WHEN STOCHASTIC MEMORYLESS BFGS MEETS NON-MONOTONE LINE SEARCH: A WAY FOR EFFICIENT OVER-PARAMETERIZED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Most existing stochastic L-BFGS methods employ constant or diminishing step-size schedules, which may limit their capability to exploit curvature information and adapt to the local geometry effectively. In this work, we propose SMBFGS, a stochastic memoryless BFGS method that incorporates the nonmonotone line search for over-parameterized learning problems. The method employs a damped quasi-Newton update to preserve the positive definiteness of the inverse Hessian approximation, thereby ensuring the descent property of resulting search directions. Its step size is then selected adaptively through the stochastic nonmonotone line search. We establish the linear convergence of SMBFGS in the general nonconvex setting under suitable interpolation assumptions. Numerical experiments on training deep neural networks demonstrate that SMBFGS outperforms representative stochastic L-BFGS methods and widely used optimizers, including Adam and AdamW, in most cases.

1 INTRODUCTION

Given a training set containing M samples, the training of a neural network can be viewed as the minimization of its empirical risk (Wright & Ma, 2022):

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

where x denotes the trainable parameters and $f_i(x)$ is the loss associated with the i-th sample. Modern over-parameterized deep models often have sufficient capacity to fit all training samples simultaneously, although over-parameterization alone does not guarantee interpolation. In the finite-sum setting, interpolation means that there exists a solution x^* that minimizes every component loss; for nonnegative losses with zero optimal value, this gives $f_i(x^*) = 0$ and $\nabla f_i(x^*) = 0$ for all i (Ma et al., 2018; Vaswani et al., 2019a). Consequently, the stochastic gradient noise vanishes at the common solution. The resulting problem is typically large-scale and nonconvex. Since full-gradient evaluations can be prohibitively expensive, practical training relies primarily on mini-batch updates, which may remain robust to stochastic gradient noise and highly nonuniform curvature (Lyu et al., 2025; Marek et al., 2026).

A broad class of mini-batch optimization methods for equation 1 can be written as

$$x_{k+1} = x_k + \eta_k d_k, \tag{2}$$

where d_k signifies the search direction and $\eta_k > 0$ represents the step size. The effectiveness of optimization methods, therefore, depends on both the quality of the direction and the choice of step sizes. Stochastic gradient methods use $d_k = -\nabla f_{B_k}(x_k)$, where $\nabla f_{B_k}(x_k)$ is a mini-batch gradient, and thus do not explicitly account for the local curvature of the objective. This limitation can be significant when the loss landscape is highly anisotropic or ill-conditioned. Quasi-Newton methods address this issue by constructing an approximation of the inverse Hessian from successive iterates and gradient differences (Wang et al., 2017). They replace the stochastic negative-gradient direction with

$$d_k = -H_k g_k, \tag{3}$$

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

where $g_k = \nabla f_{B_k}(x_k)$, and H_k is a positive definite matrix representing the inverse Hessian approximation. These methods can therefore incorporate second-order information while retaining substantially lower computational and storage costs than exact Newton-type schemes.

Extending quasi-Newton methods to stochastic optimization, however, introduces several difficulties. Firstly, gradient differences are contaminated by sampling noise and may fail to provide reliable curvature information. Secondly, positive definiteness of the quasi-Newton matrix must be preserved to obtain a descent direction. A further recurring limitation is the adjustment of step size. Theoretical choices of a fixed step size often depend on smoothness or curvature constants that are unavailable in practice, so extensive tuning is usually required. Diminishing schedules can support convergence analysis, but they decay and become overly conservative in the later stages of training. Thus, neither strategy provides direct iteration-wise adaptation to the sampled loss and the quality of the current quasi-Newton direction.

Stochastic line-search methods determine the step size using information available at the current iteration, thereby avoiding a fully prescribed schedule. A monotone rule (Vaswani et al., 2019b) typically measures sufficient decrease relative to $f_{B_k}(x_k)$, which may be restrictive because the local behavior of mini-batch objectives can vary considerably during training. A stochastic nonmonotone line search (Galli et al., 2023) relaxes this requirement by selecting the smallest integer $\ell_k \geq 0$ such that

$$\begin{aligned}
&\eta_k = \eta_{k,0}\delta^{\ell_k}, \qquad \delta \in (0,1),\\
&f_{B_k}(x_k + \eta_k d_k) \leq C_k + c\eta_k g_k^\top d_k, \qquad c \in (0,1),\\
&\widetilde{C}_k = \frac{\xi Q_k C_{k-1} + f_{B_k}(x_k)}{\xi Q_k + 1},\\
&C_k = \max\{\widetilde{C}_k,\, f_{B_k}(x_k)\}, \qquad Q_{k+1} = \xi Q_k + 1, \qquad \xi \in [0,1].
\end{aligned}$$

Here, B_k denotes the current mini-batch, $g_k = \nabla f_{B_k}(x_k)$, and $\eta_{k,0}$ is the trial step size. The reference value C_k combines the current mini-batch loss with information accumulated from previous iterations. Since $C_k \geq f_{B_k}(x_k)$, the rule permits controlled increases relative to the current sampled loss, while the term $c\eta_k g_k^\top d_k$ retains a sufficient-decrease requirement along a descent direction. This flexibility is well suited to stochastic quasi-Newton methods: useful curvature-based directions can be retained, whereas overly aggressive trial steps are reduced through backtracking.

Motivated by these considerations, we propose SMBFGS, a stochastic memoryless BFGS method incorporating a stochastic nonmonotone line search for over-parameterized learning problems. The main contributions of this work are summarized as follows.

- We introduce SMBFGS, which combines a stochastic memoryless BFGS direction with a nonmonotone line search. The method uses the same mini-batch stochastic secant pair and a damped update to obtain a positive-definite inverse Hessian approximation without storing multiple curvature pairs.
- We establish the linear convergence of SMBFGS for possibly nonconvex objectives under suitable interpolation assumptions. This result provides a theoretical justification for combining stochastic quasi-Newton directions with nonmonotone line-search globalization in the over-parameterized regime.
- We evaluate SMBFGS on several deep neural network training tasks and compare it with representative first-order, line-search, and stochastic quasi-Newton methods. The results demonstrate that SMBFGS provides stable optimization behavior and competitive training and test performance across the considered problems.

The rest of this paper is structured as follows. In Section 2, we review related work on stochastic quasi-Newton methods. Section 3 describes the proposed SMBFGS algorithm. In Section 4, we analyze the convergence of SMBFGS for nonconvex problems. Section 5 presents the numerical experiments, and Section 6 concludes the paper.

2 Related works

Existing stochastic quasi-Newton methods can be grouped according to how they control errors in stochastic gradients and curvature estimates (Mokhtari & Ribeiro, 2020).

- **Direct stochastic quasi-Newton methods.** Schraudolph et al. constructed curvature pairs from two gradients evaluated on the same sample (Schraudolph et al., 2007), whereas Bordes et al. used a diagonal quasi-Newton matrix to rescale the stochastic gradient (Bordes et al., 2009). Sohl-Dickstein et al. took a different approach by maintaining quadratic models of the component functions and minimizing their aggregate approximation (Sohl-Dickstein et al., 2014). To improve the reliability of curvature information, Byrd et al. periodically computed subsampled Hessian-vector products (Byrd et al., 2016). For nonconvex problems, Wang et al. employed a damped limited-memory BFGS update (Wang et al., 2017), while Chen et al. further incorporated regularization into the update (Chen et al., 2019). Despite these modifications, they generally rely on prescribed diminishing step sizes.
- **Variance-reduced stochastic quasi-Newton methods.** Moritz et al. combined limited-memory BFGS with a variance-reduced gradient estimator and subsampled Hessian-vector products (Moritz et al., 2016). Subsequent studies improved this framework through nonuniform mini-batch sampling (Zhao et al., 2017), momentum acceleration (Chang et al., 2019), or a recursive variance-reduced gradient estimator (Zhang et al., 2021). These methods permit fixed step sizes and provide improved convergence guarantees, but require periodic reference-gradient evaluations or additional inner-outer iterations. Moreover, variance-reduction methods have been found to offer limited practical benefits for large-scale, nonconvex deep-learning problems (Defazio & Bottou, 2019).
- **Batch management and adaptive step selection.** Berahas et al. constructed curvature pairs from the overlap between consecutive batches (Berahas et al., 2016). Their later robust version accommodates inconsistent batch information and is analyzed with both fixed and diminishing step sizes (Berahas & Takáč, 2020). Bollapragada et al. combined progressive batching with a stochastic Armijo line search (Bollapragada et al., 2018). However, the corresponding convergence analysis is based on fixed step sizes. In the context of interpolation, Meng et al. further showed that stochastic BFGS can converge linearly with constant step sizes and batch sizes (Meng et al., 2020).

Overall, previous methods mainly stabilize stochastic quasi-Newton updates through curvature modification, variance reduction, or batch control. SMBFGS instead combines a damped memoryless BFGS direction with a stochastic nonmonotone line search, avoiding both a prescribed decay schedule and periodic full-gradient evaluations. To the best of our knowledge, this is the first work to combine stochastic L-BFGS with a nonmonotone line search, thereby achieving globalization in over-parameterized settings.

3 METHODS

3.1 PRELIMINARIES ON MEMORYLESS BFGS

Given the latest displacement and gradient-difference vectors $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$, respectively, the inverse L-BFGS update (Nocedal & Wright, 2006) is

$$\begin{aligned} H_k = {} & (V_{k-1}^T \cdots V_{k-m}^T) H_{k,0} (V_{k-m} \cdots V_{k-1}) \\ & + (V_{k-1}^T \cdots V_{k-m+1}^T) \rho_{k-m} s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_{k-1}) \\ & \vdots \\ & + \rho_{k-1} s_{k-1} s_{k-1}^T \end{aligned}, \tag{4}$$

where $H_{k,0} \succ 0$ is the initial inverse-Hessian approximation, $\rho_k = \frac{1}{s_k^\top y_k}$, $V_k = I - \rho_k y_k s_k^T$, and m signifies the storage length. Under the curvature condition $s_k^\top y_k > 0$ in equation 4, H_k remains symmetric positive definite. Consequently, the quasi-Newton direction defined in equation 3 is a strict descent direction whenever $\nabla f(x_k) \neq 0$.

L-BFGS retains only a small number of recent curvature pairs, whereas its memoryless form stores only $\{s_{k-1}, y_{k-1}\}$ (i.e., in equation 4, $m = 1$) and sets $H_{k,0} = I$ (Liu & Nocedal, 1989). Hence, $H_k \nabla f(x_k)$ can be evaluated by a one-pair two-loop recursion without forming a dense matrix, requiring only $\mathcal{O}(d)$ storage and arithmetic. This property makes the method attractive for high-dimensional learning problems. In the stochastic setting, however, sampling noise may render the

raw curvature pair unreliable or violate $s_k^\top y_k > 0$. The mechanism used by SMBFGS to address this issue is presented next.

