Sharpened Quasi-Newton Methods: Faster Superlinear Rate and Larger Local Convergence Neighborhood

Qiujiang Jin [1] **Alec Koppel** [2] **Ketan Rajawat** [3] **Aryan Mokhtari** [1]

Abstract

Non-asymptotic analysis of quasi-Newton methods have gained traction recently. In particular, several works have established a non-asymptotic superlinear rate of $\mathcal{O}((1/\sqrt{t})^t)$ for the (classic) BFGS method by exploiting the fact that its error of Newton direction approximation approaches zero. Moreover, a greedy variant of BFGS was recently proposed which accelerates its convergence by directly approximating the Hessian, instead of the Newton direction, and achieves a fast local quadratic convergence rate. Alas, the local quadratic convergence of Greedy-BFGS requires way more updates compared to the number of iterations that BFGS requires for a local superlinear rate. This is due to the fact that in Greedy-BFGS the Hessian is directly approximated and the Newton direction approximation may not be as accurate as the one for BFGS. In this paper, we close this gap and present a novel BFGS method that has the best of both worlds in that it leverages the approximation ideas of both BFGS and Greedy-BFGS to properly approximate the Newton direction and the Hessian matrix simultaneously. Our theoretical results show that our method outperforms both BFGS and Greedy-BFGS in terms of convergence rate, while it reaches its quadratic convergence rate with fewer steps compared to Greedy-BFGS. Numerical experiments on various datasets also confirm our theoretical findings.

[1]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA. [2]Amazon, Bellevue, WA, USA. [3]Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, UP, INDIA. Correspondence to: Qiujiang Jin <qiujiang@austin.utexas.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

1. Introduction

In this paper, we focus on the use of quasi-Newton methods to solve the following unconstrained problem

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

where $f:\mathbb{R}^d\to\mathbb{R}$ is strongly convex and its gradient is Lipschitz continuous; see details in Section 3.2. We denote the unique optimal solution of (1) by x_*.

First-order algorithms, i.e., gradient-based methods, are widely used for solving (1), and it is well-known that their iterates converge to x_* at a linear rate (i.e., the error decays exponentially fast). A major advantage of first-order methods is their low computational cost of $\mathcal{O}(d)$, where d is the problem dimension. However, the convergence rate of these methods depends on the problem curvature and hence they could be slow in ill-conditioned problems. Second-order methods that leverage the objective function Hessian to improve their curvature estimation often arise as a natural alternative to accelerate convergence in ill-posed problems, and they achieve fast local convergence rates (Bennett, 1916; Ortega & Rheinboldt, 1970; Conn et al., 2000; Nesterov & Polyak, 2006). Specifically, Newton's method achieves a local quadratic convergence rate when applied to solve (1) with the additional assumption that the Hessian is Lipschitz (Boyd & Vandenberghe, 2004, Chapter 9). A major obstacle in the implementation of Newton's method though is its requirement to solve a linear system at each iteration, which makes its computational cost $\mathcal{O}(d^3)$.

Quasi-Newton (QN) methods serve as a middle ground between first- and second-order methods, as they improve the linear rate of first-order methods and converge superlineraly, and simultaneously their computation cost is $\mathcal{O}(d^2)$ which improves the $\mathcal{O}(d^3)$ cost of Newton-type methods. Their main idea is to construct a positive definite matrix that approximates the Hessian required in Newton's method. Since the update of Hessian approximation matrix in QN methods only requires a set of matrix-vector multiplications, their computational cost per iteration is $\mathcal{O}(d^2)$. There are several types of QN methods that differ in their Hessian approximation updates, including Symmetric Rank-One (SR1) method (Conn et al., 1991), the Broyden method (Broy-

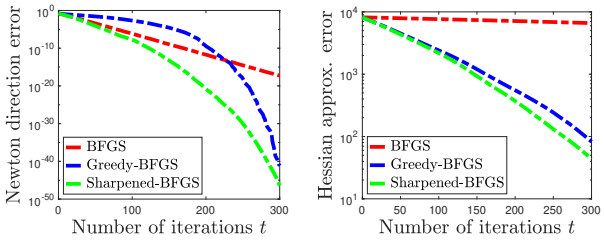


Figure 1. Comparison of BFGS, Greedy-BFGS, and the proposed Sharpened-BFGS algorithms in terms of Newton direction error (top) and Hessian approximation error (bottom) for a quadratic problem with dimension $d = 400$ and condition number $\kappa = 100$.

den, 1965; Broyden et al., 1973; Gay, 1979), the Davidon-Fletcher-Powell (DFP) method (Davidon, 1959; Fletcher & Powell, 1963), the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), the limited-memory BFGS (L-BFGS) method (Nocedal, 1980; Liu & Nocedal, 1989), and the Greedy-QN method (Rodomanov & Nesterov, 2021a).

Perhaps the most important property of QN methods is their local superlinear convergence. Specifically, Rodomanov & Nesterov (2021a) introduced and analyzed a novel Greedy-QN method which is based on the classical Broyden class of QN methods and uses a greedily-selected vector to maximize certain measure of progress (see Section 2 for more details). Greedy-QN achieves a non-asymptotic quadratic convergence rate of $(1-1/d\kappa)^{t^2/2}(d\kappa)^t$, where κ is the problem condition number. Note that this bound is equivalent to $((1-1/d\kappa)^{t/2}(d\kappa))^t$ which shows that the fast quadratic convergence starts when $t \geq d\kappa \ln(d\kappa)$. It is also worth noting that in comparison with standard QN methods, greedy QN requires more information, including the diagonal elements of the Hessian at each iteration. In a follow-up work, Rodomanov & Nesterov (2021b) proved a non-asymptotic superlinear convergence rate for standard QN methods including the DFP and BFGS methods. They showed that BFGS and DFP achieve a local superlinear convergence rate of $(d\kappa/t)^{t/2}$ and $(d\kappa^2/t)^{t/2}$, respectively, under the assumptions that the objective function is strongly convex, smooth and strongly self-concordant. Later, Rodomanov & Nesterov (2021c) improved their results to the convergence rates of $((d\ln\kappa)/t)^{t/2}$ for BFGS and $((d\kappa\ln\kappa)/t)^{t/2}$ for DFP. As noted in Table 1, the convergence rate of BFGS is slower than the one for Greedy-BFGS, but the superlinear rate starts at a smaller time index.

Contributions. As mentioned above, (standard) BFGS aims at approximating the Newton direction and obtains a fast convergence rate from the beginning, but it fails to perfectly approximate the Hessian. On the other hand, Greedy-BFGS's goal is to directly approximate the Hessian matrix and therefore at first its convergence is slower than BFGS, but once its Hessian approximation improves it converges substantially faster than BFGS; see Figure 1. Considering these points, a natural question that arises is:

> *Is it possible to achieve the best of two worlds and develop a QN method that exhibits faster local convergence by approximating both the Newton direction and the Hessian matrix?*

In this paper, we address this question by proposing a novel Sharpened-BFGS method which utilizes ideas of the classic BFGS and Greedy-BFGS. The proposed Sharpened-BFGS method exploits the initial fast convergence of BFGS by approximating the Newton direction, while developing an accurate approximation of the Hessian by following the Greedy-BFGS idea which allows for a quadratic convergence rate. As stated in Table 1, our method outperforms both BFGS and Greedy-BFGS in terms of convergence rate, while it reaches the superlinear convergence rate with fewer steps compared to the greedy method. We should add that the computational cost per iteration of Sharpened-BFGS is the same as its standard and greedy counterparts.

Related Work. Jin & Mokhtari (2020) established a non-asymptotic superlinear convergence rate of $(1/t)^{t/2}$ for standard QN methods under the assumptions that the objective function is strongly convex, smooth and its Hessian is Lipschitz continuous at the optimal solution. They also established a similar result for self-concordant functions. Their local convergence rate does not depend on the problem parameters such as d or κ, but their results require both Hessian approximation error and the distance to the optimal solution to be sufficiently small. Moreover, Ye et al. (2021) obtained the explicit local superlinear convergence rate of the SR1 method. Further, Lin et al. (2021a) extended the non-asymptotic local superlinear convergence rate of the Broyden family QN methods for solving nonlinear equations. It is worth noting that recently, Lin et al. (2021b) proposed a randomized version of Greedy-BFGS which obtains a convergence rate of $(d\kappa(1-\frac{1}{d})^{\frac{t}{2}})^t$. This randomized technique can be also utilized for our proposed Sharpened BFGS method to improve its convergence rate dependency in terms of κ. Due to space limitation, we present and analyze randomized Sharpened-BFGS in Appendix H.

2. Preliminaries

In this section, we review some basics of QN methods that we require for developing our method. Consider $x_t \in \mathbb{R}^d$ as the iterate associated with time index t and $\nabla f(x_t) \in \mathbb{R}^d$ as the objective function gradient evaluated at x_t. The general form of a QN update is given by

$$x_{t+1} = x_t - \eta_t G_t^{-1} \nabla f(x_t), \tag{2}$$

Algorithm	Superlinear Rate	t_0
Standard BFGS	$(\frac{d\ln\kappa}{t})^{\frac{t}{2}}$	$d\ln\kappa$
Greedy-BFGS	$\left(d\kappa(1-\frac{1}{d\kappa})^{\frac{t}{2}}\right)^t$	$d\kappa\ln(d\kappa)$
Sharpened-BFGS	$(1-\frac{1}{d\kappa})^{\frac{t(t-1)}{4}}(\frac{d\kappa}{t})^{\frac{t}{2}}$	$d\kappa$

Table 1. Convergence rate comparison of different variants of BFGS, where d is the dimension, κ is the condition number and t_0 is the time index at which the superlinear convergence begins.

where $\eta_t > 0$ is the step size (learning rate) and $G_t \in \mathbb{R}^{d\times d}$ is the matrix approximating the Hessian $\nabla^2 f(x_t) \in \mathbb{R}^{d\times d}$. In general, η_t is determined by some line search algorithms so that the iteration generated converge to the optimal solution globally. In this paper, we focus on the local convergence analysis of QN algorithms, which requires the use of a unit step size $\eta_t = 1$. Hence, in the rest of the paper, we assume that the iterates $\{x_t\}_{t=1}^{\infty}$ stay in a local neighborhood of x_* and $\eta_t = 1$ is always admissible.

2.1. BFGS Operator and Algorithm

The essence of a QN method is its update for the Hessian approximation matrix G_t. There are various ways for updating G_t, but in this paper we focus on the BFGS method. Before stating the BFGS method, we first introduce it as an algorithm for approximating linear operators. This perspective turns out to be advantageous for unifying it with its greedy variant. To do so, consider $A \in \mathbb{R}^{d\times d}$ as a positive definite linear operator, and suppose $G \in \mathbb{R}^{d\times d}$ is the operator that approximates A and is updated according to the BFGS update. Then, the BFGS update rule for approximating operator A along the direction $u \in \mathbb{R}^d\backslash\{0\}$ is

$$BFGS(A, G, u) = G_+ := G - \frac{Guu^\top G}{u^\top Gu} + \frac{Auu^\top A}{u^\top Au}. \quad (3)$$

Note that this update tries to move from G to G_+ in a way that operators A and G_+ are equal to each other in the direction of vector u, i.e., $Au = G_+u$.

Remark 2.1. *As noted in (2), we need to compute the inverse of the Hessian approximation matrix at each step. Hence, we need a direct update for the Hessian inverse approximation matrices. By exploiting the Sherman-Morrison-Woodbury formula, one can show that the Hessian inverse approximation matrix $H = G^{-1}$ update can be written as*

$$H_+ = \left(I - \frac{uu^\top A}{u^\top Au}\right) H \left(I - \frac{Auu^\top}{u^\top Au}\right) + \frac{uu^\top}{u^\top Au}. \quad (4)$$

Hence, the computational cost of BFGS is $\mathcal{O}(d^2)$, as it only requires computation of matrix-vector multiplication.

When we focus on minimizing a function and the ultimate linear operator that we aim to approximate is its curvature, then we select the direction as $u = x_{t+1} - x_t$ and the desired operator as the average Hessian $A = J_t := \int_0^1 \nabla^2 f(x_t + \tau(x_{t+1} - x_t))d\tau$. This way we ensure that the new Hessian approximation matrix G_{t+1} satisfies the secant condition, i.e.,

$$G_{t+1}(x_{t+1} - x_t) = J_t(x_{t+1} - x_t) = \nabla f(x_{t+1}) - \nabla f(x_t),$$

If we define the variable and gradient differences as

$$s_t := x_{t+1} - x_t, \qquad y_t := \nabla f(x_{t+1}) - \nabla f(x_t), \quad (5)$$

then the classic BFGS update is equivalent to

$$G_{t+1} = G_t - \frac{G_t s_t s_t^\top G_t}{s_t^\top G_t s_t} + \frac{y_t y_t^\top}{s_t^\top y_t}. \quad (6)$$

A major advantage of the BFGS update in (6) is that it forces the new Hessian approximation matrix G_{t+1} to satisfy the secant condition, which implies $G_{t+1}s_t = y_t$. This condition ultimately ensures that the BFGS direction $G_t^{-1}\nabla f(x_t)$ approaches the Newton direction $\nabla^2 f(x_t)^{-1}\nabla f(x_t)$; see Chapter 6 of (Nocedal & Wright, 2006) for details.

