# A Generic Acceleration Framework for Stochastic Composite Optimization

Andrei Kulunchakov, Julien Mairal

# A Generic Acceleration Framework for Stochastic Composite Optimization

Andrei Kulunchakov
Inria*
andrei.kulunchakov@inria.fr

Julien Mairal
Inria*
julien.mairal.@inria.fr

June 3, 2019

**Abstract**

In this paper, we introduce various mechanisms to obtain accelerated first-order stochastic optimization algorithms when the objective function is convex or strongly convex. Specifically, we extend the Catalyst approach originally designed for deterministic objectives to the stochastic setting. Given an optimization method with mild convergence guarantees for strongly convex problems, the challenge is to accelerate convergence to a noise-dominated region, and then achieve convergence with an optimal worst-case complexity depending on the noise variance of the gradients. A side contribution of our work is also a generic analysis that can handle inexact proximal operators, providing new insights about the robustness of stochastic algorithms when the proximal operator cannot be exactly computed.

## 1 Introduction

In this paper, we consider stochastic composite optimization problems of the form

$$\min_{x\in\mathbb{R}^p} \{F(x) := f(x) + \psi(x)\} \quad \text{with} \quad f(x) = \mathbb{E}_\xi[\tilde{f}(x,\xi)], \tag{1}$$

where $f$ is a convex $L$-smooth function (meaning differentiable with $L$-Lipschitz continuous gradient) and $\psi$ is a possibly non-smooth convex lower-semicontinuous function. For instance, $\psi$ may be the $\ell_1$-norm, which is known to induce sparsity, or an indicator function that may take the value $+\infty$ outside of a convex set and 0 inside [22]. The random variable $\xi$ corresponds to data samples. When the amount of training data is finite, the expectation $\mathbb{E}_\xi[\tilde{f}(x,\xi)]$ can be replaced by a finite sum, a setting that has attracted a lot of attention in machine learning recently, see, *e.g.*, [14, 15, 20, 26, 36, 43, 53] for incremental algorithms and [1, 27, 31, 34, 47, 55, 56] for accelerated variants.

Yet, as noted in [8], one is typically not interested in the minimization of the empirical risk—that is, a finite sum of functions—with high precision, but instead, one should focus on the expected risk involving the true (unknown) data distribution. When one can draw an infinite number of samples from this distribution, the true risk (1) may be minimized by using appropriate stochastic optimization techniques. Unfortunately, fast methods designed for deterministic objectives would not apply to this setting; methods based on stochastic approximations admit indeed optimal "slow" rates that are typically $O(1/\sqrt{k})$ for convex functions and $O(1/k)$ for strongly convex ones, depending on the exact assumptions made on the problem, where $k$ is the number of noisy gradient evaluations [39].

Better understanding the gap between deterministic and stochastic optimization is one goal of this paper. Specifically, we are interested in Nesterov's acceleration of gradient-based approaches [40, 41]. In a nutshell, gradient descent or its proximal variant applied to a $\mu$-strongly convex $L$-smooth function achieves an

*Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, Grenoble, 38000, France.

exponential convergence rate $O((1-\mu/L)^k)$ in the worst case in function values, and a sublinear rate $O(L/k)$ if the function is simply convex ($\mu = 0$). By interleaving the algorithm with clever extrapolation steps, Nesterov showed that faster convergence could be achieved, and the previous convergence rates become $O((1-\sqrt{\mu/L})^k)$ and $O(L/k^2)$, respectively. Whereas no clear geometrical intuition seems to appear in the literature to explain why acceleration occurs, proof techniques to show accelerated convergence [5, 41, 51] and extensions to a large class of other gradient-based algorithms are now well established [1, 11, 34, 42, 47].

Yet, the effect of Nesterov's acceleration to stochastic objectives remains poorly understood since existing unaccelerated algorithms such as stochastic mirror descent [39] and their variants already achieve the optimal asymptotic rate. Besides, negative results also exist, showing that Nesterov's method may be unstable when the gradients are computed approximately [13, 17]. Nevertheless, several approaches such as [4, 12, 16, 18, 19, 24, 29, 30, 52] have managed to show that acceleration may be useful to forget faster the algorithm's initialization and reach a region dominated by the noise of stochastic gradients; then, "good" methods are expected to asymptotically converge with a rate exhibiting an optimal dependency in the noise variance [39], but with no dependency on the initialization. A major challenge is then to achieve the optimal rate for these two regimes.

In this paper, we consider an optimization method $\mathcal{M}$ with the following property: given an auxiliary strongly convex objective function $h$, we assume that $\mathcal{M}$ is able to produce iterates $(z_t)_{t\geq 0}$ with expected linear convergence to a noise-dominated region—that is, such that

$$\mathbb{E}[h(z_t) - h^\star] \leq C(1-\tau)^t(h(z_0) - h^\star) + B\sigma^2, \tag{2}$$

where $C, \tau, B > 0$, $h^\star$ is the minimum function value, and $\sigma^2$ is an upper bound on the variance of stochastic gradients accessed by $\mathcal{M}$, which we assume to be uniformly bounded. Whereas such an assumption has limitations, it remains the most standard one for stochastic optimization (see [10, 44] for more realistic settings in the smooth case). The class of methods satisfying (2) is relatively large. For instance, when $h$ is $L$-smooth, the stochastic gradient descent method (SGD) with constant step size $1/L$ and iterate averaging satisfies (2) with $\tau = \mu/L$, $B = 1/L$, and $C = 1$, see [29].

**Main contribution.** In this paper, we extend the Catalyst approach [34] to stochastic problems. Under mild conditions, our approach is able to turn $\mathcal{M}$ into a converging algorithm with a worst-case expected complexity that decomposes into two parts: the first one exhibits an accelerated convergence rate in the sense of Nesterov and shows how fast one forgets the initial point; the second one corresponds to the stochastic regime and typically depends (optimally in many cases) on $\sigma^2$. Note that even though we only make assumptions about the behavior of $\mathcal{M}$ on strongly convex sub-problems (2), we also treat the case where the objective (1) is convex, but not strongly convex.

To illustrate the versatility of our approach, we consider the stochastic finite-sum problem [7, 23, 32, 54], where the objective (1) decomposes into $n$ components $\tilde{f}(x,\xi) = \frac{1}{n}\sum_{i=1}^n \tilde{f}_i(x,\xi)$ and $\xi$ is a stochastic perturbation, coming, *e.g.*, from data augmentation or noise injected during training to improve generalization or privacy (see [29, 36]). The underlying finite-sum structure may also result from clustering assumptions on the data [23], or from distributed computing [32], a setting beyond the scope of our paper. Whereas it was shown in [29] that classical variance-reduced stochastic optimization methods such as SVRG [53], SDCA [47], SAGA [14], or MISO [36], can be made robust to noise, the analysis of [29] is only able to accelerate the SVRG approach. With our acceleration technique, all of the aforementioned method can be modified such that they find a point $\hat{x}$ satisfying $\mathbb{E}[F(\hat{x}) - F^\star] \leq \varepsilon$ with global iteration complexity, for the $\mu$-strongly convex case,

$$\tilde{O}\left(\left(n + \sqrt{n\frac{L}{\mu}}\right)\log\left(\frac{F(x_0)-F^\star}{\varepsilon}\right) + \frac{\sigma^2}{\mu\varepsilon}\right). \tag{3}$$

The term on the left is the optimal complexity for finite-sum optimization [1, 2], up to logarithmic terms in $L, \mu$ hidden in the $\tilde{O}(.)$ notation, and the term on the right is the optimal complexity for $\mu$-strongly convex stochastic objectives [18] where $\sigma^2$ is due to the perturbations $\xi$. As Catalyst [34], the price to pay compared to non-generic direct acceleration techniques [1, 29] is a logarithmic factor.

**Other contributions.** In this paper, we generalize the analysis of Catalyst [34, 45] to handle various new cases. Beyond the ability to deal with stochastic optimization problems, our approach (i) improves Catalyst by allowing sub-problems of the form (2) to be solved approximately *in expectation*, which is more realistic than the deterministic requirement made in [34] and which is also critical for stochastic optimization, (ii) leads to a new accelerated stochastic gradient descent algorithms for composite optimization with similar guarantees as [18, 19, 29], (iii) handles the analysis of accelerated proximal gradient descent methods with inexact computation of proximal operators, improving the results of [46] while also treating the stochastic setting.

# 2 Relation with Inexact and Stochastic Proximal Point Methods

Catalyst is based on the inexact accelerated proximal point algorithm [21], which consists in solving approximately a sequence of sub-problems and updating two sequences $(x_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ by

$$x_k \approx \operatorname*{argmin}_{x\in\mathbb{R}^p} \Big\{ h_k(x) := F(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \Big\} \quad \text{and} \quad y_k = x_k + \beta_k(x_k - x_{k-1}), \tag{4}$$

where $\beta_k$ in $(0,1)$ is obtained from Nesterov's acceleration principles [41], and $\kappa$ is a well chosen regularization parameter. The method $\mathcal{M}$ is used to obtain an approximate minimizer of $h_k$ by using an appropriate computational budget; when $\mathcal{M}$ converges linearly, it may be shown that the resulting algorithm (4) enjoys a better worst-case complexity than if $\mathcal{M}$ was used directly on $f$, see [34].

Since asymptotic linear convergence is out of reach when $f$ is a stochastic objective, a classical strategy consists in replacing $F(x)$ in (4) by a finite-sum approximation obtained by random sampling, leading to deterministic sub-problems. Typically without Nesterov's acceleration (with $y_k = x_k$), this strategy is often called the stochastic proximal point method [3, 6, 28, 50, 49]. The point of view we adopt in this paper is different and is based on the minimization of surrogate functions $h_k$ related to (4), but which are more general and may take other forms than $F(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2$.

# 3 Preliminaries: Basic Multi-Stage Schemes

In this section, we present two simple multi-stage mechanisms to improve the worst-case complexities of stochastic optimization methods, before introducing acceleration principles.

**Basic restart with mini-batching or decaying step sizes.** Consider an optimization method $\mathcal{M}$ with convergence rate (2) and assume that there exists a hyper-parameter to control a trade-off between the bias $B\sigma^2$ and the computational complexity. Specifically, we assume that the bias can be reduced by an arbitrary factor $\eta < 1$, while paying a factor $1/\eta$ in terms of complexity per iteration (or $\tau$ may be reduced by a factor $\eta$, thus slowing down convergence). This may occur in two cases:

- by using a mini-batch of size $1/\eta$ to sample gradients, which replaces $\sigma^2$ by $\eta\sigma^2$;
- or the method uses a step size proportional to $\eta$ that can be chosen arbitrarily small.

For instance, stochastic gradient descent with constant step size and iterate averaging is compatible with both scenarios [29]. Then, consider a target accuracy $\varepsilon$ and define the sequences $\eta_k = 1/2^k$ and $\varepsilon_k = 2B\sigma^2\eta_k$ for $k \geq 0$. We may now solve successively the problem up to accuracy $\varepsilon_k$—*e.g.*, with a constant number $O(1/\tau)$ steps of $\mathcal{M}$ when using mini-batches of size $1/\eta_k = 2^k$ to reduce the bias—and by using the solution of iteration $k$–1 as a warm restart. As shown in Appendix B, the scheme converges and the worst-case complexity to achieve the accuracy $\varepsilon$ in expectation is

$$O\left(\frac{1}{\tau}\log\left(\frac{C(F(x_0) - F^\star)}{\varepsilon}\right) + \frac{B\sigma^2\log(2C)}{\tau\varepsilon}\right). \tag{5}$$

For instance, one may run SGD with constant step size $\eta_k/L$ at stage $k$ with iterate averaging as in [29], which yields $B = 1/L$, $C = 1$, and $\tau = L/\mu$. Then, the left term is the classical complexity $O((L/\mu)\log(1/\varepsilon))$ of the (unaccelerated) gradient descent algorithm for deterministic objectives, whereas the right term is the optimal complexity for stochastic optimization in $O(\sigma^2/\mu\varepsilon)$. Similar restart principles appear for instance in [4] in the design of a multistage accelerated SGD algorithm.

**Restart: from sub-linear to linear rate with strong convexity.** A natural question is whether asking for a linear rate in (2) for strongly convex problems is a strong requirement. Here, we show that a sublinear rate is in fact sufficient for our needs by generalizing a restart technique introduced in [19] for stochastic optimization, which was previously used for deterministic objectives in [25].

Specifically, consider an optimization method $\mathcal{M}$ such that the convergence rate (2) is replaced by

$$\mathbb{E}[h(z_t) - h^\star] \leq \frac{D\|z_0 - z^\star\|^2}{2t^d} + \frac{B\sigma^2}{2}, \tag{6}$$

where $D, d > 0$ and $z^\star$ is a minimizer of $h$. Assume now that $h$ is $\mu$-strongly convex with $D \geq \mu$ and consider restarting $s$ times the method $\mathcal{M}$, each time running $\mathcal{M}$ for $t' = \lceil (2D/\mu)^{1/d} \rceil$ iterations. Then, it may be shown (see Appendix B) that the relation (2) holds with $t = st'$, $\tau = \frac{1}{2t'}$, and $C = 1$. If a mini-batch or step size mechanism is available, we may then proceed as before and obtain a converging scheme with complexity (5), *e.g.*, by using mini-batches of exponentially increasing sizes once the method reaches a noise-dominated region, and by using a restart frequency of order $O(1/\tau)$.

# 4 Generic Multi-Stage Approaches with Acceleration

We are now in shape to introduce a generic acceleration framework that generalizes (4). Specifically, given some point $y_{k-1}$ at iteration $k$, we consider a surrogate function $h_k$ related to a parameter $\kappa > 0$, an approximation error $\delta_k \geq 0$, and an optimization method $\mathcal{M}$ that satisfy the following properties:

($\mathcal{H}_1$) $h_k$ is $(\kappa + \mu)$-strongly convex, where $\mu$ is the strong convexity parameter of $f$;

($\mathcal{H}_2$) $\mathbb{E}[h_k(x)|\mathcal{F}_{k-1}] \leq F(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2$ for $x = \alpha_{k-1}x^\star + (1-\alpha_{k-1})x_{k-1}$, which is deteministic given the past information $\mathcal{F}_{k-1}$ up to iteration $k$–1 and $\alpha_{k-1}$ is given in Alg. 1;

($\mathcal{H}_3$) $\mathcal{M}$ can provide the exact minimizer $x_k^\star$ of $h_k$ and a point $x_k$ (possibly equal to $x_k^\star$) such that $\mathbb{E}[F(x_k)] \leq \mathbb{E}[h_k^\star] + \delta_k$ where $h_k^\star = \min_x h_k(x)$.

