)$ is due to the gradient evaluation. To investigate the



(a) Cost of eigenvalue estimations (b) β_k of Min-AM ($d = 4000$) (c) Eigenvalue estimates ($d = 4000$)

Figure 6: (a) The ratio of the time of eigenvalue estimations to the total running time during iterations; (b) the mixing parameter β_k of Min-AM and the optimal choice $2/(\mu + L)$ when the problem dimension $d = 4000$; (c) exact eigenvalues of the Hessian and eigenvalue estimates computed by Min-AM when the problem dimension $d = 4000$.

practical performance of the eigenvalue estimation, we applied the basic Min-AM to solve problem (172) of dimension $d = 1000, 2000, 4000$, where each $A \in \mathbb{R}^{d\times d}$ was generated following Gaussian distribution, and the maximal iteration number is $max_iter = 1001$ for Min-AM. As can be found in Figure 6(a), the additional computational time incurred by the eigenvalue estimations is marginal when d is large and k is small. Figure 6(b) and Figure 6(c) show the case that $d = 4000$. (The cases $d = 1000$ and $d = 2000$ have similar results.) We find that β_k is quickly adapted to the optimal value $2/(\mu + L)$ within very small number of iterations. So we can adaptively choose β_k in the beginning, and fix the obtained β_k and disable the eigenvalue estimation procedure in the later iterations to reduce the computational cost. Figure 6(c) shows that the eigenvalue estimates computed by Min-AM (at the last iteration) well approximate the exact eigenvalues.

F.2 Regularized logistic regression problem

The regularized logistic regression problem is formulated as

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{T}\sum_{i=1}^{T} \log\left(1 + \exp(-y_i x^{\mathrm{T}} \xi_i)\right) + \frac{w}{2}\|x\|_2^2, \tag{173}$$

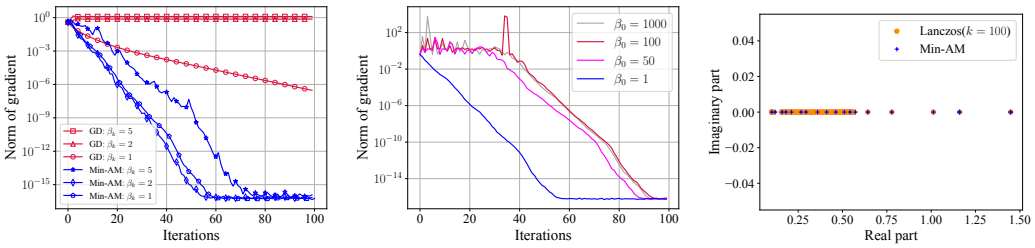
where $\xi_i \in \mathbb{R}^d$ is the i-th data sample and $y_i = \pm 1$ is the corresponding label. The regularization parameter $w = 0.1$. We used the datasets "madelon" and "a9a" from LIBSVM [14] that are two-class classifications:

- madelon: training data size: 2000; feature size: 500;
- a9a: training data size: 32561; feature size: 123.

To set proper μ and L for Nesterov's accelerated gradient (NAG) method and check the eigenvalue estimates from Min-AM, we applied the standard Lanczos algorithm [24] to compute 100 Ritz values of $\nabla^2 f(x^*)$ as the estimates of the true eigenvalues of $\nabla^2 f(x^*)$, where the minimizer x^* was obtained by solving (173). For the test on madelon, Lanczos algorithm gave $\mu = 1.01 \times 10^{-1}$, $L = 1.45$; for the test on a9a, Lanczos algorithm gave $\mu = 1.00 \times 10^{-1}$, $L = 8.64 \times 10^{-1}$.

For the gradient descent (GD), the step size was tuned and set as 1, which was proper for both datasets. We used the Polak-Ribière variant of nonlinear conjugate gradient (NCG), and the step size was determined by line search with cubic interpolation, where the strong Wolfe conditions were checked. The L-BFGS used the Barzilai–Borwein step size as the initial guess of the approximate inverse Hessian. For AM and ST-AM, the mixing parameter was set as 1. The AM-I performed similarly to AM in this test, so we only report the results of AM. For the restarted Min-AM, we set $\tau = 10^{-16}$ for the madelon test and $\tau = 10^{-32}$ for the a9a test. $m = 100$ for both datasets. η was set as a large number since Min-AM converged for these tests. Besides, we computed at most 20 eigenvalue estimates in Min-AM (between two successive restarts) to determine β_k.

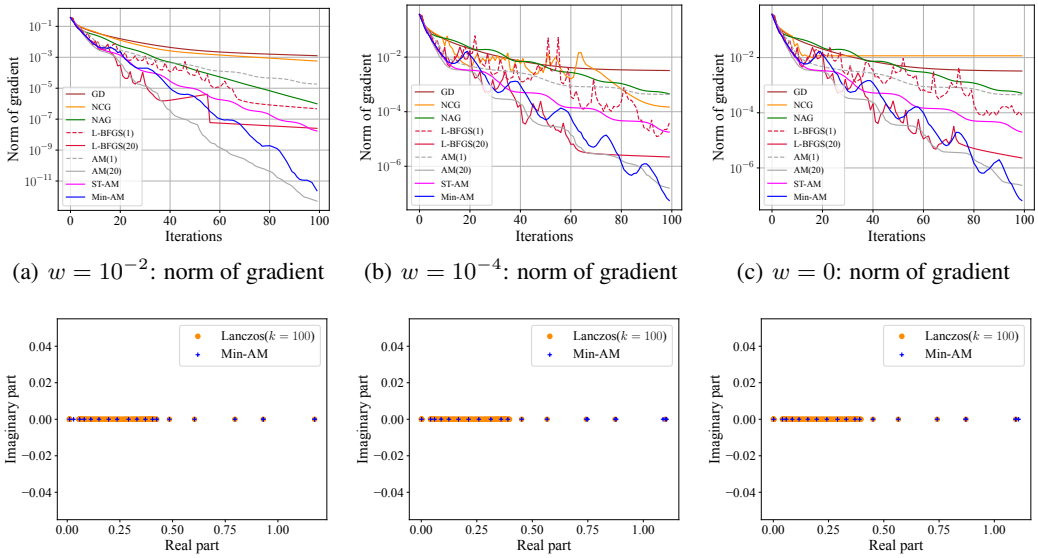
Results about the choice of β_k. In Figure 7, we show additional results of the tests on madelon dataset. (1) To test the effect of β_k on Min-AM, we disabled the adaptive choice of β_k and used