2.2. Greedy-BFGS Algorithm

As mentioned in the previous section, BFGS does a good a job in approximating the Newton direction, but its Hessian approximation may not approach the true Hessian. To be precise, consider the following metric which captures the difference between positive definite matrices $A, G \in \mathbb{R}^{d\times d}$

$$\sigma(A, G) := Tr(A^{-1}G) - d, \quad (7)$$

where $Tr(X)$ is the trace of matrix X, i.e., the sum of the diagonal elements of X. Note that if $A \preceq G$, we can use $\sigma(A, G)$ as a potential function that measures the distance between two matrices A and G. Note that $\sigma(A, G) = 0$ if and only if $A = G$. Using the above potential function, in the next lemma, we state the error of Hessian approximation for the BFGS operator in (3). The proof can be found in (Rodomanov & Nesterov, 2021a).

Lemma 2.1. *Consider positive definite matrices $A, G \in \mathbb{R}^{d\times d}$ and suppose that $G_+ = BFGS(A, G, u)$ as defined in (3) and $u \in \mathbb{R}^d\backslash\{0\}$. If $A \preceq G$, then we have*

$$\sigma(A, G) - \sigma(A, G_+) \geq \frac{u^\top Gu}{u^\top Au} - 1. \quad (8)$$

This result shows how fast the gap between the Hessian approximation and the true Hessian decreases after one step of BFGS. The result in Lemma 2.1 also shows that the selection of direction u can influence the decrease in the trace potential function $\sigma(A, G)$ after one BFGS update. Note that for an arbitrary direction $u \in \mathbb{R}^d\backslash\{0\}$, there is no guarantee that the Hessian approximation matrix converges

to the exact Hessian matrix. In fact, if we set $u = x^+ - x$ as done in the classic BFGS update, there is no guarantee that $\sigma(A, G)$ converges to 0. This observation reveals the following question: How can we select u to maximize the progress in decreasing $\sigma(A, G)$ and ensuring that $\sigma(A, G)$ converges to 0, i.e., G converges to A?

Rodomanov & Nesterov (2021a) answered this question by proposing a greedy selection scheme for determination of the best choice of u. To better explain this concept, consider a quadratic problem, where the objective function Hessian is fixed and denoted by the positive definite matrix A. In this case, to maximize the right hand side of (8), which shows the progress for the BFGS update, one could select u as

$$\bar{u}(A, G) := \underset{u \in \{e_i\}_{i=1}^d}{\arg\max} \frac{u^\top G u}{u^\top A u}, \tag{9}$$

where $\{e_i\}$ is the vector whose i-th element is 1 and its remaining elements are 0. If we choose $u = \bar{u}(A, G)$ in each iteration of BFGS update (3), we obtain the Greedy-BFGS algorithm in (Rodomanov & Nesterov, 2021a). The advantage of this greedily selected is that it ensures the trace potential function $\sigma(A, G)$ is strictly decreasing and converges to 0 linearly as specified in the following lemma.

Lemma 2.2 ((Rodomanov & Nesterov, 2021a)). *Consider positive definite matrices $A, G \in \mathbb{R}^{d\times d}$ that satisfy $A \preceq G$ and $\mu I \preceq A \preceq LI$, where $0 < \mu \leq L$ are two constants. Suppose that $\bar{G}_+ = BFGS(A, G, \bar{u}(A, G))$ where $\bar{u}(A, G) \in \mathbb{R}^d$ is greedily selected as defined in (9). Then,*

$$\sigma(A, \bar{G}_+) \leq \left(1 - \frac{\mu}{dL}\right) \sigma(A, G). \tag{10}$$

This result shows that by following the Greedy-BFGS update the error of Hessian approximation, in terms of the metric $\sigma(.,.)$ defined in (7), converges to zero linearly and eventually the sequence of Hessian approximations approaches the true Hessian. Note that, for the non-quadratic case, a similar argument holds, but the algorithm should be slightly modified as the computation of the average Hessian J_t is costly and instead one might use the current Hessian $\nabla^2 f(x_t)$. We discuss this point in detail in the following section, when we present our Sharpened-BFGS method.

3. Sharpened-BFGS

In this section, we propose the Sharpened-BFGS algorithm which benefits from the update of BFGS for Newton direction approximation and the Greedy-BFGS update to approximate the Hessian matrix. In a nutshell, the update of Sharpened-BFGS first adjusts the Hessian approximation according to the BFGS update by setting $u = x_{t+1} - x_t$, and then improves the Hessian approximation by following the greedy update, and selecting the vector u in a greedy fashion. To introduce our method, we first focus on a quadratic program where the Hessian is fixed. We then build on our intuition from the quadratic case to develop the general version of our method for the problem in (1).

Algorithm 1 Sharpened-BFGS applied to (11).

Require: Initial point x_0 and initial matrix $G_0 = LI$.
for $t = 0, 1, 2, \ldots$ **do**
  Update the variable: $x_{t+1} = x_t - G_t^{-1} \nabla f(x_t)$;
  Compute $s_t = x_{t+1} - x_t$;
  Compute $\bar{G}_t = BFGS(A, G_t, s_t)$;
  Compute $\bar{u} = \bar{u}(A, \bar{G}_t)$ according to (9);
  Compute $G_{t+1} = BFGS(A, \bar{G}_t, \bar{u})$;
end for

3.1. Quadratic Programming

Consider a special case of (1) where the objective function is quadratic and given by

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{2} x^\top A x + b^\top x, \tag{11}$$

where $A \in \mathbb{R}^{d\times d}$ is a symmetric positive definite matrix satisfying $\mu I \preceq A \preceq LI$ and $b \in \mathbb{R}^d$. The Sharpened-BFGS algorithm applied to (11) is shown in Algorithm 1. We observe that the proposed algorithm involves two BFGS updates per iteration. Intuitively, we improve the Hessian approximation along the classical BFGS direction and subsequently along the Greedy-BFGS direction. Notice that the initial Hessian approximation matrix is $G_0 = LI$. Hence, the initial Hessian inverse approximation matrix is simply $H_0 = (1/L)I$. For the quadratic problem, the sequence generated by Sharpened-BFGS converges to the optimal solution globally, as we show in Theorems 3.2 and 3.4. Hence, the initial point x_0 can be any vector in $\mathbb{R}^d$.

To formally show how Sharpened-BFGS exploits the fast properties of both BFGS and Greedy-BFGS, we first define the Newton decrement as $\lambda_f(x) := \sqrt{\nabla f(x)^\top \nabla^2 f(x)^{-1} \nabla f(x)}$. In our results, we report convergence in terms of $\lambda_f(x)$ and we use the notation $\lambda_t := \lambda_f(x_t)$. We next state the following intermediate result that shows for the class of quasi-Newton updates defined in (2) (with step size $\eta = 1$) on a quadratic program, how fast $\lambda_f(x)$ converges to zero. The proof of this result can be found in (Rodomanov & Nesterov, 2021b).

Lemma 3.1. *Consider the quadratic function in (11) and the sequence of iterates generated according to the update in (2) with step size $\eta_t = 1$. Then, we have that*

$$\lambda_{t+1} = \theta(A, G_t, x_{t+1} - x_t)\lambda_t, \tag{12}$$

where

$$\theta(A, G, u) := \left(\frac{u^\top (G - A) A^{-1} (G - A) u}{u^\top G A^{-1} G u}\right)^{\frac{1}{2}}. \tag{13}$$

First, note that $\theta(A, G, u)$ captures the closeness of G and A along the direction of u, where $u \in \mathbb{R}^d \backslash \{0\}$. The above result shows that the contraction factor for the convergence of the Newton decrement is related to the gap between $G_t(x_{t+1} - x_t)$ and $A(x_{t+1} - x_t)$. In the following theorem, we characterize a global upper bound on $\theta(A, G_t, x_{t+1} - x_t)$ for the Sharpened-BFGS method.

Theorem 3.2. *Consider the Sharpened-BFGS method in Algorithm 1 applied to the quadratic problem (11). Then,*

$$\theta(A, G_t, x_{t+1} - x_t) \leq 1 - \frac{\mu}{L}, \qquad \forall t \geq 0, \tag{14}$$

and therefore

$$\lambda_t \leq \left(1 - \frac{\mu}{L}\right)^t \lambda_0, \qquad \forall t \geq 0. \tag{15}$$

Proof. Check Appendix B. □

The above result shows that the iterates generated by Sharpened-BFGS converge to the solution at a linear rate of $1 - \mu/L$. However, this is not a tight bound and simply follows from the fact that eigenvalues of G_t and A are uniformly bounded. In the next lemma, we present that the sequence $\theta(A, G_t, x_{t+1} - x_t)$ eventually approaches zero and hence the iterates of Sharpened-BFGS converge superlinearly.

Lemma 3.3. *Consider Sharpened-BFGS in Algorithm 1 applied to the quadratic function (11). Further, define $\theta_t := \theta(A, G_t, x_{t+1} - x_t)$ and $\sigma_t := \sigma(A, G_t)$. Then,*

$$\sigma_{t+1} \leq \left(1 - \frac{\mu}{dL}\right)\left(\sigma_t - \theta_t^2\right) \tag{16}$$

for any $t \geq 0$. Moreover, we have

$$\sum_{i=0}^{t-1} \frac{\theta_i^2}{(1 - \frac{\mu}{dL})^i} \leq \sigma_0, \qquad \forall t \geq 1. \tag{17}$$

Proof. Check Appendix C. □

First, note that (16) shows that in Sharpened-BFGS σ_t converges to zero as in the Greedy-BFGS algorithm. Moreover comparing the bound in (16) with the one in (10) shows that in Sharpened-BFGS σ_t converges faster than Greedy-BFGS, as $\theta_t^2 > 0$. Second, the result in (17) shows that the sequence θ_t converges to zero. Hence, one can leverage this result to show a tighter upper bound for θ_t compared to the one in (14) and show a faster rate than the one in (15) for the Sharpened-BFGS. This goal is accomplished in the following Theorem.

Theorem 3.4. *Consider Sharpened-BFGS described in Algorithm 1 applied to the quadratic function (11). Then, for $t \geq 1$ we have*

$$\lambda_t \leq \left(1 - \frac{\mu}{dL}\right)^{\frac{t(t-1)}{4}} \left(\frac{dL}{t\mu}\right)^{\frac{t}{2}} \lambda_0. \tag{18}$$

Algorithm 2 General Sharpened-BFGS

Require: Initial point x_0 and initial matrix $G_0 = LI$.
for $t = 0, 1, 2, \ldots$ **do**
  Update $x_{t+1} = x_t - G_t^{-1} \nabla f(x_t)$;
  Compute $s_t = x_{t+1} - x_t$;
  Set $J_t = \int_0^1 \nabla^2 f(x_t + \tau s_t) d\tau$;
  Compute $\bar{G}_t = BFGS(J_t, G_t, s_t)$;
  Compute $r_t = \|x_{t+1} - x_t\|_{x_t}$;
  Compute $\hat{G}_t = (1 + Mr_t/2)^2 \bar{G}_t$;
  Compute $\bar{u} = \bar{u}(\nabla^2 f(x_{t+1}), \hat{G}_t)$ according to (9);
  Compute $G_{t+1} = BFGS(\nabla^2 f(x_{t+1}), \hat{G}_t, \bar{u})$;
end for

Proof. Check Appendix D. □

If we analyze the superlinear convergence rate in (18), we observe that there are two terms that contribute to the rate. The first is the quadratic rate $(1 - \frac{\mu}{dL})^{\frac{t(t-1)}{4}}$ and the second is $(\frac{dL}{t\mu})^{\frac{t}{2}}$. Notice that for the second term $(\frac{dL}{t\mu})^{\frac{t}{2}}$, the superlinear convergence kicks in only after $t \geq d\frac{L}{\mu}$. Hence, by combining the results of Theorem 3.2 and 3.4, we obtain that during the initial iterations $t < d\frac{L}{\mu}$ Sharpened-BFGS converges linearly and for $t > d\frac{L}{\mu}$ the rate becomes faster than quadratic rate and λ_t approaches zero at a rate of $\mathcal{O}((1 - \frac{\mu}{dL})^{t^2} (\frac{dL}{\mu t})^t)$.

3.2. General Strongly-Convex and Smooth Setting

In this section, we extend our algorithm and its analysis to non-quadratic convex programs. To do so, We first state the required assumptions on the objective function to establish the superlinear convergence rate of Sharpened-BFGS.