The generic acceleration framework is presented in Algorithm 1. Note that the conditions on $h_k$ bear similarities with estimate sequences introduced by Nesterov [41]. However, the choices of $h_k$ and the proof technique are significantly different, as we will see with various examples below. We also assume at the moment that the exact minimizer $x_k^\star$ of $h_k$ is available, which differs from the Catalyst framework [34]; the case with approximate minimization will be presented in Section 4.1.

**Proposition 1** (Convergence analysis for Algorithm 1)**.** *Consider Algorithm 1. Then,*

$$\mathbb{E}[F(x_k) - F^\star] \leq \begin{cases} (1-\sqrt{q})^k \left(2(F(x_0) - F^\star) + \sum_{j=1}^k (1-\sqrt{q})^{-j}\delta_j\right) & \text{if } \mu \neq 0 \\ \frac{2}{(k+1)^2}\left(\kappa\|x_0 - x^\star\|^2 + \sum_{j=1}^k \delta_j (j+1)^2\right) & \text{otherwise} \end{cases}. \tag{8}$$

The proof of the proposition is given in Appendix C and is based on an extension of the analysis of Catalyst [34]. Next, we present various application cases leading to algorithms with acceleration.

**Algorithm 1** Generic Acceleration Framework with Exact Minimization of $h_k$

**Input:** $x_0$ (initial estimate); $\mathcal{M}$ (optimization method); $\mu$ (strong convexity constant); $\kappa$ (parameter for $h_k$); $K$ (number of iterations); $(\delta_k)_{k\geq 0}$ (approximation errors);
**Initialization:** $y_0 = x_0$; $q = \frac{\mu}{\mu+\kappa}$; $\alpha_0 = 1$ if $\mu = 0$ or $\alpha_0 = \sqrt{q}$ if $\mu \neq 0$;
**for** $k = 1, \dots, K$ **do**
  Consider a surrogate $h_k$ satisfying $(\mathcal{H}_1)$, $(\mathcal{H}_2)$ and obtain $x_k, x_k^\star$ using $\mathcal{M}$ satisfying $(\mathcal{H}_3)$;
  Compute $\alpha_k$ in $(0,1)$ by solving the equation $\alpha_k^2 = (1-\alpha_k)\alpha_{k-1}^2 + q\alpha_k$.
  Update the extrapolated sequence

$$y_k = x_k^\star + \beta_k(x_k^\star - x_{k-1}) + \frac{(\kappa+\mu)(1-\alpha_k)}{\kappa}(x_k - x_k^\star) \quad \text{with} \quad \beta_k = \frac{\alpha_{k-1}(1-\alpha_{k-1})}{\alpha_{k-1}^2 + \alpha_k}. \tag{7}$$

**end for**
**Output:** $x_k$ (final estimate).

**Accelerated proximal gradient method.** When $f$ is deterministic and the proximal operator of $\psi$ (see Appendix A for the definition) can be computed in closed form, choose $\kappa = L - \mu$ and define

$$h_k(x) := f(y_{k-1}) + \nabla f(y_{k-1})^\top (x - y_{k-1}) + \frac{L}{2}\|x - y_{k-1}\|^2 + \psi(x). \tag{9}$$

Consider $\mathcal{M}$ that minimizes $h_k$ in closed form: $x_k = x_k^\star = \mathrm{Prox}_{\psi/L}\left[y_{k-1} - \frac{1}{L}\nabla f(y_{k-1})\right]$. Then, $(\mathcal{H}_1)$ is obvious; $(\mathcal{H}_2)$ holds from the convexity of $f$, and $(\mathcal{H}_3)$ with $\delta_k = 0$ follows from classical inequalities for $L$-smooth functions [41]. Finally, we recover accelerated convergence rates [5, 41].

**Accelerated proximal point algorithm.** We consider $h_k$ given in (4) with exact minimization (thus an unrealistic setting, but conceptually interesting) with $\kappa = L - \mu$. Then, the assumptions $(\mathcal{H}_1)$, $(\mathcal{H}_2)$, and $(\mathcal{H}_3)$ are satisfied with $\delta_k = 0$ and we recover the accelerated rates of of [21].

**Accelerated stochastic gradient descent with prox.** A more interesting choice of surrogate is

$$h_k(x) := f(y_{k-1}) + g_k^\top (x - y_{k-1}) + \frac{\kappa+\mu}{2}\|x - y_{k-1}\|^2 + \psi(x), \tag{10}$$

where $\kappa \geq L - \mu$ and $g_k$ is an unbiased estimate of $\nabla f(y_{k-1})$—that is, $\mathbb{E}[g_k|\mathcal{F}_{k-1}] = \nabla f(y_{k-1})$—with variance bounded by $\sigma^2$, following classical assumptions from the stochastic optimization literature [18, 19, 24]. Then, $(\mathcal{H}_1)$ and $(\mathcal{H}_2)$ are satisfied given that $f$ is convex. To characterize $(\mathcal{H}_3)$, consider $\mathcal{M}$ that minimizes $h_k$ in closed form: $x_k = x_k^\star = \mathrm{Prox}_{\psi/(\kappa+\mu)}[y_{k-1} - \frac{1}{\kappa+\mu}g_k]$, and define $u_{k-1} := \mathrm{Prox}_{\psi/(\kappa+\mu)}[y_{k-1} - \frac{1}{\kappa+\mu}\nabla f(y_{k-1})]$, which is deterministic given $\mathcal{F}_{k-1}$. Then, from (10),

$$\begin{aligned} f(x_k) &\leq h_k(x_k) + (\nabla f(y_{k-1}) - g_k)^\top (x_k - y_{k-1}) \qquad \text{(from } L\text{-smoothness of } f) \\ &= h_k^\star + (\nabla f(y_{k-1}) - g_k)^\top (x_k - u_{k-1}) + (\nabla f(y_{k-1}) - g_k)^\top (u_{k-1} - y_{k-1}). \end{aligned}$$

When taking expectations, the last term on the right disappears since $\mathbb{E}[g_k|\mathcal{F}_{k-1}] = \nabla f(y_{k-1})$:

$$\begin{aligned} \mathbb{E}[f(x_k)] &\leq \mathbb{E}[h_k^\star] + \mathbb{E}[\|g_k - \nabla f(y_{k-1})\|\|x_k - u_{k-1}\|] \\ &\leq \mathbb{E}[h_k^\star] + \frac{1}{\kappa+\mu}\mathbb{E}\left[\|g_k - \nabla f(y_{k-1})\|^2\right] \leq \mathbb{E}[h_k^\star] + \frac{\sigma^2}{\kappa+\mu}, \end{aligned} \tag{11}$$

where we used the non-expansiveness of the proximal operator [38]. Therefore, $(\mathcal{H}_3)$ holds with $\delta_k = \sigma^2/(\kappa+\mu)$. The resulting algorithm is similar to [29] and offers the same guarantees. The novelty of our approach is then a unified convergence proof for the deterministic and stochastic cases.

**Corollary 2** (Complexity of proximal stochastic gradient algorithm, $\mu > 0$)**.** *Consider Algorithm 1 with $h_k$ defined in (10). When $f$ is $\mu$-strongly convex, choose $\kappa = L - \mu$. Then,*

$$\mathbb{E}[F(x_k) - F^\star] \leq \left(1 - \sqrt{\frac{\mu}{L}}\right)^k (F(x_0) - F^\star) + \frac{\sigma^2}{\sqrt{\mu L}},$$

which is of the form (2) with $\tau = \sqrt{\mu/L}$ and $B = \sigma^2/(\sqrt{\mu L})$. Interestingly, the optimal complexity $O\left(\sqrt{L/\mu}\log((F(x_0) - F^\star)/\varepsilon) + \sigma^2/\mu\varepsilon\right)$ can be obtained by using the first restart strategy presented in Section 3, see Eq. (5), either by using increasing mini-batches or decreasing step sizes.

When the objective is convex, but not strongly convex, Proposition 1 gives a bias term $O(\sigma^2 k/\kappa)$ that increases linearly with $k$. Yet, the following corollary exhibits an optimal rate with finite horizon, when both $\sigma^2$ and an upper-bound on $\|x_0 - x^\star\|^2$ are available. Even though non-practical, the result shows that our analysis recovers the optimal dependency in the noise level, as [19, 29] and others.

**Corollary 3** (Complexity of proximal stochastic gradient algorithm, $\mu = 0$)**.** *Consider a fixed budget $K$ of iterations of Algorithm 1 with $h_k$ defined in (10). When $\kappa = \max(L, \sigma(K+1)^{3/2}/\|x_0 - x^\star\|)$,*

$$\mathbb{E}[F(x_K) - F^\star] \leq \frac{2L\|x_0 - x^\star\|^2}{(K+1)^2} + \frac{3\sigma\|x_0 - x^\star\|}{\sqrt{K+1}}.$$

While all the previous examples use the choice $x_k = x_k^\star$, we will see in Section 4.2 cases where we may choose $x_k \neq x_k^\star$. Before that, we introduce a variant when $x_k^\star$ is not available.

## 4.1 Variant with Inexact Minimization

In this variant, presented in Algorithm 2, $x_k^\star$ is not available and we assume that $\mathcal{M}$ also satisfies:

($\mathcal{H}_4$) given $\varepsilon_k \geq 0$, $\mathcal{M}$ can provide a point $x_k$ such that $\mathbb{E}[h_k(x_k) - h_k^\star] \leq \varepsilon_k$.

**Algorithm 2** Generic Acceleration Framework with Inexact Minimization of $h_k$

**Input:** same as Algorithm 2;
**Initialization:** $y_0 = x_0$; $q = \frac{\mu}{\mu+\kappa}$; $\alpha_0 = 1$ if $\mu = 0$ or $\alpha_0 = \sqrt{q}$ if $\mu \neq 0$;
**for** $k = 1, \ldots, K$ **do**
  Consider a surrogate $h_k$ satisfying ($\mathcal{H}_1$), ($\mathcal{H}_2$) and obtain $x_k$ satisfying ($\mathcal{H}_3$) and ($\mathcal{H}_4$);
  Compute $\alpha_k$ in $(0,1)$ by solving the equation $\alpha_k^2 = (1 - \alpha_k)\alpha_{k-1}^2 + q\alpha_k$.
  Update the extrapolated sequence $y_k = x_k + \beta_k(x_k - x_{k-1})$ with $\beta_k$ defined in (7);
**end for**
**Output:** $x_k$ (final estimate).

The next proposition, proven in Appendix C, gives us some insight on how to achieve acceleration.

**Proposition 4** (Convergence analysis for Algorithm 2)**.** *Consider Alg. 2. Then, for any $\gamma \in (0, 1]$,*

$$\mathbb{E}[F(x_k) - F^\star] \leq \begin{cases} \left(1 - \frac{\sqrt{q}}{2}\right)^k \left(2(F(x_0) - F^\star) + 4\sum_{j=1}^k \left(1 - \frac{\sqrt{q}}{2}\right)^{-j} \left(\delta_j + \frac{\varepsilon_j}{\sqrt{q}}\right)\right) & \text{if } \mu \neq 0 \\ \frac{2e^{1+\gamma}}{(k+1)^2}\left(\kappa\|x_0 - x^\star\|^2 + \sum_{j=1}^k (j+1)^2\delta_j + \frac{(j+1)^{3+\gamma}\varepsilon_j}{\gamma}\right) & \text{if } \mu = 0. \end{cases}$$

To maintain the accelerated rate, the sequence $(\delta_k)_{k\geq 0}$ needs to converge at a similar speed as in Proposition 1, but the dependency in $\varepsilon_k$ is slightly worse. Specifically, when $f$ is $\mu$-strongly convex, we may have both $(\varepsilon_k)_{k\geq 0}$ and $(\delta_k)_{k\geq 0}$ decreasing at a rate $O((1-\rho)^k)$ with $\rho > \sqrt{q}/2$, but we pay a factor $(1/\sqrt{q})$ compared to (8). When $\mu = 0$, the accelerated $O(1/k^2)$ rate is preserved whenever $\varepsilon_k = O(1/k^{4+2\gamma})$ and $\delta_k = O(1/k^{3+\gamma})$, but we pay a factor $O(1/\gamma)$ compared to (8).

**Catalyst [34].** When using $h_k$ defined in (4), we recover the convergence rates of [34]. In such a case $\delta_k = \varepsilon_k$ since $\mathbb{E}[F(x_k)] \leq \mathbb{E}[h_k(x_k)] \leq \mathbb{E}[h_k^\star] + \delta_k$. In order to analyze the complexity of minimizing each $h_k$ with $\mathcal{M}$ and derive the global complexity of the multi-stage algorithm, the next proposition, proven in Appendix C, characterizes the quality of the initialization $x_{k-1}$.

**Proposition 5** (Warm restart for Catalyst)**.** *Consider Alg. 2 with $h_k$ defined in (4). Then, for $k \geq 2$,*

$$\mathbb{E}[h_k(x_{k-1}) - h_k^\star] \leq \frac{3\varepsilon_{k-1}}{2} + 54\kappa \max\left(\|x_{k-1} - x^\star\|^2, \|x_{k-2} - x^\star\|^2, \|x_{k-3} - x^\star\|^2\right), \tag{12}$$

where $x_{-1} = x_0$. Following [34], we may now analyze the global complexity. For instance, when $f$ is $\mu$-strongly convex, we may choose $\varepsilon_k = O((1-\rho)^k(F(x_0) - F^\star))$ with $\rho = \sqrt{q}/3$. Then, it is possible to show that Proposition (4) yields $\mathbb{E}[F(x_k) - F^\star] = O(\varepsilon_k/q)$ and from the inequality $\frac{\mu}{2}\|x_k - x^\star\|^2 \leq F(x_k) - F^\star$ and (12), we have $\mathbb{E}[h_k(x_{k-1}) - h_k^\star] = O(\frac{\kappa}{\mu q}\varepsilon_{k-1}) = O(\varepsilon_{k-1}/q^2)$. Consider now a method $\mathcal{M}$ that behaves as (2). When $\sigma = 0$, $x_k$ can be obtained in $O(\log(1/q)/\tau) = \tilde{O}(1/\tau)$ iterations of $\mathcal{M}$ after initializing with $x_{k-1}$. This allows us to obtain the global complexity $\tilde{O}((1/\tau\sqrt{q})\log(1/\varepsilon))$. For example, when $\mathcal{M}$ is the proximal gradient descent method, $\kappa = L$ and $\tau = (\mu + \kappa)/(L + \kappa)$ yield the global complexity $\tilde{O}(\sqrt{L/\mu}\log(1/\varepsilon))$ of an accelerated method.