3.2 THE DEVELOPED SMBFGS ALGORITHM

SMBFGS combines a damped memoryless BFGS direction with a stochastic nonmonotone line search. At iteration k, a mini-batch B_k is sampled, and $g_k = \nabla f_{B_k}(x_k)$ is computed. To avoid directly mixing gradients from different mini-batches, the curvature pair is constructed using the preceding batch B_{k-1} at two consecutive iterates:

$$s_{k-1} = x_k - x_{k-1}, \qquad y_{k-1} = \nabla f_{B_{k-1}}(x_k) - \nabla f_{B_{k-1}}(x_{k-1}). \tag{5}$$

This construction requires one additional stochastic-gradient evaluation, but reduces the direct influence of batch-to-batch variation on the gradient difference (Wang et al., 2017). Then, we apply Powell damping (Powell, 1978):

$$\theta_{k-1} = \begin{cases} \dfrac{0.75\|s_{k-1}\|^2}{\|s_{k-1}\|^2 - s_{k-1}^\top y_{k-1}}, & s_{k-1}^\top y_{k-1} < 0.25\|s_{k-1}\|^2 \\ 1 & \text{otherwise} \end{cases}, \bar{y}_{k-1} = \theta_{k-1} y_{k-1} + (1-\theta_{k-1}) s_{k-1}. \tag{6}$$

Consequently, $s_{k-1}^\top \bar{y}_{k-1} \geq 0.25\|s_{k-1}\|^2 > 0$ whenever $s_{k-1} \neq 0$. Replacing y_{k-1} by $\bar{y}_{k-1}$ in equation 4 therefore preserves positive definiteness. Only this most recent damped pair is retained, and $d_k = -H_k g_k$ is evaluated by the one-pair two-loop recursion. When no pair is available, $H_k = I$ and hence $d_k = -g_k$.

The initial trial step is chosen by the stochastic Polyak rule (Galli et al., 2023)

$$\widehat{\eta}_{k,0} = \min\left\{ \frac{f_{B_k}(x_k) - f_{B_k}^*}{c_p \|g_k\|^2}, \eta_{\max} \right\}, \tag{7}$$

where $f_{B_k}^* = \min_x f_{B_k}(x)$. For the nonnegative losses considered in our experiments, $f_{B_k}^*$ is set to zero. $\widehat{\eta}_{k,0}$ only initializes the line search; acceptance is determined using the actual quasi-Newton direction.

Specifically, let $Q_0 = 0$ and $r_0 = 0$. The nonmonotone reference value and the accepted step size are defined by

$$\begin{aligned} \widetilde{C}_k &= \frac{\xi Q_k C_{k-1} + f_{B_k}(x_k)}{\xi Q_k + 1}, \qquad C_k = \max\{\widetilde{C}_k, f_{B_k}(x_k)\}, \\ \eta_k &= \widehat{\eta}_{k,0} \delta^{\,r_k + \ell_k}, \qquad\qquad Q_{k+1} = \xi Q_k + 1, \end{aligned} \tag{8}$$

where ℓ_k is the smallest nonnegative integer satisfying

$$f_{B_k}(x_k + \eta_k d_k) \leq C_k + c\eta_k g_k^\top d_k. \tag{9}$$

After accepting the step, SMBFGS sets

$$x_{k+1} = x_k + \eta_k d_k, \quad r_{k+1} = \max\{r_k + \ell_k - 1, 0\}, \quad r_0 = 0.$$

The reset counter r_k transfers recent backtracking information to the next iteration. It prevents the line search from repeatedly restarting from an overly aggressive Polyak candidate, while the subtraction of one in its update allows the trial step to recover gradually after successful iterations. In particular, r_k decreases when $\ell_k = 0$, remains unchanged when $\ell_k = 1$, and increases when multiple backtracking steps are required. See Appendix A.1 for the pseudocode of SMBFGS.

4 CONVERGENCE ANALYSIS

This section studies the convergence behavior of SMBFGS in the over-parameterized regime. We first introduce the assumptions required by the analysis and then establish the corresponding convergence properties.

For brevity, we write $f_k(x) := f_{B_k}(x)$ and $g_k(x) := \nabla f_{B_k}(x)$. Let $\{\mathcal{F}_k\}_{k\geq 0}$ denote the filtration generated by all information revealed before the mini-batch B_k is sampled, and define $\mathbb{E}_k[\cdot] := \mathbb{E}[\cdot \mid \mathcal{F}_k]$. Unless otherwise stated, all equalities and inequalities involving random quantities are understood to hold almost surely.

4.1 BASIC ASSUMPTIONS

Throughout the analysis, we assume that the loss function F is differentiable and bounded below. Then the following assumptions are required for the proof.

Assumption 4.1 (Smoothness)**.** The objective function F is L-smooth, that is,

$$\|\nabla F(x) - \nabla F(y)\| \le L\|x - y\|, \qquad \forall x, y \in \mathbb{R}^d.$$

Moreover, each sampled function f_k is L_k-smooth, and there exists a constant $L_{\max} > 0$ such that

$$L \le L_{\max}, \qquad L_k \le L_{\max}, \qquad \forall k \ge 0.$$

Assumption 4.2 (Stochastic-gradient-related directions)**.** There exist constants $c_1, c_2 > 0$, independent of k, such that the search directions satisfy

$$\|d_k\| \le c_1\|g_k\|, \quad g_k^T d_k \le -c_2\|g_k\|^2, \qquad \forall k \ge 0$$

Assumption 4.3 (Conditional unbiasedness)**.** The sampled function value and stochastic gradient are conditionally unbiased, namely,

$$\mathbb{E}_k[f_k(x_k)] = F(x_k), \qquad \mathbb{E}_k[g_k] = \nabla F(x_k), \qquad \forall k \ge 0.$$

Assumption 4.4 (Strong growth condition)**.** There exists a constant $\rho \ge 1$ such that

$$\mathbb{E}_k\big[\|g_k\|^2\big] \le \rho\|\nabla F(x_k)\|^2, \qquad \forall k \ge 0.$$

For any square-integrable random vectors u and v, define the scalar conditional variance and covariance by $\mathrm{Var}_k(v) := \mathbb{E}_k[\|v\|^2] - \|\mathbb{E}_k[v]\|^2$, and $\mathrm{Cov}_k(u, v) := \mathbb{E}_k[u^T v] - \mathbb{E}_k[u]^T\mathbb{E}_k[v]$. Then the following assumption is required (Lapucci & Pucci, 2025).

Assumption 4.5 (Direction-gradient covariance)**.** There exists a constant $c_3 > 0$ such that

$$\mathrm{Cov}_k(d_k, g_k) \ge -c_3\,\mathrm{Var}_k(g_k), \qquad \forall k \ge 0.$$

Moreover, assume that $\sigma := c_2 - c_3\left(1 - \frac{1}{\rho}\right) > 0$, where ρ is given in Assumption 4.4.

Assumption 4.6 (Common-minimizer and common-value interpolation)**.** There exists a global minimizer x^* of F such that, for every $k \ge 0$,

$$x^* \in \operatorname*{argmin}_{x\in\mathbb{R}^d} f_k(x), \qquad f_k(x^*) = F^*.$$

Consequently,

$$f_k(x) - F^* \ge 0, \qquad \forall x \in \mathbb{R}^d, \quad \forall k \ge 0.$$

Assumption 4.7 (Polyak-Łojasiewicz condition)**.** The objective function F admits a global minimizer x^*. Let

$$F^* := F(x^*).$$

There exists a constant $\mu > 0$ such that

$$\|\nabla F(x)\|^2 \ge 2\mu\big(F(x) - F^*\big), \qquad \forall x \in \mathbb{R}^d.$$

4.2 CONVERGENCE PROPERTIES

The following theorem demonstrates the linear convergence of SMBFGS in the general nonconvex case. Before stating the main result, define $A := \frac{L_{\max}c_1^2}{2cc_2}$, $\eta_{\min} := \frac{2\delta c_2(1-c)}{c_1^2 L_{\max}}$, and $\chi := \frac{1}{\eta_{\min}} + A - 2\sigma\mu$, where c is the parameter in equation 9, and the other parameters follow from the above assumptions.

Theorem 4.1 (Linear convergence)**.** *Suppose that Assumptions 4.1–4.7 hold and that* $0 < \chi < \frac{1}{\eta_{\max}}$. *Let* $a > 0$ *be an auxiliary weight satisfying* $0 < a < \frac{1}{\eta_{\max}} - \chi$, *and assume that* $0 \le \xi < \frac{a}{A+a}$. *Define* $\nu := \eta_{\max}(\chi + a), b := \left(1 + \frac{A}{a}\right)\xi$, *and* $q := \max\{\nu, b\}$. *Then* $q \in (0, 1)$. *Let the sequence* $\{x_k\}$ *be generated by SMBFGS. Moreover, define*

$$\Delta_k := F(x_k) - F^*, \qquad \Gamma_k := C_k - F^*,$$

and the Lyapunov sequence

$$\Psi_k := \Delta_k + a\eta_{\max}\Gamma_{k-1}.$$

Then

$$\mathbb{E}\big[F(x_k) - F^*\big] \le q^k\mathbb{E}[\Psi_0], \qquad \forall k \ge 0.$$

Thus, the expected objective gap converges linearly to zero.

Theorem 4.1 is particularly relevant to practical over-parameterized learning settings. In such a regime, a sufficiently expressive model may interpolate the training data, so that the sampled losses admit a common zero-loss solution, while sufficiently wide neural networks may exhibit a local PŁ geometry around a global minimizer (Liu et al., 2022). Under these conditions, the theorem shows that SMBFGS can combine curvature-informed search directions with a stochastic nonmonotone line search while retaining linear convergence in expectation. This result provides theoretical support for the use of adaptive nonmonotone quasi-Newton methods in over-parameterized learning. The proof can be found in Appendix A.2.

5 Experimental study

This section evaluates the empirical performance of SMBFGS on several deep neural network training tasks. We first describe the experimental setup, including the datasets, network architectures, and baseline methods, and then compare the methods in terms of training loss, test accuracy, and computational efficiency.

5.1 Experimental settings

We evaluate SMBFGS on a collection of image-classification tasks covering datasets and network architectures of varying scales. Specifically, our experiments employ four datasets: MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), CIFAR10 (Krizhevsky, 2009), and CIFAR100 (Krizhevsky, 2009). Moreover, we consider five network architectures: MLP (Luo et al., 2019), ResNet-34 (He et al., 2016), VGG-11 (Simonyan & Zisserman, 2015), DenseNet-121 (Huang et al., 2017), and WideResNet (Zagoruyko & Komodakis, 2016).