Assumption 3.1. *The objective function f is twice differentiable. It is strongly convex with parameter $\mu > 0$ and its gradient ∇f is Lipschitz continuous with parameter $L > 0$.*

Assumption 3.2. *The objective function f is strongly self-concordant with $M > 0$, i.e., for any $x, y, z, w \in \mathbb{R}^d$, we have $\nabla^2 f(y) - \nabla^2 f(x) \preceq M\|y - x\|_z \nabla^2 f(w)$, where $\|y - x\|_z := \sqrt{(y - x)^\top \nabla^2 f(z)(y - x)}$.*

The strongly self-concordant functions form a subclass of the famous self-concordant functions class introduced in (Nesterov, 1989; Nesterov & Nemirovskii, 1994), which plays a fundamental role in the local analysis of Newton's method. The concept of strong self-concordance was first proposed by Rodomanov & Nesterov (2021a) to establish the explicit quadratic convergence rate of the greedy QN method. Note that a strongly convex function with Lipschitz continuous Hessian is strongly self-concordant; see Example 4.1 in (Rodomanov & Nesterov, 2021a).

The general Sharpened-BFGS method is presented in Algorithm 2. We observe that Algorithm 2 is fundamentally

similar to the Algorithm 1 for the quadratic case, but there are still some differences between them. In general, similar to Algorithm 1, we first update the Hessian approximation matrix along the standard BFGS direction and then along the Greedy-BFGS direction. The only difference between Algorithm 1 and 2 is that we add the correction term $r_t = \|x_{t+1}-x_t\|_{x_t}$ in Steps 6 and 7 of Algorithm 2. The reason for this modification is that the trace potential function $\sigma(A, G)$ is only well-defined under the condition $A \preceq G$. Suppose currently the condition $\nabla^2 f(x) \preceq G$ holds. We add the correction term to ensure that after one BFGS update, the new point x_+ and the new Hessian approximation matrix G_+ still satisfy the condition $\nabla^2 f(x_+) \preceq G_+$. Since the Hessian of the general convex function is not fixed, there is no guarantee that the quasi-Newton update can preserve the property of $\nabla^2 f(x) \preceq G$ without that correction term. The initial Hessian approximation matrix is still $G_0 = LI$. We should also add that Step 4 does not require computing $J_t = \int_0^1 \nabla^2 f(x_t + \tau s_t) d\tau$ explicitly. As we discussed in Section 2.1, we can compute the standard BFGS update in Step 5 according to (6).

Remark 3.1. *The computational cost per iteration of Algorithm 1 is $\mathcal{O}(d^2)$. The difference between Algorithm 2 and 1 is in Steps 6 and 7 of Algorithm 2. The computational cost of calculating the vector $r_t = \|x_{t+1} - x_t\|_{x_t}$ and the matrix $\hat{G}_t = (1 + Mr_t/2)^2 \bar{G}_t$ is also $\mathcal{O}(d^2)$. Hence, the computational cost per iteration of Algorithm 2 is also $\mathcal{O}(d^2)$.*

The convergence rate analysis of the Sharpened-BFGS method is inspired by the counterpart of the quadratic function, but there are still some differences between these two analyses as we need to take into account the variation of the Hessian for the non-quadratic case. Most importantly, for the general (non-quadratic) case, we can only obtain local convergence results as we state in Theorems 3.6 and 3.8. In other words, the initial point x_0 should be within a local neighborhood of the optimal solution x_* to guarantee the convergence of Sharpened-BFGS. Similar to Section 3.1, we first establish the relationship between θ defined in (13) and the Newton decrement $\lambda_f(x)$ after one iteration of quasi-Newton update for minimizing a general convex function. The proof can be found in (Rodomanov & Nesterov, 2021b).

Lemma 3.5. *Consider problem (1) and suppose Assumptions 3.1–3.2 are satisfied. Then, the iterates x_t generated according to the update in (2) with step size $\eta_t = 1$ satisfy*

$$\lambda_{t+1} \leq \left(1 + \frac{Mr_t}{2}\right) \theta(J_t, G_t, x_{t+1} - x_t)\lambda_t, \qquad (19)$$

where $J_t := \int_0^1 \nabla^2 f(x_t + \tau(x_{t+1} - x_t))d\tau$ and $r_t := \|x_{t+1} - x_t\|_{x_t}$.

Notice that the above lemma is in parallel to Lemma 3.1 for the quadratic case. Now, we establish a local upper bound for the measurement $\theta(J_t, G_t, x_{t+1}-x_t)$ and prove the local linear convergence rate of the general Sharpened-BFGS method, which is similar to the results in Theorem 3.2.

Theorem 3.6. *Consider Sharpened-BFGS in Algorithm 2 applied to the objective function f satisfying Assumption 3.1 and 3.2. Moreover, suppose that the initial point x_0 satisfies*

$$\lambda_0 \leq \frac{C_0 \mu}{ML}, \qquad (20)$$

where $C_0 = \frac{1}{4} \ln \frac{3}{2}$. Then, for any $t \geq 0$ we have

$$\theta(J_t, G_t, x_{t+1} - x_t) \leq 1 - \frac{2\mu}{3L}, \qquad (21)$$

which leads to

$$\lambda_t \leq (1 - \frac{\mu}{2L})^t \lambda_0. \qquad (22)$$

Proof. Check Appendix E. □

The above theorem presents that in a local neighborhood of the optimal solution, the iterates generated by Sharpened-BFGS achieve a linear convergence rate of $1 - \mu/2L$, which is obtained by a loose bound on θ_t. As mentioned in Section 3.1, our ultimate target is to improve this to the superlinear rate. In the following lemma, we establish inequalities similar to (16) and (17) to establish a superlinear convergence rate for Sharpened-BFGS.

Lemma 3.7. *Consider Sharpened-BFGS in Algorithm 2 applied to the objective function f satisfying Assumptions 3.1 and 3.2. Moreover, suppose that the initial point x_0 satisfies*

$$\lambda_0 \leq \frac{C_0 \mu}{ML}, \qquad (23)$$

where $C_0 = \frac{1}{4} \ln \frac{3}{2}$. Further, consider the definitions $\theta_t := \theta(\nabla^2 f(x_t), G_t, x_{t+1} - x_t)$ and $\sigma_t := \sigma(\nabla^2 f(x_t), G_t)$. Then, for any $t \geq 0$ it holds that

$$\sigma_{t+1} \leq (1 - \frac{\mu}{2dL}) \left[(1 + \frac{M\lambda_t}{2})^4 (\sigma_t + 4Md\lambda_t) - \frac{1}{4}\theta_t^2 \right]. \qquad (24)$$

Moreover, we have

$$\sum_{i=0}^{t-1} \frac{\theta_i^2}{(1 - \frac{\mu}{2dL})^i} \leq 8(\sigma_0 + 4Md\lambda_0), \qquad \forall t \geq 1. \qquad (25)$$

Proof. Check Appendix F. □

The conclusions of the above lemma are similar to the ones for the quadratic case in Lemma 3.3, except that a local condition is required. Specifically, the result in (24) implies that in Sharpened-BFGS σ_t converges to zero as long as λ_0 is sufficiently small. Similarly this rate is faster than the one for Greedy-BFGS as $\theta_t^2 > 0$. Second, the result

in (25) implies that the sequence θ_t converges to zero at a fast rate as its weighted sum by a factor larger than 1 that is exponentially growing is finite. Hence, locally this result provides a tighter upper bound on θ_t compared to the one in (21) and shows a faster rate than the one in (22) for Sharpened-BFGS. We leverage these points to establish the convergence rate of Sharpened-BFGS for non-quadratic problems.

Theorem 3.8. *Consider the Sharpened-BFGS method in Algorithm 2 applied to the objective function f satisfying Assumptions 3.1-3.2. Suppose the initial point x_0 satisfies*

$$\lambda_0 \leq \frac{C_1 \mu}{dML}, \tag{26}$$

where $C_1 = \frac{\ln 2}{20}$. Then, $\forall t \geq 1$, we have

$$\lambda_t \leq 2\left(1-\frac{\mu}{2dL}\right)^{\frac{t(t-1)}{4}}\left(\frac{8dL}{t\mu}\right)^{\frac{t}{2}}\lambda_0. \tag{27}$$

Proof. Check Appendix G. □

We observe that the superlinear convergence rate of Theorem 3.8 is very similar to the result of Theorem 3.4. As we discussed in the last paragraph of Section 3.1, we can summarize the Theorem 3.6 and 3.8 into one convergence result. Hence, the iteration generated by the Sharpened-BFGS method applied to the unconstrained optimization problem as specified in Algorithm 2 satisfies the following local convergence rate. When the iteration number $t \leq \Theta(d\frac{L}{\mu})$, the linear convergence rate in (21) holds. When $t \geq \Theta(d\frac{L}{\mu})$, we can reach the superlinear convergence rate in (27).

4. Discussions

In this section, we compare the convergence results of Sharpened-BFGS with the ones for Greedy-BFGS and standard BFGS. We specifically focus on the case that the objective function satisfies Assumption 3.1 and 3.2. To simplify the comparisons, we replace all the universal constants with 1 in the convergence results and only compare the parameters μ, L, M and d defined in Assumption 3.1 and 3.2. We denote the condition number by $\kappa = L/\mu \geq 1$.

Sharpened-BFGS. According to our result, if we set $G_0 = LI$ and the initial point x_0 satisfies

$$\lambda_f(x_0) = \mathcal{O}\left(\frac{1}{dM\kappa}\right),$$

then the iterates generated by Sharpened-BFGS satisfy:

$$\frac{\lambda_f(x_t)}{\lambda_f(x_0)} \leq \min\left\{\left(1-\frac{1}{\kappa}\right)^t, \left(1-\frac{1}{d\kappa}\right)^{\frac{t(t-1)}{4}}\left(\frac{d\kappa}{t}\right)^{\frac{t}{2}}\right\}.$$

Hence, for $t < d\kappa$, the first upper bound is smaller and the Newton decrement converges at a linear rate of $(1-\frac{1}{\kappa})^t$, and for $t \geq d\kappa$ the second term becomes smaller and we observe a superlinear rate of $(1-\frac{1}{d\kappa})^{\frac{t(t-1)}{4}}(\frac{d\kappa}{t})^{\frac{t}{2}}$, which is faster than quadratic rate.

Greedy-BFGS. Next, we present the convergence result for Greedy-BFGS in (Rodomanov & Nesterov, 2021a). If we set $G_0 = LI$ and the initial point x_0 satisfies

$$\lambda_f(x_0) = \mathcal{O}\left(\frac{1}{dM\kappa}\right),$$

then the iterates of Greedy-BFGS satisfy:

$$\frac{\lambda_f(x_t)}{\lambda_f(x_0)} \leq \min\left\{\left(1-\frac{1}{\kappa}\right)^t, \left(1-\frac{1}{d\kappa}\right)^{\frac{t(t-1)}{2}}\left(\frac{1}{2}\right)^t\right\}.$$

We observe that the superlinear convergence appears after $d\kappa\ln(d\kappa)$ iterations for Greedy-BFGS, while for Sharpened-BFGS it takes $d\kappa$ steps to reach the superlinear convergence. Hence, Sharpened-BFGS achieves the superlinear rate with fewer iterations compared to Greedy-BFGS. Moreover, eventually the superlinear convergence rate of Sharpened-BFGS is faster than the one for Greedy-BFGS. This is because both of the methods achieve a quadratic convergence rate of the form $(1-\frac{1}{d\kappa})^{t^2}$. However, when t is sufficiently large, we have $(\frac{d\kappa}{t})^{\frac{t}{2}} \ll (\frac{1}{2})^t$.

