AccSPS Learning Rate: Accelerated Convergence through Decision-Adjusted Levels for Stochastic Polyak Stepsize

Jingtao Qin, *Student Member, IEEE,* Anbang Liu, *Student Member, IEEE,* Mikhail Bragin, *Senior Member, IEEE,* Nanpeng Yu, *Senior Member, IEEE,*

***Abstract*—Recently, the application of stochastic gradient descent (SGD) with Polyak stepsizes has gained attention and exhibited promising performance for machine learning problems. However, when the interpolation condition (where each loss attains a minimum at a globally optimal solution) is not satisfied, SGD with Polyak stepsizes encounters a primary limitation that impedes its overall effectiveness: a lack of knowledge of the optimal loss. In this study, we introduce a non-diminishing accelerated stochastic Polyak stepsize (AccSPS) with level adjustment coupled with approximate variance-reduced gradient (AVRG) descent to overcome this limitation. Our approach incorporates a decision-based level adjustment method to obtain an accurate estimation of the optimal loss. To significantly reduce memory requirements, we adopt a variance-reduced method that keeps a snapshot of average gradients after specific iterations. Theoretical analyses establish the convergence rate to the exact minimum in the non-interpolated setting. Numerical studies demonstrate the superior performance of AccSPS, showcasing a significantly lower loss when compared to state-of-the-art algorithms like DecSPS, AdaGrad, Adam, and AMSGrad, up to several orders of magnitude.**

***Note to Practitioners*—This paper addresses a common challenge in training machine learning models—namely, the difficulty of tuning learning rates when the optimal loss value is unknown. Traditional approaches, such as stochastic gradient descent (SGD) with Polyak stepsizes, work well when every training example achieves its minimum at a known optimal solution. However, in many practical settings this condition does not hold, limiting the effectiveness of these methods. The proposed solution introduces an accelerated variant of SGD, named AccSPS, which dynamically adjusts its stepsize without needing the exact optimal loss value. It does this by using a decision-based level adjustment method that estimates the optimal loss during training, coupled with a technique that reduces the memory burden by averaging gradients at strategic points. As a result, this approach not only simplifies the tuning process but also improves the overall performance of the training algorithm. Numerical experiments show that AccSPS can achieve significantly lower loss values compared to widely used algorithms like DecSPS, AdaGrad, Adam, and AMSGrad. While the method has demonstrated promising performance in theoretical and experimental studies, future work will focus on further refining these techniques for even broader applications in machine learning.**

***Index Terms*—Stochastic Polyak stepsize, level adjustment, variance-reduced gradient, non-interpolation.**

This work was funded by the National Science Foundation under Grant No. 2324940. Jingtao Qin and Nanpeng Yu are with the Electrical and Computer Engineering Department at the University of California, Riverside. (email: jingtao.qin@email.ucr.edu, nyu@ece.ucr.edu). Anbang Liu is with the Department of Data and System Engineering at the University of Hong Kong. (email:anbang@hku.hk). Mikhail Bragin is with Southern California Edison. (Corresponding author: Nanpeng Yu).

I. Introduction

In this paper, we consider solving machine learning problems, which can be formulated as follows:

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

where $f_i(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ are loss functions and x are the parameters of the model. The goal is to minimize the average loss $f(x)$. The nature of the function f varies based on the model under consideration, exhibiting characteristics of strongly convex, convex, or non-convex. Such problems are common in machine learning. The set of optimal solutions $x^\star$ is denoted as $\mathcal{X}^\star \in \mathbb{R}^d$ and we assume that $\mathcal{X}^\star$ is a closed non-empty set. The optimum value of f and f_i are denoted as $f^\star := \inf_x f(x)$ and $f_i^\star := \inf_x f_i(x)$, respectively.

A. Background and literature review

The success of a machine learning model depends on the efficiency of optimization of (1), and the foundational principles behind rest upon a series of solution updates. During the course of several decades, there have been numerous studies that improve solution-updating directions (e.g., stochastic gradients, incremental gradients) and stepsizes (also known as learning rate, e.g., constant stepsizes, Polyak stepsizes) for fast convergence, which will be discussed next.

Updating directions. Stochastic gradient descent (SGD), introduced in [1], is widely used for machine learning problems (1) where solutions are updated as $x_{k+1} = x_k - \eta_k \nabla f_i(x^k)$ with directions $\nabla f_i(x^k)$ representing gradients for component i. The high variance in gradient estimates can make SGD inefficient for large datasets where more stable and accurate gradient estimates are preferred. To address this issue, stochastic variance-reduced gradient (SVRG) descent [2]–[4] was developed to reduce variance by maintaining snapshots of past average gradients.

Stepsizes. Ensuring the convergence of SGD and SVRG hinges on a critical parameter—the stepsize η_k. Adopting diminishing stepsizes, [1], [5]–[7] has been one of the traditional strategies to guarantee convergence to the exact solution. This stands in contrast to a constant stepsize, which only ensures convergence to a neighborhood of the optimum. Besides, a variety of adaptive stepsize methods, including Adam [8], RMSprop [9], AdaGrad [10]–[12], and AMSGrad [13] have gained widespread popularity and have been incorporated as foundational optimizers in machine learning libraries.

Following the pioneering work by Polyak [14] originally developed to set stepsizes as $\eta_k = \frac{f(x^k)-f^\star}{||g^k||^2}$ for deterministic subgradient methods [15], [16], the "stochastic" variants of Polyak stepsizes have gained momentum in recent studies for machine learning [17]–[24]. For a smooth problem (1), subgradients g^k at x^k always exist and are equal to gradients $\nabla f(x^k)$. Compared to the deterministic Polyak stepsize, which requires the assessment of the function value $f(x^k)$ and its subgradient g^k, the stochastic Polyak stepsize developed in [19], [20], [22]–[24] simplifies this requirement by only necessitating the evaluation of $f_i(x^k)$ and $\nabla f_i(x^k)$. However, the primary challenge with stochastic versions of the Polyak stepsize arises in non-interpolated settings due to the lack of information about the optimal loss $f^\star$, which will be discussed next.

Previous work on estimations of $f^\star$. Various estimation techniques have been employed since the Polyak stepsize relies on the generally unknown value of $f^\star$. In [17], a variant of the Polyak stepsize named the $L4$ algorithm is introduced, which requires online estimation of $f^\star$. It substitutes $f^\star$ with $\gamma f_{\min}$, where $f_{\min}$ represents the minimum loss achieved up to that iteration. The method, however, has no theoretical convergence guarantees and exhibits no robust empirical performance.

A comparable approach is presented in [18] to estimate $f^\star$ where a scheduled SGD is run first to obtain a loss value as the estimation of $f^\star$, which is subsequently maintained without further adjustments. Furthermore, the theoretical proof specifically concentrates on strongly convex smooth functions. In [25], a moving-target Polyak step size is proposed, utilizing n auxiliary variables to record the past loss values for each data point. However, it requires careful hyperparameter tuning and may face challenges when applied to problems with large regularization.

When estimating f^*, a notable distinction occurs under interpolated versus non-interpolated settings. Under interpolated setting, where at $x^\star$ the following relation holds $f^\star = f_i^\star = 0$, the estimation problem is much simplified [19]–[21]. [20] presented a bounded stepsize $\eta_k = \min\left\{\frac{f_i(x^k)-f_i^\star}{c||\nabla f_i(x^k)||^2}, \eta_b\right\}$ to guarantee convergence. However, the non-interpolated scenario, often encountered in under-parameterized or regularized problems (e.g., incorporating an l_2 regularization term to (1)), presents a significant challenge since both $f^\star$ and $f_i^\star$ may exceed zero rendering estimations $f_i^\star$ a bit less informative for estimating $f^\star$. Addressing this gap, [22] introduced DecSPS, an adaptation that calculates the stepsize as $\eta_k = \frac{1}{c_k}\min\left\{\frac{f_i(x^k)-l_i^\star}{||\nabla f_i(x^k)||^2}, c_{k-1}\eta_{k-1}\right\}$ setting a diminishing boundary to ensure convergence even when $f^\star$ and $f_i^\star$ are unknown. This model leverages a constant lower bound $l_i^\star \leq f_i^\star$, typically zero, to approximate $f_i^\star$. This strategy, while advancing towards resolving the convergence issues, due to its diminishing upper bound for the stepsize, may limit the ability to exploit the nature behind the Polyak stepsize fully.

Level adjustment for accurate estimations of $f^\star$. To get accurate approximations of $f^\star$, the optimal function value $f^\star$ can be replaced with dynamically-adjusted level values $\hat{f}_k$. There have been two level-adjustment research directions. The "path-based" level adjustment approaches were developed in [26]–[28] for subgradients methods [29], [30]. While these methods have demonstrated convergence, their reliance on hyperparameters and customizability for specific problem instances remains problematic. In another direction, "decision-based" level adjustment methods have been developed in [31], [32]. The original key idea for decision-based level adjustment is to detect the divergence of a sequence of solutions $\{x^{k_j}, x^{k_j+1}, \ldots, x^{k_j+m_j}\}$ from the exact solution $x^\star$ [31]; the result has then been strengthened to detect the violation of the Polyak stepsize directly [32]. Within both "decision-based" methods, level values are readjusted when a violation is detected, and both approaches showed an advantage over their "path-based" counterparts as demonstrated by [31], [32].

B. Main contributions

As previously mentioned, in the non-interpolated setting, SGD with Polyak stepsize encounters a primary limitation: the absence of $f^\star$, significantly impeding convergence to the true optimum. In this paper, we propose an approximate variance-reduced gradient descent method by synergistically combining the Polyak stepsize and the decision-guided level adjustment approach [32] to address this limitation in the non-interpolation setting: Namely, the dynamically adjusted level values converge to the true $f^\star$, leading to the solution converging to the exact solution with approximate variance-reduced gradients. To the best of our knowledge, we are the first to develop the Polyak stepsizing rule with a *decision-based* level adjustment approach to address problems arising in the field of machine learning. Furthermore, this level-adjusted Polyak rule can be embedded within the surrogate Lagrangian relaxation (SLR) framework to tackle large-scale mixed-integer programs (MIPs) [33]–[36].

Specifically, our work makes the following contributions:

• In Section II, to significantly reduce memory requirements, we extend the approximate gradient method and the level adjustment approach proposed in [32] to the stochastic variance-reduced minibatch setting. Then we propose a novel non-diminishing stepsize termed accelerated stochastic Polyak stepsize (AccSPS) and introduce the detailed algorithm to update solutions. The non-diminishing nature of AccSPS comes from adapting to approximate variance-reduced gradients as well as from gauging distances from function value to the series of level values converging to the true $f^\star$ from above.

• In Section III, we theoretically derive the convergence rates for AccSPS employing the approximate variance-reduced gradient descent, both with and without the knowledge of $f^\star$.

• In Section IV, we showcase an illustrative example highlighting the dynamics of AccSPS and emphasizing the significance of $f^\star$. We illustrate that, with knowledge of $f^\star$, SPS can achieve convergence to the exact solution, even in the presence of correlation between η_k and $\nabla f_i(x^k)$ mentioned in [22]. Moreover, the dynamically adjusted estimation of $f^\star$ ensures AccSPS exhibits a faster convergence to the exact minimum compared to DecSPS, particularly when lacking knowledge of $f^\star$.