To demonstrate the superiority of SMBFGS, we compare it against five competing methods: SGD (Robbins & Monro, 1951), Adam (Kingma & Ba, 2015), AdamW (Loshchilov & Hutter, 2019), PBQN-FO (Bollapragada et al., 2018), and SLS (Vaswani et al., 2019b). The first three are widely used in deep learning. PBQN-FO is a progressive batching quasi-Newton method for stochastic optimization, and SLS is a stochastic line search method. All results in this section are based on five independent runs.

Due to space constraints, we relegate the detailed problem list, model parameters, and algorithm-specific hyperparameters to Appendix A.3. We conduct seven experiments in total. For conciseness, the main text presents the results of five experiments, while the complete results are provided in Appendix A.3.

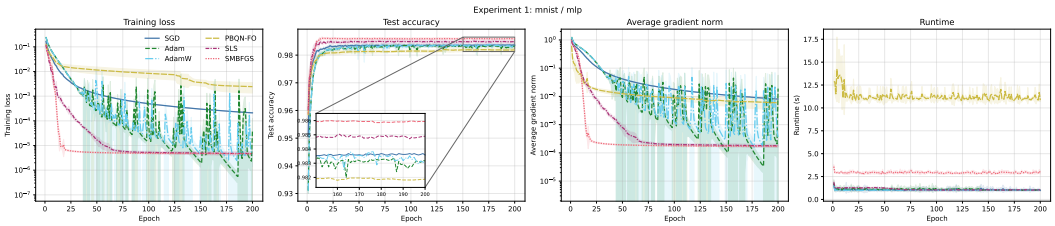
5.2 Comparison results

We compare the methods from two complementary perspectives. Figure 1 reports the training loss, test accuracy, average gradient norm, and per-epoch runtime against the number of epochs. Figure 2 further presents the training loss and test accuracy against cumulative training time, thereby accounting for differences in computational cost. The curves show the mean over five independent runs, and the shaded regions represent one standard deviation.

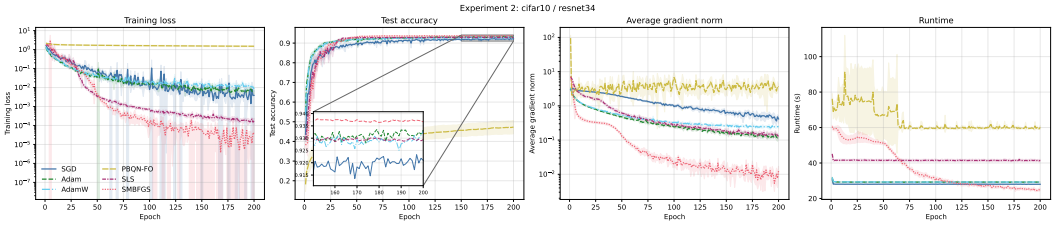
Epoch-wise results. As shown in Figure 1, SMBFGS exhibits strong and stable optimization behavior across the five tasks. On MNIST with MLP, it rapidly enters a low-loss regime and attains the highest final mean test accuracy. On CIFAR-10 with either ResNet-34 or DenseNet-121, SMBFGS achieves the lowest terminal training loss and gradient norm, together with the highest final mean test accuracy. A similar advantage is observed on SVHN with WideResNet, where SMBFGS reaches a training loss comparable to the lowest value obtained by the baselines, while producing a smaller gradient norm and a higher test accuracy. These improvements are particularly pronounced on the two CIFAR-10 tasks, for which PBQN-FO stagnates at substantially larger loss values.

On CIFAR-100 with DenseNet-121, SMBFGS again produces the lowest training loss and gradient norm, whereas SLS attains a slightly higher test accuracy. Thus, although SMBFGS provides the most accurate optimization of the training objective in this experiment, the result also illustrates that a smaller training error does not necessarily translate into better test performance. The improved progress per epoch comes with a higher per-epoch cost than the first-order baselines. This overhead

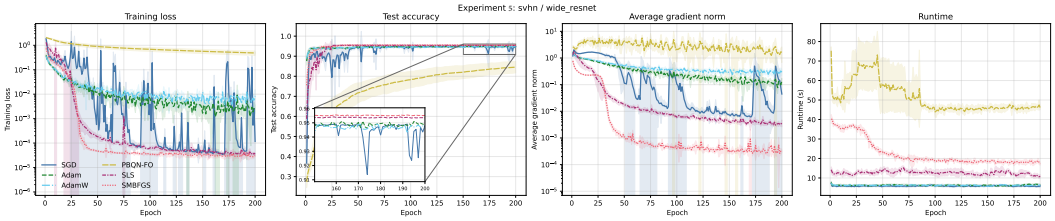
is expected because forming a same-batch curvature pair requires an additional gradient evaluation, while the line search may require repeated sampled-loss evaluations. Nevertheless, SMBFGS is generally less expensive per epoch than PBQN-FO.



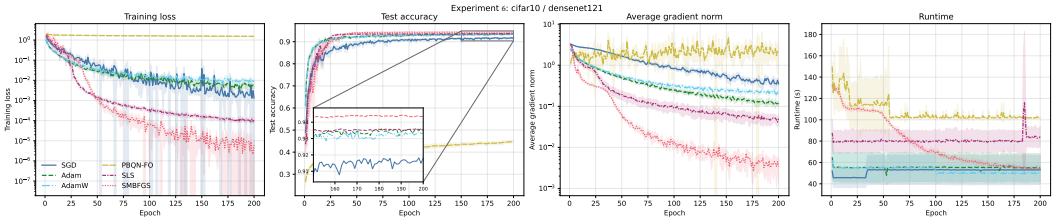
(a) MNIST with MLP



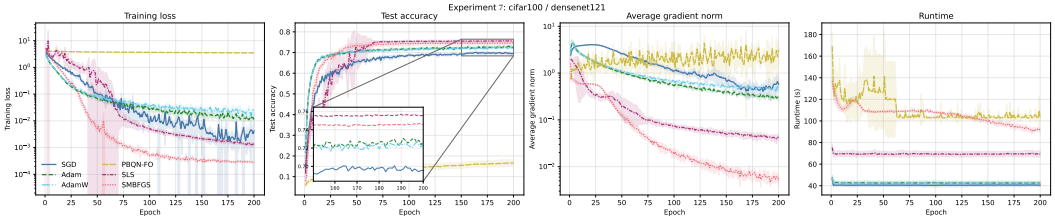
(b) CIFAR-10 with ResNet-34



(c) SVHN with WideResNet



(d) CIFAR-10 with DenseNet-121



(e) CIFAR-100 with DenseNet-121

Figure 1: Epoch-wise comparison across the five dataset-architecture configurations.

Table 1 provides a quantitative summary of the terminal performance. As indicated by the bold entries, SMBFGS achieves or ties for the best mean training loss in six of the seven experiments and

obtains the highest mean test accuracy in five. The only exception in training loss occurs on CIFAR-10 with VGG-11, where SLS performs better. SLS also attains the highest test accuracy on the two CIFAR-100 tasks, while SMBFGS ranks second and simultaneously reaches a lower training loss.

Among the baseline methods, SLS therefore exhibits the strongest overall performance. Its stochastic monotone line search selects the step size adaptively using the sampled function value and gradient, which provides a clear advantage over prescribed learning-rate schedules. Nevertheless, its search direction remains the negative stochastic gradient, and every accepted step must satisfy a monotone sufficient-decrease condition on the sampled objective. By comparison, SMBFGS incorporates local curvature information into the search direction through a single curvature pair and employs a nonmonotone acceptance criterion to accommodate controlled fluctuations in the mini-batch loss. In the direct comparison, SMBFGS achieves a higher mean test accuracy than SLS in five experiments and a lower or tied mean training loss in six. Although the two CIFAR-100 results show that this advantage is not universal, the overall evidence suggests that combining a compact curvature approximation with nonmonotone step-size adaptation can be more effective than relying on first-order line-search information alone.

By contrast, PBQN-FO remains competitive only on the relatively small MNIST problem and deteriorates substantially on the deeper convolutional networks. Thus, the results should not be interpreted as evidence that curvature information alone is sufficient; rather, they highlight the importance of incorporating such information through a computationally economical update and an appropriate globalization mechanism. Overall, the tabulated results corroborate the graphical observations and support SMBFGS as a favorable compromise among optimization accuracy, generalization performance, and computational complexity.

Table 1: Final training loss and test accuracy over five independent runs. Values are reported as mean $\pm$ standard deviation. The best and second-best mean results in each row are shown in boldface and underlined, respectively.