BFGS. Now, we present the convergence result for BFGS provided in (Rodomanov & Nesterov, 2021c). If we set $G_0 = LI$ and the initial point x_0 satisfies

$$\lambda_f(x_0) = \max\left\{\mathcal{O}\left(\frac{1}{M\kappa}\right), \mathcal{O}\left(\frac{1}{Md\ln\kappa}\right)\right\},$$

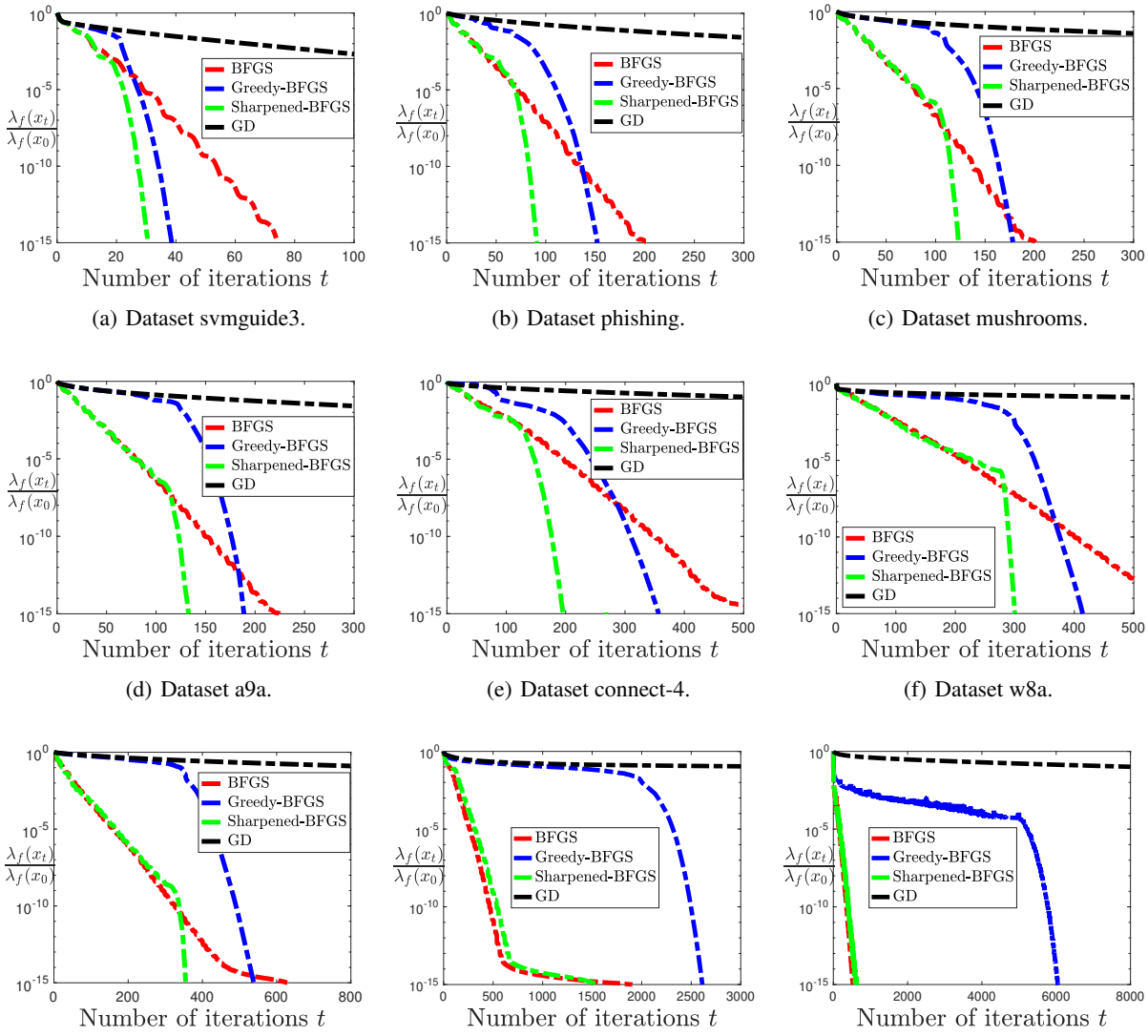
then the iterates of BFGS satisfy:

$$\frac{\lambda_f(x_t)}{\lambda_f(x_0)} \leq \min\left\{\left(1-\frac{1}{\kappa}\right)^t, \left(\frac{d\ln\kappa}{t}\right)^{\frac{t}{2}}\right\}.$$

We observe that the superlinear convergence of BFGS starts after $d\ln\kappa$ steps, while it takes $d\kappa$ iterations for the appearance of the superlinear convergence of Sharpened-BFGS. However, the superlinear convergence rate of Sharpened-BFGS is faster than BFGS as for large t we have

$$\left(1-\frac{1}{d\kappa}\right)^{\frac{t(t-1)}{4}}\left(\frac{d\kappa}{t}\right)^{\frac{t}{2}} \ll \left(\frac{d\ln\kappa}{t}\right)^{\frac{t}{2}}.$$

For the broad strokes, see the quantitative comparisons summarized in Table 1.



(a) Dataset svmguide3. (b) Dataset phishing. (c) Dataset mushrooms.

(d) Dataset a9a. (e) Dataset connect-4. (f) Dataset w8a.

(g) Dataset protein. (h) Dataset colon-cancer. (i) Dataset gisette.

Figure 2. Comparison of BFGS, Greedy-BFGS, Sharpened-BFGS and gradient descent (GD) on different datasets.

5. Numerical Experiments

In this section, we present our numerical experiments on different datasets to compare the performance of Sharpened-BFGS with BFGS and Greedy-BFGS. We focus on the following logistic regression problem with l_2 regularization

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{N}\sum_{i=1}^{N} \ln\left(1+e^{-y_i z_i^\top x}\right) + \frac{\mu}{2}\|x\|^2, \quad (28)$$

where $\{z_i\}_{i=1}^N$ are the data points and $\{y_i\}_{i=1}^N$ are their corresponding labels. We assume that $z_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ for all $1 \leq i \leq N$. This objective function is strongly convex with parameter $\mu > 0$. We normalize all data points such that $\|z_i\| = 1$ for all $1 \leq i \leq N$. Hence, the gradient of the function $f(x)$ is smooth with parameter $L = 1/4 + \mu$. The logistic regression objective function is also strongly self-concordant; check Section 5.1 in (Rodomanov & Nesterov, 2021a). Therefore, the objective function $f(x)$ defined in (28) satisfies Assumptions 3.1-3.2.

We conduct our experiments on eight datasets. All the parameters (sample size N, dimension d and regularization parameter μ) of these different datasets are summarized in Table 2. The regularization parameter μ is chosen from the set $\mathcal{A} = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ to achieve the best performance. The algorithms that we study are (i) Sharpened-BFGS, (ii) standard BFGS, (iii) Greedy-BFGS, and (iv) gradient descent (GD). We initialize all the algorithms with the same initial point $x_0 = (1/d^{3/2}) * \vec{1}$, where

Dataset	N	d	μ
svmguide3	1243	21	0.01
phishing	11055	68	0.001
mushrooms	8124	112	0.001
a9a	32561	123	0.001
connect-4	67557	126	0.0001
w8a	49749	300	0.0001
protein	17766	357	0.0001
colon-cancer	62	2000	0.00001
gisette	6000	5000	0.00001

Table 2. Sample size N, dimension d, regularization parameter μ.

$\vec{\mathbf{1}} \in \mathbb{R}^d$ is the one vector. We set the initial Hessian approximation matrix as LI and set the stepsize to 1 for all QN methods. The step size of gradient descent is set as $1/L$ to achieve its linear convergence rate on each dataset. In practice, we found it is better not to apply the correction strategy in the hybrid and greedy methods (i.e., simply set $\hat{G}_t = \bar{G}_t$ in step 7 of Algorithm 2). The convergence rates of the ratio $\lambda_f(x_t)/\lambda_f(x_0)$ versus the number of iterations t are presented in Figure 2.

We observe that Sharpened-BFGS outperforms both the classical and greedy BFGS methods. More specifically, in the initial phase of the convergence process, Sharpened-BFGS exploits the Newton direction approximation of BFGS and has a fast convergence like BFGS, while Greedy-BFGS has a slower convergence at the beginning as it Hessian approximation is not accurate yet. Then, once the time index increases and almost approaches d, and d greedy updates are accomplished, the Hessian approximation of Greedy-BFGS becomes more accurate. As a result, Greedy-BFGS achieves a very fast convergence rate at this turning point. Similarly, Sharpened-BFGS follows the same fast convergence of Greedy-BFGS almost at the same time index, as it also exploits the Hessian approximation update rule in Greedy-BFGS. This behavior is consistent over all the considered datasets in our experiments, as illustrated in Figure 2. We should also add that these empirical observations are consistent with our theoretical findings and the performance comparisons of these algorithms in Section 4.

6. Conclusions

In this paper, we proposed a novel quasi-Newton method called Sharpened-BFGS for solving unconstrained convex optimization problems, where the objective function is strongly convex with μ, its gradient is smooth with L, and it is strongly self-concordant with M. Sharpened-BFGS benefits from the Newton direction approximation of BFGS as well as Hessian approximation of Greedy-BFGS. Using these properties, we proved that the proposed Sharpened-BFGS achieves a superlinear convergence rate of $\mathcal{O}((1-\frac{\mu}{dL})^{\frac{t(t-1)}{4}}(\frac{dL}{t\mu})^{\frac{t}{2}})$, which is faster than quadratic rate. We also compared the convergence results of our method with the classical BFGS and Greedy-BFGS methods and highlighted how Sharpened-BFGS takes advantage of the Newton direction approximation in BFGS and the Hessian approximation in Greedy-BFGS. We also numerically illustrated the advantages of our proposed method against BFGS and Greedy-BFGS.

Acknowledgement

This research of Q. Jin and A. Mokhtari is supported in part by NSF Grants 2007668, 2019844, and 2112471, ARO Grant W911NF2110226, the Machine Learning Lab (MLL) at UT Austin, and the Wireless Networking and Communications Group (WNCG) Industrial Affiliates Program.

References

Bennett, A. A. Newton's method in general analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 2(10):592, 1916.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

Broyden, C. G. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19 (92):577–593, 1965.

Broyden, C. G. The convergence of single-rank quasi-Newton methods. *Mathematics of Computation*, 24(110): 365–382, 1970.

Broyden, C. G., Jr., J. E. D., Broyden, and More, J. J. On the local and superlinear convergence of quasi-Newton methods. *IMA J. Appl. Math*, 12(3):223–245, June 1973.

Conn, A. R., Gould, N. I. M., and Toint, P. L. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical programming*, 50(1-3): 177–195, 1991.

Conn, A. R., Gould, N. I., and Toint, P. L. *Trust region methods*, volume 1. Siam, 2000.

Davidon, W. Variable metric method for minimization. Technical report, Argonne National Lab., Lemont, Ill., 1959.

Fletcher, R. A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322, 1970.

Fletcher, R. and Powell, M. J. A rapidly convergent descent method for minimization. *The computer journal*, 6(2): 163–168, 1963.

Gay, D. M. Some convergence properties of Broyden's method. *SIAM Journal on Numerical Analysis*, 16(4): 623–630, 1979.

Goldfarb, D. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24 (109):23–26, 1970.

Jin, Q. and Mokhtari, A. Non-asymptotic superlinear convergence of standard quasi-newton methods. *arXiv preprint arXiv:2003.13607*, 2020.

Lin, D., Ye, H., and Zhang, Z. Explicit superlinear convergence of broyden's method in nonlinear equations. *arXiv preprint arXiv:2109.01974*, 2021a.

Lin, D., Ye, H., and Zhang, Z. Greedy and random quasi-newton methods with faster explicit superlinear convergence. *Advances in Neural Information Processing Systems 34*, 2021b.

Liu, D. C. and Nocedal, J. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Nesterov, J. E. Self-concordant functions and polynomial-time methods in convex programming. *Report, Central Economic and Mathematic Institute, USSR Acad. Sci*, 1989.

Nesterov, Y. and Nemirovskii, A. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Nesterov, Y. and Polyak, B. T. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Nocedal, J. Updating quasi-Newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Ortega, J. M. and Rheinboldt, W. C. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.

Rodomanov, A. and Nesterov, Y. Greedy quasi-newton methods with explicit superlinear convergence. *SIAM Journal on Optimization*, 31(1):785–811, 2021a.

Rodomanov, A. and Nesterov, Y. Rates of superlinear convergence for classical quasi-newton methods. *Mathematical Programming*, pp. 1–32, 2021b.

Rodomanov, A. and Nesterov, Y. New results on superlinear convergence of classical quasi-newton methods. *Journal of Optimization Theory and Applications*, 188(3):744–769, 2021c.

Shanno, D. F. Conditioning of quasi-Newton methods for function minimization. *Mathematics of computation*, 24 (111):647–656, 1970.

Ye, H., Lin, D., Zhang, Z., and Chang, X. Explicit superlinear convergence rates of the sr1 algorithm. *arXiv preprintarXiv:2105.07162*, 2021.