• In Section V, we validate our theoretical findings through numerical studies. By testing on various datasets, we demonstrate the effectiveness and superior performance of AccSPS compared to state-of-the-art algorithms in terms of achieving a lower loss for a variety of convex machine learning problems. Additionally, we extend our experiments to non-convex datasets demonstrating the advantage over existing methods for Matrix Factorization and image classification problems.

II. Accelerated stochastic Polyak stepsize with level value adjustment

In this section, the approximate function value and approximate variance-reduced gradient in the stochastic minibatch setting are presented first. The Polyak stepsize and the accelerated level adjustment approach are then presented, followed by the procedure of the entire algorithm.

A. Approximate variance-reduced gradient descent

To reduce computational requirements, we partition the entire dataset into predetermined batches $\mathcal{S}$ and randomly select one batch $\mathcal{B}_i \in \mathcal{S}, i = 1, \ldots, m$ during each iteration. Here m is the number of batches, and b_i is the number of components in batch $\mathcal{B}_i$. With this setting, we define the approximate variance-reduced gradient as follows.

Definition II.1. Consider that we keep a snapshot of average gradient $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{m} b_i \nabla f_{\mathcal{B}_i}(\tilde{x})$ and an average function value $\tilde{\nu} = \frac{1}{n}\sum_{i=1}^{m} b_i f_{\mathcal{B}_i}(\tilde{x})$ after every s iterations, with $\tilde{x} = x^{k-1}$. Assume a batch gradient $\nabla f_{\mathcal{B}_j}$ is calculated at iteration k, we define the approximate variance-reduced gradient as:

$$\tilde{g}^k = \frac{b_j}{n}\nabla f_{\mathcal{B}_j}(x^k) - \frac{b_j}{n}\nabla f_{\mathcal{B}_j}(\tilde{x}) + \tilde{\mu}. \quad (2)$$

The approximate variance-reduced gradient satisfies the following inequality,

$$f^\star - F_{B_j}(x^k) \geq \langle x^\star - x^k, \tilde{g}^k \rangle, \quad (3)$$

where $F_{B_j}(x^k)$ is the approximate function value at iteration k:

$$F_{B_j}(x^k) = \frac{b_j}{n} f_{\mathcal{B}_j}(x^k) - \frac{b_j}{n} f_{\mathcal{B}_j}(\tilde{x}) + \tilde{\nu} - \langle \tilde{x} - x^k, \tilde{\mu} - \frac{b_j}{n}\nabla f_{\mathcal{B}_j}(\tilde{x})\rangle. \quad (4)$$

The solution is updated by taking a series of steps η_k along the approximate variance-reduced gradient directions $\tilde{g}^k$ as:

$$x^{k+1} = x^k - \eta_k \tilde{g}^k. \quad (5)$$

The approach to calculating the stepsize η_k will be presented in the next subsection. The proof of the property (3) of the approximate variance-reduced gradient can be found in Section A-A of the Appendix.

B. Accelerated stochastic Polyak stepsize with approximate variance-reduced gradients

Building upon the deterministic Polyak stepsizes as detailed in [32], we extend the method to a stochastic Polyak stepsize context:

$$\text{AccSPS}: \ \eta_k = \gamma \cdot \frac{F_{B_j}(x^k) - \hat{f}_k}{||\tilde{g}^k||^2}, 0 < \gamma < \bar{\gamma} < 2, \quad (6)$$

where $\hat{f}_k$ is the level value used to estimate $f^\star$, γ and $\bar{\gamma}$ are the scaling constants. The value of γ controls the value of step size. For problems where the gradients exhibit large variance, a smaller λ helps ensure more stable updates. In contrast, for problems with smaller gradient variance, a larger λ can be used to accelerate convergence. To avoid numerical issues resulting from potential division by a small number, an upper bound U is used for AccSPS. The level value estimation procedure will be introduced in the next subsection.

In contrast to the deterministic Polyak stepsize [15], AccSPS eliminates the need for evaluating function values $f(x^k)$ as well as subgradients g^k. Moreover, AccSPS operates without requiring knowledge of $f^\star$. As compared to the stochastic Polyak stepsize presented in [20], [22], AccSPS only requires additional previously obtained average function values $\tilde{\nu}$ and average gradient $\tilde{\mu}$, which are periodically updated and stored without incurring significant memory requirements.

C. Decision-guided level adjustment

To operationalize the level adjustment for stochastic Polyak stepsizes, the following result (due [32]) is used:

Theorem II.2. *Consider the sequence of solutions generated iteratively according to* (5) *is* $\{x^{k_j}, x^{k_j+1}, \ldots, x^{k_j+m_j-1}\}$, *with the corresponding approximate variance-reduced gradients as* $\{\tilde{g}^{k_j}, \tilde{g}^{k_j+1}, \ldots, \tilde{g}^{k_j+m_j-1}\}$. *The associated step sizes* $\{\eta_{k_j}, \eta_{k_j+1}, \cdots, \eta_{k_j+m_j-1}\}$ *are computed using level values* $\{\hat{f}_{k_j}, \hat{f}_{k_j+1}, \ldots, \hat{f}_{k_j+m_j-1}\}$ *via* (6). *If the following Polyak Stepsize Violation detection (PSVD) problem:*

$$\begin{cases} \langle x - x^{k_j}, \tilde{g}^{k_j} \rangle \leq -\frac{1}{\bar{\gamma}}\eta_{k_j}||\tilde{g}^{k_j}||^2, \\ \langle x - x^{k_j+1}, \tilde{g}^{k_j+1} \rangle \leq -\frac{1}{\bar{\gamma}}\eta_{k_j+1}||\tilde{g}^{k_j+1}||^2, \\ \cdots \\ \langle x - x^{k_j+m_j-1}, \tilde{g}^{k_j+m_j-1} \rangle \leq -\frac{1}{\bar{\gamma}}\eta_{k_j+m_j-1}||\tilde{g}^{k_j+m_j-1}||^2, \end{cases} \quad (7)$$

with $x \in \mathbb{R}^d$ *being continuous decision variables admits no feasible solution, then the level value can be updated as:*

$$\hat{f}_{k_j+m_j} = \frac{\gamma}{\bar{\gamma}}\hat{f}_{k_j+m_j-1} + (1 - \frac{\gamma}{\bar{\gamma}}) \min_{\kappa \in [k_j, k_j+m_j-1]} F_{B_j}(x^\kappa). \quad (8)$$

Moreover, $\hat{f}_{k_j} \leq \hat{f}_{k_j+m_j} \leq f^\star$.

There should be a reasonable gap between γ and $\bar{\gamma}$, which ensures that when the PSVD problem is infeasible, the level value can be sufficiently updated. The sensitivity to parameter $\bar{\gamma}$ is studied in Appendix B-A. In our numerical experiments, we set $\bar{\gamma} = 1.5\gamma$.

Theorem II.2 helps detect the violation of the Polyak stepsizes, as proved in [32], each time a system of linear inequalities (7) is violated, a tighter level value $\hat{f}_{k_j+m_j}$ is calculated. We adopt the dual simplex algorithm in the commercial solver Gurobi to check its feasibility. Machine learning applications frequently utilize a large number of parameters. As a result, (7) may contain a large number of decision variables and a significant number of constraints. To detect violation of (7), the number of variables considered is reduced to n_v, and feasibility checks of (7) are performed every M iterations; sensitivity with respect to both parameters is empirically validated in Section B-A of the Appendix demonstrating significant reduction of the CPU time without significant slow-down of level adjustments or convergence speed. A promising avenue for future research is to exploit the incremental introduction of constraints across iterations by employing the PSVD solution from one iteration as a warm start for the next.

D. AccSPS algorithm with level adjustment

A complete procedure of AccSPS for solving (1) is outlined in Algorithm 1. The level value $\hat{f}_0$ are initialized to 0, the approximate function value is initialized as $F_{B_j}(x^0) = \frac{1}{n}\sum_{i=1}^m b_i f_{\mathcal{B}_i}(x^0)$, and the approximate variance-reduced gradient is initialized as $\tilde{g}^0 = \frac{1}{n}\sum_{i=1}^m b_i \nabla f_{\mathcal{B}_i}(x^0)$. At each iteration, $\tilde{g}^k$ and $F_{B_j}(x^k)$ are updated according to (2) and (4). Then the PSVD problem is solved every M predetermined number of iterations. When the PSVD problem is infeasible, the level value $\hat{f}_k$ is adjusted following (8). Finally, the stepsize η_k and the solution x_{k+1} are updated according to (5)-(6).

Algorithm 1 (Loopless) AccSPS

```
1: Input: initial solution $x^0$
2: Initialize $\hat{f}_0 = 0$, $\tilde{x} = x^0$, $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^m b_i \nabla f_{\mathcal{B}_i}(\tilde{x})$, $\tilde{\nu} = \frac{1}{n}\sum_{i=1}^m b_i f_{\mathcal{B}_i}(\tilde{x})$.
3: for $k = 0$ to $K$ do
4:     if $(k+1) \bmod s = 0$ then
5:         Update $\tilde{x} = x^{k-1}$, $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^m b_i \nabla f_{\mathcal{B}_i}(\tilde{x})$, $\tilde{\nu} = \frac{1}{n}\sum_{i=1}^m b_i f_{\mathcal{B}_i}(\tilde{x})$.
6:     end if
7:     Update $\tilde{g}^k, F_{B_j}(x^k)$ using (2) and (4)
8:     $\eta_k \leftarrow \min\{\gamma \frac{F_{B_j}(x^k) - \hat{f}_k}{\|\tilde{g}^k\|^2}, U\}$
9:     if $k \bmod M = 0$ then
10:        Solve PSVD problem (7)
11:        if (7) is not feasible then
12:            Update level value $\hat{f}_k$ using (8)
13:        end if
14:    end if
15:    $x^{k+1} \leftarrow x^k - \eta_k \tilde{g}^k$
16: end for
```

III. Convergence analysis

This section presents major theoretical convergence results. Specifically, we provide convergence results for AccSPS within the approximate variance-reduced gradient framework, namely, convergence results are derived for Polyak stepsizes both with known $f^\star$ and unknown $f^\star$.

In the following convergence proofs, it is assumed that $f^\star = f(x^\star)$ and the approximate variance-reduced gradient is bounded from above:

Assumption III.1. (Approximate variance-reduced gradient boundness). For any k, there exists a scalar G, such that

$$\|\tilde{g}^k\| \le G. \tag{9}$$

Assumption III.1 is a common assumption in the analysis of stochastic Polyak stepsize methods. For example, both [20] and [21] make similar bounded gradient assumptions to establish convergence results in the context of non-smooth optimization problems. In practice, real-world problems may involve unbounded or highly variable gradients. To address this, one can employ gradient clipping or stepsize clipping strategies to effectively enforce boundedness during training.

In the following, the convergence rate with the Polyak stepsize is shown. The cases where $f^\star$ is known are considered first, followed by the cases where $f^\star$ is unknown. As in [32], to guarantee convergence, we assume that the following condition is satisfied.

Condition III.2. (Convergence condition). For any k, there exists $\varepsilon_k > 0$ such that if $F_{B_j}(x^k) < f(x^k)$, then $\hat{f}_k + \varepsilon_k \le F_{B_j}(x^k)$.

In the above, $\{\varepsilon_k\}_{k=1}^\infty$ can be any sequence that satisfies $\sum_{k=1}^\infty \varepsilon_k = \infty$. According to the definition in (4), $F_{B_j}(x^k)$ is always less than or equal to $f(x^k)$, a result that follows directly from the properties of convexity.