(a) Fixed β_k (b) Adaptive β_k (c) Eigenvalue estimates ($\beta_0 = 100$)

Figure 7: Regularized logistic regression with $w = 0.1$ on madelon dataset. (a) Gradient norms of GD (with constant step size β_k) and Min-AM (with fixed mixing parameter β_k). (b) Min-AM with adaptive choice of β_k; gradient norms of Min-AM with different initializations of β_k are shown. (c) Min-AM with adaptive choice of β_k; the Ritz values of $\nabla^2 f(x^*)$ from k-step Lanczos algorithm, and eigenvalue estimates from Min-AM with $\beta_0 = 100$ are shown.

fixed β_k chosen from $\{1, 2, 5\}$. It is observed in Figure 7(a) that GD does not converge if the step size β_k is chosen from $\{2, 5\}$, which suggests that $\theta_k > 1$ in Theorem 2 and Theorem 6. However, Min-AM still converges in these cases, which validates our discussion in Remark 8 in Appendix C.1: the minimization problem on the right-hand side of (112) in Theorem 6 dominates the convergence when m_k is large. It also indicates that Min-AM is less sensitive to β_k than GD. (2) When adaptive choice of β_k is used, we investigate the effect of the initialization β_0 on the convergence and the quality of eigenvalue estimates. It is found in Figure 7(b) that even with the improper initialization $\beta_0 = 50, 100, 1000$, the iterations still finally converge to an acceptable precision. In fact, GD, AM, and Min-AM diverge if using fixed β_k chosen from $\{50, 100, 1000\}$. Hence, Min-AM with adaptive choice of β_k is easy to apply since it discards the requirement of manually tuning the mixing parameter. In Figure 7(c), we find that the eigenvalue estimates can still roughly approximate the largest and the smallest Ritz values computed by the Lanczos algorithm, which verifies Theorem 3.



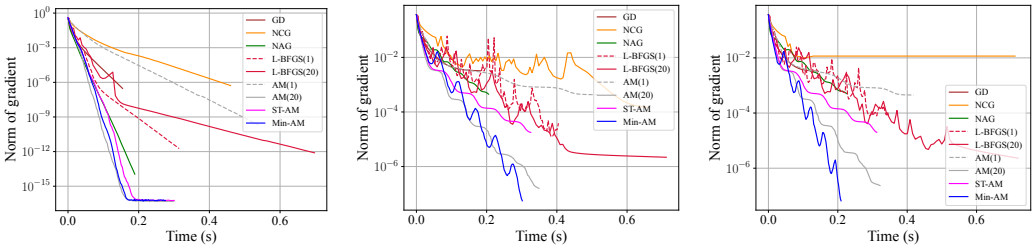
(a) $w = 10^{-2}$: norm of gradient (b) $w = 10^{-4}$: norm of gradient (c) $w = 0$: norm of gradient

(d) $w = 10^{-2}$: eigenvalue estimates (e) $w = 10^{-4}$: eigenvalue estimates (f) $w = 0$: eigenvalue estimates

Figure 8: The norm of gradient and eigenvalue estimates for different w. Left column: $w = 10^{-2}$; middle column: $w = 10^{-4}$; right column: $w = 0$. "Lanczos($k = 100$)" denotes the Ritz values computed by Lanczos algorithm, and "Min-AM" denotes the eigenvalue estimates from Min-AM.

Results of the problem with different regularization parameters. We also tested Min-AM for the problem (173) with different settings of the regularization parameter w. The results in Figure 8 also

show that Min-AM significantly improves the convergence of AM(1) and is comparable to AM(20) when w is small. The comparison between the Ritz values computed by Lanczos algorithm and the eigenvalue estimates computed by Min-AM validates Theorem 3. Since the Ritz values approximate the true eigenvalues of $\nabla^2 f(x^*)$, it is expected that Min-AM can give promising eigenvalue estimates of $\nabla^2 f(x^*)$, which accounts for the efficiency of Min-AM.



(a) $w = 10^{-1}$: norm of gradient (b) $w = 10^{-4}$: norm of gradient (c) $w = 0$: norm of gradient

Figure 9: The norm of gradient with respect to running time.

Running time. In Figure 9, we report the convergence of each method with respect to the running time. Min-AM has comparable performance to AM(20) in these tests, while using a smaller memory size.

F.3 Deep neural network training problem

The experiments about the stochastic Min-AM (sMin-AM) focused on training deep neural networks. Since Min-AM can be viewed as a special case of sMin-AM with $\delta_k^{(1)} = \delta_k^{(2)} = 0, \alpha_k = 1$, the basic Min-AM was also covered.