(a) Final training loss

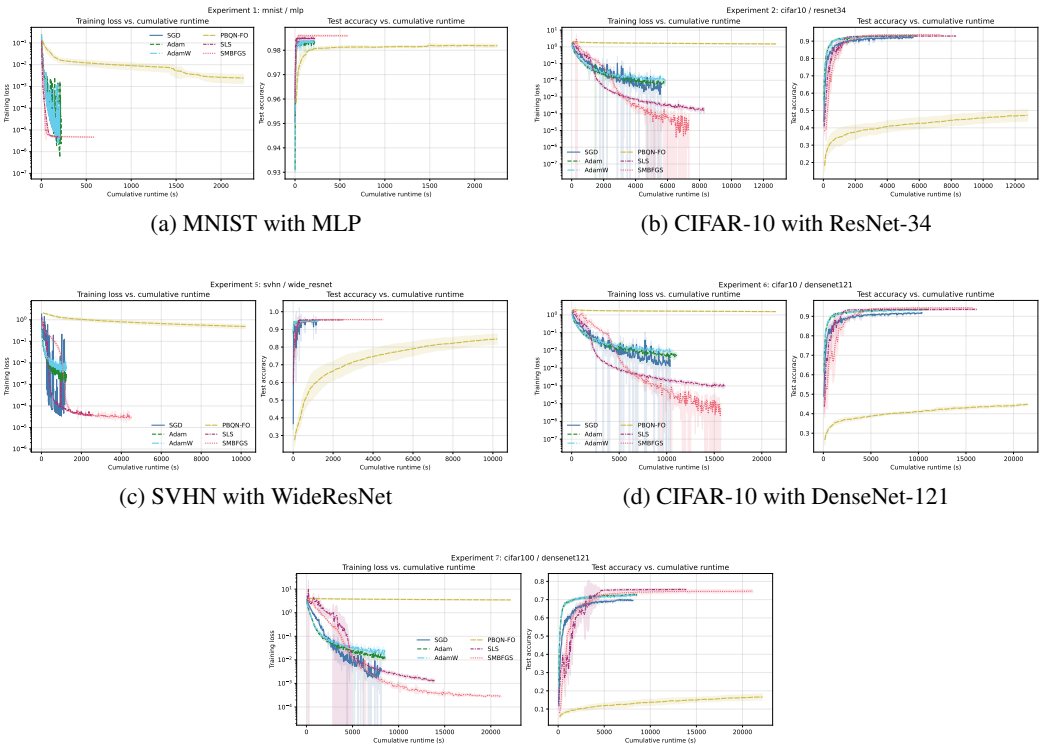
Experiment	SGD	Adam	AdamW	SLS	PBQN-FO	SMBFGS
Exp. 1: MNIST + MLP	$\underline{0.0002} \pm 0.0000$	$\mathbf{0.0000} \pm 0.0001$	$\mathbf{0.0000} \pm 0.0000$	$\mathbf{0.0000} \pm 0.0000$	0.0024 ± 0.0012	$\mathbf{0.0000} \pm 0.0000$
Exp. 2: CIFAR-10 + ResNet-34	0.0039 ± 0.0034	0.0059 ± 0.0015	0.0082 ± 0.0025	$\underline{0.0002} \pm 0.0001$	1.4934 ± 0.1007	$\mathbf{0.0000} \pm 0.0001$
Exp. 3: CIFAR-10 + VGG-11	$\underline{0.0091} \pm 0.0027$	0.1381 ± 0.0171	0.1024 ± 0.0447	$\mathbf{0.0054} \pm 0.0006$	1.2082 ± 0.0530	0.0113 ± 0.0053
Exp. 4: CIFAR-100 + ResNet-34	$\underline{0.0011} \pm 0.0005$	0.0137 ± 0.0026	0.0280 ± 0.0044	0.0013 ± 0.0002	4.2083 ± 0.1098	$\mathbf{0.0004} \pm 0.0000$
Exp. 5: SVHN + WideResNet	$\underline{0.0001} \pm 0.0001$	0.0027 ± 0.0010	0.0086 ± 0.0077	$\mathbf{0.0000} \pm 0.0000$	0.4864 ± 0.1085	$\mathbf{0.0000} \pm 0.0000$
Exp. 6: CIFAR-10 + DenseNet-121	0.0013 ± 0.0003	0.0049 ± 0.0011	0.0086 ± 0.0026	$\underline{0.0001} \pm 0.0000$	1.5312 ± 0.0350	$\mathbf{0.0000} \pm 0.0000$
Exp. 7: CIFAR-100 + DenseNet-121	0.0042 ± 0.0053	0.0109 ± 0.0016	0.0200 ± 0.0073	$\underline{0.0013} \pm 0.0002$	3.4986 ± 0.1101	$\mathbf{0.0003} \pm 0.0000$

(b) Final test accuracy (%)

Experiment	SGD	Adam	AdamW	SLS	PBQN-FO	SMBFGS
Exp. 1: MNIST + MLP	98.36 ± 0.03	98.32 ± 0.08	98.31 ± 0.05	$\underline{98.49} \pm 0.06$	98.19 ± 0.08	$\mathbf{98.59} \pm 0.03$
Exp. 2: CIFAR-10 + ResNet-34	92.10 ± 0.39	$\underline{93.33} \pm 0.13$	93.28 ± 0.26	92.94 ± 0.24	47.22 ± 3.77	$\mathbf{93.66} \pm 0.12$
Exp. 3: CIFAR-10 + VGG-11	$\underline{87.74} \pm 0.26$	85.83 ± 0.21	86.59 ± 0.46	87.37 ± 0.28	57.73 ± 2.22	$\mathbf{87.95} \pm 0.46$
Exp. 4: CIFAR-100 + ResNet-34	70.59 ± 0.44	69.96 ± 0.22	69.69 ± 0.23	$\mathbf{74.69} \pm 0.20$	5.05 ± 1.02	$\underline{71.04} \pm 0.47$
Exp. 5: SVHN + WideResNet	94.79 ± 0.16	94.85 ± 0.16	94.66 ± 0.45	$\underline{95.35} \pm 0.14$	84.55 ± 3.19	$\mathbf{95.52} \pm 0.17$
Exp. 6: CIFAR-10 + DenseNet-121	91.81 ± 0.21	93.51 ± 0.28	93.33 ± 0.33	$\underline{93.54} \pm 0.46$	44.84 ± 0.65	$\mathbf{94.35} \pm 0.17$
Exp. 7: CIFAR-100 + DenseNet-121	69.54 ± 0.79	72.69 ± 0.29	72.39 ± 0.86	$\mathbf{75.55} \pm 0.59$	16.65 ± 1.85	$\underline{74.52} \pm 0.89$

Cumulative-time results. Figure 2 provides a more direct assessment of computational efficiency. Owing to their inexpensive updates, SGD, Adam, and AdamW often make faster initial progress in terms of cumulative time. Their loss curves, however, tend to level off at larger values, and their final test accuracies are lower on most of the displayed tasks. Despite its additional curvature and line-search costs, SMBFGS compensates through greater progress per epoch. On MNIST, CIFAR-10, and SVHN, it ultimately reaches the lowest or nearly lowest training loss and the highest final mean test accuracy within the reported computational budget.

The CIFAR-100 experiment presents a more nuanced trade-off: SLS reaches a slightly higher test accuracy in less cumulative time, whereas SMBFGS obtains a lower training loss. PBQN-FO incurs the largest computational cost and does not achieve comparable accuracy on the displayed convolutional-network tasks. In summary, the cumulative-time results indicate that SMBFGS does not uniformly dominate every method during the early training stage, but its additional computation is offset by substantially improved optimization accuracy and competitive test performance.



(a) MNIST with MLP

(b) CIFAR-10 with ResNet-34

(c) SVHN with WideResNet

(d) CIFAR-10 with DenseNet-121

(e) CIFAR-100 with DenseNet-121

Figure 2: Runtime-based comparison across the five dataset–architecture configurations. In each subfigure, the left and right panels show training loss and test accuracy against cumulative runtime, respectively.

Overall, the results suggest that SMBFGS achieves a favorable balance among optimization performance, computational cost, and memory usage. In contrast to conventional stochastic L-BFGS methods that retain multiple curvature pairs, SMBFGS stores only one pair and therefore requires only $O(d)$ curvature storage. It also achieves lower training losses and competitive or higher test accuracies than Adam and AdamW on most of the evaluated tasks. These observations indicate that SMBFGS provides a practical curvature-aware alternative for large-scale, over-parameterized neural-network training.

6 Conclusion

This work introduced SMBFGS, a stochastic memoryless BFGS method with a nonmonotone line search for over-parameterized learning problems. SMBFGS combines same-batch curvature estimation, Powell damping, and a stochastic Polyak initial step size. The damped update preserves the positive definiteness of the inverse-Hessian approximation, while the line search adjusts the step size according to the sampled loss and the current quasi-Newton direction. Under suitable interpolation conditions, we proved linear convergence of SMBFGS in expectation without requiring convexity.

Experiments on deep neural network training demonstrate that SMBFGS provides stable optimization behavior and competitive performance against existing classical stochastic quasi-Newton methods and widely used first-order optimizers. These results suggest that combining curvature information with stochastic nonmonotone globalization is a promising approach to large-scale nonconvex learning. Future work will investigate weaker assumptions beyond common-value interpolation, reduce the computational overhead of curvature estimation, and explore the application of SMBFGS to parameter-efficient fine-tuning of large-scale pretrained models.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing. We have not used generative AI tools for algorithm design. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ACKNOWLEDGMENTS

Acknowledgment and funding information have been omitted to preserve anonymity and will be included in the camera-ready version if the paper is accepted.

REFERENCES

Albert S. Berahas and Martin Takáč. A robust multi-batch l-bfgs method for machine learning. *Optimization Methods and Software*, 35(1):191–219, 2020.

Albert S Berahas, Jorge Nocedal, and Martin Takác. A multi-batch l-bfgs method for machine learning. *Advances in Neural Information Processing Systems*, 29, 2016.

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*, pp. 620–629. PMLR, 2018.

Antoine Bordes, Léon Bottou, and Patrick Gallinari. Sgd-qn: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research*, 10:1737–1754, 2009.

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.

Daqing Chang, Shiliang Sun, and Changshui Zhang. An accelerated linearly convergent stochastic l-bfgs algorithm. *IEEE transactions on neural networks and learning systems*, 30(11):3338–3346, 2019.

Huiming Chen, Ho-Chun Wu, Shing-Chow Chan, and Wong-Hing Lam. A stochastic quasi-newton method for large-scale nonconvex optimization with applications. *IEEE transactions on neural networks and learning systems*, 31(11):4776–4790, 2019.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Leonardo Galli, Holger Rauhut, and Mark Schmidt. Don't be so monotone: Relaxing stochastic line search in over-parameterized models. *Advances in Neural Information Processing Systems*, 36: 34752–34764, 2023.

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*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2261–2269. Ieee, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Matteo Lapucci and Davide Pucci. Convergence conditions for stochastic line search based optimization of over-parametrized models. *Optimization*, pp. 1–20, 2025.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2019.

Jiameng Lyu, Jinxing Xie, Shilin Yuan, and Yuan Zhou. A minibatch stochastic gradient descent-based learning metapolicy for inventory systems with myopic optimal policy. *Management Science*, 71(7):5572–5588, 2025.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. In *International Conference on Machine Learning*, pp. 3325–3334. PMLR, 2018.

Martin Marek, Sanae Lotfi, Aditya Somasundaram, Andrew Wilson, and Micah Goldblum. Small batch size training for language models: When vanilla sgd works, and why gradient accumulation is wasteful. *Advances in Neural Information Processing Systems*, 38:148837–148862, 2026.

Si Yi Meng, Sharan Vaswani, Issam Hadj Laradji, Mark Schmidt, and Simon Lacoste-Julien. Fast and furious convergence: Stochastic second order methods under interpolation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1375–1386. PMLR, 2020.

Aryan Mokhtari and Alejandro Ribeiro. Stochastic quasi-newton methods. *Proceedings of the IEEE*, 11:1906–1922, 2020.

Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic l-bfgs algorithm. In *Artificial intelligence and statistics*, pp. 249–258. PMLR, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Michael JD Powell. Algorithms for nonlinear constraints that use lagrangian functions. *Mathematical programming*, 14(1):224–248, 1978.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Jascha Sohl-Dickstein, Ben Poole, and Surya Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-newton methods. In *International Conference on Machine Learning*, pp. 604–612. PMLR, 2014.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1195–1204. PMLR, 2019a.

Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. *Advances in neural information processing systems*, 32, 2019b.

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.