Our results improve upon Catalyst [34] in two aspects that are crucial for stochastic optimization: (i) we allow the sub-problems to be solved in expectation, whereas Catalyst requires the stronger condition $h_k(x_k) - h_k^\star \leq \varepsilon_k$; (ii) Proposition 5 removes the requirement of [34] to perform a full gradient step for initializing the method $\mathcal{M}$ in the composite case (see Prop. 12 in [34]).

**Proximal gradient descent with inexact prox [46].** The surrogate (10) with inexact minimization can be treated in the same way as Catalyst, which provides a unified proof for both problems. Then, we recover the results of [46], while allowing inexact minimization to be performed in expectation.

**Stochastic Catalyst.** With Proposition 5, we are in shape to consider stochastic problems when using a method $\mathcal{M}$ that converges linearly as (2) with $\sigma^2 \neq 0$ for minimizing $h_k$. As in Section 3, we also assume that there exists a mini-batch/step-size parameter $\eta$ that can reduce the bias by a factor $\eta < 1$ while paying a factor $1/\eta$ in terms of inner-loop complexity. As above, we discuss the strongly-convex case and choose the same sequence $(\varepsilon_k)_{k\geq 0}$. In order to minimize $h_k$ up to accuracy $\varepsilon_k$, we set $\eta_k = \min(1, \varepsilon_k/(2B\sigma^2))$ such that $\eta_k B\sigma^2 \leq \varepsilon_k/2$. Then, the complexity to minimize $h_k$ with $\mathcal{M}$ when using the initialization $x_{k-1}$ becomes $\tilde{O}(1/\tau\eta_k)$, leading to the global complexity

$$\tilde{O}\left(\frac{1}{\tau\sqrt{q}}\log\left(\frac{F(x_0) - F^\star}{\varepsilon}\right) + \frac{B\sigma^2}{q^{3/2}\tau\varepsilon}\right). \tag{13}$$

Details about the derivation are given in Appendix B. The left term corresponds to the Catalyst accelerated rate, but it may be shown that the term on the right is sub-optimal. Indeed, consider $\mathcal{M}$ to be ISTA with $\kappa = L - \mu$. Then, $B = 1/L$, $\tau = O(1)$, and the right term becomes $\tilde{O}((\sqrt{L/\mu})\sigma^2/\mu\varepsilon)$, which is sub-optimal by a factor $\sqrt{L/\mu}$. Whereas this result is a negative one, suggesting that Catalyst is not robust to noise, we show in Section 4.2 how to circumvent this for a large class of algorithms.

**Accelerated stochastic proximal gradient descent with inexact prox.** Finally, consider $h_k$ defined in (10) but the proximal operator is computed approximately, which, to our knowledge, has never been analyzed in the stochastic context. Then, it may be shown (see Appendix B for details) that Proposition 4 holds with $\delta_k = 2\varepsilon_k + 3\sigma^2/(2(\kappa + \mu))$. Then, an interesting question is how small should $\varepsilon_k$ be to guarantee the optimal dependency with respect to $\sigma^2$ as in Corollary 2. In the strongly-convex case, Proposition 4 simply gives $\varepsilon_k = O(\sqrt{q}\sigma^2/(\kappa + \mu))$ such that $\delta_k \approx \varepsilon_k/\sqrt{q}$.

### 4.2 Exploiting methods $\mathcal{M}$ providing strongly convex surrogates

Among various application cases, we have seen an extension of Catalyst to stochastic problems. To achieve convergence, the strategy requires a mechanism to reduce the bias $B\sigma^2$ in (2), *e.g.*, by using mini-batches or decreasing step sizes. Yet, the approach suffers from two issues: (i) some of the parameters are based on unknown quantities such as $\sigma^2$; (ii) the worst-case complexity exhibits a sub-optimal dependency in $\sigma^2$, typically of order $1/\sqrt{q}$ when $\mu > 0$. Whereas practical workarounds for the first point are discussed in Section 5, we now show how to solve the second one in many cases, by using Algorithm 1 with a particular surrogate $h_k$ provided by the optimization method. Consider indeed a method $\mathcal{M}$ satisfying (2) and which is able, after $T$ steps, to produce a point $x_k$ such that

$$\mathbb{E}[H_k(x_k) - h_k^\star] \leq C(1-\tau)^T(H_k(x_{k-1}) - H_k^\star + \xi_{k-1}) + B\sigma^2 \quad \text{with} \quad H_k(x) = F(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2, \quad (14)$$

where $h_k$ is a function satisfying $(\mathcal{H}_1)$, $(\mathcal{H}_2)$, and that can be minimized in closed form and $\xi_{k-1} = O(\mathbb{E}[F(x_{k-1}) - F^\star])$; $(\mathcal{H}_3)$ is also satisfied with $\delta_k = C(1-\tau)^T(H_k(x_{k-1}) - H_k^\star + \xi_{k-1}) + B\sigma^2$ since $\mathbb{E}[F(x_k)] \leq \mathbb{E}[H_k(x_k)] \leq \mathbb{E}[h_k^\star] + \delta_k$. In other words, $\mathcal{M}$ is used to perform *approximate minimization* of $H_k$, but we consider cases where $\mathcal{M}$ also provides *another surrogate* $h_k$ with closed-form minimizer that satisfies the conditions required to use Algorithm 1, which has better convergence guarantees than Algorithm 2 (same convergence rate up to a better factor).

As shown in Appendix D, even though (14) looks technical, a large class of optimization techniques are able to provide the condition (14), including many variants of proximal stochastic gradient descent methods with variance reduction such as SAGA [14], MISO [36], SDCA [47], or SVRG [53].

Whereas (14) seems to be a minor modification of (2), an important consequence is that it will allow us to gain a factor $1/\sqrt{q}$ in complexity when $\mu > 0$, corresponding precisely to the sub-optimality factor. Indeed, we may notice that Therefore, even though the surrogate $H_k$ needs only be minimized approximately, the condition (14) allows us to use Algorithm 1 instead of Algorithm 2. The dependency with respect to $\delta_k$ being better than $\varepsilon_k$ (by $1/\sqrt{q}$), we have then the following result:

**Proposition 6** (Stochastic Catalyst with Optimality Gaps, $\mu > 0$)**.** *Consider Algorithm 1 with a method $\mathcal{M}$ and surrogate $h_k$ satisfying (14) when $\mathcal{M}$ is used to minimize $H_k$ by using $x_{k-1}$ as a warm restart. Assume that $f$ is $\mu$-strongly convex and that there exists a parameter $\eta$ that can reduce the bias $B\sigma^2$ by a factor $\eta < 1$ while paying a factor $1/\eta$ in terms of inner-loop complexity.*

*Choose $\delta_k = O((1-\sqrt{q}/2)^k(F(x_0)-F^\star))$ and $\eta_k = \min(1, \delta_k/(2B\sigma^2))$. Then, the complexity to solve (14) and compute $x_k$ is $\tilde{O}(1/\tau\eta_k)$, and the global complexity to obtain $\mathbb{E}[F(x_k) - F^\star] \leq \varepsilon$ is*

$$\tilde{O}\left(\frac{1}{\tau\sqrt{q}}\log\left(\frac{F(x_0)-F^\star}{\varepsilon}\right) + \frac{B\sigma^2}{q\tau\varepsilon}\right).$$

The term on the left is the accelerated rate of Catalyst for deterministic problems, whereas the term on the right is potentially optimal for strongly convex problems, as illustrated in the next table. We provide indeed practical choices for the parameters $\kappa$, leading to various values of $B, \tau, q$, for the proximal stochastic gradient descent method with iterate averaging as well as variants of SAGA,MISO,SVRG that can cope with stochastic perturbations, which are discussed in Appendix D. All the values below are given up to universal constants to simplify the presentation.

| Method $\mathcal{M}$ | $\kappa$ | $\tau$ | $B$ | $q$ | Complexity after Catalyst |
|---|---|---|---|---|---|
| prox-SGD | $L-\mu$ | $\frac{1}{2}$ | $\frac{1}{L}$ | $\frac{\mu}{L}$ | $\tilde{O}\left(\sqrt{\frac{L}{\mu}}\log\left(\frac{F_0}{\varepsilon}\right) + \frac{\sigma^2}{\mu\varepsilon}\right)$ |
| SAGA/MISO/SVRG with $\frac{L}{n} \geq \mu$ | $\frac{L}{n}-\mu$ | $\frac{1}{n}$ | $\frac{1}{L}$ | $\frac{\mu n}{L}$ | $\tilde{O}\left(\sqrt{n\frac{L}{\mu}}\log\left(\frac{F_0}{\varepsilon}\right) + \frac{\sigma^2}{\mu\varepsilon}\right)$ |

In this table, $F_0 := F(x_0) - F^\star$ and the methods SAGA/MISO/SVRG are applied to the stochastic finite-sum problem discussed in Section 1 with $n$ $L$-smooth functions. As in the deterministic case, we

note that when $L/n \leq \mu$, there is no acceleration for SAGA/MISO/SVRG since the complexity of the unaccelerated method $\mathcal{M}$ is $\tilde{O}\left(n \log\left(F_0/\varepsilon\right) + \sigma^2/\mu\varepsilon\right)$, which is independent of the condition number and already optimal [29]. In comparison, the logarithmic terms in $L, \mu$ that are hidden in the notation $\tilde{O}$ do not appear for a variant of the SVRG method with direct acceleration introduced in [29]. Here, our approach is more generic. Note also that $\sigma^2$ for prox-SGD and SAGA/MISO/SVRG cannot be compared to each other since the source of randomness is larger for prox-SGD, see [7, 29].

# 5 Experiments

In this section, we perform numerical evaluations by following [29], which was notably able to make SVRG and SAGA robust to stochastic noise, and accelerate SVRG. More details and experiments are given in Appendix E.

**Datasets, formulations and methods.** We consider $\ell_2$-logistic regression and support vector machine with the squared hinge loss, as in [29], see Appendix E. Studying the squared hinge loss is interesting since its gradients are unbounded on the optimization domain, which may break the bounded noise assumption. The regularization parameter $\mu$ acts as the strong convexity constant and is chosen among the smallest values one would try when performing parameter search. Specifically, we consider $\mu = 1/10n$ and $\mu = 1/100n$, where $n$ is the number of training points. Following [7, 29, 54], we consider DropOut perturbations [48] with rate $\delta = 0$ (no noise), $\delta = 0.01$ and $\delta = 0.1$, and consider three datasets used in [29], alpha, gen, ckn-cifar, see Appendix E.

We consider the variants of SVRG and SAGA of [29], which use decreasing step sizes when $\delta > 0$ (otherwise, they do not converge). We use the suffix "-d" each time decreasing step sizes are used. We also consider Katyuasha [1] when $\delta = 0$, and the accelerated SVRG method of [29].

**Practical questions and implementation.** In all setups, we choose the parameter $\kappa$ according to theory, which are described in the previous section, following Catalyst [33]. For composite problems, Proposition 5 suggests to use $x_{k-1}$ as a warm start for inner-loop problems. For smooth ones, [34] shows that in fact, other choices such as $y_{k-1}$ are appropriate and lead to similar complexity results. In our experiments with smooth losses, we use $y_{k-1}$, which has shown to perform consistently better.

The strategy for $\eta_k$ discussed in Proposition 6 suggests to use constant step-sizes for a while in the inner-loop, typically of order $1/(\kappa + L)$ for the methods we consider, before using an exponentially decreasing schedule. Unfortunately, even though theory suggests a rate of decay in $(1 - \sqrt{q}/2)^k$, it does not provide useful insight on when decaying should start since the theoretical time requires knowing $\sigma^2$. A similar issue arise in stochastic optimization techniques involving iterate averaging [9]. We adopt a similar heuristic as in this literature and start decaying after $k_0$ epochs, with $k_0 = 30$. Finally, we discuss the number of iterations of $\mathcal{M}$ to perform in the inner-loop. When $\eta_k = 1$, the theoretical value is of order $\tilde{O}(1/\tau) = \tilde{O}(n)$, and we choose exactly $n$ iterations (one epoch), as in Catalyst [34]. After starting decaying the step-sizes ($\eta_k < 1$), we use $\lceil n/\eta_k \rceil$, according to theory.

**Experiments and conclusions.** We run each experiment five time with a different random seed and average the results. All curves also display one standard deviation. Appendix E contains numerous experiments, where we vary the amount of noise, the type of approach (SVRG vs. SAGA), the amount of regularization $\mu$, and choice of loss function. In Figure 1, we show a subset of these curves. Most of them show that acceleration may be useful even in the stochastic optimization regime, consistently with [29], but that all acceleration methods may no perform well for very ill-conditioned problems with $\mu = 1/1000n$, which are unrealistic in the context of empirical risk minimization.

## Acknowledgments

This work was supported by the ERC grant SOLARIS (number 714381).