Appendix

A. Preliminary Lemmas

In this subsection, we develop some technical preliminaries which are critical in the path towards establishing our main convergence results. We begin with the following lemma regarding the BFGS operator defined in Sec. 2.1

Lemma A.1. *Consider positive definite matrices $A, G \in \mathbb{R}^{d\times d}$ and suppose that $G_+ = BFGS(A, G, u)$ as defined in (3) for any $u \in \mathbb{R}^d\backslash\{0\}$. Then, the following results hold:*

1. *For any constants $\xi, \eta \geq 1$, we have*

$$\frac{1}{\xi}A \preceq G \preceq \eta A \quad \Rightarrow \quad \frac{1}{\xi}A \preceq G_+ \preceq \eta A. \tag{29}$$

2. *If $A \preceq G$, then we have*

$$\sigma(A, G) - \sigma(A, G_+) \geq \theta^2(A, G, u). \tag{30}$$

3. *If $\frac{1}{\xi}A \preceq G$ and $\theta(A, G, u) \leq \xi$, where $\xi \geq 1$ is a constant, then*

$$\sigma(A, G) - \sigma(A, G_+) \geq \frac{1}{4\xi^2}\theta^2(A, G, u) - \ln \xi. \tag{31}$$

Proof. Check Lemma 2.1 in (Rodomanov & Nesterov, 2021b) for the proof of (29) and check Lemma 2.2 in (Rodomanov & Nesterov, 2021b) for the proof of (30). Now we prove (31). We denote $Det(A)$ as the determinant of the matrix $A \in \mathbb{R}^{d\times d}$. Applying results of Lemma 2.4 in (Rodomanov & Nesterov, 2021b), we obtain that

$$\psi(A, G) - \psi(A, G_+) \geq \omega(\frac{\theta(A, G, u)}{\xi}), \tag{32}$$

where

$$\psi(A, G) := Tr(A^{-1}(G - A)) - \ln Det(A^{-1}G) = \sigma(A, G) - \ln Det(A^{-1}G), \tag{33}$$

and

$$\omega(t) := t - \ln(1 + t), \quad \forall t \geq -1. \tag{34}$$

Thus, we obtain that

$$\begin{aligned}\psi(A, G) - \psi(A, G_+) &= \sigma(A, G) - \sigma(A, G_+) - \ln Det(A^{-1}G) + \ln Det(A^{-1}G_+)\\ &= \sigma(A, G) - \sigma(A, G_+) + \ln Det(G^{-1}G_+).\end{aligned} \tag{35}$$

From Lemma 6.2 of (Rodomanov & Nesterov, 2021b), we have that

$$Det(G^{-1}G_+) = \frac{u^\top Au}{u^\top Gu}. \tag{36}$$

Hence, we get that

$$\psi(A, G) - \psi(A, G_+) = \sigma(A, G) - \sigma(A, G_+) + \ln \frac{u^\top Au}{u^\top Gu}. \tag{37}$$

Substituting (37) into the (32), we obtain that

$$\sigma(A, G) - \sigma(A, G_+) \geq \omega(\frac{\theta(A, G, u)}{\xi}) - \ln \frac{u^\top Au}{u^\top Gu}. \tag{38}$$

Notice that the function $\omega(t)$ satisfies the following property,

$$\omega(t) \geq \frac{t^2}{2(t + 1)}, \quad \forall t \geq 0. \tag{39}$$

Thus, we derive that

$$\omega(\frac{\theta(A,G,u)}{\xi}) \geq \frac{\theta^2(A,G,u)/\xi^2}{2(\theta(A,G,u)/\xi+1)} = \frac{\theta^2(A,G,u)}{2\xi(\theta(A,G,u)+\xi)} \geq \frac{\theta^2(A,G,u)}{4\xi^2}, \tag{40}$$

where the second inequality is due to the condition $\theta(A,G,u) \leq \xi$. From condition $\frac{1}{\xi}A \preceq G$, we know that for any $u \in \mathbb{R}^d \backslash \{0\}$

$$\frac{u^\top A u}{u^\top G u} \leq \xi. \tag{41}$$

Substituting (40) and (41) into (38), we achieve conclusion (31).

□

In the following lemma, we show that the Hessians of the strongly self-concordant function at two different points.

Lemma A.2. *Suppose the objective function $f(x)$ is strongly self-concordant with constant $M > 0$. Consider $x, y \in \mathbb{R}^d$, $r = \|y - x\|_x$ and $J = \int_0^1 \nabla^2 f(x + \tau(y-x))d\tau$. Then, we have that*

$$\frac{\nabla^2 f(x)}{1+Mr} \preceq \nabla^2 f(y) \preceq (1+Mr)\nabla^2 f(x). \tag{42}$$

$$\frac{\nabla^2 f(x)}{1+\frac{Mr}{2}} \preceq J \preceq (1+\frac{Mr}{2})\nabla^2 f(x). \tag{43}$$

$$\frac{\nabla^2 f(y)}{1+\frac{Mr}{2}} \preceq J \preceq (1+\frac{Mr}{2})\nabla^2 f(y). \tag{44}$$

Proof. Check Lemma 4.2 in (Rodomanov & Nesterov, 2021a). □

Lemma A.3. *Suppose the objective function $f(x)$ satisfies the Assumption 3.1 and 3.2. Consider the following update*

$$x_{t+1} = x_t - G_t^{-1}\nabla f(x_t), \tag{45}$$

where $G_t \in \mathbb{R}^{d\times d}$ is the s.p.d. Hessian approximation matrix satisfying that

$$\nabla^2 f(x_t) \preceq G_t \preceq \eta \nabla^2 f(x_t), \tag{46}$$

where $\eta \geq 1$ is some constant. Suppose the following condition holds

$$M\lambda_t \leq 2. \tag{47}$$

Denote that $r_t = \|x_{t+1} - x_t\|_{x_t}$ and $J_t = \int_0^1 \nabla^2 f(x_t + \tau(x_{t+1} - x_t))d\tau$. Then, we have

$$r_t \leq \lambda_t, \tag{48}$$

$$\theta(J_t, G_t, x_{t+1} - x_t) \leq \frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta}. \tag{49}$$

Proof. From (45), we have that

$$\begin{aligned} r_t = \|x_{t+1} - x_t\|_{x_t} &= \left(\nabla f(x_t)^\top G_t^{-1}\nabla^2 f(x_t) G_t^{-1}\nabla f(x_t)\right)^{\frac{1}{2}} \\ &\leq \left(\nabla f(x_t)^\top G_t^{-1}\nabla f(x_t)\right)^{\frac{1}{2}} \leq \left(\nabla f(x_t)^\top \nabla^2 f(x_t)^{-1}\nabla f(x_t)\right)^{\frac{1}{2}} = \lambda_t, \end{aligned} \tag{50}$$

where the inequalities hold due to (46). Therefore, (48) holds. Now we condition (49). Using (46) and (43) of Lemma A.2, we obtain that

$$\frac{1}{1+\frac{Mr_t}{2}} J_t \preceq \nabla^2 f(x_t) \preceq G_t \preceq \eta \nabla^2 f(x_t) \preceq \eta(1+\frac{Mr_t}{2})J_t. \tag{51}$$

Using $r_t \leq \lambda_t$ from (48), we get that

$$\frac{1}{1+\frac{M\lambda_t}{2}} J_t \preceq G_t \preceq \eta(1+\frac{M\lambda_t}{2})J_t. \tag{52}$$

Hence, we have

$$-\left(1-\frac{1}{\eta(1+\frac{M\lambda_t}{2})}\right) J_t^{-1} \preceq G_t^{-1} - J_t^{-1} \preceq \frac{M\lambda_t}{2} J_t^{-1}. \tag{53}$$

Notice that

$$\left(1-\frac{1}{\eta(1+\frac{M\lambda_t}{2})}\right) \leq 1 - \frac{1-\frac{M\lambda_t}{2}}{\eta} = \frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta}. \tag{54}$$

Since $M\lambda_t \leq 2$ and $\eta \geq 1$, we have

$$\frac{M\lambda_t}{2} = 1 - (1 - \frac{M\lambda_t}{2}) \leq 1 - \frac{1-\frac{M\lambda_t}{2}}{\eta} = \frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta}. \tag{55}$$

Therefore, we have

$$-\frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta} J_t^{-1} \preceq G_t^{-1} - J_t^{-1} \preceq \frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta} J_t^{-1}. \tag{56}$$

Hence, we get

$$(G_t^{-1} - J_t^{-1}) J_t (G_t^{-1} - J_t^{-1}) \preceq \left(\frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta}\right)^2 J_t^{-1}, \tag{57}$$

$$s_t^\top G_t (G_t^{-1} - J_t^{-1}) J_t (G_t^{-1} - J_t^{-1}) G_t s_t \leq \left(\frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta}\right)^2 s_t^\top G_t J_t^{-1} G_t s_t, \tag{58}$$

where $s_t = x_{t+1} - x_t$ is the variable difference. Therefore, by the definition of θ in (13), we prove conclusion (49),

$$\begin{aligned} \theta(J_t, G_t, x_{t+1} - x_t) &= \left(\frac{s_t^\top (G_t - J_t) J_t^{-1} (G_t - J_t) s_t}{s_t^\top G_t J_t^{-1} G_t s_t}\right)^{\frac{1}{2}} \\ &= \left(\frac{s_t^\top G_t (J_t^{-1} - G_t^{-1}) J_t (J_t^{-1} - G_t^{-1}) G_t s_t}{s_t^\top G_t J_t^{-1} G_t s_t}\right)^{\frac{1}{2}} \leq \frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta}. \end{aligned} \tag{59}$$

□

B. Proof of Theorem 3.2

First, we use induction to prove the following condition

$$A \preceq G_t \preceq \frac{L}{\mu} A, \quad \forall t \geq 0. \tag{60}$$

From $\mu I \preceq A \preceq LI$, we observe that the initial Hessian approximation matrix $G_0 = LI$ satisfies $A \preceq G_0 \preceq \frac{L}{\mu} A$. Hence, condition (60) holds for $t = 0$. We assume that condition (60) holds for $t = k$, i.e., $A \preceq G_k \preceq \frac{L}{\mu} A$, where $k \geq 0$. Applying (29) of Lemma A.1 to the update in step 4 of Algorithm 1, we obtain that $A \preceq \bar{G}_k \preceq \frac{L}{\mu} A$. Applying (29) of Lemma A.1 again to the update in step 6 of Algorithm 1, we obtain that $A \preceq G_{k+1} \preceq \frac{L}{\mu} A$. Therefore, condition (60) holds for $t = k+1$. By induction, we prove that condition (60) holds for any $t \geq 0$. Moreover, this condition implies that for any $t \geq 0$, we have

$$0 \preceq A^{-1} - G_t^{-1} \preceq (1 - \frac{\mu}{L}) A^{-1}. \tag{61}$$

Hence, we obtain that

$$(G_t - A) A^{-1} (G_t - A) = G_t (A^{-1} - G_t^{-1}) A (A^{-1} - G_t^{-1}) G_t \preceq (1 - \frac{\mu}{L})^2 G_t A^{-1} G_t, \tag{62}$$

$$s_t^\top (G_t - A)A^{-1}(G_t - A)s_t \leq (1 - \frac{\mu}{L})^2 s_t G_t A^{-1} G_t s_t, \tag{63}$$

where $s_t = x_{t+1} - x_t$ is the variable difference. By the definition of θ in (13), we have that

$$\theta(A, G_t, x_{t+1} - x_t) = \left(\frac{s_t^\top (G_t - A)A^{-1}(G_t - A)s_t}{s_t G_t A^{-1} G_t s_t} \right)^{\frac{1}{2}} \leq 1 - \frac{\mu}{L}. \tag{64}$$

Therefore, (14) holds for any $t \geq 0$. Applying (12) of Lemma 3.1, we prove that

$$\lambda_{t+1} = \theta(A, G_t, x_{t+1} - x_t)\lambda_t \leq (1 - \frac{\mu}{L})\lambda_t, \qquad \forall t \geq 0. \tag{65}$$

Hence, we prove the linear convergence rate of (15). □

C. Proof of Lemma 3.3

The initial Hessian approximation matrix $G_0 = LI \succeq A$. Applying the same induction technique used in the proof of Theorem 3.2, we can prove that for any $t \geq 0$

$$G_t \succeq A, \qquad \bar{G}_t \succeq A, \tag{66}$$

where $\bar{G}_t$ is defined in step 4 of Algorithm 1. Using (30) of Lemma A.1, we have that

$$\sigma(A, G_t) - \sigma(A, \bar{G}_t) \geq \theta^2(A, G_t, x_{t+1} - x_t), \quad \forall t \geq 0. \tag{67}$$

Applying (10) of Lemma 2.2 to the step 6 of Algorithm 1, we obtain that

$$\sigma(A, G_{t+1}) \leq (1 - \frac{\mu}{dL})\sigma(A, \bar{G}_t), \quad \forall t \geq 0. \tag{68}$$

We prove conclusion (16) by combining and regrouping the above two inequalities. Now we prove condition (17). Recall and define the following shorthanded notations

$$\theta_t = \theta(A, G_t, x_{t+1} - x_t), \qquad \sigma_t = \sigma(A, G_t), \qquad c = \frac{\mu}{dL}. \tag{69}$$

Condition (16) is equivalent to

$$\sigma_t \leq (1 - c)\sigma_{t-1} - (1 - c)\theta_{t-1}^2, \quad \forall t \geq 1. \tag{70}$$

Applying the above inequality recursively, we can derive that

$$\begin{aligned} \sigma_t &\leq (1 - c)\sigma_{t-1} - (1 - c)\theta_{t-1}^2 \\ &\leq (1 - c)^2 \sigma_{t-2} - (1 - c)^2 \theta_{t-2}^2 - (1 - c)\theta_{t-1}^2 \\ &\leq (1 - c)^t \sigma_0 - \sum_{i=0}^{t-1} (1 - c)^{t-i} \theta_i^2. \end{aligned} \tag{71}$$