John Wright and Yi Ma. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications*. Cambridge University Press, 2022.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

Qingsong Zhang, Feihu Huang, Cheng Deng, and Heng Huang. Faster stochastic quasi-newton methods. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4388–4397, 2021.

Renbo Zhao, William Benjamin Haskell, and Vincent YF Tan. Stochastic l-bfgs: Improved convergence rates and practical acceleration strategies. *IEEE Transactions on Signal Processing*, 66(5): 1155–1169, 2017.

A Appendix

A.1 Algorithmic framework

The complete procedure of SMBFGS is shown in Algorithm 1. At each iteration, SMBFGS samples a mini-batch B_k and evaluates the corresponding loss f_k and gradient g_k. At the first iteration, the inverse-Hessian approximation is initialized as the identity matrix.

Thereafter, the method constructs the displacement s_{k-1} and a same-batch gradient difference y_{k-1}, and then applies Powell damping to obtain a modified curvature vector $\bar{y}_{k-1}$ satisfying the positive-curvature condition. Using only the most recent curvature pair $(s_{k-1}, \bar{y}_{k-1})$, the memoryless BFGS direction $d_k = -H_k g_k$ is computed through the two-loop recursion. SMBFGS then forms a stochastic Polyak initial step size, caps it by $\eta_{\max}$, and applies the inherited reset factor. Starting from this candidate, backtracking is performed until the stochastic nonmonotone sufficient-decrease condition is satisfied, where the reference value C_k combines the current mini-batch loss with information inherited from previous iterations. Finally, the iterate, reset counter, and reference weight are updated, and the procedure is repeated with a newly sampled mini-batch.

A.2 Convergence proof

Discussion of the assumptions. The assumptions in Section 4 are tailored to the over-parameterized regime and are not intended to cover arbitrary nonconvex stochastic optimization problems (Galli et al., 2023; Lapucci & Pucci, 2025). Conditional unbiasedness, smoothness, and bounded effective trial stepsizes are standard conditions in stochastic line-search analyses, whereas the PŁ condition allows the objective function to be nonconvex while retaining a gradient-dominated geometry (Liu et al., 2022). For the proposed quasi-Newton direction, the stochastic-gradient-related properties can be ensured through the damping and safeguarding mechanisms together with uniform spectral bounds on the quasi-Newton matrices. The strong-growth and direction–gradient covariance conditions are technical sufficient conditions used to transfer descent for the sampled objective to expected descent for the full objective (Lapucci & Pucci, 2025).

To facilitate the convergence analysis, we first present several properties of the initial step size and the line search technique before giving the formal convergence theorem. According to Galli et al. (2023), the following lemma holds.

Lemma A.1 (Bounded initial step size)**.** *Suppose that Assumption 4.1 is satisfied. Then the initial step size computed from equation 7 is bounded. That is, there exists a positive constant $\underline{\eta} > 0$, s.t.*

$$0 < \underline{\eta} \leq \eta_{k,0} \leq \eta_{max}.$$

Lemma A.2 identifies an interval of trial stepsizes that are guaranteed to satisfy the stochastic nonmonotone Armijo condition. Consequently, it ensures finite termination of the backtracking procedure and provides the basis for the uniform stepsize and backtracking bounds established in Proposition A.1.

Algorithm 1 Stochastic Memoryless BFGS with Nonmonotone Line Search (SMBFGS)

Require: Dataset $\mathcal{D}$, initial point x_0, batch size B, maximum epoch E, and parameters $\eta_{\max} > 0$, $c, c_p, \delta \in (0,1)$, and $\xi \in [0,1)$

Initialize $Q_0 \leftarrow 0$ and $r_0 \leftarrow 0$
for $k = 0, 1, \ldots, E-1$ **do**
  Sample a mini-batch $B_k \subseteq \mathcal{D}$ with $|B_k| = B$
  Compute $f_k \leftarrow f_{B_k}(x_k)$ and $g_k \leftarrow \nabla f_{B_k}(x_k)$
  if $k = 0$ **then**
    $H_k \leftarrow I$ and $C_{-1} \leftarrow f_k$
  else
    $s_{k-1} \leftarrow x_k - x_{k-1}$
    Compute

$$y_{k-1} \leftarrow \nabla f_{B_{k-1}}(x_k) - \nabla f_{B_{k-1}}(x_{k-1})$$

    Compute the damping coefficient

$$\theta_{k-1} \leftarrow \begin{cases} \dfrac{0.75\|s_{k-1}\|^2}{\|s_{k-1}\|^2 - s_{k-1}^\top y_{k-1}}, & s_{k-1}^\top y_{k-1} < 0.25\|s_{k-1}\|^2, \\ 1, & \text{otherwise} \end{cases}$$

    $\bar{y}_{k-1} \leftarrow \theta_{k-1} y_{k-1} + (1-\theta_{k-1}) s_{k-1}$
    $\rho_{k-1} \leftarrow 1/(s_{k-1}^\top \bar{y}_{k-1})$
  end if
  Set the search direction $d_k \leftarrow -H_k g_k$ ▷ This step is implemented by a two-loop recursion.
  Compute the stochastic Polyak candidate

$$\widehat{\eta}_{k,0} \leftarrow \min\left\{ \frac{f_k - f_{B_k}^*}{c_p \|g_k\|^2}, \eta_{\max} \right\}$$

  Compute the nonmonotone reference

$$\widetilde{C}_k \leftarrow \frac{\xi Q_k C_{k-1} + f_k}{\xi Q_k + 1}, \qquad C_k \leftarrow \max\{\widetilde{C}_k, f_k\}$$

  Set $\ell_k \leftarrow 0$ and $\eta_k \leftarrow \widehat{\eta}_{k,0} \delta^{\, r_k + \ell_k}$
  while $f_{B_k}(x_k + \eta_k d_k) > C_k + c\eta_k g_k^\top d_k$ **do**
    $\ell_k \leftarrow \ell_k + 1$
    $\eta_k \leftarrow \widehat{\eta}_{k,0} \delta^{\, r_k + \ell_k}$
  end while
  $x_{k+1} \leftarrow x_k + \eta_k d_k$
  $r_{k+1} \leftarrow \max\{r_k + \ell_k - 1, 0\}$
  $Q_{k+1} \leftarrow \xi Q_k + 1$
end for

Lemma A.2 (Guaranteed acceptance interval)**.** *Suppose that f_k satisfies the smoothness assumption in Assumption 4.1, and that the search direction d_k satisfies Assumption 4.2 for some constants $c_1, c_2 > 0$. Let $c \in (0,1)$, and define*

$$\eta_{\text{low}}^k := \frac{2c_2(1-c)}{c_1^2 L_k}.$$

Then, for every $\eta \in \left[0, \eta_{\text{low}}^k\right]$, the stochastic nonmonotone Armijo condition

$$f_k(x_k + \eta d_k) \le C_k + c\eta g_k^T d_k$$

is satisfied.

Proof. By the L_k-smoothness of f_k, the descent lemma yields, for every $\eta \geq 0$,

$$f_k(x_k + \eta d_k) \leq f_k(x_k) + \eta \nabla f_k(x_k)^T d_k + \frac{L_k \eta^2}{2} \|d_k\|^2$$
$$= f_k(x_k) + \eta g_k^T d_k + \frac{L_k \eta^2}{2} \|d_k\|^2.$$

Subtracting $f_k(x_k) + c\eta g_k^T d_k$ from both sides gives

$$f_k(x_k + \eta d_k) - \left[f_k(x_k) + c\eta g_k^T d_k\right] \leq (1-c)\eta g_k^T d_k + \frac{L_k \eta^2}{2} \|d_k\|^2.$$

By Assumption 4.2,

$$g_k^T d_k \leq -c_2 \|g_k\|^2, \qquad \|d_k\| \leq c_1 \|g_k\|.$$

Therefore,

$$f_k(x_k + \eta d_k) - \left[f_k(x_k) + c\eta g_k^T d_k\right] \leq -c_2(1-c)\eta \|g_k\|^2 + \frac{c_1^2 L_k \eta^2}{2} \|g_k\|^2$$
$$= \eta \left(-c_2(1-c) + \frac{c_1^2 L_k \eta}{2}\right) \|g_k\|^2.$$

Now, let $0 \leq \eta \leq \eta_{\text{low}}^k = \frac{2c_2(1-c)}{c_1^2 L_k}$. Then

$$\frac{c_1^2 L_k \eta}{2} \leq c_2(1-c),$$

and consequently

$$-c_2(1-c) + \frac{c_1^2 L_k \eta}{2} \leq 0.$$

It follows that

$$f_k(x_k + \eta d_k) \leq f_k(x_k) + c\eta g_k^T d_k.$$

Finally, by the definition of the nonmonotone reference value,

$$C_k \geq f_k(x_k).$$

Hence,

$$f_k(x_k + \eta d_k) \leq C_k + c\eta g_k^T d_k,$$

which is precisely the stochastic nonmonotone Armijo condition. □

Proposition A.1 establishes iteration-independent bounds on the accepted stepsizes and a uniform upper bound on the number of backtracking reductions. These estimates prevent the stepsizes from vanishing and keep the per-iteration line-search cost uniformly bounded.

Proposition A.1 (Uniform stepsize and backtracking bounds)**.** *Suppose that the assumptions of Lemma A.2 hold at every iteration. Let $c \in (0,1)$, $\delta \in (0,1)$. Given an initial trial stepsize $\eta_{k,0} > 0$, and it satisfy $\underline{\eta} \leq \eta_{k,0} \leq \eta_{\max}, \forall k \geq 0$, for some constant $\eta_{\max} > 0$.*

Define

$$\eta_k := \eta_{k,0} \delta^{j_k},$$

where j_k is the smallest integer $j \in \mathbb{N}_0$ such that

$$f_k\big(x_k + \eta_{k,0}\delta^j d_k\big) \leq C_k + c\eta_{k,0}\delta^j g_k^T d_k.$$

Define

$$\eta_{\text{low}}^k := \frac{2c_2(1-c)}{c_1^2 L_k}, \qquad \eta_{\text{low}} := \frac{2c_2(1-c)}{c_1^2 L_{\max}}, \quad \tilde{\eta} := \min\left\{\underline{\eta}, \delta\eta_{\text{low}}\right\}.$$

Moreover, let

$$j^* := \max\left\{0, \left\lceil \log_{1/\delta}\left(\frac{\eta_{\max}}{\eta_{\text{low}}}\right)\right\rceil\right\}.$$

Then the stochastic nonmonotone backtracking procedure is well defined and terminates after at most j^* reductions, namely,

$$j_k \le j^*, \qquad \forall k \ge 0.$$

Furthermore, the accepted stepsize satisfies

$$\eta_k \ge \min\left\{\eta_{k,0}, \delta\eta_{\text{low}}^k\right\},$$

and consequently

$$\boxed{0 < \tilde{\eta} \le \eta_k \le \eta_{\max}, \forall k \ge 0.}$$

In particular, at most $j^* + 1$ trial function evaluations are required by the line search at each iteration.