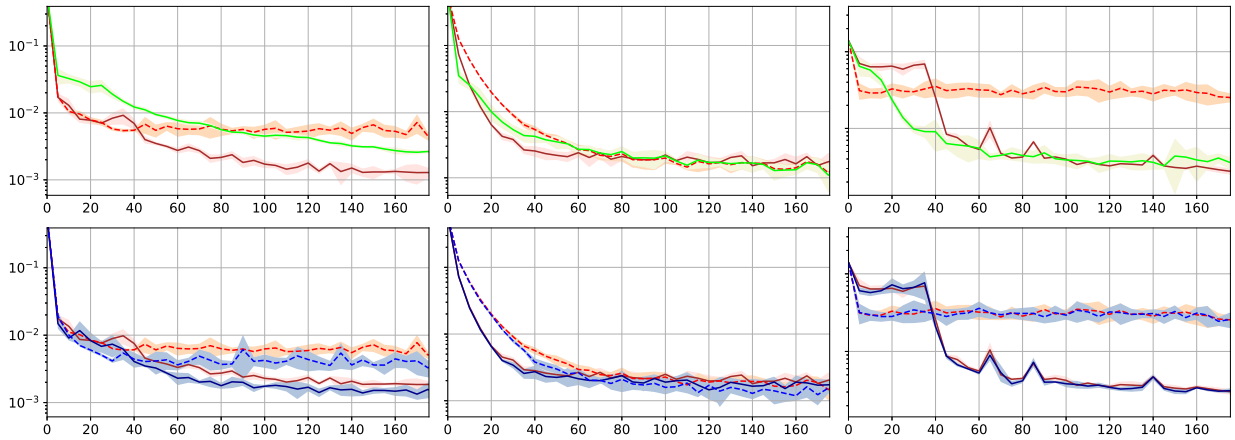

Figure 1: Accelerating SVRG-like (top) and SAGA (bottom) methods for $\ell_2$-logistic regression with $\mu = 1/(100n)$ (bottom) for $\delta = 0.1$. All plots are on a logarithmic scale for the objective function value, and the $x$-axis denotes the number of epochs. The colored tubes around each curve denote a standard deviations across 5 runs. They do not look symmetric because of the logarithmic scale.

# References


[1] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of Symposium on Theory of Computing (STOC)*, 2017.

[2] Y. Arjevani and O. Shamir. Dimension-free iteration complexity of finite sum optimization problems. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[3] H. Asi and J. C. Duchi. Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity. *preprint arXiv:1810.05633*, 2018.

[4] N. S. Aybat, A. Fallah, M. Gurbuzbalaban, and A. Ozdaglar. A universally optimal multistage accelerated stochastic gradient method. *preprint arXiv:1901.08022*, 2019.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[6] D. P. Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical Programming*, 129(2):163, 2011.

[7] A. Bietti and J. Mairal. Stochastic optimization with variance reduction for infinite datasets with finite-sum structure. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[8] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[9] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning, 2016. URL `https://arxiv.org/abs/1606.04838`. quantization overview.

[10] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[11] A. Chambolle and T. Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI Journal of Computational Mathematics*, 1: 29–54, 2015.

[12] M. B. Cohen, J. Diakonikolas, and L. Orecchia. On acceleration with noise-corrupted gradients. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2018.

[13] A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19 (3):1171–1183, 2008.

[14] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[15] A. Defazio, T. Caetano, and J. Domke. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2014.

[16] O. Devolder. Stochastic first order methods in smooth convex optimization. Technical report, Université catholique de Louvain, 2011.

[17] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

[18] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4): 1469–1492, 2012.

[19] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization II: Shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

[20] R. M. Gower, P. Richtárik, and F. Bach. Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching. *preprint arXiv:1805.02632*, 2018.

[21] O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2 (4):649–664, 1992.

[22] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II.* Springer, 1996.

[23] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[24] C. Hu, W. Pan, and J. T. Kwok. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems (NIPS)*. 2009.

[25] A. Iouditski and Y. Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions. *preprint arXiv:1401.1792*, 2014.

[26] J. Konečnỳ and P. Richtárik. Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3:9, 2017.

[27] D. Kovalev, S. Horvath, and P. Richtarik. Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. *preprint arXiv:1901.08689*, 2019.

[28] B. Kulis and P. L. Bartlett. Implicit online learning. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2010.

[29] A. Kulunchakov and J. Mairal. Estimate sequences for stochastic composite optimization: Variance reduction, acceleration, and robustness to noise. *preprint arXiv:1901.08788*, 2019.

[30] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133 (1):365–397, 2012.

[31] G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, 171(1–2):167–215, 2018.

[32] G. Lan and Y. Zhou. Random gradient extrapolation for distributed and stochastic optimization. *SIAM Journal on Optimization*, 28(4):2753–2782, 2018.

[33] H. Lin, J. Mairal, and Z. Harchaoui. A Universal Catalyst for First-Order Optimization. In *28th International Conference on Neural Information Processing Systems, NIPS'15*, pages 3384–3392, Montreal, Canada, Dec. 2015. MIT Press. URL `https://hal.inria.fr/hal-01160728`. main paper (9 pages) + appendix (21 pages).

[34] H. Lin, J. Mairal, and Z. Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research (JMLR)*, 18(212):1–54, 2018.

[35] H. Lin, J. Mairal, and Z. Harchaoui. An inexact variable metric proximal point algorithm for generic quasi-Newton acceleration. *preprint arXiv:1610.00960*, 2019.

[36] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[37] J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[38] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletins de la Socitété Mathématique de France*, 93(2):273–299, 1965.

[39] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[40] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[41] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

[42] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[43] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2017.

[44] L. M. Nguyen, P. H. Nguyen, M. van Dijk, P. Richtárik, K. Scheinberg, and M. Takáč. SGD and Hogwild! convergence without the bounded gradients assumption. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2018.

[45] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst acceleration for gradient-based non-convex optimization. *preprint arXiv:1703.10993*, 2018.

[46] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[47] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1):105–145, 2016.

[48] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15 (1):1929–1958, 2014.

[49] P. Toulis, D. Tran, and E. Airoldi. Towards stability and optimality in stochastic gradient descent. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

[50] P. Toulis, T. Horel, and E. M. Airoldi. Stable Robbins-Monro approximations through stochastic proximal updates. *preprint arXiv:1510.00967*, 2018.

[51] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. 2008. unpublished.

[52] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research (JMLR)*, 11(Oct):2543–2596, 2010.

[53] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[54] S. Zheng and J. T. Kwok. Lightweight stochastic optimization for minimizing finite sums with infinite data. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2018.

[55] K. Zhou. Direct acceleration of SAGA using sampled negative momentum. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[56] K. Zhou, F. Shang, and J. Cheng. A simple stochastic variance reduced algorithm with fast convergence rates. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2018.

# A Useful Results and Definitions

In this section, we present auxiliary results and definitions.

**Definition 7** (Proximal operator)**.** *Given a convex lower-semicontinuous function $\psi$ defined on $\mathbb{R}^p$, the proximal operator of $\psi$ is defined as the unique solution of the strongly-convex problem*

$$Prox_{\psi}[y] = \operatorname*{argmin}_{x\in\mathbb{R}^p} \left\{ \frac{1}{2}\|y-x\|^2 + \psi(x) \right\}.$$

**Lemma 8** (Convergence rate of the sequences $(\alpha_k)_{k\geq 0}$ and $(A_k)_{k\geq 0}$)**.** *Consider the sequence in $(0,1)$ defined by the recursion*

$$\alpha_k^2 = (1-\alpha_k)\alpha_{k-1}^2 + q\alpha_k \quad \textit{with} \quad 0 \leq q < 1,$$

*and define $A_k = \prod_{t=1}^{k}(1-\alpha_t)$. Then,*

- *if $q = 0$ and $\alpha_0 = 1$, then, for all $k \geq 1$,*

$$\frac{2}{(k+2)^2} \leq A_k = \alpha_k^2 \leq \frac{4}{(k+2)^2}.$$

- *if $\alpha_0 = \sqrt{q}$, then for all $k \geq 1$,*

$$A_k = (1-\sqrt{q})^k \quad \textit{and} \quad \alpha_k = \sqrt{q}.$$

- *if $\alpha_0 = 1$, then for all $k \geq 1$,*

$$A_k \leq \min\left((1-\sqrt{q})^k, \frac{4}{(k+2)^2}\right) \quad \textit{and} \quad \alpha_k \geq \max\left(\sqrt{q}, \frac{\sqrt{2}}{k+2}\right).$$

*Proof.* We prove the three points, one by one.

**First point.** Let us prove the first point when $q = 0$ and $\alpha_0 = 1$. The relation $A_k = \alpha_k^2$ is obvious for all $k \geq 1$ and the relation $\alpha_k^2 \leq \frac{4}{(k+2)^2}$ holds for $k = 0$. By induction, let us assume that we have the relation $\alpha_{k-1}^2 \leq \frac{4}{(k+1)^2}$ and let us show that it propagates for $\alpha_k^2$. Assume, by contradiction, that $\alpha_k^2 > \frac{4}{(k+2)^2}$, meaning that $\alpha_k > \frac{2}{(k+2)}$. Then,

$$\alpha_k^2 = (1-\alpha_k)\alpha_{k-1}^2 \leq (1-\alpha_k)\frac{4}{(k+1)^2} < \frac{4k}{(k+2)(k+1)^2} = \frac{4}{(k+2)(k+2+\frac{1}{k})} < \frac{4}{(k+2)^2},$$

and we obtain a contradiction. Therefore, $\alpha_k^2 \leq \frac{4}{(k+2)^2}$ and the induction hypothesis allows us to conclude for all $k \geq 1$. Then, note [45] that we also have for all $k \geq 1$,

$$A_k = \prod_{t=1}^{k}(1-\alpha_t) \geq \prod_{t=1}^{k}\left(1 - \frac{2}{t+2}\right) = \frac{2}{(k+1)(k+2)} \geq \frac{2}{(k+2)^2}.$$

**Second point.** The second point is obvious by induction.

**Third point.** For the third point, we simply assume $\alpha_0 = 1$ such that $\alpha_0 \geq \sqrt{q}$. Then, the relation $\alpha_k \geq \sqrt{q}$ and therefore $A_k \leq (1-\sqrt{q})^k$ are easy to show by induction. Then, consider the sequence defined recursively by $u_k^2 = (1-u_k)u_{k-1}^2$ with $u_0 = 1$. From the first point, we have that $\frac{\sqrt{2}}{k+2} \leq u_k \leq \frac{2}{k+2}$. We will show that $\alpha_k \geq u_k$ for all $k \geq 0$, which will be sufficient to conclude since then we would have $A_k \leq \prod_{t=1}^{k}(1-u_t) \leq \frac{4}{(k+2)^2}$. First, we note that $\alpha_0 = u_0$; then, assume that $\alpha_{k-1} \geq u_{k-1}$ and also assume by contradiction that $\alpha_k > u_k$. This implies that

$$u_k^2 = (1-u_k)u_{k-1}^2 \leq (1-u_k)\alpha_{k-1}^2 < (1-\alpha_k)\alpha_{k-1}^2 \leq \alpha_k^2,$$

which contradicts the assumption $\alpha_k > u_k$. This allows us to conclude by induction. □

**Lemma 9** (Convergence rate of sequences $\Theta_k = \prod_{i=1}^{k}(1-\theta_i)$)**.** *Consider the sequence $\theta_j = \frac{\gamma}{(1+j)^{1+\gamma}}$ with $\gamma$ in $(0,1]$. Then,*

$$e^{-(1+\gamma)} \leq \Theta_k \leq 1. \tag{15}$$

*Proof.* We use the classical inequality $\log(1+u) \geq \frac{u}{1+u}$ for all $u > -1$:

$$-\log(\Theta_k) = -\sum_{j=1}^{k}\log\left(1 - \frac{\gamma}{(1+j)^{1+\gamma}}\right) \leq \sum_{j=1}^{k}\frac{\gamma}{(1+j)^{1+\gamma}-\gamma} \leq \sum_{j=1}^{k}\frac{\gamma}{j^{1+\gamma}},$$

when noting that the function $g(x) = (1+x)^{1+\gamma} - x^{1+\gamma}$ is greater than $\gamma$ for all $x \geq 1$, since $g(1) \geq 1 \geq \gamma$ and $g$ is non-decreasing. Then,

$$-\log(\Theta_k) \leq \sum_{j=1}^{k}\frac{\gamma}{j^{1+\gamma}} \leq \gamma + \gamma\int_{x=1}^{k}\frac{1}{x^{1+\gamma}}dx = \gamma + 1 - \frac{1}{k^\gamma} \leq \gamma + 1.$$

Then, we immediately obtain (15).

□

# B Details about Complexity Results

## B.1 Details about (5)

Consider the complexity (2) with $h = f$. To achieve the accuracy $2B\sigma^2$, it is sufficient to run the method $\mathcal{M}$ for $t_0$ iterations, such that

$$C(1-\tau)^{t_0}(F(x_0) - F^\star) \leq B\sigma^2.$$

It is then easy to see that this inequality is satisfied as soon as $t_0$ is greater than $\frac{1}{\tau}\log(C(F(x_0)-F^\star)/B\sigma^2)$. Since $\varepsilon \leq B\sigma^2$ and using the concavity of the logarithm function, it is also sufficient to choose $t_0 = \frac{1}{\tau}\log(C(F(x_0)-F^\star)/\varepsilon)$.

Then, we perform $K$ restart stages such that $\varepsilon_K \leq \varepsilon$. Each stage is initialized with a point $x_k$ satisfying $\mathbb{E}[F(x_k)-F^\star] \leq \varepsilon_{k-1}$, and the goal of each stage is to reduce the error by a factor $1/2$. Given that $\eta_k$ increases the computational cost, the complexity of the $k$-th stage is then upper-bounded by $\frac{2^k}{\tau}\log(2C)$, leading to the global complexity

$$O\left(\frac{1}{\tau}\log\left(\frac{C(F(x_0)-F^\star)}{\varepsilon}\right)+\sum_{k=1}^{K}\frac{2^k}{\tau}\log(2C)\right) \quad \text{with} \quad K=\left\lceil \log_2\left(\frac{2B\sigma^2}{\varepsilon}\right)\right\rceil,$$

and (5) follows by elementary calculations.