The convergence condition above is easy to satisfy. At each iteration, we can easily check whether the gap between $F_{B_j}(x^k)$ and $\hat{f}^k$ is greater than a pre-set value ε_k. In our experiments, we choose ε_k as a small constant. If not (which will not happen in our experiments), we move to the next mini-batch and recalculate $F_{B_j}(x^k)$. In the worst case, $\hat{f}_k + \varepsilon_k \le F_{B_j}(x^k)$ is not satisfied after solving all the mini-batch, then $F_{B_j}(x^k) = f(x^k)$.

Theorem III.3. *Suppose that solution x^k is updated by the approximate variance-reduced gradient with the stepsize calculated as* (6)*. If $\hat{f}_k = f^\star$, then $\forall k$*

$$\min_{\kappa \in \{0,1,\ldots,k\}} \{F_{B_j}(x^\kappa) - f^\star\} \le \frac{G}{\sqrt{k+1}\sqrt{2\gamma - \gamma^2}} \left\|x^0 - x^\star\right\|_2. \tag{10}$$

For $k > \frac{G^2 \cdot \|x^0 - x^\star\|_2^2}{\varepsilon_k^2 (2\gamma - \gamma^2)}$,

$$\min_{\kappa \in \{0,1,\ldots,k\}} \{f(x^\kappa) - f^\star\} \le \frac{G}{\sqrt{k+1}\sqrt{2\gamma - \gamma^2}} \left\|x^0 - x^\star\right\|_2. \tag{11}$$

The proof is presented in Section A-C of the Appendix. In the above theorem, it is shown that with the Polyak stepsize and the optimal value $f^\star$, the best function value obtained converges to the optimal value with a rate of $\mathcal{O}(1/\sqrt{k})$.

Next, we consider the situation where the level value is an underestimate of $f^\star$ and is dynamically adjusted. It has been shown in Theorem 3 of [32] that the level value converges to the optimal function value $\lim_{k\to\infty} \hat{f}_k = f^\star$. To derive the convergence rates, we first define a gap (approaching zero due to Thereom 4 of [32]) between the level value and the optimal function.

Theorem III.4. *Suppose that at iteration k_j, a new level value $\hat{f}_{k_j}$ is generated. Denote $\mu_{k_j} = f^\star - \hat{f}_{k_j} > 0$. If the solution is updated based on the approximate variance-reduced gradient and the stepsize is calculated per* (6) *before the next level* *update at k_{j+1}, then*

$$\min_{\kappa\in\{k_j,k_j+1,\dots,k\}} F_{B_j}(x^{\kappa}) - f^\star \leq \frac{\left\|x^{k_j}-x^\star\right\|_2^2}{(2-\gamma)\frac{\gamma}{G^2}\min(\mu_{k_j},\varepsilon_k)K} + \frac{\gamma U(f^\star-\hat{f}_{k_j})}{(2-\gamma)\frac{\gamma}{G^2}\min(\mu_{k_j},\varepsilon_k)}, \tag{12}$$

where $\frac{\gamma}{G^2}\min(\mu_{k_j},\varepsilon_k) > 0$ and $K = k - k_j + 1$.

The proof is presented in Section A-D of the Appendix. The first term of the right-hand side decreases as k increases. The value of the second term depends on the distance between the optimal function value and the level value. In the following lemma and theorem, we can establish the convergence rate for $k \to \infty$.

Lemma III.5. *As $k \to \infty$, the PSVD problem (7), there will become infeasible infinitely often. Denote the sequence of the iterations, when the level value is updated, as Υ, i.e.,*

$$\Upsilon = \{k_j\}_{j=1}^{\infty}. \tag{13}$$

The difference between any two consecutive elements of Υ is finite, i.e.,

$$k_{\kappa+1} - k_\kappa < \infty, \forall \kappa \geq 1. \tag{14}$$

The above Lemma has been shown in Lemma 4 of [32]. From the above lemma, we can always select a subsequence Ψ of Υ that $\lim_{i\to\infty}\Psi_{[i+1]} - \Psi_{[i]} = \infty$, where $\Psi_{[i]}$ denote the i^{th}element of Ψ.

Theorem III.6. *If the L-smooth condition of f holds and Ψ is any subsequence of Υ, the following inequality holds,*

$$\min_{\kappa\in\{l(k),l(k)+1,\dots,k\}} F_{B_j}(x^{\kappa}) - f^\star \leq \frac{\left\|x^{l(k)}-x^\star\right\|_2^2}{(2-\gamma)\gamma\min(\frac{\varepsilon_k}{G^2},\frac{1}{2L})\mathcal{K}(k)} + \frac{\gamma U(f^\star-\hat{f}_{l(k)})}{(2-\gamma)\gamma\min(\frac{\varepsilon_k}{G^2},\frac{1}{2L})}, \tag{15}$$

where $l(k)$ is the largest of the values in Ψ that is less than or equal to k and $\mathcal{K}(k) = k - l(k)$.

The proof is presented in Section A-E of the Appendix. As can be seen from (15), as $k \to \infty$, the first term of the right-hand side converges to zero with a rate of $\mathcal{O}(1/\mathcal{K}(k))$. The second term also converges to zero since the level value converges to the optimal function value.

In the stochastic optimization literature, it is common to assume that either $f(x)$ or each component function $f_i(x)$ is L-smooth, particularly in methods such as SGD and SVRG. While this assumption is commonly used to establish convergence rates in expectation, our analysis adopts a worst-case perspective: at each iteration, the convergence guarantee holds independently of the specific mini-batch, provided that Condition III.2 is satisfied. This underscores a fundamental distinction in how our method leverages the L-smoothness assumption compared to existing approaches.

IV. Dynamics in AccSPS: illustrative example

In this section, we study the dynamics of AccSPS in a small-scale non-interpolated regime to provide visualization. Here we consider a two-dimensional regularized problem as $f = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}(x-x_i^\star)^T H_i (x-x_i^\star)\right) + \frac{\lambda}{2}||x||^2$, where H_i is a random symmetric positive definite (SPD) matrix generated using the standard Gaussian matrix $A_i \in \mathbb{R}^{d\times 3d}$ as $H_i = A_i A_i^T/(3d)$ and $\lambda = 2$. The number of iterations is set to 300 for all methods and the data points $n = 2$.

As discussed in [22], when utilizing SPS [20], even with knowledge of $f_i^\star$, the convergence of $f(x^k)$ is biased due to the *correlation between η_k and $\nabla f_i(x^k)$* in the non-interpolated setting. Nevertheless, we briefly demonstrate that by incorporating knowledge of $f^\star$ and making certain modifications, SPS can converge to the exact solution. Specifically, we enforce a diminishing lower bound to maintain the positivity of the stepsize and substitute $f_i^\star$ with $f^\star$. In this scenario, the stepsize is defined as $\eta_k = \max\left\{\frac{1}{c_k}, \min\left\{\frac{f_i(x^k)-f^\star}{c_k||\nabla f_i(x^k)||^2}, \eta_b\right\}\right\}$, where $c_k = \mathcal{O}(\sqrt{k})$. To distinguish from the original SPS [20], we call this modified SPS as mSPS.

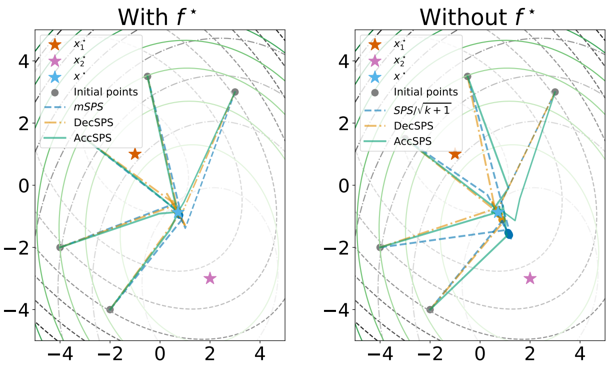


Fig. 1. Dynamic of AccSPS compared to mSPS (SPS) and DecSPS. In the case of SPS, we incorporate a diminishing multiplicative constant to anticipate a behavior similar to SGD. We visualize the contour lines of the respective landscapes, including the average landscape. The initial solutions are denoted by grey dots, while the final solutions for various methods are represented by dots in the same color.

In the following analysis, we first adopt conditions outlined by [22] to solve the two-dimensional regularized problem, specifically, employing a batch size of 1 and possessing the knowledge of $f^\star$. In this configuration, we substitute the level value in (6) with $f^\star$. As shown in the left subfigure of Figure 1, with the knowledge of $f^\star$, all three methods can converge to the exact solution.

In the right subfigure of Figure 1, we illustrate that, with a dynamically adjusted level value $\hat{f}_k$, AccSPS can converge to the exact solution even without knowledge of $f^\star$, as demonstrated by the proof in Theorem III.6. In contrast, SPS converges to a biased solution, and DecSPS converges to a very small neighborhood surrounding the exact solution due to a lack of knowledge of $f^\star$.

V. Numerical studies

In this section, we evaluate the performance of AccSPS on convex and non-convex problems in the non-interpolated setting. For convex problems, we choose binary classification tasks, with regularized logistic loss $f(x) = \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i\cdot a_i^T x)) + \frac{\lambda}{2}||x||^2$, where $a_i \in \mathbb{R}^d$ represents the feature vector, and $y_i \in \{-1,1\}$ is the target label for data point i. For non-convex problems, we address the regression problem for deep matrix factorization, as outlined in [20]: $\min_{W_1,W_2} \mathbb{E}_{x\sim N(0,I)}||W_2W_1x - Ax||^2$, where W_1, W_2 are weight matrices and A is a badly conditioned matrix. Additionally, we address image classification using deep neural networks. Here we use Gurobi 10.0.1 [37] to solve PSVD problems. All experiments were conducted on a desktop with an 8-core Intel Core i7-9800X CPU, except for image classification tasks, which were performed on a server equipped with a 32-core AMD Ryzen Threadripper 3970X 3.7GHz CPU and four NVIDIA GeForce RTX 2080 Ti GPUs with 10 Gigabit memories.

For the binary classification problem, we use two real-world datasets: A1A and Breast Cancer from [38] to evaluate the performance of AccSPS, with different regularization levels and batch sizes greater than 1 (A1A: batch size 128, $\lambda = 0.001$, Breast Cancer: batch size 32, $\lambda = 0.005$). Sensitivities of AccSPS to parameters γ, $\bar{\gamma}$, n_v, and M are shown in Section B-A of the Appendix. For the updating frequency of the average gradient, we choose $s = m/2$. For the pre-set value ε_k, we choose $\varepsilon_k = 1e^{-4}$.