F.3.1 Hyperparameter setting of the stochastic Min-AM

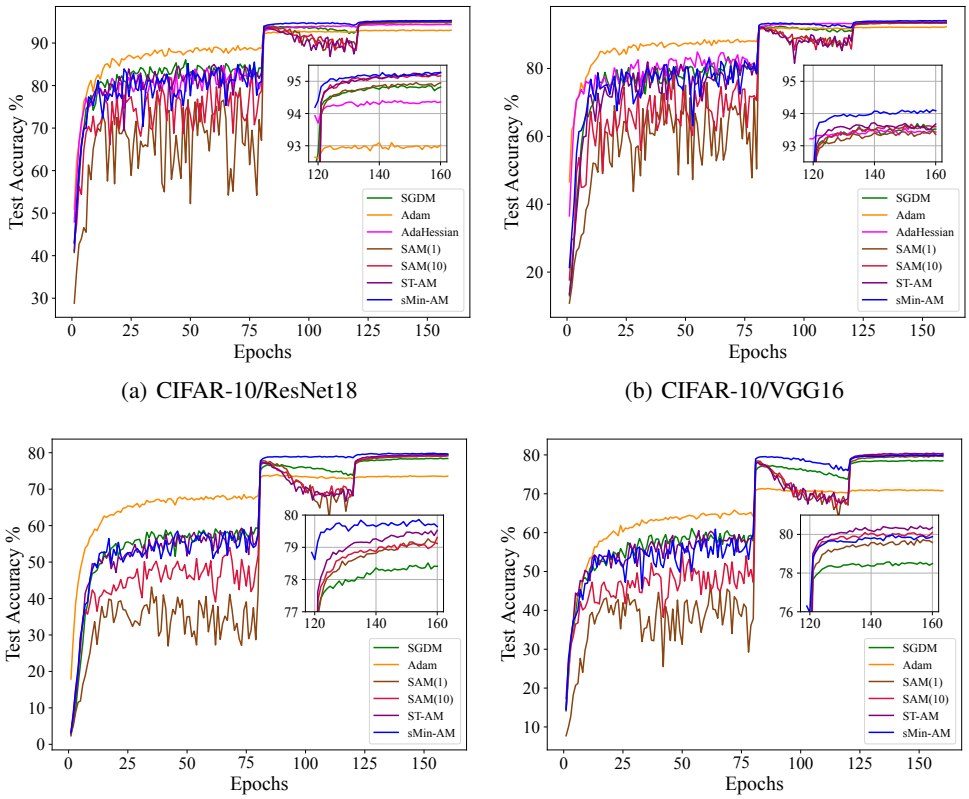
The main hyperparameters of sMin-AM are $\alpha_k, \beta_k, \delta_k^{(1)}, \delta_k^{(2)}$. We set $\delta_k^{(1)} = \delta_k^{(2)} = 1$, which ensures that H_k^A is positive definite. We fixed $\alpha_k = 1$ and only tuned $\beta_k \in (0, 1]$ in the experiments.

F.3.2 Experiments on CIFAR-10 and CIFAR-100

The experiments on CIFAR-10 and CIFAR-100 followed the same settings of SAM [62] and ST-AM [63] for direct comparisons. Both CIFAR-10 and CIFAR-100 contain a training dataset (50K images) and a test dataset (10K images), where CIFAR-10 has 10 classes and CIFAR-100 has 100 classes for classification. The basic setting of network training followed the standard setting of training ResNet [30]: The batch size was 128 as commonly suggested; for training with N iterations, the learning rate of the optimizer was decayed at the $(\lfloor \frac{N}{2} \rfloor)$-th iteration and the $(\lfloor \frac{3}{4}N \rfloor)$-th iteration. The experiments were run with 3 random seeds and the averaged results along with the standard deviations were reported. The final accuracy on the test dataset was used as the evaluation metric. The accuracy reported in Table 1(a) is the final test accuracy of training with 160 epochs.

We compared sMin-AM with SGDM, Adam, AdaHessian, stochastic AM (SAM), and short-term recurrence AM (ST-AM). SGDM is the default optimizer for training many deep neural networks, such as ResNet [30], WideResNet [66], DenseNet [34], and ResNeXt [64]. Adam is an adaptive learning rate method that uses diagonal approximation to the Hessian based on moving average. AdaHessian also uses adaptive learning rates like Adam, but it relies on Hessian-vector products to obtain the diagonal approximation. SAM and ST-AM are variants of AM. Both can be used to train neural networks. The SAM with memory size m is denoted as SAM(m). ST-AM keeps two vector pairs and has the same memory size as SAM(2).

We tuned the hyperparameters of all the optimizers (including sMin-AM) following the same way for fair comparison. The tuning procedures were conducted on CIFAR-10/ResNet20. For each optimizer, the hyperparameter setting that achieved the highest final test accuracy for training ResNet20 on CIFAR-10 was unchanged in the other tests of training neural networks on CIFAR.



(a) CIFAR-10/ResNet18

(b) CIFAR-10/VGG16

(c) CIFAR-100/ResNeXt50

(d) CIFAR-100/DenseNet121

Figure 10: Test accuracy during the network training on CIFAR-10 and CIFAR-100.

For SGDM, the momentum was set as 0.9, which is the default setting in the literature [30, 34]. The initial learning rate and weight decay were tuned and were set as 0.1, 0.0005, respectively. The learning rate decay was 0.1. This setting is also recommended in WideResNet [66].

For Adam, the initial learning rate and weight decay were tuned and set as 0.001, 0.0005, respectively. The momentum terms were $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as commonly suggested [65, 68]. The learning rate decay was 0.1.