## B.2 Obtaining (5) from (6)

Since $h$ is $\mu$-strongly convex, we notice that (6) implies the rate

$$\mathbb{E}[h(z_t)-h^\star] \leq \frac{D(h(z_0)-h^\star)}{\mu t^d}+\frac{B\sigma^2}{2},$$

by using the strong convexity inequality $h(z_0) \geq h^\star + \frac{\mu}{2}\|z_0-z^\star\|^2$. After running the algorithm for $t' = \lceil (2D/\mu)^{1/d}\rceil$ iterations, we can show that

$$\mathbb{E}[h(z_{t'})-h^\star] \leq \frac{h(z_0)-h^\star}{2}+\frac{B\sigma^2}{2}.$$

Then, when restarting the procedure $s$ times (using the solution of the previous iteration as initialization), and denoting by $h_{st'}$ the last iterate, it is easy to show that

$$\mathbb{E}[h(x_{st'})-h^\star] \leq \frac{h(x_0)-h^\star}{2^s}+\frac{B\sigma^2}{2}\left(\sum_{i=0}^{s-1}\frac{1}{2^i}\right) \leq \frac{h(z_0)-h^\star}{2^s}+B\sigma^2.$$

Then, calling $t = st'$, we can use the inequality $2^{-u} \leq 1-\frac{u}{2}$ for $u$ in $[0,1]$, due to convexity, and

$$\mathbb{E}[h(z_t)-h^\star] \leq (h(z_0)-h^\star)\left(2^{-1/t'}\right)^t + B\sigma^2 = (h(z_0)-h^\star)\left(1-\frac{1}{2t'}\right)^k + B\sigma^2,$$

which gives us (2) with $C=1$ and $\tau = \frac{1}{2t'}$. It is then easy to obtain (5) by following similar steps as in Section B.1, by noticing that the restart frequency is of the same order $O(1/\tau)$.

## B.3 Details about (13)

**Inner-loop complexity.** Since $\eta_k$ is chosen such that the bias $\eta_k B\sigma^2$ is smaller than $\varepsilon_k$, the number of iterations of $\mathcal{M}$ to solve the sub-problem is $\tilde{O}(\tau) = O(\log(1/q)\tau)$, as in the deterministic case, and the complexity is thus $\tilde{O}(\tau/\eta_k)$.

**Outer-loop complexity.** Since $\mathbb{E}[F(x_k)-F^\star] \leq O((1-\sqrt{q}/3)^k(F(x_0)-F^\star))/q$ according to Proposition 4, it suffices to choose

$$K = O\left(\frac{1}{\sqrt{q}}\log\left(\frac{F(x_0)-F^\star}{q\varepsilon}\right)\right)$$

iterations to guarantee $\mathbb{E}[F(x_K)-F^\star] \leq \varepsilon = O(\varepsilon_K/q) = O((1-\sqrt{q}/3)^K(F(x_0)-F^\star)/q)$.

**Global complexity.** The total complexity to guarantee $\mathbb{E}[F(x_k) - F^\star] \leq \varepsilon$ is then

$$\begin{aligned}
C &= \sum_{k=1}^{K} \tilde{O}\left(\frac{\tau}{\eta_k}\right) \\
&\leq \tilde{O}\left(\sum_{k=1}^{K} \tau + \sum_{k=1}^{K} \frac{B\sigma^2\tau}{\varepsilon_k}\right) \\
&= \tilde{O}\left(\sum_{k=1}^{K} \tau + \sum_{k=1}^{K} \frac{B\sigma^2\tau}{\left(1 - \frac{\sqrt{q}}{3}\right)^k (F(x_0) - F^\star)}\right) \\
&= \tilde{O}\left(\frac{\tau}{\sqrt{q}} \log\left(\frac{F(x_0) - F^\star}{\varepsilon}\right) + \frac{B\sigma^2\tau}{\sqrt{q}\left(1 - \frac{\sqrt{q}}{3}\right)^{K+1} (F(x_0) - F^\star)}\right) \\
&= \tilde{O}\left(\frac{\tau}{\sqrt{q}} \log\left(\frac{F(x_0) - F^\star}{\varepsilon}\right) + \frac{B\sigma^2\tau}{q^{3/2}\varepsilon}\right),
\end{aligned}$$

where the last relation uses the fact that $\varepsilon = O(\varepsilon_K/q) = O((1 - \sqrt{q}/3)^K (F(x_0) - F^\star)/q)$.

### B.4 Complexity of accelerated stochastic proximal gradient descent with inexact prox

Assume that $h_k(x_k) - h_k^\star \leq \varepsilon_k$. Then, following similar steps as in (11),

$$\begin{aligned}
\mathbb{E}[F(x_k)] &\leq \mathbb{E}[h_k(x_k)] + \mathbb{E}[(g_k - \nabla f(y_{k-1}))^\top (x_k - y_{k-1})] \\
&= \mathbb{E}[h_k(x_k)] + \mathbb{E}[(g_k - \nabla f(y_{k-1}))^\top (x_k - u_{k-1})] \\
&= \mathbb{E}[h_k(x_k)] + \mathbb{E}[(g_k - \nabla f(y_{k-1}))^\top (x_k - x_k^\star)] + \mathbb{E}[(g_k - \nabla f(y_{k-1}))^\top (x_k^\star - u_{k-1})] \\
&\leq \mathbb{E}[h_k(x_k)] + \mathbb{E}[(g_k - \nabla f(y_{k-1}))^\top (x_k - x_k^\star)] + \frac{\sigma^2}{\kappa + \mu} \\
&\leq \mathbb{E}[h_k(x_k)] + \frac{\mathbb{E}[\|g_k - \nabla f(y_{k-1})\|^2]}{2(\kappa + \mu)} + \frac{(\kappa + \mu)\mathbb{E}[\|x_k - x_k^\star\|^2]}{2} + \frac{\sigma^2}{\kappa + \mu} \\
&\leq \mathbb{E}[h_k(x_k)] + \mathbb{E}[h_k(x_k) - h_k^\star] + \frac{3\sigma^2}{2(\kappa + \mu)} \\
&\leq \mathbb{E}[h_k^\star] + 2\varepsilon_k + \frac{3\sigma^2}{2(\kappa + \mu)}.
\end{aligned}$$

And thus, $\delta_k = 2\varepsilon_k + \frac{3\sigma^2}{2(\kappa+\mu)}$.

## C Proofs of Main Results

### C.1 Proof of Propositions 1 and 4

*Proof.* In order to treat both propositions jointly, we introduce the quantity

$$w_k = \begin{cases} x_k & \text{for variant } \mathcal{A} \\ x_k^\star & \text{for variant } \mathcal{B} \end{cases},$$

and, for all $k \geq 1$,

$$v_k = w_k + \frac{1 - \alpha_{k-1}}{\alpha_{k-1}}(w_k - x_{k-1}), \tag{16}$$

with $v_0 = x_0$, as well as $\eta_k = \frac{\alpha_k - q}{1-q}$ for all $k \geq 0$.

Note that the following relations hold for all $k \geq 1$, keeping in mind that $\alpha_k^2 = (1-\alpha_k)\alpha_{k-1}^2 + q\alpha_k$:

$$\begin{aligned} 1 - \eta_k &= \frac{1-\alpha_k}{1-q} = \frac{(\kappa+\mu)(1-\alpha_k)}{\kappa} \\ \eta_k &= \frac{\alpha_k - q}{1-q} = \frac{\alpha_k^2 - q\alpha_k}{\alpha_k - q\alpha_k} = \frac{\alpha_{k-1}^2(1-\alpha_k)}{\alpha_k - \alpha_k^2 + (1-\alpha_k)\alpha_{k-1}^2} = \frac{\alpha_{k-1}^2}{\alpha_{k-1}^2 + \alpha_k}. \end{aligned}$$

Then, based on the previous relations, we have

$$\begin{aligned} y_k &= w_k + \beta_k(w_k - x_{k-1}) + \frac{(\kappa+\mu)(1-\alpha_k)}{\kappa}(x_k - w_k) \\ &= w_k + \frac{\alpha_{k-1}(1-\alpha_{k-1})}{\alpha_{k-1}^2 + \alpha_k}(w_k - x_{k-1}) + (1-\eta_k)(x_k - w_k) \\ &= w_k + \frac{\eta_k(1-\alpha_{k-1})}{\alpha_{k-1}}(w_k - x_{k-1}) + (1-\eta_k)(x_k - w_k) \\ &= \eta_k v_k + (1-\eta_k)x_k, \end{aligned}$$

which is similar to the relation used in [34] when $w_k = x_k$. Then, the proof differs from [34] since we introduce the surrogate function $h_k$. For all $x$ in $\mathbb{R}^p$,

$$\begin{aligned} h_k(x) &\geq h_k^\star + \frac{\kappa+\mu}{2}\|x - x_k^\star\|^2 \quad \text{(by strong convexity, see } \mathcal{H}_1) \\ &= h_k^\star + \frac{\kappa+\mu}{2}\|x - w_k\|^2 + \underbrace{\frac{\kappa+\mu}{2}\|w_k - x_k^\star\|^2 + (\kappa+\mu)\langle x - w_k, w_k - x_k^\star\rangle}_{-\Delta_k(x)}. \end{aligned} \tag{17}$$

Introduce now the following quantity for the convergence analysis:

$$z_{k-1} = \alpha_{k-1}x^\star + (1-\alpha_{k-1})x_{k-1},$$

and consider $x = z_{k-1}$ in (17) while taking expectations, noting that all random variables indexed by $k$–1 are deterministic given $\mathcal{F}_{k-1}$,

$$\begin{aligned} \mathbb{E}[F(x_k)] &\leq \mathbb{E}[h_k^\star] + \delta_k \qquad \text{(by } \mathcal{H}_3) \\ &\leq \mathbb{E}[h_k(z_{k-1})] - \mathbb{E}\left[\frac{\kappa+\mu}{2}\|z_{k-1} - w_k\|^2\right] + \mathbb{E}[\Delta_k(z_{k-1})] + \delta_k \\ &\leq \mathbb{E}[F(z_{k-1})] + \mathbb{E}\left[\frac{\kappa}{2}\|z_{k-1} - y_{k-1}\|^2\right] - \mathbb{E}\left[\frac{\kappa+\mu}{2}\|z_{k-1} - w_k\|^2\right] + \mathbb{E}[\Delta_k(z_{k-1})] + \delta_k, \end{aligned} \tag{18}$$

where the last inequality is due to $(\mathcal{H}_2)$.

Let us now open a parenthesis and derive a few relations that will be useful to find a Lyapunov function. To use more compact notation, define $X_k = \mathbb{E}[\|x^\star - x_k\|^2]$, $V_k = \mathbb{E}[\|x^\star - v_k\|^2]$ and $F_k = \mathbb{E}[F(x_k) - F^\star]$, and note that

$$\begin{aligned} \mathbb{E}[F(z_{k-1})] &\leq \alpha_{k-1}f^\star + (1-\alpha_{k-1})\mathbb{E}[F(x_{k-1})] - \frac{\mu\alpha_{k-1}(1-\alpha_{k-1})}{2}X_{k-1} \\ \mathbb{E}[\|z_{k-1} - w_k\|^2] &= \alpha_{k-1}^2 V_k \\ \mathbb{E}[\|z_{k-1} - y_{k-1}\|^2] &\leq \alpha_{k-1}(\alpha_{k-1} - \eta_{k-1})X_{k-1} + \alpha_{k-1}\eta_{k-1}V_{k-1}. \end{aligned} \tag{19}$$

The first relation is due to the convexity of $f$; the second one can be obtained from the definition of $v_k$ in (16) after simple calculations; the last one can be obtained as in the proof of Theorem 3 in [34] (end of page 16).

We may now close the parenthesis, come back to (18) and we use the relations (19):

$$F_k + \frac{(\kappa+\mu)\alpha_{k-1}^2}{2} V_k \leq (1-\alpha_{k-1})F_{k-1} - \frac{\mu\alpha_{k-1}(1-\alpha_{k-1})}{2} X_{k-1} + \\ \frac{\kappa}{2}\alpha_{k-1}(\alpha_{k-1}-\eta_{k-1})X_{k-1} + \frac{\kappa}{2}\alpha_{k-1}\eta_{k-1}V_{k-1} + \delta_k + \mathbb{E}[\Delta_k(z_{k-1})].$$

It is then easy to see that the terms involving $X_{k-1}$ cancel each other since $\eta_{k-1} = \alpha_{k-1} - \frac{\mu}{\kappa}(1-\alpha_{k-1})$.

**Lyapunov function.** We may finally define the Lyapunov function

$$S_k = (1-\alpha_k)F_k + \frac{\kappa\alpha_k\eta_k}{2} V_k. \tag{20}$$

and we obtain

$$\frac{S_k}{1-\alpha_k} \leq S_{k-1} + \delta_k + \mathbb{E}[\Delta_k(z_{k-1})], \tag{21}$$

For variant Algorithm 1, we have $\Delta_k(z_{k-1}) = 0$ since $w_k = x_k^\star$, and we obtain the following relation by unrolling the recursion:

$$S_k \leq A_k \left( S_0 + \sum_{j=1}^{k} \frac{\delta_j}{A_{j-1}} \right) \qquad \text{with} \qquad A_j = \prod_{i=1}^{j}(1-\alpha_i). \tag{22}$$

**Specialization to $\mu > 0$.** When $\mu > 0$, we have $\alpha_0 = \sqrt{q}$ and

$$\begin{aligned} S_0 &= (1-\sqrt{q})(F(x_0) - F^\star) + \frac{\kappa\sqrt{q}(\sqrt{q}-q)}{2(1-q)}\|x_0 - x^\star\|^2 \\ &= (1-\sqrt{q})(F(x_0) - F^\star) + \frac{(\kappa+\mu)\sqrt{q}(\sqrt{q}-q)}{2}\|x_0 - x^\star\|^2 \\ &= (1-\sqrt{q})(F(x_0) - F^\star) + \frac{\mu(1-\sqrt{q})}{2}\|x_0 - x^\star\|^2 \\ &\leq 2(1-\sqrt{q})(F(x_0) - F^\star), \end{aligned} \tag{23}$$

by using the strong convexity inequality $F(x_0) \geq F^\star + \frac{\mu}{2}\|x_0 - x^\star\|^2$. Then, noting that $\mathbb{E}[F(x_k) - F^\star] \leq \frac{S_k}{1-\sqrt{q}}$ and $A_k = (1-\sqrt{q})^k$ (Lemma 8), we immediately obtain the first part of (8) from (22).