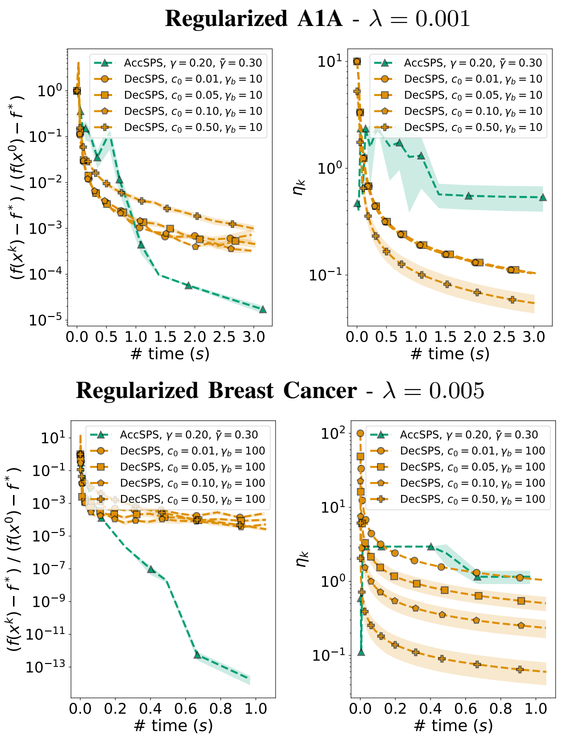


Fig. 2. Performance of AccSPS compared to DecSPS. Experiments are repeated 5 times and mean values and standard deviations are plotted.

Comparison with DecSPS. Now we evaluate the performance of AccSPS against DecSPS, with insights presented in Figure 2. For a fair comparison, we report the relative loss of both methods over wall-clock time rather than iterations. For DecSPS, [22] suggests using a large upper bound for the stepsize under light regularization. The left subfigures distinctly illustrate AccSPS's superior performance over DecSPS. Meanwhile, the right subfigures emphasize the noteworthy characteristic of AccSPS—a dynamically adjusting stepsize that responds adaptively to both loss and gradient dynamics. The "plateau" regions for the stepsizes occur because, for the Breast Cancer dataset, the approximate variance-reduced gradients quickly approach zero. This results in a significant increase in stepsize, which is then upper bounded by U.

Comparison with other optimizers. We demonstrate the effectiveness of AccSPS by comparing it with other widely used optimizers. Following [22], the comparison is performed with respect to SGD, AdaGrad [11], Adam [8], and AMSGrad [13]. For these optimizers, we use a linear-decay schedule for the step sizes. The selection of different regularization levels for the A1A dataset is outlined in Table I. Parameters for the non-AccSPS optimizers are fine-tuned to achieve optimal performance.

To illustrate the convergence dynamics, we present the relative loss at different wall-clock times in Table I. While during initial iterations, certain non-AccSPS optimizers may attain a lower relative loss than AccSPS, as the iterations progress, the relative loss obtained by AccSPS becomes markedly smaller outperforming other methods by several orders of magnitude.

TABLE I
EVALUATION OF OPTIMIZERS ON DIFFERENT DATASETS

Regularized A1A Dataset

Optimizer	$\lambda = 0.001$		$\lambda = 0.01$	
	$t = 1.5s$	$t = 3.0s$	$t = 1.5s$	$t = 3.0s$
SGD	$3.51e^{-4}$	$2.04e^{-4}$	$2.79e^{-4}$	$2.52e^{-5}$
AdaGrad	$5.76e^{-4}$	$3.16e^{-4}$	$1.70e^{-4}$	$4.79e^{-5}$
Adam	$2.91e^{-3}$	$6.99e^{-4}$	$1.43e^{-3}$	$2.58e^{-4}$
AMSGrad	$1.41e^{-3}$	$5.44e^{-4}$	$1.12e^{-3}$	$3.05e^{-4}$
DecSPS	$7.73e^{-4}$	$3.25e^{-4}$	$2.50e^{-4}$	$7.91e^{-5}$
AccSPS	$\mathbf{5.71e^{-5}}$	$\boxed{\mathbf{1.72e^{-5}}}$	$\mathbf{5.73e^{-12}}$	$\boxed{\mathbf{2.58e^{-14}}}$

Regularized Breast Cancer Dataset

Optimizer	$\lambda = 0.005$		$\lambda = 0.05$	
	$t = 0.5s$	$t = 1.0s$	$t = 0.5s$	$t = 0.8s$
SGD	$9.84e^{-6}$	$3.44e^{-6}$	$1.67e^{-5}$	$2.79e^{-6}$
AdaGrad	$3.33e^{-5}$	$2.43e^{-5}$	$1.93e^{-5}$	$3.78e^{-6}$
Adam	$9.47e^{-5}$	$9.42e^{-6}$	$2.49e^{-5}$	$2.26e^{-5}$
AMSGrad	$6.19e^{-5}$	$7.94e^{-6}$	$2.93e^{-5}$	$8.03e^{-6}$
DecSPS	$8.60e^{-5}$	$5.20e^{-5}$	$\mathbf{1.35e^{-5}}$	$1.04e^{-5}$
AccSPS	$\mathbf{5.52e^{-13}}$	$\boxed{\mathbf{1.82e^{-14}}}$	$2.25e^{-5}$	$\boxed{\mathbf{1.83e^{-14}}}$

Non-convex deep matrix factorization. Following the experimental setup detailed in [17], [39], we select $A \in \mathbb{R}^{20\times 12}$ with a condition number $\kappa(A) = 10^{10}$ and generate a consistent dataset consisting of 5,000 samples. We vary the rank k of the matrix $W_1 \in \mathbb{R}^{k\times 12}$ and $W_2 \in \mathbb{R}^{20\times k}$ to regulate the interpolation degree of the problem. For $k < 20$, the interpolation condition is not satisfied, and $f^\star > 0$. As illustrated in Figure 3, AccSPS demonstrates superior performance compared to DecSPS. AccSPS requires a smaller scaling constant γ to achieve fast convergence. For all the cases considered with $k = 6$, AccSPS achieves the fastest decrease in loss thus outperforming DecSPS.

Non-convex image classification using deep neural networks. Here, we address non-convex multi-class image classification problems using deep neural networks. Our experiments are conducted on the CIFAR-10 dataset, utilizing

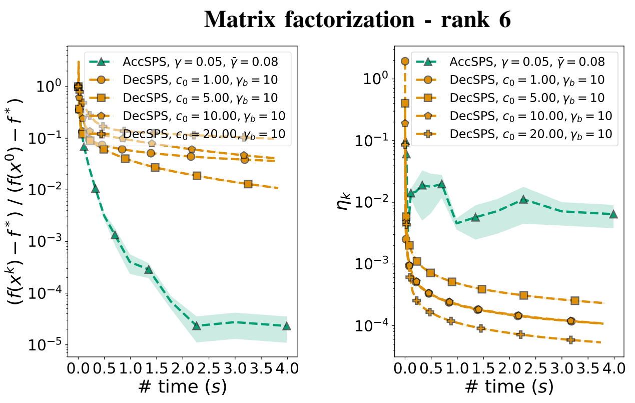


Fig. 3. Performance of AccSPS compared to DecSPS on matrix factorization datasets. Experiments are repeated 5 times and mean values and standard deviations (std) are plotted. Different scale factors are applied to std for better visualization.

the standard ResNet-18 architecture [40]. We use the cross-entropy loss as the loss function and add l_2 regularization to the objective function (1). We split the CIFAR-10 dataset into 50,000 training samples and 10,000 test samples with a batch size of 128. In addition to DecSPS, we include SPS [20], SVRG [2], AdaSVRG [41], SGD, and Adam as benchmark methods. For the optimizers with non-adaptive stepsize formulas (SVRG, AdaSVRG, SGD, and Adam), we employ a linear-decay schedule for the stepsize. For a fair comparison, we set γ and the upper bound η_b of AccSPS and SPS to the same values. To save computation time for AccSPS, SVRG, and AdaSVRG, we use a larger batch size when updating the average gradient $\tilde{\mu}$ and set $s = m$.

As shown in Figure 4, AccSPS achieves the fastest convergence rate among all optimizers. Compared to DecSPS, our proposed AccSPS achieves lower training loss and higher test accuracy, demonstrating its superior ability to handle non-convex deep learning problems. Compared to SPS, AccSPS maintains much smaller step sizes when the regularization level is high. This is due to AccSPS's ability to dynamically adjust the level value, leading to more efficient optimization. Compared to SVRG, AdaSVRG, SGD, and Adam, AccSPS achieves a much faster reduction in training loss.

VI. Conclusion

To reduce computational and memory requirements when training machine learning models, we introduced an approximate variance-reduced gradient method that employs an approximate steepest gradient descent direction. By applying incremental averaging, this approach effectively reduces the variance of stochastic updates, resulting in more stable convergence under noisy conditions. To achieve fast convergence, we employed Polyak stepsize with dynamically adjusted level values $\hat{f}_k$ to iteratively estimate the optimal value $f^\star$ for the entire problem using the decision-guided level adjustment approach. Leveraging the approximate variance-reduced gradient and the level value selection scheme, we then provide a theoretical analysis of an accelerated variant of the resulting stochastic Polyak stepsize (AccSPS). This approach demonstrates convergence to the exact solution without relying on the interpolation assumption in convex stochastic problems. Numerical studies indicate that our proposed AccSPS outperforms several benchmark algorithms such as DecSPS, AdaGrad, Adam, AMSGrad, and SVRG. Key potential directions of future research involve establishing proofs in non-convex settings and expanding experimental applications to larger deep-learning models and datasets.

Appendix A
Proofs of Main Theorems

Within this section, we provide the proofs for the primary theorems outlined in the main document.

A. *Proof of property* (3) *in Definition II.1*

Proof. Multiplying the approximate variance-reduced gradient $\tilde{g}^k$ by $x^\star - x^k$ leads to equation (16):

Since we assume the dataset is partitioned into m predetermined batches, it holds that $\frac{1}{n}\sum_{i=1}^{m} b_i f_{\mathcal{B}_i}(x^\star) = f(x^\star)$, assume $f(x^\star) = f^\star$, then:

$$\langle x^\star - x^k, \tilde{g}^k \rangle \le f^\star - F_{B_j}(x^k) \tag{17}$$

and thus completes the proof. □

B. *Proof of Theorem II.2*

Before the proof of Theorem II.2, we first prove the following two lemmas.