For AdaHessian, the initial learning rate was 0.15, and the momentum terms $\beta_1 = 0.9, \beta_2 = 0.999$; eps = 1×10^{-4}, and the hessian-power was 1. The weight decay was $0.0005/0.15$, and the learning rate decay was 0.1.

For SAM, the initial mixing parameter $\beta_0 = 1$, the initial damping term $\alpha_0 = 1$, and the regularization parameter $c_1 = 0.01$. The weight decay was 0.0015. The decay rate for β_k, α_k was 0.06.

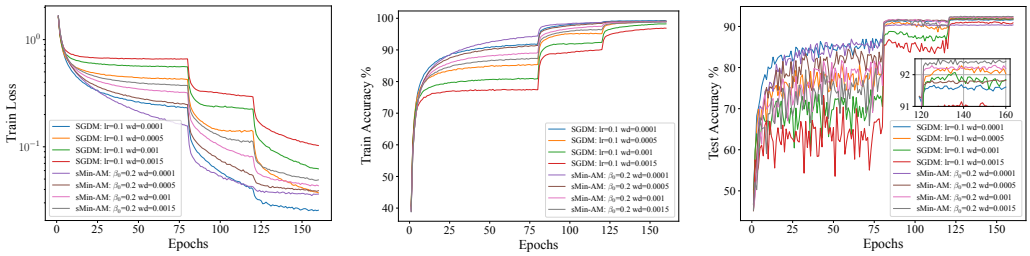
For ST-AM, the initial mixing parameter $\beta_0 = 1$, the initial damping term $\alpha_0 = 1$, and the regularization parameters were $c_1 = 1, c_2 = 1 \times 10^{-7}$. The weight decay was 0.001, and the decay rate for β_k, α_k was 0.1.

For sMin-AM, the initial mixing parameter $\beta_0 = 0.2$, the damping term $\alpha_k = 1$, and the regularization parameters were $\delta_k^{(1)} = 1, \delta_k^{(2)} = 1$. The weight decay was 0.0015, and the decay rate for β_k was 0.1.

Note that our hyperparameter settings of the baseline methods were the same as those in SAM [62] and ST-AM [63], so we used their results for reference.

Convergence behaviour in the training process. In Figure 10, we plot the test accuracy of training four networks on CIFAR-10 and CIFAR-100, for each optimizer. It shows Adam has fast convergence in the beginning, but stagnates in the later training. Compared with SGDM, SAM, and ST-AM, it is

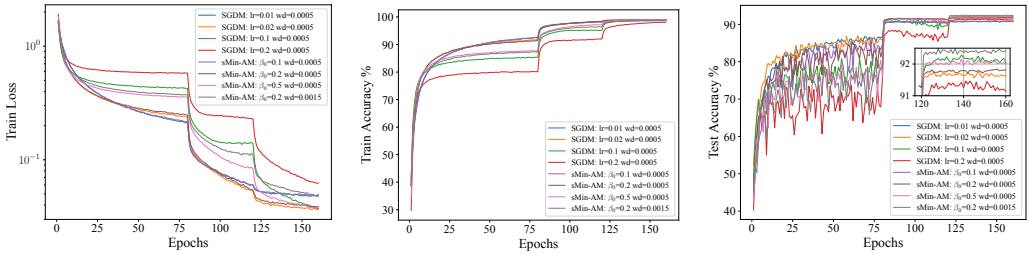
observed that sMin-AM converges faster to an acceptable accuracy. The process of sMin-AM is also much more stable than SAM(1) and SAM(10).



(a) Train loss (b) Train accuracy (c) Test accuracy

Figure 11: Effect of weight decay on training CIFAR-10/ResNet20. "lr" and "wd" are abbreviations of learning rate and weight decay.

Effect of weight decay. Figure 11 shows the effect of weight decay on the training process. It suggests that a larger weight decay can slow down the training in terms of training loss. When using the same weight decay, sMin-AM often has faster convergence than SGDM. The results also justify our settings of the weight decay.



(a) Train loss (b) Train accuracy (c) Test accuracy

Figure 12: Effect of learning rate or mixing parameter on training CIFAR-10/ResNet20. "lr" and "wd" are abbreviations of learning rate and weight decay.

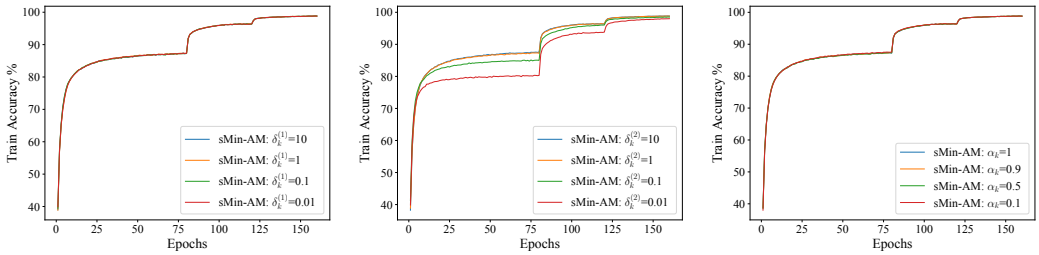
Effect of learning rate/mixing parameter. Figure 12 shows the behaviours of SGDM with different learning rate settings and sMin-AM with different mixing parameters. Small learning rate or small mixing parameter can lead to fast convergence in the beginning, but the training may stagnate early and the test performance is not satisfactory.