**Specialization to $\mu = 0$.** When $\mu = 0$, we have $\alpha_0 = 1$ and $S_0 = \frac{\kappa}{2}\|x_0 - x^\star\|^2$. Then, according to Lemma 8 and (22), for $k \geq 1$,

$$\mathbb{E}[F(x_k) - F^\star] \leq \frac{S_k}{1-\alpha_k} \leq \frac{\kappa\|x_0 - x^\star\|^2}{2} A_{k-1} + \sum_{j=1}^{k} \frac{\delta_j A_{k-1}}{A_{j-1}}, \tag{24}$$

and we obtain the second part of (8) noting that $A_{k-1} \leq \frac{4}{(k+1)^2}$ and that $A_{j-1} \geq \frac{2}{(j+1)^2}$. Then, Proposition 1 is proven.

**Proof of Proposition 4.** When $w_k = x_k$, we need to control the quantity $\Delta_k(z_{k-1})$. Consider any scalar $\theta_k$ in $(0,1)$. Then,

$$\begin{aligned}
\Delta_k(z_{k-1}) &= -\frac{\kappa+\mu}{2}\|x_k - x_k^\star\|^2 - (\kappa+\mu)\langle z_{k-1} - x_k, x_k - x_k^\star\rangle \\
&= -\frac{\kappa+\mu}{2}\|x_k - x_k^\star\|^2 - (\kappa+\mu)\alpha_{k-1}\langle x^\star - v_k, x_k - x_k^\star\rangle \\
&\le -\frac{\kappa+\mu}{2}\|x_k - x_k^\star\|^2 + (\kappa+\mu)\alpha_{k-1}\|x^\star - v_k\|\|x_k - x_k^\star\| \\
&\le \left(\frac{1}{\theta_k} - 1\right)\frac{\kappa+\mu}{2}\|x_k - x_k^\star\|^2 + \frac{\theta_k(\kappa+\mu)\alpha_{k-1}^2}{2}\|x^\star - v_k\|^2 \quad \text{(Young's inequality)} \\
&\le \left(\frac{1}{\theta_k} - 1\right)(h_k(x_k) - h_k^\star) + \frac{\theta_k(\kappa+\mu)\alpha_{k-1}^2}{2}\|x^\star - v_k\|^2 \quad \text{(since } \theta_k \le 1\text{)} \\
&\le \left(\frac{1}{\theta_k} - 1\right)(h_k(x_k) - h_k^\star) + \frac{\theta_k(\kappa+\mu)(\alpha_k^2 - \alpha_k q)}{2(1-\alpha_k)}\|x^\star - v_k\|^2 \\
&= \left(\frac{1}{\theta_k} - 1\right)(h_k(x_k) - h_k^\star) + \frac{\theta_k\kappa\alpha_k\eta_k}{2(1-\alpha_k)}\|x^\star - v_k\|^2
\end{aligned}$$

Then, we take expectations and, noticing that the quadratic term involving $\|x^\star - v_k\|$ is smaller than $\theta_k S_k/(1-\alpha_k)$ in expectation (from the definition of $S_k$ in (20)), we obtain

$$\mathbb{E}[\Delta_k(z_{k-1})] \le \left(\frac{1}{\theta_k} - 1\right)\varepsilon_k + \frac{\theta_k S_k}{1-\alpha_k},$$

and from (21),

$$S_k \le \frac{(1-\alpha_k)}{(1-\theta_k)}\left(S_{k-1} + \delta_k + \left(\frac{1}{\theta_k} - 1\right)\varepsilon_k\right).$$

By unrolling the recursion, we obtain

$$S_k \le \frac{A_k}{\Theta_k}\left(S_0 + \sum_{j=1}^{k}\frac{\Theta_{j-1}}{A_{j-1}}\left(\delta_j - \varepsilon_j + \frac{\varepsilon_j}{\theta_j}\right)\right) \quad \text{with} \quad A_j = \prod_{i=1}^{j}(1-\alpha_i) \quad \text{and} \quad \Theta_j = \prod_{i=1}^{j}(1-\theta_i). \tag{25}$$

**Specialization to $\mu > 0$.** When $\mu > 0$, we have $\alpha_k = \sqrt{q}$ for all $k \ge 0$. Then, we may choose $\theta_k = \frac{\sqrt{q}}{2}$; then, $1 - \sqrt{q} \le \left(1 - \frac{\sqrt{q}}{2}\right)^2$ and $\frac{A_k}{\Theta_k} \le \left(1 - \frac{\sqrt{q}}{2}\right)^k$ for all $k \ge 0$. By using the relation (23), we obtain

$$\begin{aligned}
S_k &\le 2\left(1 - \frac{\sqrt{q}}{2}\right)^k (1-\sqrt{q})(F(x_0) - F^\star) + 2\sum_{j=1}^{k}\left(\frac{1-\sqrt{q}}{1-\frac{\sqrt{q}}{2}}\right)^{k-j+1}\left(\delta_j - \varepsilon_j + \frac{\varepsilon_j}{\sqrt{q}}\right) \\
&\le (1-\sqrt{q})\left(2\left(1 - \frac{\sqrt{q}}{2}\right)^k (F(x_0) - F^\star) + 4\sum_{j=1}^{k}\left(\frac{1-\sqrt{q}}{1-\frac{\sqrt{q}}{2}}\right)^{k-j}\left(\delta_j - \varepsilon_j + \frac{\varepsilon_j}{\sqrt{q}}\right)\right), \\
&\le (1-\sqrt{q})\left(2\left(1 - \frac{\sqrt{q}}{2}\right)^k (F(x_0) - F^\star) + 4\sum_{j=1}^{k}\left(1 - \frac{\sqrt{q}}{2}\right)^{k-j}\left(\delta_j - \varepsilon_j + \frac{\varepsilon_j}{\sqrt{q}}\right)\right),
\end{aligned}$$

where the second inequality uses $\frac{1}{1-\frac{\sqrt{q}}{2}} \le 2$. Since $(1-\sqrt{q})\mathbb{E}[F(x_k) - F^\star] \le S_k$, we obtain the first part of Proposition (4).

**Specialization to $\mu = 0$.** When $\mu = 0$, we have $\alpha_0 = 1$ and $S_0 = \frac{\kappa}{2}\|x_0 - x^\star\|^2$. We may then choose $\theta_k = \frac{\gamma}{(k+1)^{1+\gamma}}$ for any $\gamma$ in $(0, 1]$, leading to $e^{-(1+\gamma)} \leq \Theta_k \leq 1$ for all $k \geq 0$ according to Lemma 9. Besides, according to the proof of Lemma 8, $\frac{2}{(k+2)^2} \leq A_k \leq \frac{4}{(k+2)^2}$ for all $k \geq 1$.

Then, from (25),

$$\begin{aligned}\mathbb{E}[F(x_k) - F^\star] &\leq \frac{A_{k-1}}{\Theta_k}\frac{\kappa\|x_0 - x^\star\|^2}{2} + \sum_{j=1}^{k} \frac{A_{k-1}\Theta_{j-1}}{\Theta_k A_{j-1}}\left(\delta_j - \varepsilon_j + \frac{\varepsilon_j}{\gamma}(1+j)^{1+\gamma}\right) \\ &\leq \frac{2e^{1+\gamma}}{(k+1)^2}\left(\kappa\|x_0 - x^\star\|^2 + \sum_{j=1}^{k}(j+1)^2(\delta_j - \varepsilon_j) + \frac{(j+1)^{3+\gamma}\varepsilon_j}{\gamma}\right),\end{aligned}$$

which yields the second part of Proposition (4). □

## C.2 Proof of Proposition 5

Assume that for $k \geq 2$, we have the relation

$$\mathbb{E}[h_{k-1}(x_{k-1}) - h_{k-1}^\star] \leq \varepsilon_{k-1}. \tag{26}$$

Then, we want to evaluate the quality of the initial point $x_{k-1}$ to minimize $h_k$.

$$\begin{aligned}h_k(x_{k-1}) - h_k^\star &= h_{k-1}(x_{k-1}) + \frac{\kappa}{2}\|x_{k-1} - y_{k-1}\|^2 - \frac{\kappa}{2}\|x_{k-1} - y_{k-2}\|^2 - h_k^\star \\ &= h_{k-1}(x_{k-1}) - h_{k-1}^\star + h_{k-1}^\star - h_k^\star + \frac{\kappa}{2}\|x_{k-1} - y_{k-1}\|^2 - \frac{\kappa}{2}\|x_{k-1} - y_{k-2}\|^2 \\ &= h_{k-1}(x_{k-1}) - h_{k-1}^\star + h_{k-1}^\star - h_k^\star - \kappa(x_{k-1} - y_{k-1})^\top(y_{k-1} - y_{k-2}) - \frac{\kappa}{2}\|y_{k-1} - y_{k-2}\|^2.\end{aligned} \tag{27}$$

Then, we may use the fact that $h_k^\star$ can be interpreted as the Moreau-Yosida smoothing of the objective $f$, defined as $G(y) = \min_{x \in \mathbb{R}^p} F(x) + \frac{\kappa}{2}\|x - y\|^2$, which gives us immediately a few useful results, as noted in [35]. Indeed, we know that $G$ is $\kappa$-smooth with $\nabla G(y_{k-1}) = \kappa(y_{k-1} - x_k^\star)$ for all $k \geq 1$ and

$$\begin{aligned}h_{k-1}^\star = G(y_{k-2}) &\leq G(y_{k-1}) + \nabla G(y_{k-1})^\top(y_{k-2} - y_{k-1}) + \frac{\kappa}{2}\|y_{k-1} - y_{k-2}\|^2 \\ &= h_k^\star + \kappa(y_{k-1} - x_k^\star)^\top(y_{k-2} - y_{k-1}) + \frac{\kappa}{2}\|y_{k-1} - y_{k-2}\|^2.\end{aligned} \tag{28}$$

Then, combining (27) and (28),

$$\begin{aligned}h_k(x_{k-1}) - h_k^\star &\leq h_{k-1}(x_{k-1}) - h_{k-1}^\star + \kappa(x_{k-1} - x_k^\star)^\top(y_{k-2} - y_{k-1}). \\ &\leq h_{k-1}(x_{k-1}) - h_{k-1}^\star + \kappa(x_{k-1} - x_{k-1}^\star)^\top(y_{k-2} - y_{k-1}) + \kappa(x_{k-1}^\star - x_k^\star)^\top(y_{k-2} - y_{k-1}) \\ &\leq h_{k-1}(x_{k-1}) - h_{k-1}^\star + \kappa(x_{k-1} - x_{k-1}^\star)^\top(y_{k-2} - y_{k-1}) + \kappa\|y_{k-1} - y_{k-2}\|^2 \\ &\leq h_{k-1}(x_{k-1}) - h_{k-1}^\star + \frac{\kappa}{2}\|x_{k-1} - x_{k-1}^\star\|^2 + \frac{3\kappa}{2}\|y_{k-1} - y_{k-2}\|^2 \\ &\leq \frac{3}{2}(h_{k-1}(x_{k-1}) - h_{k-1}^\star) + \frac{3\kappa}{2}\|y_{k-1} - y_{k-2}\|^2,\end{aligned}$$

where the third inequality uses the non-expansiveness of the proximal operator; the fourth inequality uses the inequality $a^\top b \leq \frac{\|a\|^2}{2} + \frac{\|b\|^2}{2}$ for vectors $a, b$, and the last inequality uses the strong convexity of $h_{k-1}$. Then, we may use the same upper-bound on $\|y_{k-1} - y_{k-2}\|$ as [34, Proposition 12], namely

$$\|y_{k-1} - y_{k-2}\|^2 \leq 36 \max\left(\|x_{k-1} - x^\star\|^2, \|x_{k-2} - x^\star\|^2, \|x_{k-3} - x^\star\|^2\right),$$

where we define $x_{-1} = x_0$ if $k = 2$.

### C.3 Proof of Proposition 6

The proof is similar to the derivation described in Section B.3.

**Inner-loop complexity.** With the choice of $\delta_k$, we have that $\xi_{k-1} = O(\delta_{k-1}/\sqrt{q})$. Besides, since we enforce $\mathbb{E}[H_k(x_k) - H_k^\star] \leq \delta_k$ for all $k \geq 0$, the result of Proposition 5 can be applied and the discussion following the proposition still applies, such that the complexity for computing $x_k$ is indeed $\tilde{O}(\tau/\eta_k)$.

**Outer-loop complexity.** Then, according to Proposition 1, it is easy to show that $\mathbb{E}[F(x_k) - F^\star] \leq O((1 - \sqrt{q}/2)^k (F(x_0) - F^\star))/\sqrt{q}$ and thus it suffices to choose

$$K = O\left(\frac{1}{\sqrt{q}} \log\left(\frac{F(x_0) - F^\star}{\sqrt{q}\varepsilon}\right)\right)$$

iterations to guarantee $\mathbb{E}[F(x_K) - F^\star] \leq \varepsilon$.

**Global complexity.** We use the exact same derivations as in Section B.3 except that we use the fact that $\varepsilon = O(\varepsilon_K/\sqrt{q}) = O((1 - \sqrt{q}/3)^K (F(x_0) - F^\star)/\sqrt{q})$ instead of $\varepsilon = O(\varepsilon_K/q)$, which gives us the desired complexity.

# D Methods $\mathcal{M}$ with Duality Gaps Based on Strongly-Convex Lower Bounds

In this section, we summarize a few results from [29] and introduce minor modifications to guarantee the condition (14). For solving a stochastic composite objectives such as (1), where $F$ is $\mu$-strongly convex, consider an algorithm $\mathcal{M}$ performing the following classical updates

$$z_t \leftarrow \mathrm{Prox}_{\eta\psi}\left[z_{k-1} - \eta g_t\right] \quad \text{with} \quad \mathbb{E}[g_t|\mathcal{F}_{k-1}] = \nabla f(z_{k-1}),$$

where $\eta \leq 1/L$, and the variance of $g_t$ is upper-bounded by $\sigma_t^2$. Inspired by estimate sequences from [41], the authors of [29] build recursively a $\mu$-strongly convex quadratic function $d_t$ of the form

$$d_t(z) = d_t^\star + \frac{\mu}{2}\|z_t - z\|^2.$$

From the proof of Proposition 1 in [29], we then have

$$\mathbb{E}[d_t^\star] \geq (1 - \eta\mu)\mathbb{E}[d_{k-1}^\star] + \eta\mu\mathbb{E}[F(z_t)] - \eta^2\mu\sigma_t^2,$$

which leads to

$$F^\star - \mathbb{E}[d_t^\star] + \eta\mu(\mathbb{E}[F(z_t)] - F^\star) \leq (1 - \eta\mu)\mathbb{E}[F^\star - d_{k-1}^\star] + \eta^2\mu\sigma_t^2,$$

which is a minor modification of Proposition 1 in [29] that is better suited to our purpose.