Lemma A.1. *Suppose that at iteration k, x^k represents the solution, $\tilde{g}^k$ denotes the approximate variance-reduced gradient at x^k, and η_k is the step size. Given $x^\star \in \mathcal{X}^\star$, if η_k fails to satisfy the following inequality:*

$$\langle x^\star - x^k, \tilde{g}^k \rangle \le -\frac{1}{\bar{\gamma}}\eta_k ||\tilde{g}^k||^2 \tag{18}$$

then:

$$\eta_k > \bar{\gamma} \cdot \frac{F_{B_j}(x^k) - f^\star}{||\tilde{g}^k||^2} \tag{19}$$

Proof. Assume that (18) is violated but $\eta_k \le \bar{\gamma} \cdot \frac{F_{B_j}(x^k) - f^\star}{||\tilde{g}^k||^2}$. From the latter, the following inequality holds:

$$-\frac{1}{\bar{\gamma}}\eta_k ||\tilde{g}^k||^2 \ge f^\star - F_{B_j}(x^k) \tag{20}$$

According to (3), the above inequality can be further written as:

$$-\frac{1}{\bar{\gamma}}\eta_k ||\tilde{g}^k||^2 \ge f^\star - F_{B_j}(x^k) \ge \langle x^\star - x^k, \tilde{g}^k \rangle \tag{21}$$

this means that (18) is satisfied, so we complete the proof by contradiction. □

Lemma A.2. *Suppose that at iteration k, x^k is the solution, and $\tilde{g}^k$ represents the approximate variance-reduced gradient at x^k. The step size η_k is computed using a level value $\hat{f}_k$ where $\hat{f}_k < f^\star$ according to* (6)*. If the inequality* (18) *is not fulfilled, then a scalar $\hat{f}'$ is defined as:*

$$\hat{f}' = \frac{\gamma}{\bar{\gamma}}\hat{f}_k + (1 - \frac{\gamma}{\bar{\gamma}})F_{B_j}(x^k) \tag{22}$$

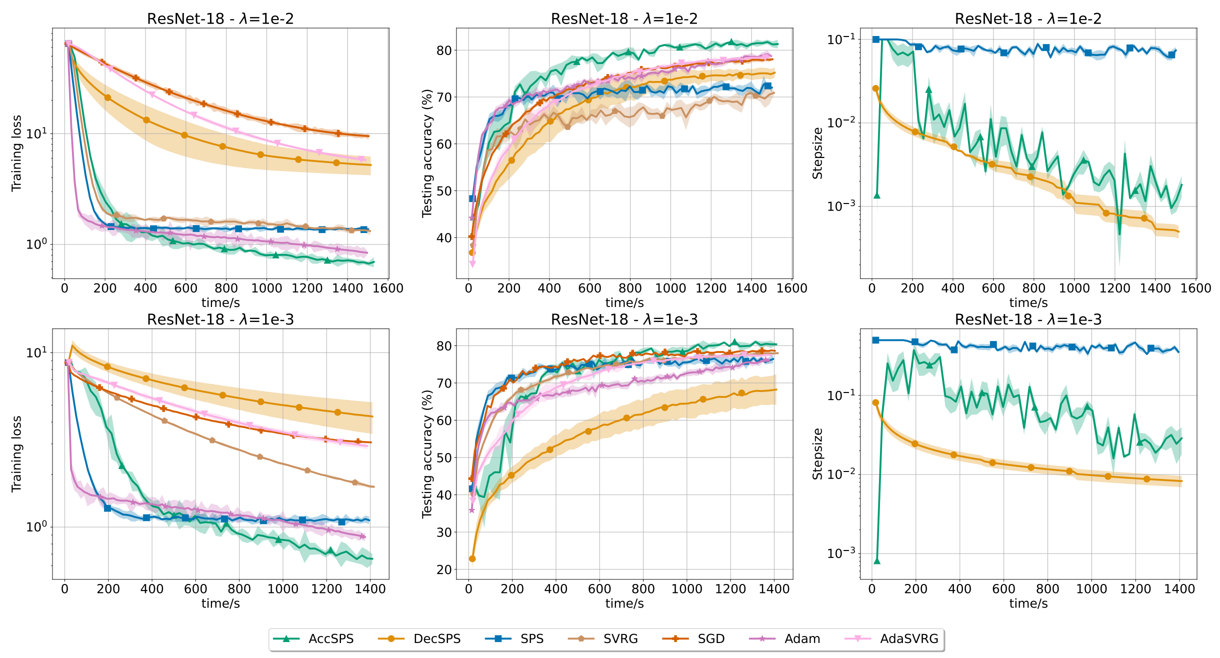


Fig. 4. performance comparison of various optimizers on CIFAR-10 dataset using ResNet-18 under different regularization levels. Experiments are repeated 5 times and mean values and standard deviations are plotted. The number of training epochs is 100 for all optimizers, and the plots are truncated to maintain consistency within the time frame. Here the testing accuracy is top-1 accuracy.

$$\begin{aligned}
\langle x^\star - x^k, \tilde{g}^k\rangle &= \langle x^\star - x^k, \frac{b_j}{n}\nabla f_{\mathcal{B}_j}(x^k) - \frac{b_j}{n}\nabla f_{\mathcal{B}_j}(\tilde{x}) + \tilde{\mu}\rangle \\
&= \langle x^\star - x^k, \frac{b_j}{n}\nabla f_{\mathcal{B}_j}(x^k) + \frac{1}{n}\sum_{i,i\neq j}^{m} b_i \nabla f_{\mathcal{B}_i}(\tilde{x})\rangle \\
&= \langle x^\star - x^k, \frac{b_j}{n}\nabla f_{\mathcal{B}_j}(x^k)\rangle + \langle x^\star - x^k, \frac{1}{n}\sum_{i,i\neq j}^{m} b_i \nabla f_{\mathcal{B}_i}(\tilde{x})\rangle \\
&\leq \frac{b_j}{n} f_{\mathcal{B}_j}(x^\star) - \frac{b_j}{n} f_{\mathcal{B}_j}(x^k) + \langle x^\star - \tilde{x} + \tilde{x} - x^k, \frac{1}{n}\sum_{i,i\neq j}^{m} b_i \nabla f_{\mathcal{B}_i}(\tilde{x})\rangle \\
&= \frac{b_j}{n} f_{\mathcal{B}_j}(x^\star) - \frac{b_j}{n} f_{\mathcal{B}_j}(x^k) + \frac{1}{n}\sum_{i,i\neq j}^{m} b_i \langle x^\star - \tilde{x}, \nabla f_{\mathcal{B}_i}(\tilde{x})\rangle + \langle \tilde{x} - x^k, \frac{1}{n}\sum_{i,i\neq j}^{m} b_i \nabla f_{\mathcal{B}_i}(\tilde{x})\rangle \\
&\leq \frac{b_j}{n} f_{\mathcal{B}_j}(x^\star) - \frac{b_j}{n} f_{\mathcal{B}_j}(x^k) + \frac{1}{n}\sum_{i,i\neq j}^{m} b_i \left(f_{\mathcal{B}_i}(x^\star) - f_{\mathcal{B}_i}(\tilde{x})\right) + \langle \tilde{x} - x^k, \tilde{\mu} - \frac{b_j}{n} f_{\mathcal{B}_j}(\tilde{x})\rangle \\
&= \frac{b_j}{n} f_{\mathcal{B}_j}(x^\star) + \frac{1}{n}\sum_{i,i\neq j}^{m} b_i f_{\mathcal{B}_i}(x^\star) - \frac{b_j}{n} f_{\mathcal{B}_j}(x^k) - \tilde{\nu} + \frac{b_j}{n} f_{\mathcal{B}_j}(\tilde{x}) + \langle \tilde{x} - x^k, \tilde{\mu} - \frac{b_j}{n} f_{\mathcal{B}_j}(\tilde{x})\rangle \\
&= \frac{1}{n}\sum_{i=1}^{m} b_i f_{\mathcal{B}_i}(x^\star) - \left(\frac{b_j}{n} f_{B_j}(x^k) - \frac{b_j}{n} f_{\mathcal{B}_j}(\tilde{x}) + \tilde{\nu} - \langle \tilde{x} - x^k, \tilde{\mu} - \frac{b_j}{n} f_{\mathcal{B}_j}(\tilde{x})\rangle\right) \\
&= \frac{1}{n}\sum_{i=1}^{m} b_i f_{\mathcal{B}_i}(x^\star) - F_{B_j}(x^k)
\end{aligned} \tag{16}$$

is a tighter estimation of $f^\star$ such that:

$$\hat{f}_k < \hat{f}' < f^\star \tag{23}$$

Proof. We start by proving $\hat{f}_k < \hat{f}'$. According to condition III.2, we have that

$$\hat{f}_k = \frac{\gamma}{\bar{\gamma}}\hat{f}_k + (1 - \frac{\gamma}{\bar{\gamma}})\hat{f}_k < \frac{\gamma}{\bar{\gamma}}\hat{f}_k + (1 - \frac{\gamma}{\bar{\gamma}})F_{B_j}(x^k) = \hat{f}' \tag{24}$$

Then we prove $\hat{f}' < f^\star$. Building upon Lemma A.1, as inequality (18) remains unsatisfied, (19) holds true. Substituting η_k in (19) with (6), we obtain:

$$\gamma \cdot \frac{F_{B_j}(x^k) - \hat{f}_k}{||\tilde{g}^k||^2} > \bar{\gamma} \cdot \frac{F_{B_j}(x^k) - f^\star}{||\tilde{g}^k||^2} \tag{25}$$

Upon canceling $||\tilde{g}^k||^2$ and reorganizing the remaining terms, the aforementioned inequality can be expressed equivalently as:

$$\hat{f}' = \frac{\gamma}{\bar{\gamma}}\hat{f}_k + (1 - \frac{\gamma}{\bar{\gamma}})F_{B_j}(x^k) < f^\star \tag{26}$$

thus we complete the proof. □

Now we can prove Theorem II.2 as:

Proof. Given that (7) has no feasible solution, it implies the existence of $x^\star \in \mathcal{X}^\star$ that is infeasible for (7). Consequently, we can deduce that there exists $\kappa \in [k_j, k_j + m_j - 1]$ such that the following inequality holds:

$$\langle x^\star - x^\kappa, \tilde{g}^\kappa \rangle > -\frac{1}{\bar{\gamma}}\eta_\kappa ||\tilde{g}^\kappa||^2 \tag{27}$$

Based on Lemma A.2, we have:

$$\frac{\gamma}{\bar{\gamma}}\hat{f}_{k_j+m_j-1} + (1 - \frac{\gamma}{\bar{\gamma}})F_{B_j}(x^\kappa) < f^\star \tag{28}$$

thus it holds that:

$$\hat{f}_{k_j+m_j} = \frac{\gamma}{\bar{\gamma}}\hat{f}_{k_j+m_j-1} + (1 - \frac{\gamma}{\bar{\gamma}}) \min_{\kappa \in [k_j, k_j+m_j-1]} F_{B_j}(x^\kappa) < f^\star \tag{29}$$

and according to Lemma A.2, $\hat{f}_{k_j+m_j} > \hat{f}_{k_j+m_j-1}$ is a tighter estimation of $f^\star$.

□

C. Proof of Theorem III.3

Theorem III.3. Suppose that x^k is updated by the approximate variance-reduced gradient with the stepsize calculated as in (6). If $\hat{f}_k = f^\star$, then for all k, the following inequality holds

$$\min_{\kappa \in \{0,1,\ldots,k\}} F_{B_j}(x^\kappa) - f^\star \leq \frac{G}{\sqrt{k+1}\sqrt{2\gamma - \gamma^2}} \left\|x^0 - x^\star\right\|_2 \tag{30}$$

For $k > \frac{G^2 \cdot \left\|x^0 - x^\star\right\|_2^2}{\varepsilon_k^2 (2\gamma - \gamma^2)}$, the following holds:

$$\min_{\kappa \in \{0,1,\ldots,k\}} f(x^\kappa) - f^\star \leq \frac{G}{\sqrt{k+1}\sqrt{2\gamma - \gamma^2}} \left\|x^0 - x^\star\right\|_2 \tag{31}$$

Proof. From the update formula and the property of the approximate variance-reduced gradient (3), the following equality holds:

$$\begin{aligned} \left\|x^{k+1} - x^\star\right\|_2^2 &= \left\|x^k - \eta_k \tilde{g}^k - x^\star\right\|_2^2 \\ &= \left\|x^k - x^\star\right\|_2^2 - 2\eta_k \langle \tilde{g}^k, x^k - x^\star \rangle + \eta_k^2 \left\|\tilde{g}^k\right\|_2^2 \\ &\leq \left\|x^k - x^\star\right\|_2^2 - 2\eta_k \left(F_{B_j}(x^k) - f(x^\star)\right) + \eta_k^2 \left\|\tilde{g}^k\right\|_2^2. \end{aligned} \tag{32}$$