Proof. Since $L_k \le L_{\max}$, we have

$$\eta_{\text{low}} = \frac{2c_2(1-c)}{c_1^2 L_{\max}} \le \frac{2c_2(1-c)}{c_1^2 L_k} = \eta_{\text{low}}^k, \qquad \forall k \ge 0.$$

By Lemma A.2, every trial stepsize

$$\eta \in \left[0, \eta_{\text{low}}^k\right]$$

is guaranteed to satisfy the stochastic nonmonotone Armijo condition.

We first prove that the backtracking procedure is well defined and that the number of backtracking reductions is uniformly bounded. By the definition of j^*,

$$\eta_{\max}\delta^{j^*} \le \eta_{\text{low}}.$$

Indeed, if $\eta_{\max} \le \eta_{\text{low}}$, then $j^* = 0$, and the inequality is immediate. Otherwise,

$$j^* \ge \log_{1/\delta}\left(\frac{\eta_{\max}}{\eta_{\text{low}}}\right),$$

which is equivalent to

$$\delta^{j^*} \le \frac{\eta_{\text{low}}}{\eta_{\max}}.$$

Since $\eta_{k,0} \le \eta_{\max}$, it follows that

$$\eta_{k,0}\delta^{j^*} \le \eta_{\max}\delta^{j^*} \le \eta_{\text{low}} \le \eta_{\text{low}}^k.$$

Therefore, Lemma A.2 guarantees that the trial stepsize

$$\eta_{k,0}\delta^{j^*}$$

satisfies the stochastic nonmonotone Armijo condition. Hence, the set of acceptable backtracking indices is nonempty, and the smallest acceptable index j_k is well defined. Moreover,

$$j_k \le j^*, \qquad \forall k \ge 0.$$

We next establish the lower bound on the accepted stepsize. We distinguish two cases.

If $j_k = 0$, then the initial trial stepsize is accepted, and

$$\eta_k = \eta_{k,0}.$$

Consequently,

$$\eta_k \ge \min\left\{\eta_{k,0}, \delta\eta_{\text{low}}^k\right\}.$$

Suppose now that $j_k \ge 1$. By the minimality of j_k, the preceding trial stepsize

$$\eta_{k,0}\delta^{j_k-1} = \frac{\eta_k}{\delta}$$

does not satisfy the stochastic nonmonotone Armijo condition, that is,

$$f_k\left(x_k + \frac{\eta_k}{\delta}d_k\right) > C_k + c\frac{\eta_k}{\delta}g_k^T d_k.$$

If

$$\frac{\eta_k}{\delta} \le \eta_{\text{low}}^k,$$

then Lemma A.2 would imply that the preceding trial stepsize satisfies the line-search condition, which would contradict the minimality of j_k. Therefore,

$$\frac{\eta_k}{\delta} > \eta_{\text{low}}^k,$$

and hence

$$\eta_k > \delta \eta_{\text{low}}^k.$$

Thus, in both cases,

$$\eta_k \ge \min\left\{\eta_{k,0}, \delta\eta_{\text{low}}^k\right\}.$$

Using

$$\eta_{k,0} \ge \underline{\eta} \qquad \text{and} \qquad \eta_{\text{low}}^k \ge \eta_{\text{low}},$$

we further obtain

$$\eta_k \ge \min\left\{\eta_{k,0}, \delta\eta_{\text{low}}^k\right\} \ge \min\left\{\underline{\eta}, \delta\eta_{\text{low}}\right\} = \tilde{\eta}.$$

Finally, since $j_k \ge 0$ and $0 < \delta < 1$, we have

$$\eta_k = \eta_{k,0}\delta^{j_k} \le \eta_{k,0} \le \eta_{\max}.$$

Combining the lower and upper bounds gives

$$0 < \tilde{\eta} \le \eta_k \le \eta_{\max}, \qquad \forall k \ge 0.$$

At iteration k, the line search evaluates the trial points corresponding to $j = 0, 1, \ldots, j_k$. It therefore requires $j_k + 1$ trial function evaluations. Since $j_k \le j^*$, at most

$$j^* + 1$$

trial function evaluations are required at each iteration. □

Remark A.1. Here, $\eta_{k,0}$ denotes the effective initial trial stepsize actually supplied to the backtracking procedure after any resetting, clipping, or pre-scaling operation.

Remark A.2. Note that Lemma A.2 and Proposition A.1 generalize Lemma 4.2 and Proposition 4.3 in Lapucci & Pucci (2025), respectively, to the nonmonotone setting, and also extend the search direction to the quasi-Newton direction.

Theorem A.1 (Linear convergence)**.** *Suppose that Assumptions 4.1–4.7 hold and that* $0 < \chi < \frac{1}{\eta_{\max}}$. *Let* $a > 0$ *be an auxiliary weight satisfying* $0 < a < \frac{1}{\eta_{\max}} - \chi$, *and assume that* $0 \le \xi < \frac{a}{A+a}$. *Define* $\nu := \eta_{\max}(\chi + a), b := \left(1 + \frac{A}{a}\right)\xi$, *and* $q := \max\{\nu, b\}$. *Then* $q \in (0, 1)$. *Moreover, define*

$$\Delta_k := F(x_k) - F^*, \qquad \Gamma_k := C_k - F^*,$$

and the Lyapunov sequence

$$\Psi_k := \Delta_k + a\eta_{\max}\Gamma_{k-1}.$$

Then

$$\boxed{\mathbb{E}\left[F(x_k) - F^*\right] \le q^k \mathbb{E}[\Psi_0], \qquad \forall k \ge 0.}$$

Thus, the expected objective gap converges linearly to zero.

Proof. For convenience, define

$$r_k := f_k(x_k) - F^*.$$

By Assumption 4.6, we have

$$r_k \ge 0.$$

Moreover, the initialization of the nonmonotone reference sequence gives

$$\Gamma_{-1} = C_{-1} - F^* \ge 0.$$

We first show that the sequence $\{\Gamma_k\}$ is nonnegative and derive a useful recursion for it. Let

$$\theta_k := \frac{\xi Q_k}{\xi Q_k + 1}.$$

Since

$$Q_0 = 0, \qquad Q_{k+1} = \xi Q_k + 1, \qquad 0 \le \xi < 1,$$

we have

$$0 \le \theta_k \le \xi.$$

Indeed, $Q_k \le 1/(1-\xi)$, and hence

$$(1-\xi)Q_k \le 1,$$

which is equivalent to

$$\frac{\xi Q_k}{\xi Q_k + 1} \le \xi.$$

By the definition of the nonmonotone reference value,

$$C_k = \max\left\{\frac{\xi Q_k C_{k-1} + f_k(x_k)}{\xi Q_k + 1}, f_k(x_k)\right\}.$$

Therefore,

$$\Gamma_k = C_k - F^* = \max\left\{(1-\theta_k)r_k + \theta_k \Gamma_{k-1}, r_k\right\}.$$

Since $r_k \ge 0$, $\Gamma_{k-1} \ge 0$, and $0 \le \theta_k \le \xi$, it follows inductively that

$$\Gamma_k \ge 0,$$

and

$$\Gamma_k \le r_k + \xi \Gamma_{k-1}. \tag{10}$$

Multiplying equation 10 by $A + a > 0$, and recalling that

$$ab = a\left(1 + \frac{A}{a}\right)\xi = (A+a)\xi,$$

we obtain

$$(A+a)\Gamma_k \le (A+a)r_k + ab\Gamma_{k-1}. \tag{11}$$

We next establish a descent relation between the expected search direction and the true gradient. By Assumption 4.3,

$$\mathbb{E}_k[g_k] = \nabla F(x_k).$$

Using the definition of conditional covariance, we have

$$\mathrm{Cov}_k(d_k, g_k) = \mathbb{E}_k[d_k^T g_k] - \mathbb{E}_k[d_k]^T \mathbb{E}_k[g_k].$$

Consequently,

$$\nabla F(x_k)^T \mathbb{E}_k[d_k] = \mathbb{E}_k[d_k^T g_k] - \mathrm{Cov}_k(d_k, g_k).$$

By Assumptions 4.2 and 4.5,

$$\begin{aligned}\nabla F(x_k)^T \mathbb{E}_k[d_k] &\le -c_2 \mathbb{E}_k[\|g_k\|^2] + c_3 \,\mathrm{Var}_k(g_k) \\ &= -c_2 \mathbb{E}_k[\|g_k\|^2] + c_3 \left(\mathbb{E}_k[\|g_k\|^2] - \|\nabla F(x_k)\|^2\right).\end{aligned}$$

Assumption 4.4 implies

$$\mathbb{E}_k[\|g_k\|^2] \le \rho \|\nabla F(x_k)\|^2,$$

or equivalently,

$$\|\nabla F(x_k)\|^2 \ge \frac{1}{\rho}\mathbb{E}_k[\|g_k\|^2].$$

Hence,

$$\nabla F(x_k)^T \mathbb{E}_k[d_k] \le -c_2 \mathbb{E}_k[\|g_k\|^2] + c_3\left(1 - \frac{1}{\rho}\right)\mathbb{E}_k[\|g_k\|^2] = -\sigma \mathbb{E}_k[\|g_k\|^2].$$

Furthermore, Jensen's inequality gives

$$\mathbb{E}_k[\|g_k\|^2] \geq \|\mathbb{E}_k[g_k]\|^2 = \|\nabla F(x_k)\|^2.$$

Since $\sigma > 0$, we conclude that

$$\nabla F(x_k)^T \mathbb{E}_k[d_k] \leq -\sigma \|\nabla F(x_k)\|^2. \tag{12}$$

We now derive a one-step bound for the true objective function. By the L-smoothness of F and the update

$$x_{k+1} = x_k + \eta_k d_k,$$

we have

$$F(x_{k+1}) \leq F(x_k) + \eta_k \nabla F(x_k)^T d_k + \frac{L\eta_k^2}{2}\|d_k\|^2.$$

Therefore,

$$\frac{\Delta_{k+1} - \Delta_k}{\eta_k} \leq \nabla F(x_k)^T d_k + \frac{L\eta_k}{2}\|d_k\|^2. \tag{13}$$