Effects of $\delta_k^{(1)}$, $\delta_k^{(2)}$, and α_k. In Figure 13, we show the results of sMin-AM with different settings of other hyperparameters, i.e., $\delta_k^{(1)}$, $\delta_k^{(2)}$, and α_k. It is observed that sMin-AM is more sensitive to $\delta_k^{(2)}$ than $\delta_k^{(1)}$ and α_k, which may be due to the fact that $\delta_k^{(2)}$ directly affects the regularization in the Min-AM update (Line 10 in Algorithm 4). When choosing $\delta_k^{(2)} \geq 1$, the effect of $\delta_k^{(2)}$ is also minor.

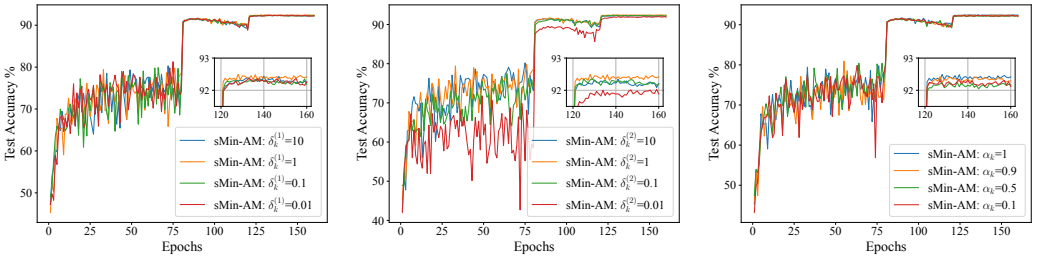
As a result, the most critical hyperparameters that affect the effectiveness of sMin-AM are the mixing parameter β_k, the regularization parameter $\delta_k^{(2)}$, and weight decay. The weight decay is a common hyperparameter that needs to be tuned for each optimizer; for $\delta_k^{(2)}$, we can choose $\delta_k^{(2)} \geq 1$ to ensure $H_k^A \succeq 0$. So except for the weight decay, we only tuned the mixing parameter in the experiments.

Check of the choices of $\delta_k^{(1)}$ and $\delta_k^{(2)}$. From Lemma 2, we know $\|H_k^A - \beta_k I\|_2 \leq -\rho_k \|p_k\|_2^2 - 2\beta_k \rho_k \|p_k\|_2 \|q_k\|_2 + \beta_k \rho_k^2 \|p_k\|_2^2 \|q_k\|_2^2$ assuming $\rho_k \leq 0$. In Figure 14, we plot the value of this bound. It is found that $\|H_k^A - \beta_k I\|_2 \leq \beta_k C_2$ for some constant C_2, which justifies our choice of the regularization parameters.

Train neural networks with fewer epochs and less time. We also tested sMin-AM for training different epochs. Figure 15 shows the comparison between sMin-AM and SGDM for training 80,



(a) Train accuracy (different $\delta_k^{(1)}$) (b) Train accuracy (different $\delta_k^{(2)}$) (c) Train accuracy (different α_k)



(d) Test accuracy (different $\delta_k^{(1)}$) (e) Test accuracy (different $\delta_k^{(2)}$) (f) Test accuracy (different α_k)

Figure 13: Effects of $\delta_k^{(1)}, \delta_k^{(2)}$, and α_k on sMin-AM for training CIFAR-10/ResNet20. The default setting is $\delta_k^{(1)} = 1, \delta_k^{(2)} = 1, \alpha_k = 1, \beta_0 = 1$, and weight decay is 0.0015. When one hyperparameter was inspected, the other hyperparameters were set as default.

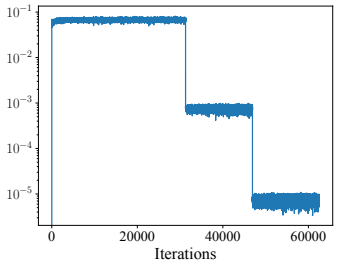
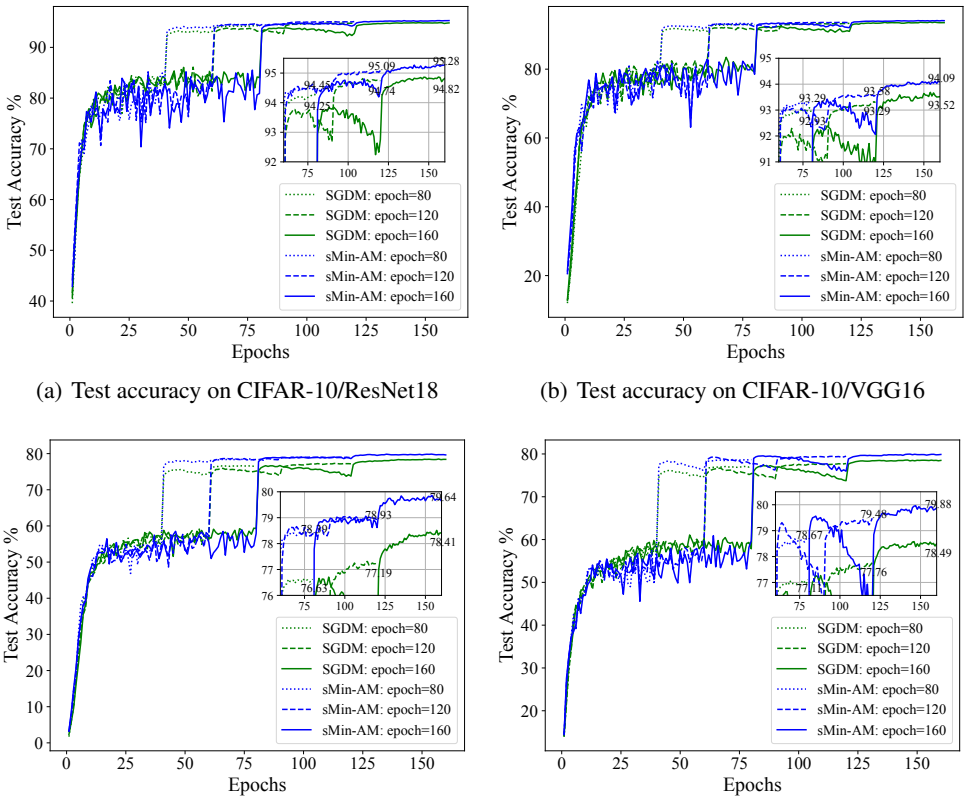