By recursively applying this inequality, we obtain:

$$\begin{aligned} \left\|x^{k+1} - x^\star\right\|_2^2 \leq \left\|x^0 - x^\star\right\|_2^2 - \sum_{\kappa=0}^{k} 2\eta_\kappa \left(F_{B_j}(x^\kappa) - f(x^\star)\right) \\ + \sum_{\kappa=0}^{k} \eta_\kappa^2 \left\|\tilde{g}^\kappa\right\|_2^2 \end{aligned} \tag{33}$$

Considering that $f^\star = f(x^\star)$ and using the stepsize (6), we have

$$\begin{aligned} \left\|x^{k+1} - x^\star\right\|_2^2 \leq \left\|x^0 - x^\star\right\|_2^2 \\ + (\gamma^2 - 2\gamma) \cdot \sum_{\kappa=0}^{k} \frac{(F_{B_j}(x^\kappa) - f^\star)^2}{\left\|\tilde{g}^\kappa\right\|_2^2} \end{aligned} \tag{34}$$

Since $\left\|x^{k+1} - x^\star\right\|_2^2 \geq 0$, we can further rewrite the inequality as:

$$\sum_{\kappa=0}^{k} (F_{B_j}(x^\kappa) - f^\star)^2 \leq G^2 \cdot \frac{\left\|x^0 - x^\star\right\|_2^2}{(2\gamma - \gamma^2)}. \tag{35}$$

Thus, we obtain

$$\min_{\kappa \in \{0,1,\ldots,k\}} F_{B_j}(x^\kappa) - f^\star \leq \frac{G}{\sqrt{k+1}\sqrt{2\gamma - \gamma^2}} \cdot \left\|x^0 - x^\star\right\|_2 \tag{36}$$

As $k \to \infty$, the right-hand side term approaches zero. Since the convergence condition III.2 is satisfied, for any κ, either $f^\star + \varepsilon_\kappa \leq F_{B_j}(x^\kappa) < f(x^\kappa)$ or $F_{B_j}(x^\kappa) = f(x^\kappa)$ is satisfied. Therefore, for $k > \frac{G^2 \cdot \left\|x^0 - x^\star\right\|_2^2}{\varepsilon_k^2 (2\gamma - \gamma^2)}$, we must have

$$\min_{\kappa \in \{0,1,\ldots,k\}} f(x^\kappa) - f^\star \leq \frac{G}{\sqrt{k+1}\sqrt{2\gamma - \gamma^2}} \cdot \left\|x^0 - x^\star\right\|_2 \tag{37}$$

The above inequality indicates that the number of iterations required to guarantee a suboptimality of δ is $\frac{G^2 \left\|x^0 - x^\star\right\|_2^2}{\delta^2 (2\gamma - \gamma^2)}$. □

D. Proof of Theorem III.4

Theorem III.4. Suppose that at iteration k_j, a new level value $\hat{f}_{k_j}$ is generated, and let $\mu_{k_j} = f^\star - \hat{f}_{k_j} > 0$. If the solution is updated using the approximate variance-reduced gradient and the stepsize is calculated according to (6) before the next level update at k_{j+1}, then the following inequality holds

$$\begin{aligned} \min_{\kappa \in \{k_j, k_j+1, \ldots, k\}} F_{B_j}(x^\kappa) - f^\star \leq \frac{\left\|x^{k_j} - x^\star\right\|_2^2}{(2-\gamma)\beta K} \\ + \frac{\gamma U (f^\star - \hat{f}_{k_j})}{(2-\gamma)\beta}, \end{aligned} \tag{38}$$

where $\beta = \frac{\gamma}{G^2}\min(\mu_{k_j}, \varepsilon_k) > 0$ and $K = k - k_j + 1$.

Proof. Since the convergence condition III.2 is satisfied, for a level value $\hat{f}_{k_j}$ and $k > k_j$, either $\hat{f}_{k_j} + \varepsilon_k \leq F_{B_j}(x^k) < f(x^k)$ or $F_{B_j}(x^k) = f(x^k)$ is satisfied. Therefore, the stepsize η_k satisfies

$$\eta_k = \gamma \cdot \frac{F_{B_j}(x^k) - \hat{f}_{k_j}}{||\tilde{g}^k||^2} \geq \frac{\gamma \varepsilon_k}{G^2}, \tag{39}$$

or

$$\eta_k = \gamma \cdot \frac{f(x^k) - \hat{f}_k}{||\tilde{g}^k||^2} \geq \gamma \cdot \frac{f^\star - \hat{f}_k}{||\tilde{g}^k||^2} = \frac{\gamma \mu_{k_j}}{G^2}, \tag{40}$$

Thus, we obtain the lower bound on the step size

$$\eta_k \geq \frac{\gamma}{G^2}\min(\mu_{k_j}, \varepsilon_k) = \beta, \tag{41}$$

Using the update formula, we have the following inequality:

$$\left\|x^{k+1}-x^\star\right\|_2^2 \le \left\|x^k-x^\star\right\|_2^2 - 2\eta_k\left(F_{B_j}(x^k)-f(x^\star)\right)+\eta_k^2\left\|\tilde{g}^k\right\|_2^2 \tag{42}$$

Rearranging the terms, we obtain

$$\begin{aligned}&2\eta_k\left(F_{B_j}(x^k)-f(x^\star)\right)\\ &\le \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\eta_k^2\left\|\tilde{g}^k\right\|_2^2\\ &\overset{(6)}{=} \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\gamma\eta_k(F_{B_j}(x^k)-\hat{f}_k)\\ &= \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\gamma\eta_k(F_{B_j}(x^k)-f(x^\star))\\ &\quad+\gamma\eta_k(f(x^\star)-\hat{f}_k)\end{aligned} \tag{43}$$

Since $f(x^\star)=f^\star$, the following inequality holds

$$(2-\gamma)\eta_k\left(F_{B_j}(x^k)-f^\star\right)\le \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2 +\gamma\eta_k(f^\star-\hat{f}_k) \tag{44}$$

By recursively applying this inequality, we obtain

$$\begin{aligned}&(2-\gamma)\sum_{\kappa\in\{k_j,\dots,k\}}\eta_\kappa\left(F_{B_j}(x^\kappa)-f^\star\right)\\ &\le \left\|x^{k_j}-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\gamma\sum_{\kappa\in\{k_j,\dots,k\}}\eta_\kappa(f^\star-\hat{f}_\kappa)\\ &\le \left\|x^{k_j}-x^\star\right\|_2^2+\gamma\sum_{\kappa\in\{k_j,\dots,k\}}\eta_\kappa(f^\star-\hat{f}_\kappa)\end{aligned} \tag{45}$$

Applying the lower bound on the step size, we obtain

$$\begin{aligned}&(2-\gamma)\beta K\cdot\left(\min_{\kappa\in\{k_j,\dots,k\}}F_{B_j}(x^\kappa)-f^\star\right)\\ &\le \left\|x^{k_j}-x^\star\right\|_2^2+\gamma\sum_{\kappa\in\{k_j,\dots,k\}}(\eta_\kappa)(f^\star-\min_{\kappa\in\{k_j,\dots,k\}}\hat{f}_\kappa)\\ &\le \left\|x^{k_j}-x^\star\right\|_2^2+\gamma\sum_{\kappa\in\{k_j,\dots,k\}}(\eta_\kappa)(f^\star-\hat{f}_{k_j})\end{aligned} \tag{46}$$

where $K=k-k_j$. Simplifying further, we obtain

$$\min_{\kappa\in\{k_j,k_j+1,\dots,k\}}F_{B_j}(x^\kappa)-f^\star\le\frac{\left\|x^{k_j}-x^\star\right\|_2^2}{(2-\gamma)\beta K}+\frac{\gamma U(f^\star-\hat{f}_{k_j})}{(2-\gamma)\beta}. \tag{47}$$

This completes the proof. □

E. Proof of Theorem III.6

Theorem III.6. Suppose the L-smooth condition of f holds and let Ψ be any subsequence of Υ. Then, for each k, the following inequality holds,

$$\begin{aligned}\min_{\kappa\in\{l(k),l(k)+1,\dots,k\}}F_{B_j}(x^\kappa)-f^\star&\le\frac{\left\|x^{l(k)}-x^\star\right\|_2^2}{(2-\gamma)\gamma\min(\frac{\varepsilon_k}{G^2},\frac{1}{2L})\mathcal{K}(k)}\\ &\quad+\frac{\gamma U(f^\star-\hat{f}_{l(k)})}{(2-\gamma)\gamma\min(\frac{\varepsilon_k}{G^2},\frac{1}{2L})},\end{aligned} \tag{48}$$

where $l(k)$ is the largest of the values in Ψ that is less than or equal to k and $\mathcal{K}(k)=k-l(k)$.

Proof. Since the convergence condition III.2 is satisfied, for a level value $\hat{f}_{k_j}$ and $k_{j+1}>k>k_j$, either $\hat{f}_{k_j}+\varepsilon_k\le F_{B_j}(x^k)<f(x^k)$ or $F_{B_j}(x^k)=f(x^k)$ is satisfied. Therefore, either the stepsize η_k satisfies

$$\eta_k=\gamma\cdot\frac{F_{B_j}(x^k)-\hat{f}_{l(k)}}{||\tilde{g}^k||^2}\ge\frac{\gamma\varepsilon_k}{G^2}, \tag{49}$$

or

$$\eta_k=\gamma\cdot\frac{f(x^k)-\hat{f}_{k_j}}{||\tilde{g}^k||^2}\ge\gamma\cdot\frac{f(x^k)-f^\star}{||\tilde{g}^k||^2}=\frac{\gamma}{2L}, \tag{50}$$

Hence, in all cases

$$\eta_k\ge\gamma\min(\frac{\varepsilon_k}{G^2},\frac{1}{2L})=\theta, \tag{51}$$

Using the same reasoning as in Section A-D, we write

$$\begin{aligned}&2\eta_k\left(F_{B_j}(x^k)-f^\star\right)\\ &\le \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\eta_k^2\left\|\tilde{g}^k\right\|_2^2\\ &\overset{(6)}{=} \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\gamma\eta_k(F_{B_j}(x^k)-\hat{f}_k)\\ &\le \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\gamma\eta_k(F_{B_j}(x^k)-\hat{f}_{l(k)})\\ &= \left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2+\gamma\eta_k(F_{B_j}(x^k)-f(x^\star))\\ &\quad+\gamma\eta_k(f^\star-\hat{f}_{l(k)})\end{aligned} \tag{52}$$

Rearranging gives

$$(2-\gamma)\eta_k\left(F_{B_j}(x^k)-f^\star\right)\le\left\|x^k-x^\star\right\|_2^2-\left\|x^{k+1}-x^\star\right\|_2^2 +\gamma\eta_k(f^\star-\hat{f}_{l(k)}). \tag{53}$$

Then, by applying the above inequality iteratively, we have

$$\sum_{\kappa=l(k)}^{k}(2-\gamma)\eta_\kappa\left(F_{B_j}(x^\kappa)-f^\star\right)\le\left\|x^{l(k)}-x^\star\right\|_2^2 +\sum_{\kappa=l(k)}^{k}\gamma\eta_\kappa(f^\star-\hat{f}_{l(k)}). \tag{54}$$

Using the lower bound and upper bound of the stepsize, it follows that

$$\begin{aligned}&(2-\gamma)\theta\mathcal{K}(k)\cdot\left(\min_{\kappa\in\{l(k),l(k)+1,\dots,k\}}F_{B_j}(x^\kappa)-f^\star\right)\\ &\le\left\|x^{l(k)}-x^\star\right\|_2^2+\mathcal{K}(k)\gamma U(f(x^\star)-\hat{f}_{l(k)}),\end{aligned} \tag{55}$$

which is

$$\begin{aligned}\min_{\kappa\in\{l(k),l(k)+1,\dots,k\}}F_{B_j}(x^\kappa)-f^\star&\le\frac{\left\|x^{l(k)}-x^\star\right\|_2^2}{(2-\gamma)\theta\mathcal{K}(k)}\\ &\quad+\frac{\gamma U(f^\star-\hat{f}_{l(k)})}{(2-\gamma)\theta}.\end{aligned} \tag{56}$$

□

APPENDIX B
ADDITIONAL EXPERIMENT RESULTS

A. *Sensitivity of AccSPS to parameter γ, $\bar{\gamma}$, n_v, and M*

In this section, we examine the sensitivity to the scaling constant γ and $\bar{\gamma}$, the number of variables n_v, as well as the updating frequency M in solving the PSVD problem (7).