The above inequality indicates that

$$\sum_{i=0}^{t-1} (1 - c)^{t-i} \theta_i^2 \leq (1 - c)^t \sigma_0 - \sigma_t \leq (1 - c)^t \sigma_0. \tag{72}$$

Dividing the term $(1 - c)^t$ on both sides of the above inequality, we can obtain that

$$\sum_{i=0}^{t-1} \frac{\theta_i^2}{(1 - c)^i} \leq \sigma_0. \tag{73}$$

Hence, we prove the result (17) since $c = \frac{\mu}{dL}$. □

D. Proof of Theorem 3.4

Using the condition $A^{-1} \preceq \frac{1}{\mu} I$ and recalling the notation $c = \frac{\mu}{dL}$, we can upper bound σ_0 by

$$\sigma_0 = \sigma(A, G_0) = Tr(A^{-1}LI) - d \leq Tr(\frac{L}{\mu} I) - d = d(\frac{L}{\mu} - 1) \leq d\frac{L}{\mu} = \frac{1}{c}. \tag{74}$$

Combining the above upper bound and (17), we derive that

$$\sum_{i=0}^{t-1} \frac{\theta_i^2}{(1-c)^i} \leq \sigma_0 \leq \frac{1}{c}. \tag{75}$$

From (12) of Lemma 3.1, we obtain that

$$\frac{\lambda_t}{\lambda_0} = \prod_{i=0}^{t-1} \frac{\lambda_{i+1}}{\lambda_i} = \prod_{i=0}^{t-1} \theta_i = \prod_{i=0}^{t-1} (1-c)^{\frac{i}{2}} \frac{\theta_i}{(1-c)^{\frac{i}{2}}} = \prod_{i=0}^{t-1} (1-c)^{\frac{i}{2}} \prod_{i=0}^{t-1} \frac{\theta_i}{(1-c)^{\frac{i}{2}}} = (1-c)^{\frac{t(t-1)}{4}} \prod_{i=0}^{t-1} \frac{\theta_i}{(1-c)^{\frac{i}{2}}}. \tag{76}$$

Using the arithmetic-geometric mean inequality and (75), we derive that

$$\prod_{i=0}^{t-1} \frac{\theta_i}{(1-c)^{\frac{i}{2}}} = \left[\prod_{i=0}^{t-1} \frac{\theta_i^2}{(1-c)^i}\right]^{\frac{1}{2}} \leq \left[\frac{1}{t} \sum_{i=0}^{t-1} \frac{\theta_i^2}{(1-c)^i}\right]^{\frac{t}{2}} \leq \left(\frac{1}{ct}\right)^{\frac{t}{2}}. \tag{77}$$

Leveraging (76) and (77), we achieve the final convergence rate of (18)

$$\frac{\lambda_t}{\lambda_0} \leq (1-c)^{\frac{t(t-1)}{4}} \left(\frac{1}{ct}\right)^{\frac{t}{2}} = (1 - \frac{\mu}{dL})^{\frac{t(t-1)}{4}} (\frac{dL}{t\mu})^{\frac{t}{2}}, \qquad \forall t \geq 1. \tag{78}$$

□

E. Proof of Theorem 3.6

First, we use induction to prove the following condition

$$\nabla^2 f(x_t) \preceq G_t \preceq \xi_t \frac{L}{\mu} \nabla^2 f(x_t), \quad \forall t \geq 0, \tag{79}$$

where

$$\xi_0 = 1 \quad \text{and} \quad \xi_t = e^{2M \sum_{i=0}^{t-1} r_i}, \quad \forall t \geq 1. \tag{80}$$

We use induction to prove (79) and (80). When $t = 0$, from Assumption 3.1 we know that

$$\nabla^2 f(x_0) \preceq G_0 = LI \preceq \frac{L}{\mu} \nabla^2 f(x_0). \tag{81}$$

Hence, (79) and (80) hold for $t = 0$. Suppose that (79) and (80) hold for $t = k$, we have that

$$\nabla^2 f(x_k) \preceq G_k \preceq \xi_k \frac{L}{\mu} \nabla^2 f(x_k), \quad \xi_k = e^{2M \sum_{i=0}^{k-1} r_i}. \tag{82}$$

Now we consider the case of $t = k + 1$. Condition (43) of Lemma A.2 indicates that

$$\frac{1}{1 + \frac{Mr_k}{2}} J_k \preceq \nabla^2 f(x_k) \preceq G_k \preceq \xi_k \frac{L}{\mu} \nabla^2 f(x_k) \preceq \xi_k \frac{L}{\mu} (1 + \frac{Mr_k}{2}) J_k. \tag{83}$$

where $J_k = \int_0^1 \nabla^2 f(x_k + \tau(x_{k+1} - x_k)) d\tau$. Applying (29) of Lemma A.1, we have that

$$\frac{1}{1 + \frac{Mr_k}{2}} J_k \preceq \bar{G}_k = BFGS(J_k, G_k, x_{k+1} - x_k) \preceq \xi_k \frac{L}{\mu} (1 + \frac{Mr_k}{2}) J_k, \tag{84}$$

where the equality is due to step 5 of Algorithm 2. Condition (44) of Lemma A.2 indicates that

$$\frac{1}{(1+\frac{Mr_k}{2})^2}\nabla^2 f(x_{k+1}) \preceq \frac{1}{1+\frac{Mr_k}{2}}J_k \preceq \bar{G}_k \preceq \xi_k\frac{L}{\mu}(1+\frac{Mr_k}{2})J_k, \preceq \xi_k\frac{L}{\mu}(1+\frac{Mr_k}{2})^2\nabla^2 f(x_{k+1}). \tag{85}$$

Multiplying the term $(1+\frac{Mr_k}{2})^2$ on both sides of the above inequality, we get that

$$\nabla^2 f(x_{k+1}) \preceq (1+\frac{Mr_k}{2})^2\bar{G}_k = \hat{G}_k \preceq \xi_k\frac{L}{\mu}(1+\frac{Mr_k}{2})^4\nabla^2 f(x_{k+1}), \tag{86}$$

where the equality is due to step 7 of Algorithm 2. Applying the fact $1+x \leq e^x$, we have

$$\xi_k(1+\frac{Mr_k}{2})^4 \leq \xi_k e^{2Mr_k} = e^{2M\sum_{i=0}^{k-1}r_i}e^{2Mr_k} = e^{2M\sum_{i=0}^{k}r_i} = \xi_{k+1}, \tag{87}$$

where the first equality is due to the induction assumption in (82) and the last equality is due to the definition in (80). Substituting (87) into (86), we have that

$$\nabla^2 f(x_{k+1}) \preceq \hat{G}_k \preceq \xi_{k+1}\frac{L}{\mu}\nabla^2 f(x_{k+1}). \tag{88}$$

Applying (29) of Lemma A.1 again and step 9 of Algorithm 2, we obtain that

$$\nabla^2 f(x_{k+1}) \preceq G_{k+1} = BFGS(\nabla^2 f(x_{k+1}), \hat{G}_k, \bar{u}(\nabla^2 f(x_{k+1}), \hat{G}_k)) \preceq \xi_{k+1}\frac{L}{\mu}\nabla^2 f(x_{k+1}). \tag{89}$$

Hence, (79) and (80) hold for $t = k+1$. Therefore, We finish the proof of (79) and (80) using induction.

Now, we use induction again to prove the result of (21) and (22). It's obvious that (22) holds for $t = 0$. Suppose that (22) holds for $0 \leq t \leq k$, we have that

$$M\lambda_t \leq M\lambda_0 \leq C_0\frac{\mu}{L} = \frac{\ln\frac{3}{2}}{4}\frac{\mu}{L} < 1 < 2, \qquad 0 \leq t \leq k, \tag{90}$$

where we use the initial condition (20) and the fact $\mu \leq L$. Conditions (79) and (90) imply that (48) and (49) of Lemma A.3 hold for all $0 \leq t \leq k$ where $\eta = \xi_t L/\mu$. Hence, we have that

$$\theta(J_t, G_t, x_{t+1}-x_t) \leq \frac{\eta - 1 + \frac{M\lambda_t}{2}}{\eta} = 1 - \frac{\mu}{L\xi_t}(1-\frac{M\lambda_t}{2}), \qquad 0 \leq t \leq k. \tag{91}$$

Applying the initial condition (20) and the induction assumption of (22) for $0 \leq t \leq k$, we observe that

$$M\sum_{i=0}^{t}\lambda_i \leq M\lambda_0\sum_{i=0}^{t}(1-\frac{\mu}{2L})^i \leq 2M\frac{L}{\mu}\lambda_0 \leq 2C_0, \qquad 0 \leq t \leq k. \tag{92}$$

Consequently,

$$e^{2M\sum_{i=0}^{t}\lambda_i} \leq e^{4C_0} = e^{\ln\frac{3}{2}} = \frac{3}{2}, \qquad 0 \leq t \leq k. \tag{93}$$

Since $M\lambda_t < 1$ from (90) and the fact that $1 - x/2 \geq e^{-x}$ for $x \in (0,1)$, we get that

$$1 - \frac{M\lambda_t}{2} \geq e^{-M\lambda_t}, \qquad 0 \leq t \leq k. \tag{94}$$

Hence, we can obtain that for $0 \leq t \leq k$,

$$\begin{aligned}\frac{1}{\xi_t}(1-\frac{M\lambda_t}{2}) &= e^{-2\sum_{i=0}^{t-1}Mr_i}(1-\frac{M\lambda_t}{2}) \geq e^{-2\sum_{i=0}^{t-1}Mr_i}e^{-M\lambda_t} \\ &\geq e^{-2\sum_{i=0}^{t-1}M\lambda_i}e^{-M\lambda_t} \geq e^{-2M\sum_{i=0}^{t}\lambda_i} \geq \frac{2}{3},\end{aligned} \tag{95}$$

where the equality holds due to the definition of (80), the first inequality holds due to (94), the second inequality holds due to (48), the third inequality holds due to $M\lambda_k \geq 0$ and the last inequality holds due to (93). Substituting (95) into (91), we get that

$$\theta(J_t, G_t, x_{t+1} - x_t) \leq 1 - \frac{\mu}{L\xi_t}(1 - \frac{M\lambda_t}{2}) \leq 1 - \frac{2\mu}{3L}, \qquad 0 \leq t \leq k. \tag{96}$$

Therefore, (21) holds for $0 \leq t \leq k$. Now consider the case of $t = k+1$. From (90) for $t = k$ and the fact that $(\ln \frac{3}{2})/8 < 1/16$, we get that

$$\frac{M\lambda_k}{2} \leq \frac{\ln \frac{3}{2}}{8}\frac{\mu}{L} \leq \frac{\mu}{16L}. \tag{97}$$

From (19) of Lemma 3.5 and (48), we have that

$$\lambda_{k+1} \leq (1 + \frac{Mr_k}{2})\theta(J_k, G_k, x_{k+1} - x_k)\lambda_k \leq (1 + \frac{M\lambda_k}{2})\theta(J_k, G_k, x_{k+1} - x_k)\lambda_k. \tag{98}$$

Substituting (96) for $t = k$ and (97) into (98), we get that

$$\lambda_{k+1} \leq (1 + \frac{\mu}{16L})(1 - \frac{2\mu}{3L})\lambda_k = (1 - \frac{29\mu}{48L} - \frac{\mu^2}{24L^2})\lambda_k \leq (1 - \frac{29\mu}{48L})\lambda_k \leq (1 - \frac{\mu}{2L})\lambda_k. \tag{99}$$

Thus, condition (22) holds for $t = k+1$ since

$$\lambda_{k+1} \leq (1 - \frac{\mu}{2L})\lambda_k \leq (1 - \frac{\mu}{2L})^{k+1}\lambda_0. \tag{100}$$

Using the same technique we can prove that condition (21) holds for $t = k+1$. Therefore, we finish proving the conclusion (21) and (22) using induction. □

F. Proof of Lemma 3.7

For brevity, we use the following shorthanded notations

$$c = \frac{\mu}{2dL}, \qquad \rho_t = 1 + \frac{M\lambda_f(x_t)}{2}, \qquad \alpha_t = \sigma_t + 4Md\lambda_t, \qquad \beta_t = \rho_t^4(1 + 8Md\lambda_t). \tag{101}$$

The initial condition (23) indicates that $M\lambda_0 \leq C_0\mu/L \leq C_0 < 2$. Hence, $r_t \leq \lambda_t$ of (48) in Lemma A.3 always holds for any $t \geq 0$. Thus, we have that

$$1 + \frac{Mr_t}{2} \leq 1 + \frac{M\lambda_t}{2} = \rho_t, \qquad \forall t \geq 0. \tag{102}$$