Figure 14: The value of $-\rho_k\|p_k\|_2^2 - 2\beta_k\rho_k\|p_k\|_2\|q_k\|_2 + \beta_k\rho_k^2\|p_k\|_2^2\|q_k\|_2^2$ during training CIFAR-10/ResNet20.

120, and 160 epochs. The results show that sMin-AM can attain comparable accuracy to SGDM in fewer epochs, which infers the faster training process of sMin-AM. In Table 2, by setting the final test accuracy (160 epochs) of SGDM as the baseline, we report the memory cost, per-epoch time, total training epochs, total training time of SAM(10), ST-AM, and sMin-AM to achieve a comparable accuracy to SGDM (within 0.05% difference). The attained final accuracy is also shown. It is found that sMin-AM largely reduces the memory cost of SAM(10) and can achieve comparable results to those of SGDM using less training time.

Discussion about the computational cost and memory cost. The per-epoch computational cost and memory cost are closely related to three factors: the network architecture, the training data, and the optimizer. From Table 2, the effects of the network architecture and the optimizer are clear. For the effect of training data, we consider the batch size. Given a specific network, the cost is composed of two parts: (i) updating network parameters by the optimizer; (ii) other necessary computations, such as the forward and back propagations of the neural network, data transfers between memory and disks, etc., where additional memory and processing time are required. If the cost of Part (ii) only occupies a small proportion of the total cost, the cost incurred by the optimizer will be of great importance. In Figure 16, we plot the ratios of memory/per-epoch time of SAM(10)/ST-AM/sMin-AM to that of SGDM, for training ResNeXt50 on CIFAR-100 with different batch sizes (16, 32, 64, and 128).



(a) Test accuracy on CIFAR-10/ResNet18

(b) Test accuracy on CIFAR-10/VGG16

(c) Test accuracy on CIFAR-100/ResNeXt50

(d) Test accuracy on CIFAR-100/DenseNet121

Figure 15: Comparison between SGDM and sMin-AM on training deep neural networks for 80, 120, and 160 epochs.

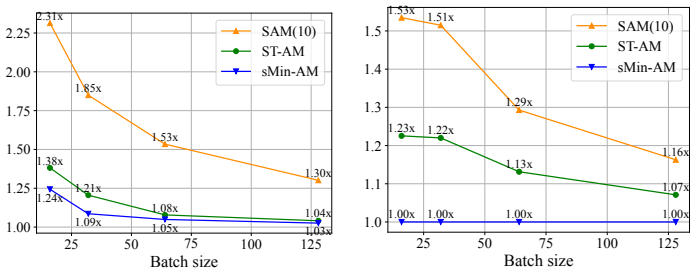
Table 2: The cost and final test accuracy compared with SGDM. Notations "m","t/e", "e", "t", "a" are abbreviations of memory, per-epoch time, training epochs, total training time, accuracy.

Cost (× SGDM) & accuracy	CIFAR-10/ResNet18 m	t/e	e	t	a(%)	CIFAR-10/VGG16 m	t/e	e	t	a(%)
SGDM	1.00	1.00	1.00	1.00	94.82	1.00	1.00	1.00	1.00	93.52
SAM(10)	1.73	1.78	0.56	1.00	94.81	2.51	2.59	1.00	2.59	93.59
ST-AM	1.05	1.46	0.56	0.82	94.84	1.55	1.91	0.88	1.67	93.56
sMin-AM	1.01	1.15	0.56	0.64	94.85	1.35	1.25	0.63	0.78	93.56
Cost (× SGDM) & accuracy	**CIFAR-100/ResNeXt50 m**	**t/e**	**e**	**t**	**a(%)**	**CIFAR-100/DenseNet121 m**	**t/e**	**e**	**t**	**a(%)**
SGDM	1.00	1.00	1.00	1.00	78.41	1.00	1.00	1.00	1.00	78.49
SAM(10)	1.30	1.16	0.50	0.58	78.37	1.16	1.19	0.50	0.60	78.84
ST-AM	1.04	1.07	0.50	0.54	78.39	1.01	1.11	0.50	0.55	78.90
sMin-AM	1.03	1.00	0.50	0.50	78.39	1.01	1.09	0.50	0.55	78.67

When a smaller batch size is used, the Part (ii) cost reduces and the proportion of Part (i) cost in the total cost increases, thus the performance improvement of sMin-AM over SAM(10)/ST-AM is more significant. To investigate the effect of model size on the cost, we tested SAM(10)/ST-AM/sMin-AM on CIFAR-10/ResNet18 and CIFAR-10/ResNet50, where the batch size is 16 so that the Part (ii) cost is not large. In Table 3, it is observed that for a network of larger scale (ResNet50), sMin-AM is more advantageous than SAM(10)/ST-AM in terms of memory cost.