Sensitivity to γ. AccSPS involves a crucial hyperparameter, the scaling constant γ. To assess its sensitivity, we conduct experiments on the A1A dataset while maintaining $\bar{\gamma} = 1.5\gamma$. As depicted in the left panel of Figure 5, the choice of $\gamma = 0.2$ yields the most rapid reduction in loss for the A1A dataset. Although the convergence analysis allows any $\gamma < 2$, we empirically found that smaller values (e.g., $\gamma = 0.2$) lead to faster and more stable convergence in practice. This conservative choice helps reduce step size variance and avoids instability due to inaccurate level estimates in early iterations.

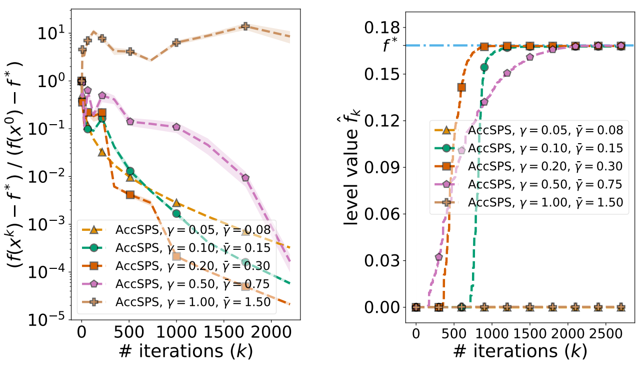


Fig. 5. Sensitivity of AccSPS to γ on the A1A dataset ($\lambda = 0.001$). Experiments are repeated 5 times and mean values and standard deviations are plotted.

Sensitivity to $\bar{\gamma}$. To test the sensitivity to $\bar{\gamma}$, the parameter γ is held constant at $\gamma = 0.2$ empirically, and various values of $\bar{\gamma}$ are selected. The results are shown in Figure 6. Based on empirical observations, it is evident that the performance of AccSPS is not significantly affected by the choice of $\bar{\gamma}$. The rightest subfigure in Figure 6 illustrates the dynamic behavior of the level value. Level values converge to $f^\star$ for all considered values of $\bar{\gamma}$. We opt for $\bar{\gamma} = 0.3$, corresponding to setting $\bar{\gamma} = 1.5\gamma$ with $\gamma = 0.2$.

Sensitivity to n_v. The number of continuous variables n_v in the PSVD problem should align with the dimension of the data points in the dataset. However, we have the flexibility to reduce n_v to save computation time.

As depicted in Figure 7, it is evident that the decreasing rates of loss of AccSPS with varying n_v remain consistently close. Moreover, the level value $\hat{f}_k$ converges to $f^\star$ across different values of n_v. Reducing the value of n_v accelerates the convergence of the level value $\hat{f}_k$, resulting in faster convergence to $f^\star$ and a reduction in computation time for solving the PSVD problem. In practical applications, we can pick $n_v = 2$ for rapid computation without impeding the decreasing rate of loss.

Table II presents the computation time for solving the PSVD problem and its percentage of the total computation time with different n_v. Here we fix the updating frequency $M = 20$. We can see that a smaller n_v results in a shorter total computation time. Additionally, the percentage of time spent on PSVD relative to the total time decreases with a smaller n_v.

Sensitivity to M. Intuitively, the ideal approach would involve solving the PSVD problem at each iteration and updating the level value if PSVD proves infeasible. Nevertheless, in practical implementation, we can assess the feasibility of the PSVD problem and update the level value every M iteration.

In Figure 9, the sensitivity of AccSPS to the updating frequency M is illustrated. We pick $n_v = 2$ for all experiments.

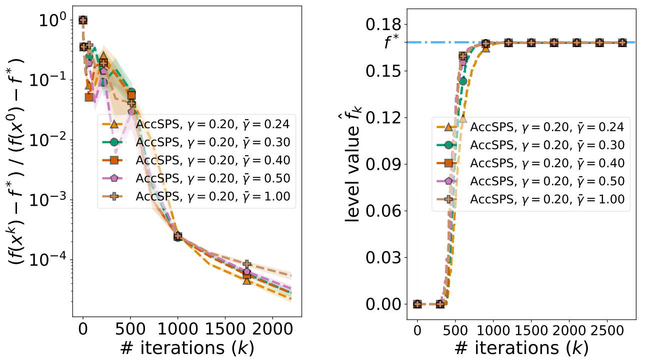


Fig. 6. Sensitivity of AccSPS to $\bar{\gamma}$ on the A1A dataset ($\lambda = 0.001$). Experiments are repeated 5 times and mean values and standard deviations are plotted.

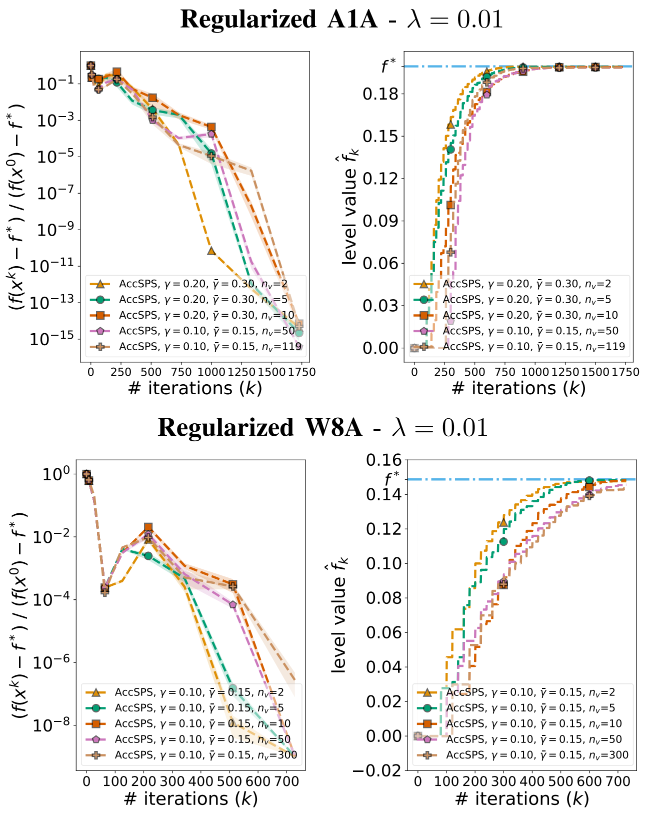


Fig. 7. Sensitivity of AccSPS to n_v. The A1A dataset has 119 dimensions for each data point and the W8A dataset has 300 dimensions for each data point. Experiments are repeated 5 times and mean values and standard deviations (std) are plotted.

TABLE II
PSVD TIME UNDER DIFFERENT n_v

Parameter	Regularized A1A ($\lambda = 0.01$)		
	PSVD time (s)	Total time (s)	Pct (%)
$n_v = 2$	0.16	0.72	22.8
$n_v = 5$	0.29	0.80	36.4
$n_v = 10$	0.67	1.24	54.1
$n_v = 50$	4.03	4.49	90.0
$n_v = d$	8.40	9.05	92.8

The updating frequency exhibits a small impact on both the decrease in loss and the convergence of the level value.

Similarly, Table III illustrates the computation time for solving the PSVD problem and its percentage of the total computation time with different M. In contrast to n_v, a larger M results in a shorter total computation time, and the percentage of time spent on PSVD relative to the total time is also reduced. In practice, we can pick $M = 20$ to reduce the computation time without compromising solution quality.

TABLE III
PSVD TIME UNDER DIFFERENT M

Parameter	Regularized Mushrooms ($\lambda = 0.1$)		
	PSVD time (s)	Total time (s)	Pct (%)
$M = 5$	1.02	1.73	58.9
$M = 10$	0.50	1.19	41.9
$M = 20$	0.26	1.09	24.4
$M = 50$	0.16	0.90	18.0
$M = 100$	0.12	0.80	14.7

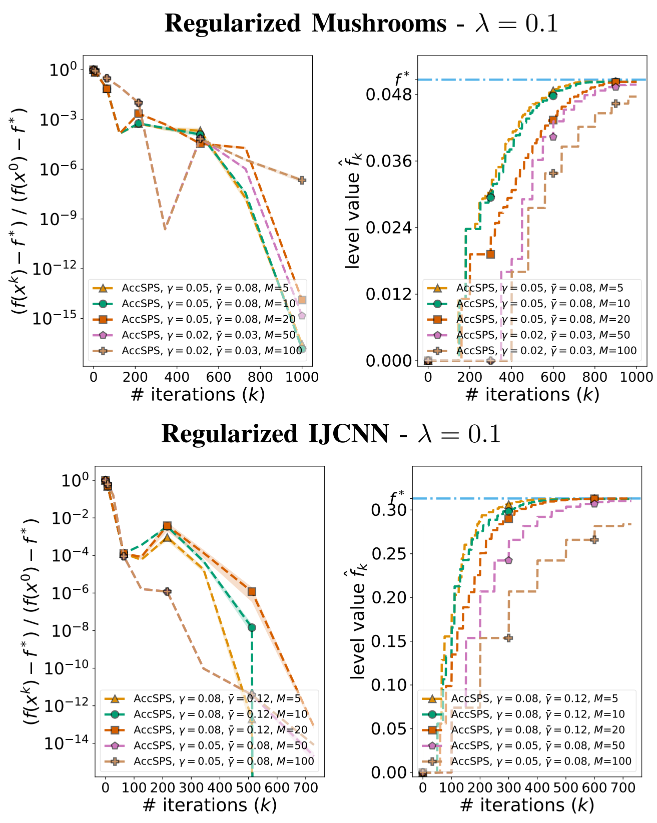


Fig. 9. Sensitivity of AccSPS to M. Experiments are repeated 5 times and mean values and standard deviations (std) are plotted.