Substituting the above inequality into (43) and (44) of Lemma A.2, we obtain that

$$\frac{\nabla^2 f(x_t)}{\rho_t} \preceq J_t \preceq \rho_t \nabla^2 f(x_t), \qquad \frac{\nabla^2 f(x_{t+1})}{\rho_t} \preceq J_t \preceq \rho_t \nabla^2 f(x_{t+1}). \tag{103}$$

From (86) of the proof of Theorem 3.6, we showed that for any $t \geq 0$, we have $\hat{G}_t \succeq \nabla^2 f(x_{t+1})$. Recall that $G_{t+1} = BFGS(\nabla^2 f(x_{t+1}), \hat{G}_t, \bar{u})$ and $\bar{u} = \bar{u}(\nabla^2 f(x_{t+1}), \hat{G}_t)$ in step 8 and 9 of Algorithm 2. Applying Lemma 2.2, we obtain that

$$\sigma_{t+1} \leq (1 - \frac{\mu}{dL})\sigma(\nabla^2 f(x_{t+1}), \hat{G}_t) \leq (1 - \frac{\mu}{2dL})\sigma(\nabla^2 f(x_{t+1}), \hat{G}_t) = (1 - c)\sigma(\nabla^2 f(x_{t+1}), \hat{G}_t). \tag{104}$$

Using the condition $\hat{G}_t = (1 + \frac{Mr_t}{2})^2\bar{G}_t$ in step 7 of Algorithm 2 and (102), we can observe that $\hat{G}_t \leq \rho_t^2\bar{G}_t$. Using this condition, (103) and the definition of σ in (7), we obtain

$$\sigma(\nabla^2 f(x_{t+1}), \hat{G}_t) = Tr(\nabla^2 f(x_{t+1})^{-1}\hat{G}_t) - d \leq \rho_t^2 Tr(\nabla^2 f(x_{t+1})^{-1}\bar{G}_t) - d \leq \rho_t^3 Tr(J_t^{-1}\bar{G}_t) - d \tag{105}$$

From (79), we know that

$$\nabla^2 f(x_t) \preceq G_t \preceq \xi_t \frac{L}{\mu}\nabla^2 f(x_t), \quad \forall t \geq 0. \tag{106}$$

Combining the above inequality and (103), we can show that,

$$\frac{1}{\rho_t} J_t \preceq G_t \preceq \xi_t \frac{L}{\mu} \rho_t J_t, \quad \forall t \geq 0. \tag{107}$$

From (21) of Theorem 3.6, we obtain that

$$\theta_t \leq 1 - \frac{2\mu}{3L} \leq 1 \leq \rho_t. \tag{108}$$

In summary, (107) shows that $G_t \succeq \frac{1}{\rho_t} J_t$ and (108) shows that $\theta_t \preceq \rho_t$. Consider (31) of Lemma A.1 and take $G = G_t$, $A = J_t$, $G_+ = BFGS(J_t, G_t, s_t) = \bar{G}_t$ in step 5 of Algorithm 2 and $\xi = \rho_t$. Applying (31) of Lemma A.1, we obtain that

$$\sigma(J_t, G_t) - \sigma(J_t, \bar{G}_t) \geq \frac{1}{4\rho_t^2} \theta_t^2 - \ln \rho_t, \tag{109}$$

which is equivalent to

$$Tr(J_t^{-1} \bar{G}_t) \leq Tr(J_t^{-1} G_t) - \frac{1}{4\rho_t^2} \theta_t^2 + \ln \rho_t, \tag{110}$$

where we use the definition of σ in (7). Substituting (110) into (105), we obtain that

$$\sigma(\nabla^2 f(x_{t+1}), \hat{G}_t) \leq \rho_t^3 \left(Tr(J_t^{-1} G_t) - \frac{1}{4\rho_t^2} \theta_t^2 + \ln \rho_t \right) - d. \tag{111}$$

Substituting (111) into (104), we have that

$$\sigma_{t+1} \leq (1-c) \left[\rho_t^3 \left(Tr(J_t^{-1} G_t) - \frac{1}{4\rho_t^2} \theta_t^2 + \ln \rho_t \right) - d \right]. \tag{112}$$

Applying (103) and the definition of σ in (7) again, we obtain that

$$Tr(J_t^{-1} G_t) \leq \rho_t Tr(\nabla^2 f(x_t)^{-1} G_t) = \rho_t (\sigma_t + d). \tag{113}$$

Substituting (113) into (112), we achieve that

$$\begin{aligned} \sigma_{t+1} &\leq (1-c) \left[\rho_t^3 \left(\rho_t(\sigma_t + d) - \frac{1}{4\rho_t^2} \theta_t^2 + \ln \rho_t \right) - d \right] \\ &= (1-c)(\rho_t^4 \sigma_t + \rho_t^4 d + \rho_t^3 \ln \rho_t - d) - \frac{1}{4}(1-c)\rho_t \theta_t^2 \\ &\leq (1-c)\rho_t^4 (\sigma_t + d + \frac{1}{\rho_t} \ln \rho_t - \frac{1}{\rho_t^4} d) - \frac{1}{4}(1-c)\theta_t^2, \end{aligned} \tag{114}$$

where the last inequality holds due to the condition $\rho_t \geq 1$. We have that

$$\begin{aligned} d + \frac{1}{\rho_t} \ln \rho_t - \frac{1}{\rho_t^4} d \leq d + \frac{d}{\rho_t} \ln \rho_t - \frac{1}{\rho_t^4} d &= \frac{\rho_t^4 + \rho_t^3 \ln \rho_t - 1}{\rho_t^4} d \leq (\rho_t^4 + \rho_t^3 \ln \rho_t - 1) d \\ &= \left[(1 + \frac{M\lambda_t}{2})^4 + (1 + \frac{M\lambda_t}{2})^3 \ln (1 + \frac{M\lambda_t}{2}) - 1 \right] d \\ &\leq (e^{2M\lambda_t} - 1 + \frac{M\lambda_t}{2} e^{\frac{3}{2} M\lambda_t}) d, \end{aligned} \tag{115}$$

where the first inequality is due to $d \geq 1$, the second inequality is due to $\rho_t \geq 1$ and the last inequality holds due to $1 + x \leq e^x$. Since the initial condition (23) holds, applying Theorem 3.6 we obtain that

$$M\lambda_t \leq M\lambda_0 \leq C_0 \frac{\mu}{L} \leq C_0 = \frac{\ln \frac{3}{2}}{4} \leq \frac{1}{8}. \tag{116}$$

Hence, (115) can be upper bounded by

$$\begin{aligned} d + \frac{1}{\rho_t}\ln\rho_t - \frac{1}{\rho_t^4}d &\le (e^{2M\lambda_t} - 1 + \frac{M\lambda_t}{2}e^{\frac{3}{2}M\lambda_t})d \\ &\le (2M\lambda_t + 4M^2\lambda_t^2 + \frac{M\lambda_t}{2}e^{\frac{3}{2}M\lambda_t})d \\ &= (2 + 4M\lambda_t + \frac{1}{2}e^{\frac{3}{2}M\lambda_t})Md\lambda_t \\ &\le (2 + \frac{1}{2} + \frac{1}{2}e^{\frac{3}{16}})Md\lambda_t \\ &\le 4Md\lambda_t, \end{aligned} \tag{117}$$

where the second inequality is due to $e^x - 1 \le x + x^2$ for $x \le \frac{1}{4}$ and the third inequality is due to (116). Substituting (117) into (114), we reach that

$$\sigma_{t+1} \le (1-c)\rho_t^4(\sigma_t + 4Md\lambda_t) - \frac{1}{4}(1-c)\theta_t^2 = (1-c)\left[(1+\frac{M\lambda_t}{2})^4(\sigma_t + 4Md\lambda_t) - \frac{1}{4}\theta_t^2\right]. \tag{118}$$

This is equivalent to the conclusion (24). Now, we move forward to prove (25). Notice that (24) is equivalent to

$$\sigma_t \le (1-c)\rho_{t-1}^4(\sigma_{t-1} + 4Md\lambda_{t-1}) - \frac{1}{4}(1-c)\theta_{t-1}^2, \quad \forall t \ge 1. \tag{119}$$

Recall the notation

$$\alpha_t = \sigma_t + 4Md\lambda_t. \tag{120}$$

Combining the above two conditions, we obtain that

$$\alpha_t \le (1-c)\rho_{t-1}^4\alpha_{t-1} - \frac{1}{4}(1-c)\theta_{t-1}^2 + 4Md\lambda_t \tag{121}$$

Notice that for any symmetric positive semi-definite matrices $A, B \in \mathbb{R}^{d\times d}$, we have

$$B \preceq Tr(A^{-1}B)A. \tag{122}$$

From (79) in the proof of Theorem 3.6, we know that $G_t \succeq \nabla^2 f(x_t)$. Taking $A = \nabla^2 f(x_t)$ and $B = G_t - \nabla^2 f(x_t)$ in the above inequality and using the definition of σ in (7), we get that

$$G_t - \nabla^2 f(x_t) \preceq Tr(\nabla^2 f(x_t)^{-1}(G_t - \nabla^2 f(x_t)))\nabla^2 f(x_t) = \sigma_t \nabla^2 f(x_t). \tag{123}$$

Hence, we obtain that

$$\nabla^2 f(x_t) \preceq G_t \preceq (1+\sigma_t)\nabla^2 f(x_t). \tag{124}$$

Applying (49) of Lemma A.3 with $\eta = 1 + \sigma_t$, we obtain that

$$\theta_t = \theta(J_t, G_t, x_{t+1} - x_t) \le \frac{\sigma_t + \frac{M\lambda_t}{2}}{1+\sigma_t} \le \sigma_t + \frac{M\lambda_t}{2} \le \sigma_t + 4Md\lambda_t, \tag{125}$$

where the second inequality is due to $\sigma_t \ge 0$ and the third inequality holds due to $d \ge 1$. Combing (19) of Lemma 3.5, (48) and the above inequality, we have that

$$\lambda_{t+1} \le (1 + \frac{Mr_t}{2})\theta_t\lambda_t \le (1 + \frac{M\lambda_t}{2})\theta_t\lambda_t \le (1 + \frac{M\lambda_t}{2})(\sigma_t + 4Md\lambda_t)\lambda_t = \rho_t\alpha_t\lambda_t, \tag{126}$$

Thus, we prove that

$$\lambda_t \le \rho_{t-1}\alpha_{t-1}\lambda_{t-1} \quad \forall t \ge 1. \tag{127}$$

Substituting (127) into (121), we have that

$$\begin{aligned} \alpha_t &\le (1-c)\rho_{t-1}^4\alpha_{t-1} + 4Md\rho_{t-1}\alpha_{t-1}\lambda_{t-1} - \frac{1}{4}(1-c)\theta_{t-1}^2 \\ &\le (1-c)\rho_{t-1}^4\alpha_{t-1} + 8(1-c)Md\rho_{t-1}^4\alpha_{t-1}\lambda_{t-1} - \frac{1}{4}(1-c)\theta_{t-1}^2 \\ &= (1-c)\rho_{t-1}^4\alpha_{t-1}(1 + 8Md\lambda_{t-1}) - \frac{1}{4}(1-c)\theta_{t-1}^2, \end{aligned} \tag{128}$$

where the second inequality is due to $\frac{1}{2} \leq 1 - \frac{\mu}{2L} = 1 - c$ and $\rho_{t-1} \geq 1$. Recall the notation $\beta_t = \rho_t^4(1 + 8Md\lambda_t)$. The above inequality can be simplified as

$$\alpha_t \leq (1-c)\beta_{t-1}\alpha_{t-1} - \frac{1}{4}(1-c)\theta_{t-1}^2. \tag{129}$$

Applying the above inequality recursively, we obtain the following result

$$\begin{aligned}
\alpha_t &\leq (1-c)\beta_{t-1}\alpha_{t-1} - \frac{1}{4}(1-c)\theta_{t-1}^2 \\
&\leq (1-c)^2\beta_{t-2}\beta_{t-1}\alpha_{t-2} - \frac{1}{4}(1-c)^2\beta_{t-1}\theta_{t-2}^2 - \frac{1}{4}(1-c)\theta_{t-1}^2 \\
&\leq (1-c)^t\alpha_0\prod_{j=0}^{t-1}\beta_j - \frac{1}{4}\sum_{i=0}^{t-1}(1-c)^{t-i}\theta_i^2\prod_{j=i+1}^{t-1}\beta_j.
\end{aligned} \tag{130}$$

Here we regulate that $\prod_{j=t}^{t-1}\beta_j$ is 1. The above inequality indicates that

$$\frac{1}{4}\sum_{i=0}^{t-1}(1-c)^{t-i}\theta_i^2\prod_{j=i+1}^{t-1}\beta_j \leq (1-c)^t\alpha_0\prod_{j=0}^{t-1}\beta_j - \alpha_t \leq (1-c)^t\alpha_0\prod_{j=0}^{t-1}\beta_j. \tag{131}$$