**With constant variance.** Assume now that $\sigma_t = \sigma$ for all $k \geq 1$. Following the iterate averaging procedure used in Theorem 1 of [29], which produces an iterate $\hat{z}_t$, we obtain

$$\mathbb{E}[F(\hat{z}_t) - d_t^\star] \leq (1 - \eta\mu)^t \left(F(z_0) - d_0^\star\right) + \eta\sigma^2, \tag{29}$$

where $d_0^\star$ can be freely specified for the analysis: it is not used by the algorithm, but it influences $d_t^\star$ through the relation $\mathbb{E}[d_t(z)] \leq \Gamma_t d_0(z) + (1 - \Gamma_t)\mathbb{E}[F(z)]$ with $\Gamma_t = (1 - \mu\eta)^k$, see Eq. (11) in [29]. In contrast, Theorem 1 in [29] would give here

$$\mathbb{E}[F(\hat{z}_t) - F^\star + d_t(z^\star) - d_t^\star] \leq (1 - \eta\mu)^t \left(2(F(z_0) - F^\star)\right) + \eta\sigma^2, \tag{30}$$

where $z^\star$ is a minimizer of $F$, which is sufficient to guarantee (2) given that $d_t(z^\star) \geq d_t^\star$.

**Application to the minimization of $H_k$.** Let us now consider applying the method to an auxiliary function $H_k$ from (14) instead of $F$, with initialization $x_{k-1}$. After running $T$ iterations, define $h_k$ to be the corresponding function $d_T$ defined above and $x_k = \hat{z}_T$. $H_k$ is $(\kappa+\mu)$-strongly convex and thus $h_k$ is also $(\kappa+\mu)$-strongly convex such that $(\mathcal{H}_1)$ is satisfied. Let us now check possible choices for $d_0^\star$ to ensure $(\mathcal{H}_2)$. For $z = \alpha_{k-1}x^\star + (1-\alpha_{k-1})x_{k-1}$, $\mathbb{E}[d_T(z)] \leq \Gamma_T d_0(z) + (1-\Gamma_T)H_k(z)$ such that we simply need to choose $d_0^\star$ such that $\mathbb{E}[d_0(z)] \leq \mathbb{E}[H_k(z)]$. Then, choose

$$d_0^\star = H_k^\star - F(x_{k-1}) + F^\star, \tag{31}$$

and

$$\begin{aligned} d_0(z) &= d_0^\star + \frac{\kappa+\mu}{2}\|x_{k-1} - z\|^2 \\ &= d_0^\star + \frac{(\kappa+\mu)\alpha_{k-1}^2}{2}\|x_{k-1} - x^\star\|^2 \\ &= d_0^\star + \frac{\mu}{2}\|x_{k-1} - x^\star\|^2 \\ &\leq d_0^\star + F(x_{k-1}) - F^\star = H_k^\star \leq H_k(z), \end{aligned}$$

such that $(\mathcal{H}_2)$ is satisfied, and finally (29) becomes

$$\mathbb{E}[H_k(x_k) - h_k^\star] \leq (1-\eta(\mu+\kappa))^T \left(H_k(x_{k-1}) - H_k^\star + F(x_{k-1}) - F^\star\right) + \eta\sigma^2,$$

which matches (14).

**Variance-reduction methods.** In [29], gradient estimators $g_t$ with variance reduction are studied, leading to variants of SAGA [14], MISO [36], and SVRG [53], which can deal with the stochastic finite-sum problem presented in Section 1. Then, the variance of $\sigma_t^2$ decreases (Proposition 2 in [29]).

Let us then consider again the guarantees of the method obtained when minimizing $F$ with $\frac{\mu}{L} \leq \frac{1}{5n}$. From Corollary 5 of [29], we have

$$\mathbb{E}[F(\hat{z}_t) - F^\star + d_t(z^\star) - d_t^\star] \leq 8\,(1-\mu\eta)^t\,(F(x_0) - F^\star) + 18\eta\sigma^2,$$

and (2) is satisfied. Consider now two cases at iteration $T$:

- if $\mathbb{E}[d_T(z^\star)] \geq F^\star$, then we have $\mathbb{E}[F(\hat{z}_T) - d_T^\star] \leq 8\,(1-\mu\eta)^T\,(F(x_0) - F^\star) + 18\eta\sigma^2$.
- otherwise, it is easy to modify Theorem 2 and Corollary 5 of [29] to obtain

$$\mathbb{E}[F(\hat{z}_T) - d_T^\star] \leq (1-\mu\eta)^T\,(2(F(x_0) - F^\star) + 6(F^\star - d_0^\star)) + 18\eta\sigma^2, \tag{32}$$

**Application to the minimization of $H_k$.** Consider now applying the method for minimizing $H_k$, with the same choice of $d_0^\star$ as (31), which ensures $(\mathcal{H}_2)$, and same definitions as above for $x_k$ and $h_k$. Note that the conditions on $\mu$ and $L$ above are satisfied when $\kappa = \frac{L}{5n} - \mu$ under the condition $\frac{L}{5n} \geq \mu$. Then, we have from the previous results, after replacing $F$ by $H_k$ making the right subsitutions

$$\mathbb{E}[H_k(x_k) - h_k^\star] \leq (1-(\mu+\kappa)\eta)^T\,(8(H_k(x_{k-1}) - H_k^\star) + 6(F(x_{k-1}) - F^\star)) + 18\eta\sigma^2,$$

and (14) is satisfied.

**Other schemes.** Whereas we have presented approaches were $d_t$ is quadratic, [29] also studies another class of algorithms where $d_t$ is composite (see Section 2.2 in [29]). The results we present in this paper can be extended to such cases, but for simplicity, we have focused on quadratic surrogates.

# E Additional Experimental Material

**Formulations.** Given training data $(a_i, b_i)_{i=1,\dots,n}$, with $a_i$ in $\mathbb{R}^p$ and $b_i$ in $\{-1, +1\}$, we consider the optimization problem

$$\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \phi(b_i a_i^\top x) + \frac{\mu}{2} \|x\|^2,$$

where $\phi$ is either the logistic loss $\phi(u) = \log(1 + e^{-u})$, or the squared hinge loss $\phi(u) = \frac{1}{2}\max(0, 1-u)^2$, which are both $L$-smooth, with $L = 0.25$ for logistic and $L = 1$ for the squared hinge loss. The regularization parameter $\mu$ acts as the strong convexity constant for the problem and is chosen among the smallest values one would try when performing parameter search, *e.g.*, by cross validation. Specifically, we consider $\mu = 1/10n$ and $\mu = 1/100n$, where $n$ is the number of training points; we also try $\mu = 1/1000n$ to evaluate the numerical stability of methods in very ill-conditioned problems. Following [7, 29, 54], we consider DropOut perturbations [48] with rate $\delta = 0$ (no noise), $\delta = 0.01$ and $\delta = 0.1$

**Datasets.** Then, we consider three datasets with various number of points $n$ and dimension $p$. The description comes from [29]:

- alpha is from the Pascal Large Scale Learning Challenge website[1] and contains $n = 250\,000$ points in dimension $p = 500$.
- gene consists of gene expression data and the binary labels $b_i$ characterize two different types of breast cancer. This is a small dataset with $n = 295$ and $p = 8\,141$.
- ckn-cifar is an image classification task where each image from the CIFAR-10 dataset[2] is represented by using a two-layer unsupervised convolutional neural network [37]. We consider here the binary classification task consisting of predicting the class 1 vs. other classes. The dataset contains $n = 50\,000$ images and the dimension of the representation is $p = 9\,216$.

All the data points are normalized to have unit $\ell_2$-norm.

**Methods.** We consider the variants of SVRG and SAGA of [29], which use decreasing step sizes when $\delta > 0$ (otherwise, they do not converge). We use the suffix "-d" each time decreasing step sizes are used. We also consider Katyuasha [1] when $\delta = 0$, and the accelerated SVRG method of [29], denoted by acc-SVRG. Then, SVRG-d, SAGA-d, acc-SVRG-d are used with the step size strategies described in [29], by using the code provided to us by the authors.

**Computing resources.** The numerical evaluation was performed by using four nodes of a CPU cluster with 56 cores of Intel CPUs each. The full set of experiments presented in this paper (with 5 runs for each setup) takes approximately half a day.

**Making plots.** We run each experiment five times and average the outputs. We display plots on a logarithmic scale for the primal gap $F(x_k) - F^\star$ (with $F^\star$ estimated as the minimum value observed from all runs). Note that for SVRG, one iteration is considered to perform two epochs since it requires accessing the full dataset every $n$ iterations on average.

## E.1 Additional experiments.

**Acceleration with no noise, $\delta = 0$.** We start evaluating the acceleration approach when there is no noise. This is essentially evaluating the original Catalyst method [34] in a deterministic setup in order to obtain a baseline comparison when $\delta = 0$. The results are presented in Figures 2 and 3 for the logistic regression problem. As predicted by theory, acceleration is more important when conditioning is low (bottom curves).

[1] `http://largescale.ml.tu-berlin.de/`

[2] `https://www.cs.toronto.edu/~kriz/cifar.html`

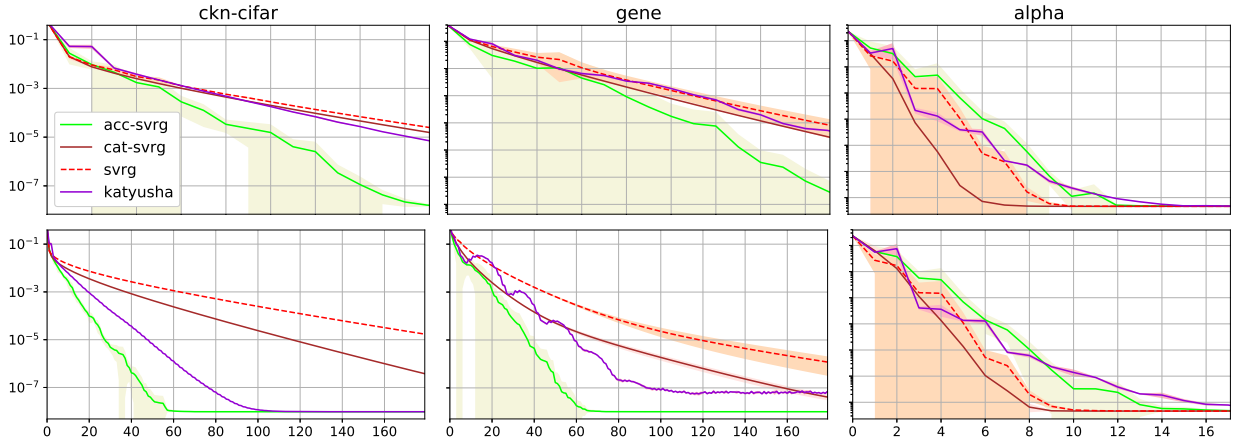


Figure 2: Accelerating SVRG-like methods for $\ell_2$-logistic regression with $\mu = 1/(10n)$ (top) and $\mu = 1/(100n)$ (bottom) for $\delta = 0$. All plots are on a logarithmic scale for the objective function value, and the $x$-axis denotes the number of epochs. The colored tubes around each curve denote a standard deviations across 5 runs. They do not look symmetric because of the logarithmic scale.

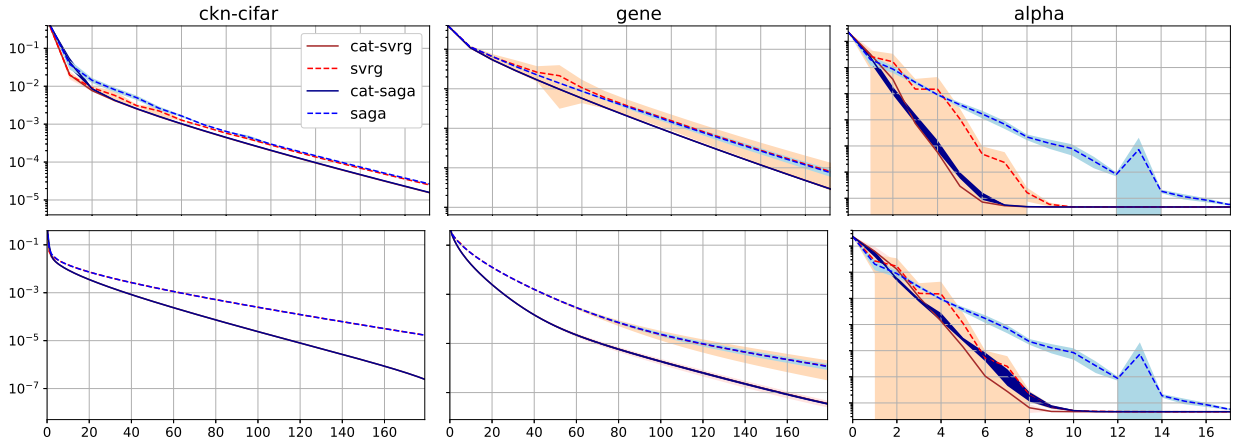


Figure 3: Same plots as in Figure 2 when comparing SVRG and SAGA, with no noise ($\delta = 0$) with $\mu = 1/(10n)$ (top) and $\mu = 1/(100n)$ (bottom) .

**Stochastic acceleration with no noise, $\delta = 0.01$ and $\delta = 0.1$.** Then, we perform a similar experiments by adding noise and report the results in Figures 4, 5, 6, 7. In general, the stochastic catalyst approach seems to perform on par with the accelerated SVRG approach of [29] and even better in one case.