By Assumption 4.2, we have $\|d_k\|^2 \leq c_1^2\|g_k\|^2$ and $-g_k^T d_k \geq c_2\|g_k\|^2$. Consequently,

$$\|d_k\|^2 \leq \frac{c_1^2}{c_2}\big(-g_k^T d_k\big).$$

Substituting this relation into equation 13 yields

$$\frac{\Delta_{k+1} - \Delta_k}{\eta_k} \leq \nabla F(x_k)^T d_k + \frac{Lc_1^2}{2c_2}\eta_k\big(-g_k^T d_k\big). \tag{14}$$

The accepted stepsize satisfies the stochastic nonmonotone Armijo condition

$$f_k(x_{k+1}) \leq C_k + c\eta_k g_k^T d_k.$$

Rearranging gives

$$c\eta_k\big(-g_k^T d_k\big) \leq C_k - f_k(x_{k+1}).$$

By Assumption 4.6,

$$f_k(x_{k+1}) \geq f_k(x^*) = F^*.$$

It follows that

$$\eta_k\big(-g_k^T d_k\big) \leq \frac{C_k - F^*}{c} = \frac{\Gamma_k}{c}.$$

Using $L \leq L_{\max}$ in equation 14, we obtain

$$\frac{\Delta_{k+1} - \Delta_k}{\eta_k} \leq \nabla F(x_k)^T d_k + \frac{L_{\max}c_1^2}{2cc_2}\Gamma_k = \nabla F(x_k)^T d_k + A\Gamma_k. \tag{15}$$

Adding $a\Gamma_k$ to both sides of equation 15, and then using equation 11, gives

$$\begin{aligned}\frac{\Delta_{k+1} - \Delta_k}{\eta_k} + a\Gamma_k &\leq \nabla F(x_k)^T d_k + (A+a)\Gamma_k \\ &\leq \nabla F(x_k)^T d_k + (A+a)r_k + ab\Gamma_{k-1}.\end{aligned} \tag{16}$$

Taking conditional expectation in equation 16, and using that x_k, Δ_k, and Γ_{k-1} are $\mathcal{F}_k$-measurable, we obtain

$$\mathbb{E}_k\left[\frac{\Delta_{k+1} - \Delta_k}{\eta_k} + a\Gamma_k\right] \leq \nabla F(x_k)^T \mathbb{E}_k[d_k] + (A+a)\mathbb{E}_k[r_k] + ab\Gamma_{k-1}.$$

Conditional unbiasedness gives

$$\mathbb{E}_k[r_k] = \mathbb{E}_k[f_k(x_k) - F^*] = F(x_k) - F^* = \Delta_k.$$

Combining equation 12 with the PL condition,

$$\|\nabla F(x_k)\|^2 \geq 2\mu\Delta_k,$$

we obtain

$$\nabla F(x_k)^T \mathbb{E}_k[d_k] \leq -2\sigma\mu\Delta_k.$$

Therefore,

$$\mathbb{E}_k\left[\frac{\Delta_{k+1}-\Delta_k}{\eta_k} + a\Gamma_k\right] \leq (A + a - 2\sigma\mu)\Delta_k + ab\Gamma_{k-1}. \tag{17}$$

Rearranging equation 17 yields

$$\mathbb{E}_k\left[\frac{\Delta_{k+1}}{\eta_k} + a\Gamma_k\right] \leq \mathbb{E}_k\left[\frac{\Delta_k}{\eta_k}\right] + (A + a - 2\sigma\mu)\Delta_k + ab\Gamma_{k-1}.$$

Since $\Delta_k \geq 0$ is $\mathcal{F}_k$-measurable and, by Proposition A.1, $\eta_k \geq \tilde{\eta} > 0$, we have

$$\mathbb{E}_k\left[\frac{\Delta_k}{\eta_k}\right] = \Delta_k \mathbb{E}_k\left[\frac{1}{\eta_k}\right] \leq \frac{\Delta_k}{\tilde{\eta}}.$$

Consequently,

$$\begin{aligned}\mathbb{E}_k\left[\frac{\Delta_{k+1}}{\eta_k} + a\Gamma_k\right] &\leq \left(\frac{1}{\tilde{\eta}} + A + a - 2\sigma\mu\right)\Delta_k + ab\Gamma_{k-1} \\ &= (\chi + a)\Delta_k + ab\Gamma_{k-1}. \end{aligned}\tag{18}$$

Multiplying equation 18 by the deterministic constant $\eta_{\max}$, we obtain

$$\mathbb{E}_k\left[\frac{\eta_{\max}}{\eta_k}\Delta_{k+1} + a\eta_{\max}\Gamma_k\right] \leq \eta_{\max}(\chi + a)\Delta_k + ab\eta_{\max}\Gamma_{k-1} = \nu\Delta_k + b\, a\eta_{\max}\Gamma_{k-1}.$$

Since Proposition A.1 also gives

$$\eta_k \leq \eta_{\max},$$

and $\Delta_{k+1} \geq 0$, we have

$$\frac{\eta_{\max}}{\eta_k}\Delta_{k+1} \geq \Delta_{k+1}.$$

Therefore,

$$\mathbb{E}_k\left[\Delta_{k+1} + a\eta_{\max}\Gamma_k\right] \leq \nu\Delta_k + b\, a\eta_{\max}\Gamma_{k-1}. \tag{19}$$

By the definition

$$\Psi_k = \Delta_k + a\eta_{\max}\Gamma_{k-1},$$

inequality equation 19 becomes

$$\mathbb{E}_k[\Psi_{k+1}] \leq \nu\Delta_k + b\, a\eta_{\max}\Gamma_{k-1}.$$

Since

$$\Delta_k \geq 0, \quad \Gamma_{k-1} \geq 0, \quad q = \max\{\nu, b\},$$

it follows that

$$\mathbb{E}_k[\Psi_{k+1}] \leq q\left(\Delta_k + a\eta_{\max}\Gamma_{k-1}\right) = q\Psi_k.$$

It remains to verify that $q \in (0, 1)$. Since

$$0 < \chi \qquad \text{and} \qquad a > 0,$$

we have

$$\nu = \eta_{\max}(\chi + a) > 0.$$

Moreover, the condition

$$a < \frac{1}{\eta_{\max}} - \chi$$

implies

$$\nu = \eta_{\max}(\chi + a) < 1.$$

Similarly,

$$0 \le b = \left(1 + \frac{A}{a}\right)\xi,$$

and the condition

$$\xi < \frac{a}{A+a}$$

gives

$$b < \left(1 + \frac{A}{a}\right)\frac{a}{A+a} = 1.$$

Hence,

$$q = \max\{\nu, b\} \in (0, 1).$$

Taking total expectation and applying the tower property, we obtain

$$\mathbb{E}[\Psi_{k+1}] = \mathbb{E}\left[\mathbb{E}_k[\Psi_{k+1}]\right] \le q\mathbb{E}[\Psi_k].$$

Applying this inequality recursively yields

$$\mathbb{E}[\Psi_k] \le q^k \mathbb{E}[\Psi_0], \qquad \forall k \ge 0.$$

Finally, since

$$0 \le \Delta_k = F(x_k) - F^* \le \Psi_k,$$

we conclude that

$$\mathbb{E}[F(x_k) - F^*] \le \mathbb{E}[\Psi_k] \le q^k \mathbb{E}[\Psi_0], \qquad \forall k \ge 0.$$

□

A.3 Experimental details

Table 2 lists the details of the benchmark problems, and Table 3 summarizes the parameter configurations of each algorithm. All experiments were conducted on a server equipped with dual Intel Xeon Gold 6248 CPUs @ 2.50GHz (80 cores total), an NVIDIA GeForce RTX 3080 Ti GPU, and 128GB RAM. The operating system was Ubuntu 22.04.3 LTS. All models were implemented using PyTorch 2.1.2 with CUDA 12.1 support on Linux.

All datasets were obtained through the standard interfaces provided by `torchvision.datasets`, with automatic downloading enabled.

Table 2: Problem list

Problem \ Data	d	M	B	$\lceil M/B \rceil$	Test size	Max. epochs (corr. iter.)
(MNIST, MLP)	535818	60000	128	469	10000	200 (93800)
(CIFAR10, ResNet-34)	21282122	50000	128	391	10000	200 (78200)
(CIFAR10, DenseNet-121)	6956298	50000	128	391	10000	200 (78200)
(CIFAR10, VGG-11)	9488266	50000	128	391	10000	200 (78200)
(CIFAR100, ResNet-34)	21328292	50000	128	391	10000	200 (78200)
(CIFAR100, DenseNet-121)	7048548	50000	128	391	10000	200 (78200)
(SVHN, WideResNet)	369498	73257	128	573	26032	200 (114600)

Note: The columns report the number of model parameters d, training-set size M, batch size B, number of mini-batches per epoch $\lceil M/B \rceil$, test-set size, and maximum number of epochs; the corresponding number of iterations is given in parentheses.

SGD, Adam, and AdamW were implemented using `torch.optim.SGD`, `torch.optim.Adam`, and `torch.optim.AdamW`, respectively. Their learning rates—the base learning rates in the cases of Adam and AdamW—were tuned separately for each dataset-architecture configuration through a preliminary grid search. Once selected, the learning rate was fixed throughout training and used consistently across the five independent runs; no learning-rate scheduler was applied. The resulting setting was $\eta = 10^{-1}$ for SGD in all experiments. For Adam and AdamW, the selected base learning rate was 10^{-4} on MNIST with MLP and 10^{-3} on the remaining tasks. Unless otherwise specified, the other optimizer parameters were kept at their default values.

Code availability. The anonymized implementation of SMBFGS and the scripts used to reproduce the experiments are available at `https://anonymous.4open.science/r/smbfgs-55FB`.

Table 3: Parameter settings of methods

Method	Hyperparameter settings
SGD	$\eta = 10^{-1}$.
Adam	$\eta = 10^{-4}$ for MNIST–MLP and 10^{-3} otherwise; $(\beta_1, \beta_2) = (0.9, 0.999)$; $\epsilon = 10^{-8}$.
AdamW	$\eta = 10^{-4}$ for MNIST–MLP and 10^{-3} otherwise; $(\beta_1, \beta_2) = (0.9, 0.999)$; $\epsilon = 10^{-8}$; $\lambda = 10^{-2}$.
PBQN-FO	$\theta = 0.9$; $m = 10$; $c_1 = 10^{-4}$; $\beta = 0.5$; $\epsilon = 10^{-2}$.
SLS	$\eta_0 = 1$; $c = 0.1$; $\beta = 0.9$; $\gamma = 1.5$.
SMBFGS	$\eta_{max} = 10$; $c = 0.5$; $c_p = 0.1$; $\delta = 0.5$; $\xi = 0.9999$.

Complete epoch-wise results. Figure 3 presents the complete epoch-wise results for all seven experiments. To avoid repeating the discussion in the main text, we focus here on Experiments 3 and 4, which provide two additional dataset–architecture configurations, and on the cross-experiment observations that become visible only from the full collection.