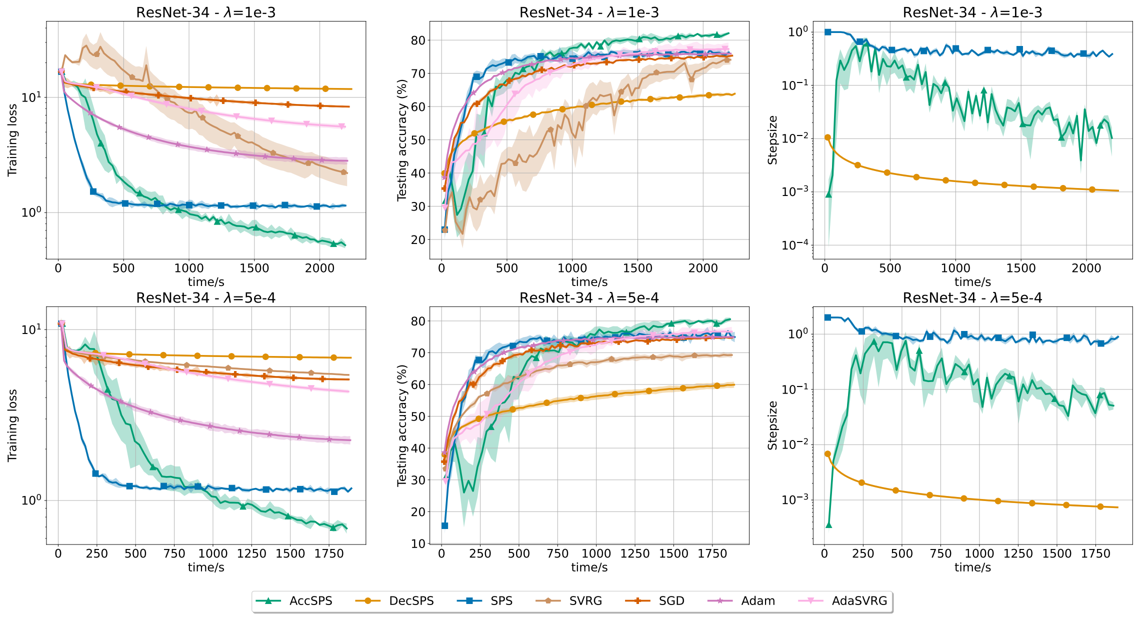


Fig. 8. Performance comparison of various optimizers on CIFAR-100 dataset using ResNet-34 under different regularization levels. Experiments are repeated 5 times and mean values and standard deviations are plotted. The number of training epochs is 100 for all optimizers, and the plots are truncated to maintain consistency within the time frame. Here the testing accuracy is top-1 accuracy.

B. Performance comparison on CIFAR-100 dataset

In this subsection, we assess and compare the performance of AccSPS and other optimizers on the CIFAR-100 dataset [42] with l_2 regularization, which has 50,000 training images and 10,000 testing images. We use a batch size $B = 256$ and consider two regularization levels $\lambda = \{1e-3, 5e-4\}$. For AccSPS, SVRG, and AdaSVRG, we use a larger batch size when updating the average gradient $\tilde{\mu}$ and set updating frequency $s = m$. For a fair comparison, we set γ and η_b to the same values for both AccSPS and SPS. As shown in Figure 8, AccSPS outperforms DecSPS on non-convex deep learning problems. Compared to SPS, AccSPS is less sensitive to γ and η_b and maintains a lower step size due to its dynamically adjusted level value. Additionally, we observe that both DecSPS and SGD exhibit a slower decrease in training loss when l_2 regularization is applied. Note that we only store 20 partial gradients (with $M = 20$ and $n_v = 5$) and function values in the CPU, resulting in negligible memory usage (less than 1 KB).

References

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, pp. 400–407, 1951.

[2] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.

[3] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Math. Program.*, vol. 162, pp. 83–112, 2017.

[4] R. M. Gower, F. Kunstner, and M. Schmidt, "Variance reduced model based methods: New rates and adaptive step sizes," in *OPT 2023: Optimization for Machine Learning*, 2023.

[5] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.

[6] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*. Springer, 2016, pp. 795–811.

[7] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, "SGD: General analysis and improved rates," in *Int. Conf. Mach. Learn.* PMLR, 2019, pp. 5200–5209.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[9] Y. Dauphin, H. De Vries, and Y. Bengio, "Equilibrated adaptive learning rates for non-convex optimization," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[10] K. Y. Levy, A. Yurtsever, and V. Cevher, "Online adaptive methods, universality and acceleration," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[11] R. Ward, X. Wu, and L. Bottou, "Adagrad stepsizes: Sharp convergence over nonconvex landscapes," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 9047–9076, 2020.

[12] S. Vaswani, I. H. Laradji, F. Kunstner, S. Y. Meng, M. Schmidt, and S. Lacoste-Julien, "Adaptive gradient methods converge faster with over-parameterization (and you can do a line-search)," 2020.

[13] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Int. Conf. Learn. Represent.*, 2018.

[14] B. T. Polyak, "Minimization of unsmooth functionals," *USSR Comput. Math. Math. Phys.*, vol. 9, no. 3, pp. 14–29, 1969.

[15] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.

[16] E. Hazan and S. Kakade, "Revisiting the polyak step size," *arXiv preprint arXiv:1905.00313*, 2019.

[17] M. Rolinek and G. Martius, "L4: Practical loss-based stepsize adaptation for deep learning," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[18] M. Prazeres and A. M. Oberman, "Stochastic gradient descent with polyak's learning rate," *J. Sci. Comput.*, vol. 89, pp. 1–16, 2021.

[19] L. Berrada, A. Zisserman, and M. P. Kumar, "Training neural networks for and by interpolation," in *Int. Conf. Mach. Learn.* PMLR, 2020, pp. 799–809.

[20] N. Loizou, S. Vaswani, I. H. Laradji, and S. Lacoste-Julien, "Stochastic polyak step-size for SGD: An adaptive learning rate for fast convergence," in *Int. Conf. Artif. Intell. Stat.* PMLR, 2021, pp. 1306–1314.

[21] R. D'Orazio, N. Loizou, I. H. Laradji, and I. Mitliagkas, "Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic polyak stepsize," *Trans. Mach. Learn. Res*, 2023. [Online]. Available: https://openreview.net/forum?id=28bQiPWxHl

[22] A. Orvieto, S. Lacoste-Julien, and N. Loizou, "Dynamics of SGD with stochastic polyak stepsizes: Truly adaptive variants and convergence to exact solution," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 26 943–26 954, 2022.

[23] X. Jiang and S. U. Stich, "Adaptive SGD with polyak stepsize and line-search: Robust convergence and variance reduction," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 26 396–26 424, 2023.

[24] X. Wang, M. Johansson, and T. Zhang, "Generalized polyak step size for first order optimization with momentum," in *Int. Conf. Mach. Learn.* PMLR, 2023, pp. 35 836–35 863.

[25] R. M. Gower, A. Defazio, and M. Rabbat, "Stochastic polyak stepsize with a moving target," *arXiv preprint arXiv:2106.11851*, 2021.

[26] U. Brannlund, *On relaxation methods for nonsmooth convex optimization.* Kungliga Tekniska Hogskolan, 1995.

[27] J.-L. Goffin and K. C. Kiwiel, *Convergence of a simple subgradient level method.* Groupe d'études et de recherche en analyse des décisions, 1998.

[28] K. C. Kiwiel, T. Larsson, and P. O. Lindberg, "The efficiency of ballstep subgradient level methods for convex optimization," *Math. Oper. Res.*, vol. 24, no. 1, pp. 237–254, 1999.

[29] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.

[30] K. Mao, Q.-K. Pan, T. Chai, and P. B. Luh, "An effective subgradient method for scheduling a steelmaking-continuous casting process," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 1140–1152, 2014.

[31] M. A. Bragin and E. L. Tucker, "Surrogate "level-based" Lagrangian relaxation for mixed-integer linear programming," *Sci. Rep.*, vol. 12, no. 1, p. 22417, 2022.

[32] A. Liu, M. A. Bragin, X. Chen, and X. Guan, "Accelerating level-value adjustment for the polyak stepsize," *J. Optim. Theory Appl.*, vol. 206, no. 3, pp. 1–36, 2025.

[33] M. A. Bragin, B. Yan, and P. B. Luh, "Distributed and asynchronous coordination of a mixed-integer linear system via surrogate Lagrangian relaxation," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1191–1205, 2020.

[34] A. Liu, P. B. Luh, K. Sun, M. A. Bragin, and B. Yan, "Integrating machine learning and mathematical optimization for job shop scheduling," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 3, pp. 4829–4850, 2023.

[35] Z. Shao, X. Cao, Q. Zhai, and X. Guan, "Risk-constrained planning of rural-area hydrogen-based microgrid considering multiscale and multi-energy storage systems," *Appl. Energy*, vol. 334, p. 120682, 2023.

[36] J. Qin, R. Yang, and N. Yu, "Physics-informed graph neural networks for collaborative dynamic reconfiguration and voltage regulation in unbalanced distribution systems," *IEEE Trans. Ind. Appl.*, vol. 61, no. 2, pp. 2538–2548, 2025.

[37] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[38] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[39] S. Vaswani, A. Mishkin, I. Laradji, M. Schmidt, G. Gidel, and S. Lacoste-Julien, "Painless stochastic gradient: Interpolation, line-search, and convergence rates," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

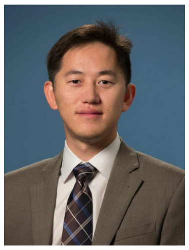
[41] B. Dubois-Taine, S. Vaswani, R. Babanezhad, M. Schmidt, and S. Lacoste-Julien, "SVRG meets AdaGrad: Painless variance reduction," *Mach. Learn.*, vol. 111, no. 12, pp. 4359–4409, 2022.

[42] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

Jingtao Qin (Student Member, IEEE) received the B.S. and M.S. degrees in Electrical Engineering from Shandong University, Jinan, China, in 2018 and 2020, respectively, and the Ph.D. degree in Electrical Engineering from the University of California, Riverside, CA, USA. He is currently a Power System Research Scientist at Hitachi America, Ltd., Santa Clara, CA, USA. His research interests include machine learning, optimization, and their applications in power systems, with a particular focus on unit commitment, contingency analysis, network reconfiguration, and voltage regulation.

Anbang Liu (Student Member, IEEE) is currently a Postdoctoral Fellow in the Department of Data and Systems Engineering at the University of Hong Kong (HKU). Prior to joining HKU, he received his Ph.D. degree from Center for Intelligent and Networked Systems at Tsinghua University in 2024. His research lies at the intersection of artificial intelligence and classical mathematical optimization, with a specific focus on developing Machine Learning approaches and Mixed Integer Programming approaches to solve challenging decision-making problems arising in manufacturing systems, supply chains, and power systems.

Mikhail A. Bragin (Senior Member, IEEE; Member, INFORMS) is currently an Energy Marketing Analysis Advisor with Southern California Edison (SCE). His research addresses complex challenges across energy systems, transportation, and advanced manufacturing. His research interests include mathematical optimization, operations research, artificial intelligence, machine learning, and quantum computing with applications to power systems, renewable energy integration, decarbonization, and manufacturing scheduling.



Nanpeng Yu (Senior Member, IEEE) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2006, and the M.S. and Ph.D. degrees in electrical engineering from Iowa State University, Ames, IA, USA, in 2007 and 2010, respectively. He is a Full Professor and Vice Chair with the Department of Electrical and Computer Engineering and Director of Energy, Economics, and Environment Research Center at University of California, Riverside, CA, USA. His current research interests include physics-informed machine learning in smart grids, electricity market design and optimization, transportation electrification, and decarbonization planning.