The difference between CIFAR-10 and CIFAR-100 has only a minor effect on the cost. For a given network architecture, e.g., VGG16, its implementations for CIFAR-10 and CIFAR-100 differ in the

dimension of the last linear layer. Since the parameters of this linear layer usually occupy a very small portion of the whole parameters, the costs (memory and per-epoch running time) on CIFAR-10 and CIFAR-100 are roughly the same, as shown in Table 4.



(a) Memory cost (b) Computation time

Figure 16: Memory and per-epoch time for training CIFAR-100/ResNeXt50 with different batch sizes, where those of SGDM are set as the units.

Table 3: The memory and per-epoch time of SAM(10)/ST-AM/sMin-AM, where the batch size is 16. The memory and per-epoch time of SGDM are set as the units; memory and per-epoch time are abbreviated as "m" and "t/e", respectively.

Cost ($\times$ SGDM)	CIFAR-10/ResNet18		CIFAR-10/ResNet50	
	m	t/e	m	t/e
SAM(10)	2.44	2.66	3.01	2.63
ST-AM	1.59	1.91	1.79	1.86
sMin-AM	1.41	1.21	1.52	1.15

Table 4: The memory and per-epoch time of SAM(10)/ST-AM/sMin-AM, where the batch size is 128. The memory and per-epoch time of SGDM are set as the units; the first element and the second element in $(\cdot, \cdot)$ denote the memory and per-epoch time, respectively.

Cost ($\times$ SGDM)	CIFAR-10				CIFAR-100			
	ResNet18	VGG16	ResNeXt50	DenseNet121	ResNet18	VGG16	ResNeXt50	DenseNet121
SAM(10)	(1.73, 1.78)	(2.51, 2.59)	(1.30, 1.18)	(1.16, 1.21)	(1.84, 1.85)	(2.52, 2.53)	(1.30, 1.16)	(1.16, 1.19)
ST-AM	(1.05, 1.46)	(1.55, 1.91)	(1.04, 1.06)	(1.02, 1.08)	(1.12, 1.47)	(1.56, 1.84)	(1.04, 1.07)	(1.01, 1.11)
sMin-AM	(1.01, 1.15)	(1.35, 1.25)	(1.02, 1.00)	(1.01, 1.01)	(1.08, 1.09)	(1.36, 1.19)	(1.03, 1.00)	(1.01, 1.09)

F.3.3 Experiment of training ResNet50 on ImageNet

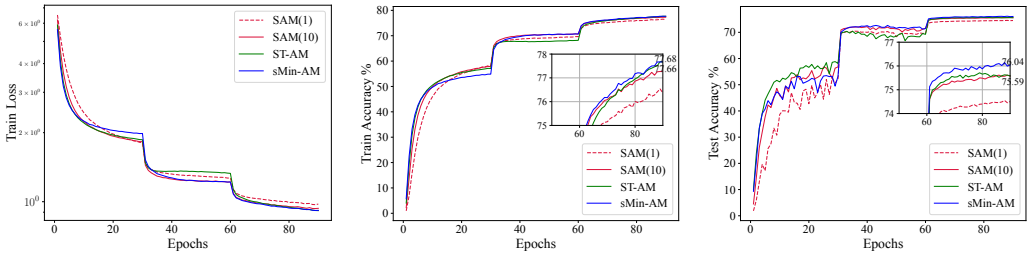
We applied sMin-AM to train ResNet50 on the ImageNet dataset, which contains 1.2M images for training and 50K images for test. The code was based on the example from PyTorch[3]. We followed the standard process of training ResNet [30]. Four GPUs were used to conduct the experiment. SGDM was used for comparison. The batch size was 256. The weight decay for each method was 0.0001. We trained ResNet50 for 90 epochs. The learning rate of SGDM and the mixing parameter of sMin-AM were decayed by 0.1 at the 30th and 60th epochs. For SGDM, the momentum was 0.9 and the learning rate was 0.1, which was the standard setting [30, 66, 34, 64]. For sMin-AM, we set the initial mixing parameter $\beta_0 = 0.5$, and the other hyperparameters were kept unchanged.

Comparisons with SAM and ST-AM. We conducted tests to compare the accuracy of sMin-AM with that of SAM/ST-AM, and results in Figure 17 show that sMin-AM also improves the test accuracy of SAM(10)/ST-AM.

F.3.4 An additional experiment: adversarial training

We conduct an additional experiment to further compare the effectiveness of SGDM, ST-AM, and sMin-AM for training deep neural networks. The considered problem is *adversarial training*, which

[3] https://github.com/pytorch/examples/tree/master/imagenet.



(a) Training loss (b) Train accuracy (c) Test accuracy

Figure 17: Comparisons between SAM, ST-AM, and sMin-AM for training ResNet50 on ImageNet. The results of the final accuracy of ST-AM and sMin-AM are shown in (b) and (c).

tries to solve the following minimax problem

$$\min_{x\in\mathbb{R}^d} \frac{1}{T}\sum_{i=1}^{T} \max_{\|\bar{\xi}_i-\xi_i\|_2\le\epsilon} f_{\bar{\xi}_i}(x), \tag{174}$$

where $\bar{\xi}_i$ is the adversarial data sample in the ϵ-ball centered at the data sample ξ_i. We used the standard PGD adversarial training process [40]: using projection gradient descent to solve the maximization problem in (174), and applying SGDM/ST-AM/sMin-AM to solve the minimization problem in (174). The tests were conducted on CIFAR-10/WideResNet34-10 and CIFAR-100/DenseNet121, where for the training dataset of CIFAR-10/CIFAR-100, 5K images were randomly selected as the validation dataset and the other 45K images were used as the new training dataset. We trained the neural network for 200 epochs, where the learning rate/mixing parameter was decayed at the 100th and the 150th epochs. The best checkpoint model on the validation dataset was chosen to be evaluated on the test dataset. For SGDM and ST-AM, we used the recommended setting of hyperparameters in [26, 63]. For sMin-AM, we set $\beta_0 = 0.3$, and other hyperparameters were kept unchanged.