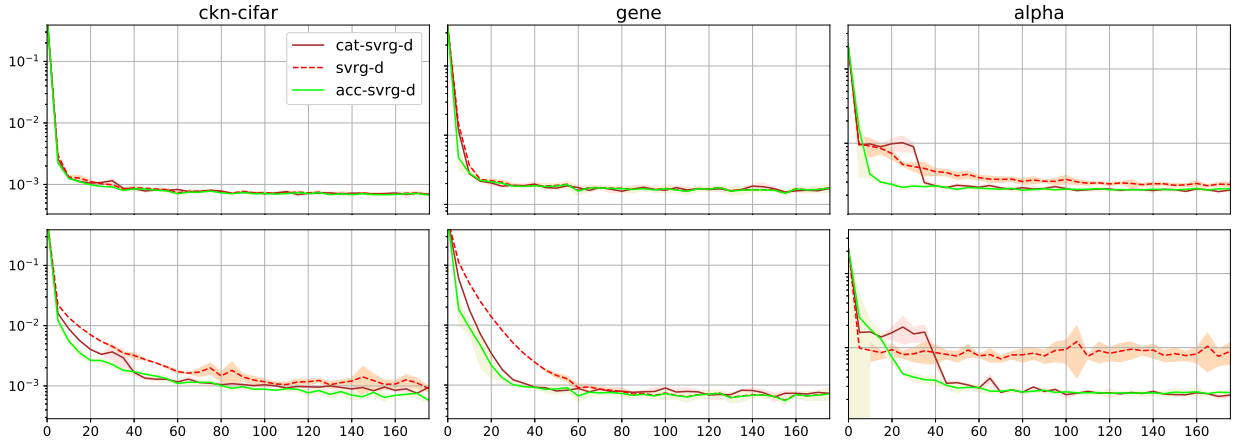


Figure 4: Same plots as in Figure 2 for $\delta = 0.01$ with $\mu = 1/(10n)$ (top) and $\mu = 1/(100n)$ (bottom).

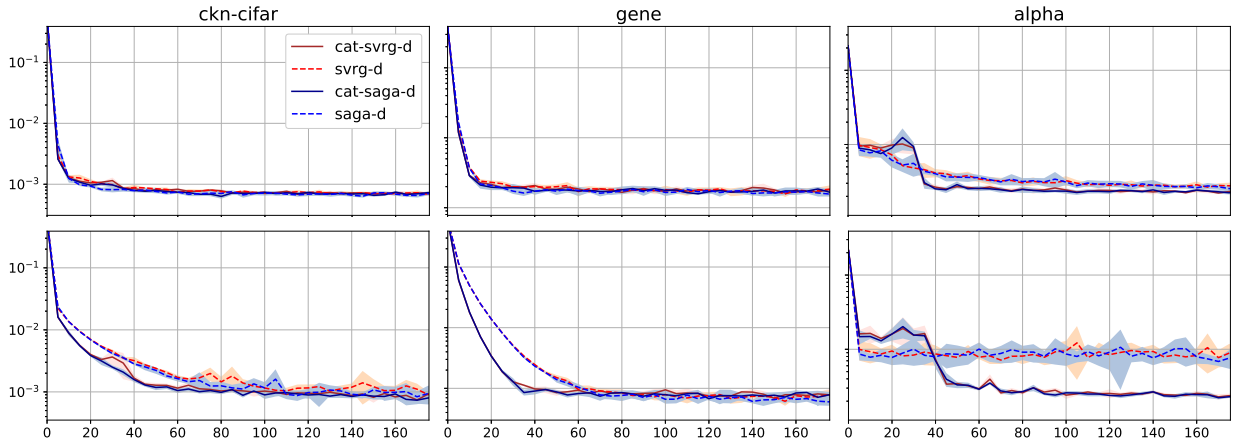


Figure 5: Same plots as in Figure 3 for $\delta = 0.01$ with $\mu = 1/(10n)$ (top) and $\mu = 1/(100n)$ (bottom).

**Evaluating the square hinge loss.** In Figure 8, we perform experiments using the square hinge loss, where the methods perform similarly as for the logistic regression case, despite the fact that the bounded noise assumption does not necessarily hold on the optimization domain for the square hinge loss.

**Evaluating ill-conditioned problems.** Finally, we study in Figure 10 how the methods behave when the problems are badly conditioned. There, acceleration seem to work on ckn-cifar, but fails on gene and alpha, suggestions that acceleration is difficult to achieve when the condition number is extremely low.

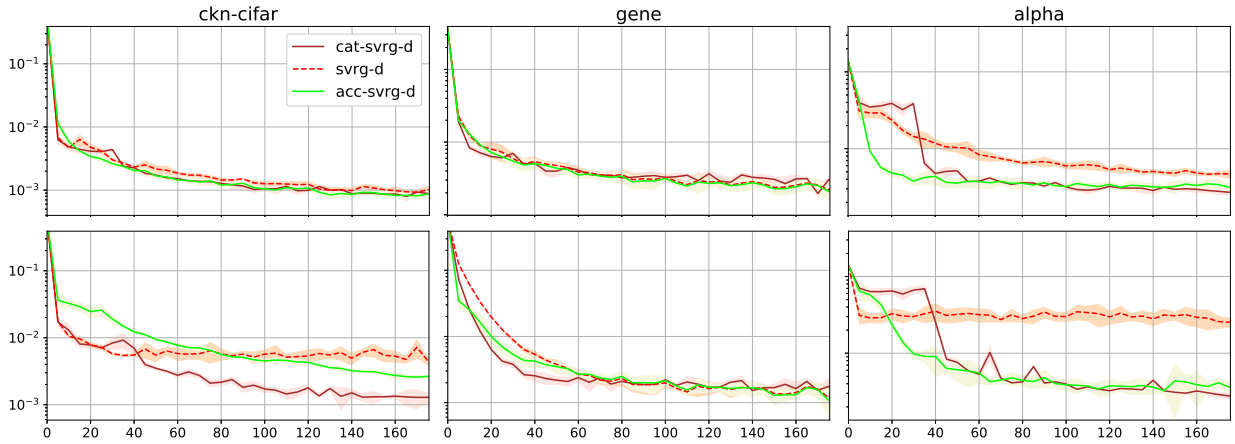


Figure 6: Same plots as in Figure 2 for $\delta = 0.1$ with $\mu = 1/(10n)$ (top) and $\mu = 1/(100n)$ (bottom).

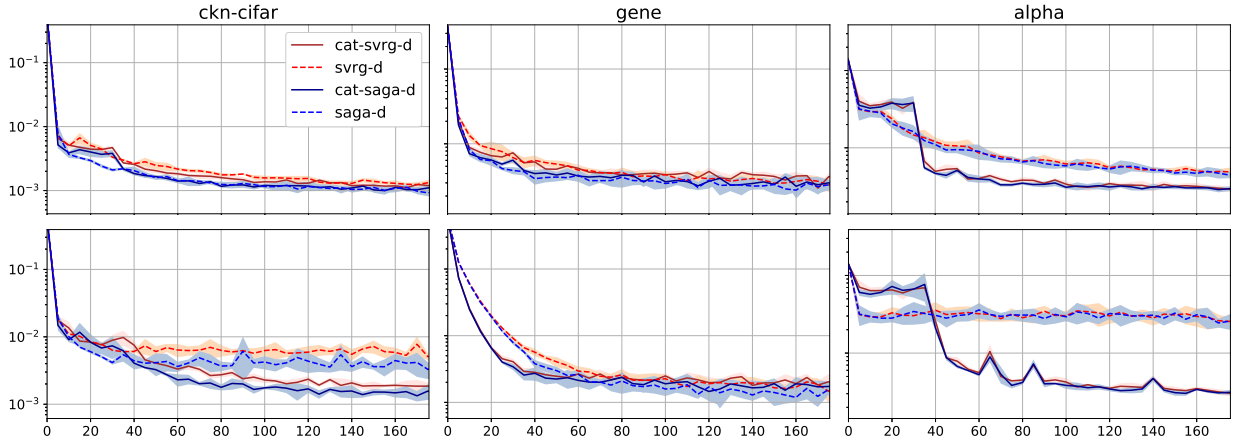


Figure 7: Same plots as in Figure 3 for $\delta = 0.1$ with $\mu = 1/(10n)$ (top) and $\mu = 1/(100n)$ (bottom).

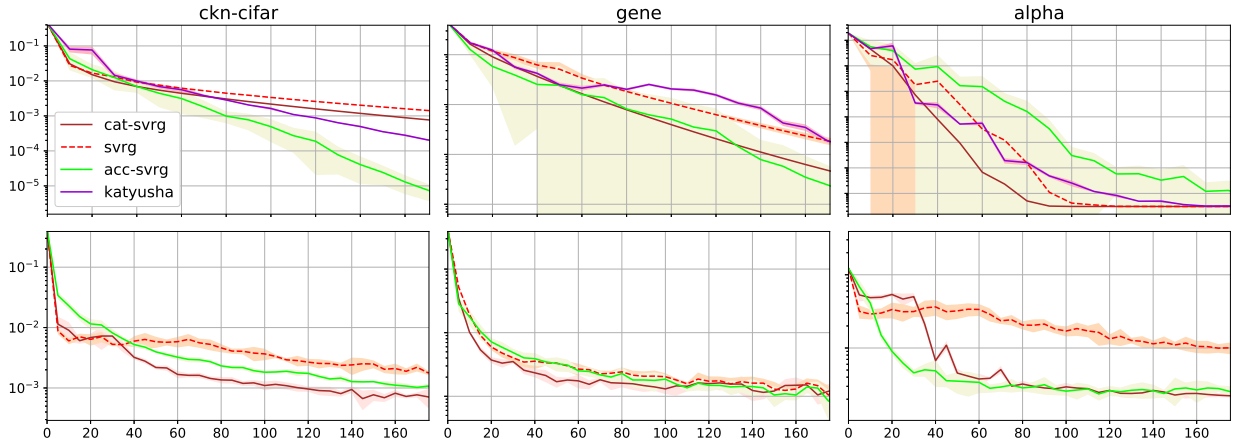


Figure 8: Accelerating SVRG-like methods when using the squared hinge loss instead of the logistic for $\delta = 0$ (top) and $\delta = 0.1$, both with $\mu = 1/(10n)$.

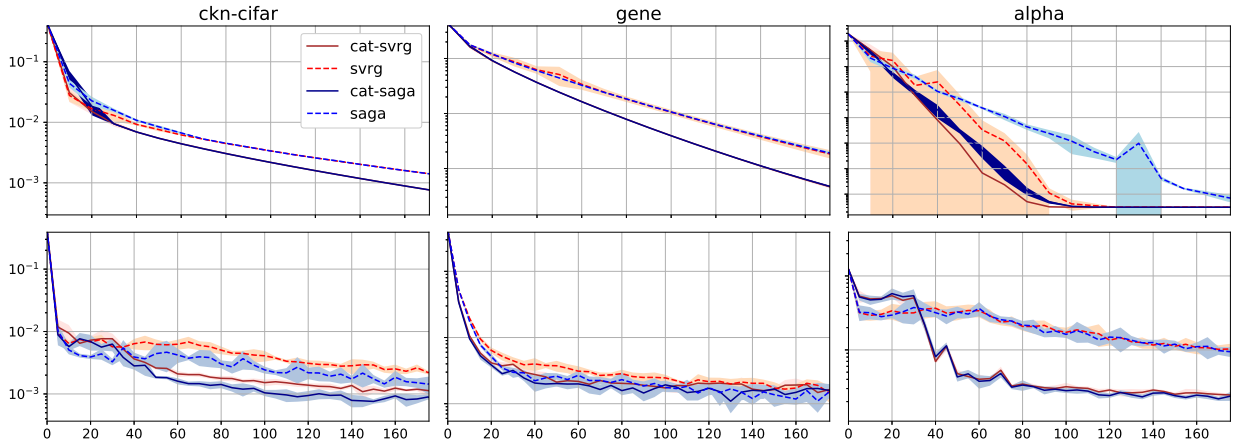


Figure 9: Same plots as in Figure 8 for SVRG and SAGA, with $\delta = 0$ (top) and $\delta = 0.1$ for $\mu = 1/(10n)$.

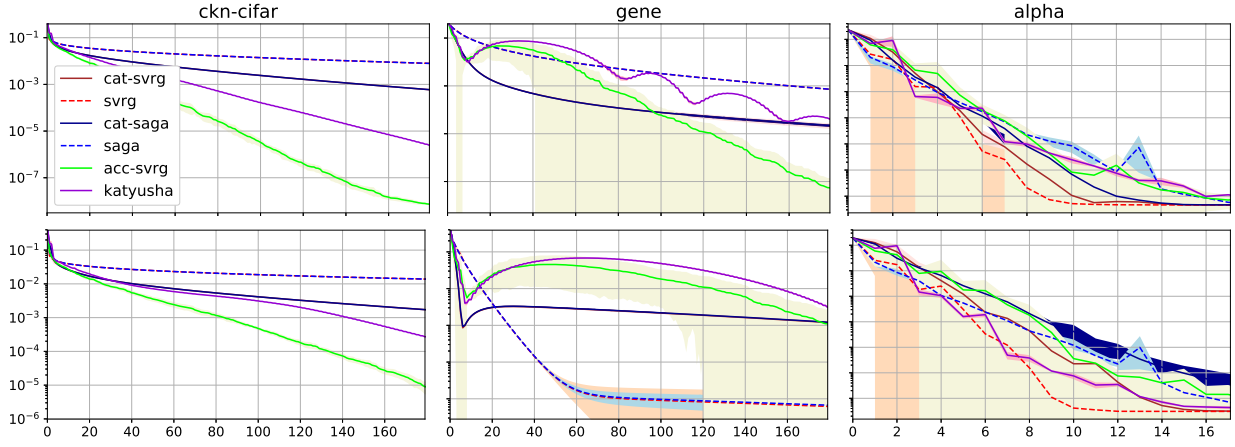


Figure 10: Illustration of potential numerical instabilities problems when the problem is very ill-conditioned. We use $\mu = 1/(1000n)$ with $\delta = 0$ for the logistic loss (top) and squared hinge (bottom).