On CIFAR-10 with VGG-11, SLS obtains the smallest final training loss, whereas SMBFGS achieves the highest mean test accuracy. In particular, the final loss of SLS is 0.0054 ± 0.0006, compared with 0.0113 ± 0.0053 for SMBFGS, while the corresponding test accuracies are $87.37\% \pm 0.28\%$ and $87.95\% \pm 0.46\%$, respectively. Consequently, this experiment provides an informative exception: SMBFGS gives the best generalization result without attaining the smallest training objective. Its per-epoch runtime is higher than that of the first-order methods and SLS, reflecting the additional work required to construct the curvature pair and perform the nonmonotone line search, but remains substantially below that of PBQN-FO.

On CIFAR-100 with ResNet-34, SMBFGS reaches the lowest final training loss, 0.0004 ± 0.0000, and its gradient norm continues to decrease below those of the competing methods. SLS, however, achieves the highest test accuracy of $74.69\% \pm 0.20\%$, compared with $71.04\% \pm 0.47\%$ for SMBFGS. The same qualitative separation is observed on CIFAR-100 with DenseNet-121: SMBFGS provides the more accurate minimization of the training objective, whereas SLS yields the higher test accuracy. Thus, for the two CIFAR-100 architectures considered here, improved optimization accuracy does not translate directly into improved generalization.

Taken together, the complete results show that the conclusions drawn from the representative experiments are not confined to a particular network family. SMBFGS remains effective across fully connected, residual, VGG, WideResNet, and DenseNet architectures. At the same time, the VGG-11 and CIFAR-100 experiments identify opportunities for further improvement, particularly in translating higher optimization accuracy into better test performance and in reducing the additional cost associated with curvature estimation and line search. These observations motivate future refinements of the curvature-pair construction and step-size adaptation strategy, with the aim of further improving the computational efficiency and generalization performance of SMBFGS.

Complete runtime-based results. Figure 4 presents the complete comparison against cumulative runtime. Since all methods were evaluated under the same computational environment, these plots provide an empirical view of the time required to reach a given loss or accuracy.

The two configurations omitted from the main-text discussion reveal different computational trade-offs. On CIFAR-10 with VGG-11, SLS reaches the lowest training loss in less cumulative time, whereas SMBFGS runs longer but ultimately attains the highest mean test accuracy. On CIFAR-100 with ResNet-34, SMBFGS continues to reduce the training loss after the first-order methods have largely plateaued. Nevertheless, SLS reaches a higher test accuracy and does so earlier. These cases further illustrate that additional progress on the training objective does not necessarily produce a proportional improvement in test performance.

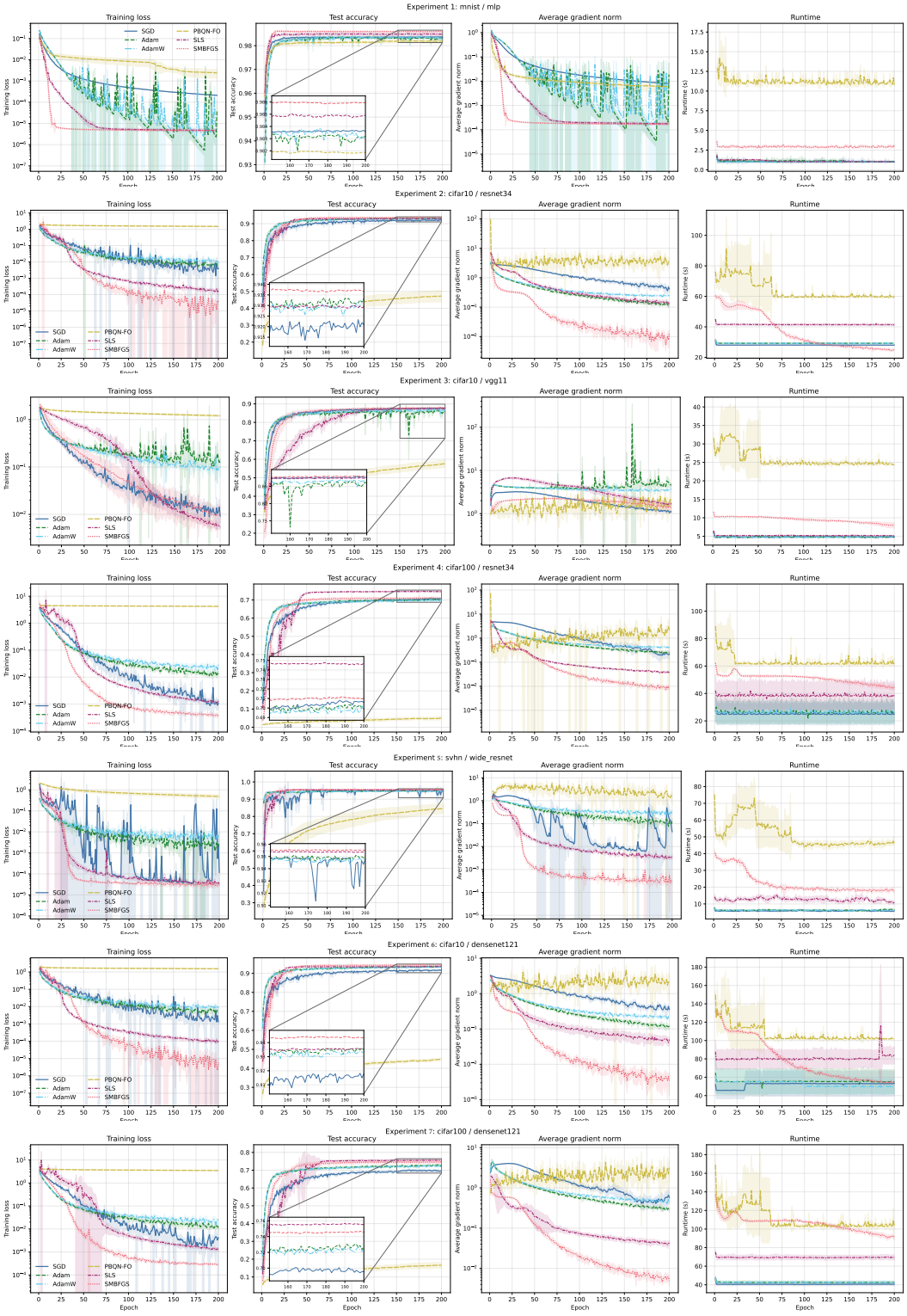


Figure 3: Epoch-wise comparison.

Across the seven experiments, SGD, Adam, and AdamW generally complete the prescribed number of epochs in less time because they require neither curvature-pair construction nor trial-point evaluations. SMBFGS incurs additional per-iteration work from computing the curvature pair and performing the nonmonotone line search. This cost is partly offset by stronger progress in the ob-

jective: on CIFAR-10 with ResNet-34 and DenseNet-121, SMBFGS reaches substantially lower training losses within a runtime comparable to, or shorter than, that required by SLS. On the smaller MNIST problem and the SVHN experiment, most methods attain high test accuracy relatively early, so the additional runtime of SMBFGS mainly yields a smaller training loss and a modest accuracy improvement.

PBQN-FO exhibits a different pattern. Except on MNIST, it consumes the largest or nearly the largest cumulative runtime while remaining far from the loss and accuracy levels attained by the other methods. In contrast, SMBFGS uses only one curvature pair and achieves a considerably more favorable balance between curvature-related cost and optimization progress. Overall, its principal computational advantage is not the shortest raw runtime, but its ability to attain a low training loss within a practical time budget. The remaining overhead also indicates room for improving the efficiency of curvature construction and line-search evaluations in future implementations.

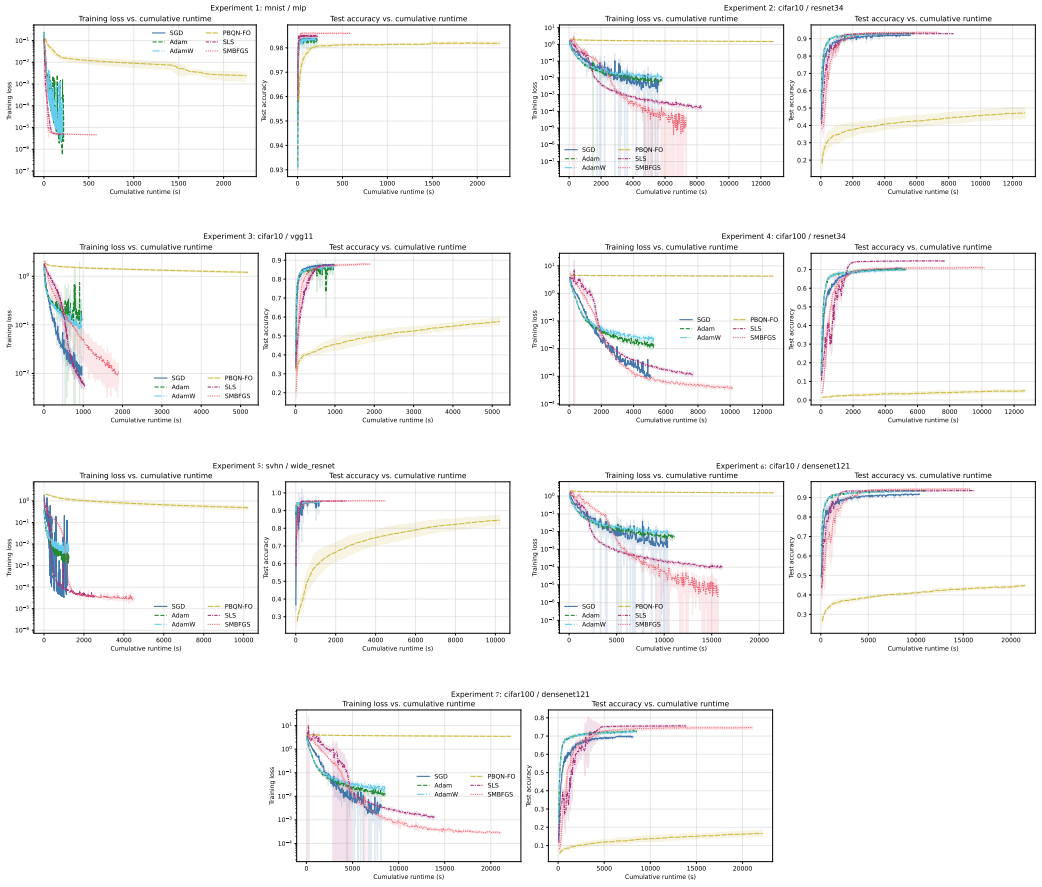


Figure 4: Runtime-based comparison.