Table 5: Clean test accuracy (%) and robust test accuracy (%) on adversarial training.

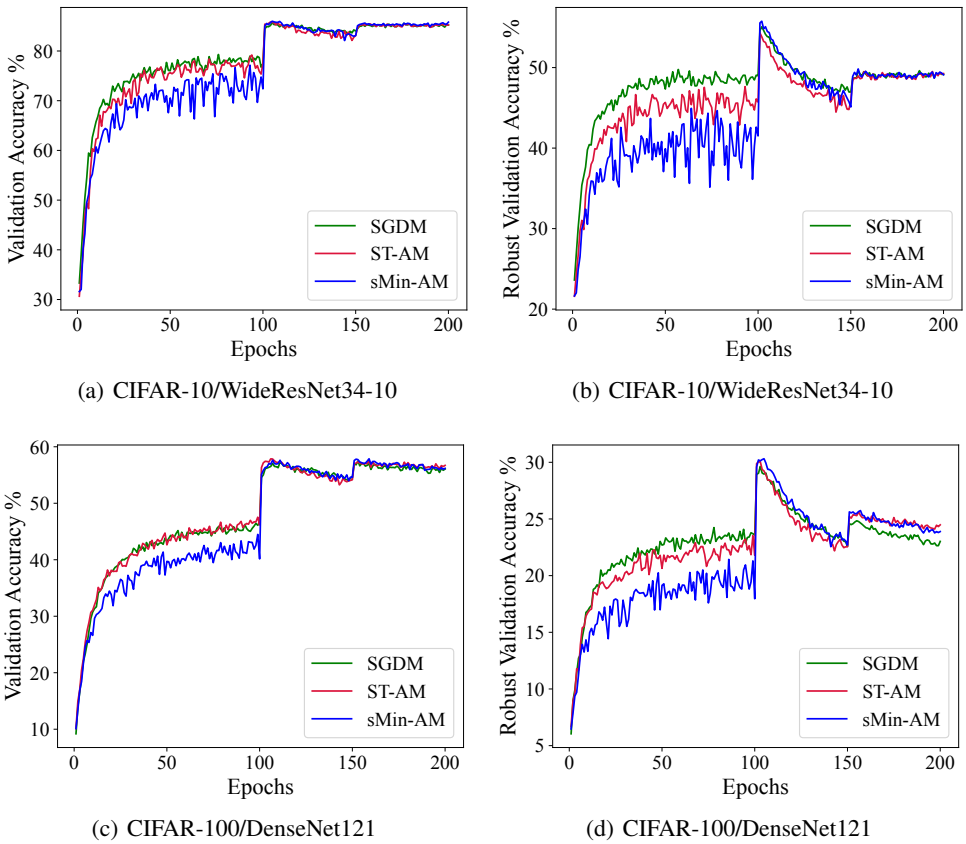
Optimizer	CIFAR-10/WideResNet34-10				CIFAR-100/DenseNet121			
	Clean	FGSM	PGD-20	C&W$_\infty$	Clean	FGSM	PGD-20	C&W$_\infty$
SGDM	85.48	66.42	54.00	53.28	59.45	39.75	30.91	29.02
ST-AM	**85.79**	66.43	53.46	52.88	**60.48**	40.39	31.19	29.56
sMin-AM	85.76	**67.49**	**54.67**	**54.05**	60.04	**40.50**	**31.71**	**29.95**

The training process of each optimizer is shown in Figure 18, where we plot the accuracy on validation dataset (called as validation accuracy), and the accuracy on adversarial data samples generated by the PGD-10 attack [40] on validation dataset (called as robust validation accuracy). It is observed that the best robust validation accuracy of sMin-AM is higher than that of SGDM/ST-AM, which indicates that better checkpoint model can be obtained by sMin-AM.

In Table 5, we report two types of test accuracy: clean test accuracy, where the clean test data was used for classification; robust test accuracy, where corrupted test data was used for classification. Three attacking methods were used for the robust test accuracy evaluation: FGSM [25], PGD-20 [40], and C&W$_\infty$ attack [13]. It shows that sMin-AM significantly improves SGDM, and outperforms ST-AM in terms of robust test accuracy, which is a desirable property in adversarial training.

G Limitations

We focus on developing the variant of AM with minimal memory size for solving optimization problems. The properties of Min-AM for deterministic optimization are established in the smooth case. The non-smooth optimization is important and one possible direction is to adapt Min-AM to the framework of proximal quasi-Newton methods [15, 7]. We leave it as an extension of Min-AM in the future work.



(a) CIFAR-10/WideResNet34-10

(b) CIFAR-10/WideResNet34-10

(c) CIFAR-100/DenseNet121

(d) CIFAR-100/DenseNet121

Figure 18: Left: accuracy on the clean validation dataset (validation accuracy). Right: accuracy on the validation dataset attacked by PGD-10 (robust validation accuracy).

Min-AM exploits the symmetry of Hessian in solving optimization problems. For general fixed-point problems, the Jacobian can be non-symmetric. However, in this case, the short-term recurrence algorithms that are equivalent to the full-memory methods generally do not exist even in solving linear systems [53].