Since $\beta_j = \rho_j^4(1 + 6Md\lambda_j) \geq 1$ for all $j \geq 1$, we obtain that

$$\prod_{j=i+1}^{t-1}\beta_j \geq 1, \qquad 0 \leq i \leq t-1. \tag{132}$$

Applying $1 + x \leq e^x$, we obtain that

$$\beta_j = (1 + \frac{M\lambda_j}{2})^4(1 + 8Md\lambda_j) \leq e^{2M\lambda_j}e^{8Md\lambda_j} = e^{10Md\lambda_j}, \quad \forall j \geq 0. \tag{133}$$

Hence, from the linear convergence result of (22) and the initial condition (23), we observe

$$\prod_{j=0}^{t-1}\beta_j \leq \prod_{j=0}^{t-1}e^{10Md\lambda_j} = e^{10Md\sum_{j=0}^{t-1}\lambda_j} \leq e^{10Md\lambda_0\sum_{j=0}^{t-1}(1-\frac{\mu}{2L})^j} \leq e^{20Md\frac{L}{\mu}\lambda_0} \leq e^{20C_1} = e^{\ln 2} = 2. \tag{134}$$

Leveraging the results in (131), (132) and (134), we obtain that

$$\frac{1}{4}\sum_{i=0}^{t-1}(1-c)^{t-i}\theta_i^2 \leq \frac{1}{4}\sum_{i=0}^{t-1}(1-c)^{t-i}\theta_i^2\prod_{j=i+1}^{t-1}\beta_j \leq (1-c)^t\alpha_0\prod_{j=0}^{t-1}\beta_j \leq 2(1-c)^t\alpha_0. \tag{135}$$

This is equivalent to

$$\sum_{i=0}^{t-1}(1-c)^{t-i}\theta_i^2 \leq 8(1-c)^t\alpha_0. \tag{136}$$

Dividing the term $(1-c)^t$ on both sides of the above inequality, we can obtain that

$$\sum_{i=0}^{t-1}\frac{\theta_i^2}{(1-c)^i} \leq 8\alpha_0. \tag{137}$$

Hence, we prove the result (25) since $c = \frac{\mu}{2dL}$ and $\alpha_0 = \sigma_0 + 4Md\lambda_0$. □

G. Proof of Theorem 3.8

Using $G_0 = LI$, initial condition (26), the definition of σ in (7) and Assumption 3.1, we obtain

$$\sigma_0 + 4Md\lambda_0 = Tr(\nabla^2 f(x_0)^{-1} G_0) - d + 4Md\lambda_0 \leq d\frac{L}{\mu} - d + 4\frac{\ln 2}{20}\frac{\mu}{L} \leq d\frac{L}{\mu} - d + 1 \leq d\frac{L}{\mu}. \tag{138}$$

Substituting (138) into (25), we have that

$$\sum_{i=0}^{t-1} \frac{\theta_i^2}{(1-c)^i} \leq 8d\frac{L}{\mu}. \tag{139}$$

Using Lemma 3.5 and (49) of Lemma A.3 and recalling the notation $\rho_t = 1 + \frac{M\lambda_t}{2}$, we obtain that

$$\frac{\lambda_t}{\lambda_0} = \prod_{i=0}^{t-1} \frac{\lambda_{i+1}}{\lambda_i} \leq \prod_{i=0}^{t-1}(1 + \frac{Mr_i}{2})\theta_i \leq \prod_{i=0}^{t-1}(1 + \frac{M\lambda_i}{2})\theta_i = \prod_{i=0}^{t-1}\rho_i \prod_{i=0}^{t-1}\theta_i. \tag{140}$$

Applying $1 + x \leq e^x$, $d \geq 1$, the linear convergence result of (22) and the initial condition (26) again, we obtain that

$$\prod_{i=0}^{t-1}\rho_i = \prod_{i=0}^{t-1}(1 + \frac{M\lambda_i}{2}) \leq e^{\frac{M}{2}\sum_{i=0}^{t-1}\lambda_i} \leq e^{\frac{M}{2}\lambda_0 \sum_{i=0}^{t-1}(1-\frac{\mu}{2L})^i} \leq e^{\frac{M}{2}\lambda_0 \frac{2L}{\mu}} \leq e^{\frac{C_1}{d}} \leq e^{C_1} = e^{\frac{ln2}{20}} \leq e^{\ln 2} = 2. \tag{141}$$

Leveraging (140) and (141), we get that

$$\frac{\lambda_t}{\lambda_0} \leq 2\prod_{i=0}^{t-1}\theta_i = 2\prod_{i=0}^{t-1}(1-c)^{\frac{i}{2}}\frac{\theta_i}{(1-c)^{\frac{i}{2}}} = 2\prod_{i=0}^{t-1}(1-c)^{\frac{i}{2}}\prod_{i=0}^{t-1}\frac{\theta_i}{(1-c)^{\frac{i}{2}}} = 2(1-c)^{\frac{t(t-1)}{4}}\prod_{i=0}^{t-1}\frac{\theta_i}{(1-c)^{\frac{i}{2}}}. \tag{142}$$

Using the arithmetic-geometric mean inequality and (139), we obtain that

$$\prod_{i=0}^{t-1}\frac{\theta_i}{(1-c)^{\frac{i}{2}}} = \left[\prod_{i=0}^{t-1}\frac{\theta_i^2}{(1-c)^i}\right]^{\frac{1}{2}} \leq \left[\frac{1}{t}\sum_{i=0}^{t-1}\frac{\theta_i^2}{(1-c)^i}\right]^{\frac{t}{2}} \leq \left(\frac{8dL}{\mu t}\right)^{\frac{t}{2}}. \tag{143}$$

Combining (142), (143) and $c = \frac{\mu}{2dL}$, we achieve the final convergence rate of (27)

$$\frac{\lambda_t}{\lambda_0} \leq 2(1-c)^{\frac{t(t-1)}{4}}\left(\frac{8dL}{\mu t}\right)^{\frac{t}{2}} = 2(1-\frac{\mu}{2dL})^{\frac{t(t-1)}{4}}(\frac{8dL}{t\mu})^{\frac{t}{2}}, \qquad \forall t \geq 1. \tag{144}$$

□

H. Randomized Sharpened-BFGS Algorithm

In this section, we extend our analysis to the randomized version of Sharpened-BFGS method. This is enlightened by the latest work of (Ye et al., 2021), where the authors proposed the modified Greedy-BFGS method based on the Cholesky factorization of the inverse Hessian approximation matrix. They presented that instead of selecting the greedy direction defined in (9) of Lemma 2.2, we consider the following Greedy-BFGS update $G_+ = BFGS(A, G, R\bar{u}(A, R))$, where R is the upper triangular matrix satisfying $A^{-1} = R^\top R$ and $\bar{u}(A, R)$ is defined as

$$\bar{u}(A, R) := \underset{u \in \{e_i\}_{i=1}^d}{\arg\max} \frac{u^\top R^{-\top} A^{-1} R^{-1} u}{u^\top u}. \tag{145}$$

Then, the linear convergence rate of $1 - 1/(d\kappa)$ in (10) of Lemma 2.2 can be improved to $1 - 1/d$, which is independent of the condition number $\kappa = L/\mu$. However, for each unit vector e_i the computational cost of the term $e_i^\top R^{-\top} A^{-1} R^{-1} e_i$ is $\mathcal{O}(d^2)$. Hence, the cost of calculating the vector $\bar{u}(A, R)$ in (145) is $\mathcal{O}(d^3)$, which makes this modified greedy update impractical to implement. Therefore, the authors of (Ye et al., 2021) proposed to replace the greedy vector in (145) by the random vector $\tilde{u} \sim \mathcal{N}(0, I_d)$ and consider the randomized BFGS update $\tilde{G}_+ = BFGS(A, G, R\tilde{u})$, where R is still the upper triangular Cholesky factorization matrix of A^{-1}. The condition-number-free linear convergence rate of $1 - 1/d$ is preserved for this randomized algorithm. This is summarized in the following lemma.

Lemma H.1 ((Ye et al., 2021)). *Consider positive definite matrices $A, G \in \mathbb{R}^{d\times d}$ that satisfy $A \preceq G$. Suppose that $\bar{G}_+ = BFGS(A, G, R^\top \tilde{u})$ where R is the upper triangular matrix with $G^{-1} = R^\top R$ and $\tilde{u} \sim \mathcal{N}(0, I_d) \in \mathbb{R}^d$ is the random vector. Then, we have*

$$\mathbb{E}\left[\sigma(A, \bar{G}_+)\right] \leq \left(1 - \frac{1}{d}\right) \mathbb{E}\left[\sigma(A, G)\right]. \tag{146}$$

Notice that the computational cost of Cholesky decomposition is in general $\mathcal{O}(d^3)$ for a matrix with dimension d. However, the expense per iteration of randomized BFGS method could be reduced to $\mathcal{O}(d^2)$ using technique highlighted in (Ye et al., 2021). Therefore, we can improve the superlinear convergence rate of our Sharpened-BFGS algorithm by replacing the Greedy-BFGS update with the randomized BFGS method proposed in (Ye et al., 2021). Meanwhile, the computational cost per iteration of randomized Sharpened-BFGS method is still $\mathcal{O}(d^2)$. This novel randomized Sharpened-BFGS method is summarized in Algorithm 3. The local linear convergence rate presented in Theorem 3.6 still holds for this randomized Sharpened-BFGS method. In the following theorem, we directly show the explicit local superlinear convergence rate for this randomized Sharpened-BFGS algorithm.

Theorem H.2. *Consider the randomized Sharpened-BFGS quasi-Newton method in Algorithm 3 applied to the objective function satisfying Assumption 3.1 and 3.2. Suppose that the initial point x_0 satisfies that*

$$\lambda_0 \leq \frac{C_1 \mu}{dML}, \quad C_1 = \frac{\ln 2}{20}. \tag{147}$$

Then, we can reach the following local superlinear convergence rate with high probability

$$\lambda_t \leq 2(1 - \frac{1}{2d})^{\frac{t(t-1)}{4}} (\frac{8dL}{t\mu})^{\frac{t}{2}} \lambda_0, \qquad \forall t \geq 1. \tag{148}$$

Proof. Here we just present the abbreviated proof to avoid repeated details since the proof of this theorem is very similar to the proof of Lemma 3.7 and Theorem 3.8. From the theory of probability, Lemma H.1 shows that there exists a constant δ such that the inequality

$$\sigma(A, \bar{G}_+) \leq (1 - \frac{1}{d})\sigma(A, G) \tag{149}$$

holds with probability at least $1 - \delta$. Here we neglect this parameter δ to simplify the proof and denote that the above inequality holds with high probability. Then, applying the same techniques from the proof of Lemma 3.7, we can show that the following condition holds with high probability for any $t \geq 0$

$$\sigma_{t+1} \leq (1 - \frac{1}{2d}) \left[(1 + \frac{M\lambda_t}{2})^4 (\sigma_t + 4Md\lambda_t) - \frac{1}{4}\theta_t^2\right], \tag{150}$$

where $\theta_t := \theta(\nabla^2 f(x_t), G_t, x_{t+1} - x_t)$ and $\sigma_t := \sigma(\nabla^2 f(x_t), G_t)$. Moreover, we have that with high probability

$$\sum_{i=0}^{t-1} \frac{\theta_i^2}{(1 - \frac{1}{2d})^i} \leq 8(\sigma_0 + 4Md\lambda_0), \qquad \forall t \geq 1. \tag{151}$$

Finally, using the same methods from the proof of Theorem 3.8, we can prove that the suplinear convergence rate of (148) holds with high probability. □

We observe that the quadratic convergence rate term is $\mathcal{O}((1 - 1/d)^{t^2})$ in the above superlinear convergence rate in (148), which is independent of the condition number κ. This condition-number-free quadratic convergence rate is the direct consequence of the linear convergence rate of (146) from Lemma H.1.

Algorithm 3 The randomized Sharpened-BFGS method.

Require: Initial point x_0 and initial Hessian approximation matrix $G_0 = LI$.

for $t = 0, 1, 2, \ldots$ **do**
  Update the variable: $x_{t+1} = x_t - G_t^{-1} \nabla f(x_t)$;
  Compute the variable difference: $s_t = x_{t+1} - x_t$;
  Set the matrix: $J_t = \int_0^1 \nabla^2 f(x_t + \tau s_t) d\tau$;
  Compute the matrix: $\bar{G}_t = BFGS(J_t, G_t, s_t)$;
  Compute the correction term: $r_t = \|x_{t+1} - x_t\|_{x_t}$;
  Compute the matrix: $\hat{G}_t = (1 + M r_t/2)^2 \bar{G}_t$;
  Compute upper triangular matrix: R_t with $\hat{G}_t^{-1} = R_t^\top R_t$;
  Choose the random direction: $\tilde{u} \sim \mathcal{N}(0, I_d)$;
  Compute $G_{t+1} = BFGS(\nabla^2 f(x_{t+1}), \hat{G}_t, R_t^\top \tilde{u})$;
end for