A framework for bilevel optimization that enables stochastic and global variance reduction algorithms

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, Thomas Moreau

A framework for bilevel optimization that enables stochastic and global variance reduction algorithms

Mathieu Dagréou*†, Pierre Ablin‡, Samuel Vaiter§, Thomas Moreau†

February 8, 2022

Abstract

Bilevel optimization, the problem of minimizing a *value function* which involves the arg-minimum of another function, appears in many areas of machine learning. In a large scale setting where the number of samples is huge, it is crucial to develop stochastic methods, which only use a few samples at a time to progress. However, computing the gradient of the value function involves solving a linear system, which makes it difficult to derive unbiased stochastic estimates. To overcome this problem we introduce a novel framework, in which the solution of the inner problem, the solution of the linear system, and the main variable evolve at the same time. These directions are written as a sum, making it straightforward to derive unbiased estimates. The simplicity of our approach allows us to develop global variance reduction algorithms, where the dynamics of all variables is subject to variance reduction. We demonstrate that SABA, an adaptation of the celebrated SAGA algorithm in our framework, has $O(\frac{1}{T})$ convergence rate, and that it achieves linear convergence under Polyak-Łojasciewicz assumption. This is the first stochastic algorithm for bilevel optimization that verifies either of these properties. Numerical experiments validate the usefulness of our method.

1 Introduction

Bilevel optimization is attracting more and more attention in the machine learning community thanks to its wide range of applications. Typical examples are hyperparameters selection [Bengio, 2000, Pedregosa, 2016, Franceschi et al., 2018, Bertrand et al., 2020], data augmentation [Cubuk et al., 2019], implicit deep learning [Bai et al., 2019] or neural architecture search [Liu et al., 2018]. Bilevel optimization aims at minimizing a function whose value depends on the result of another optimization problem, that is:

$$\begin{aligned} &\min_{x\in\mathbb{R}^d} h(x) = F(z^*(x), x) \ , \\ &\text{such that } z^*(x) \in \arg\min_{z\in\mathbb{R}^p} G(z, x) \ , \end{aligned} \tag{1}$$

where F and G are two real valued functions defined on $\mathbb{R}^p \times \mathbb{R}^d$. G is called the *inner function*, F is the *outer function* and h is the *value function*. Similarly, z is the *inner variable* and x is the *outer variable*. In most cases, the function z^* can only be approximated by an optimization algorithm, which makes bilevel optimization problems challenging theoretically and practically. Under appropriate hypotheses, the function h is differentiable, and the chain rule and implicit function theorem give for any $x \in \mathbb{R}^d$

$$\nabla h(x) = \nabla_2 F(z^*(x), x) + \nabla^2_{21} G(z^*(x), x) v^*(x) \ , \tag{2}$$

*Corresponding author: mathieu.dagreou@inria.fr
†Université Paris-Saclay, Inria, CEA, Palaiseau, 91120, France
‡CNRS, Université Paris-Dauphine, PSL-University, Paris, France
§CNRS & Université Côte d'Azur, LJAD, Nice, France

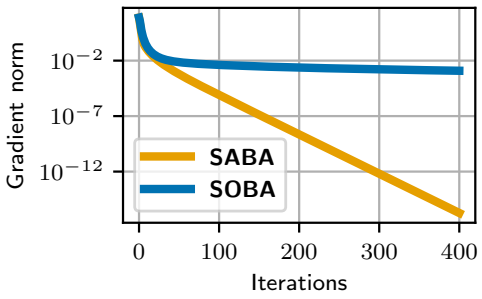


Figure 1: Convergence curves of the two proposed methods on a toy problem. SABA is a stochastic method that achieves fast convergence on the value function.

where $v^*(x) \in \mathbb{R}^p$ is the solution of a linear system

$$v^*(x) = -\left[\nabla^2_{11} G(z^*(x), x)\right]^{-1} \nabla_1 F(z^*(x), x) \ . \tag{3}$$

In the light of (2) and (3), it turns out that the derivation of the gradient of h at each iteration is cumbersome because it involves two subproblems: the resolution of the inner problem to find an approximation of $z^*(x)$ and the resolution of a linear system to find an approximation of $v^*(x)$. It makes the practical implementation of first order methods like gradient descent for (1) challenging.

As is the case in many machine learning problems, we suppose in this paper that F and G are empirical means:

$$F(z, x) = \frac{1}{m} \sum_{j=1}^{m} F_j(z, x), \quad G(z, x) = \frac{1}{n} \sum_{i=1}^{n} G_i(z, x) \ .$$

This structure suggests the use of stochastic methods to solve (1). For single-level problems (that is, classical optimization problems where one function should be minimized), using Stochastic Gradient Descent (SGD; Robbins and Monro 1951, Bottou 2010) and variants is natural because individual gradients are straightforward unbiased estimators of the gradient. In the bilevel framework, we want to develop algorithms that make progress on problem (1) by using only a few functions F_j and G_i at a time. However, since ∇h involves the inverse of the Hessian of G, building such stochastic algorithms is quite challenging, one of the difficulties being that there is no straightforward unbiased estimator of ∇h. Still, in settings where m or n are large, where computing even a single evaluation of F or G is extremely expensive, stochastic methods are the only scalable algorithms.

Variance reduction [Johnson and Zhang, 2013, Schmidt et al., 2017] is a popular technique to obtain fast stochastic algorithms. In a single-level setting, these methods build an approximation of the gradient of the objective function using only stochastic gradients. Contrary to SGD, the variance of the approximation goes to 0 as the algorithm progresses, allowing for faster convergence. For instance, the SAGA method [Defazio et al., 2014] achieves linear convergence if the objective function satisfies a Polyak-Łojasciewicz inequality, and $O(\frac{1}{T})$ convergence rate on smooth non-convex functions [Reddi et al., 2016]. The extension of these methods to bilevel optimization is a natural idea to develop faster algorithms. However, this idea is hard to implement for the same reasons as before: it is complicated to derive unbiased estimators of ∇h, let alone variance reduction ones.

Contributions. We introduce a **novel framework for bilevel optimization** in Section 2, where the inner variable, the solution of the linear system (3) and the outer variable evolve jointly. The evolution directions are written as sums of derivatives of F_j and G_i, which allows us to derive simple unbiased stochastic estimators. In this framework, we propose SOBA, a natural extension of SGD (Section 2.1), and SABA (Section 3.4), a natural extension of the variance reduction algorithm SAGA [Defazio et al., 2014].

In Section 3 we analyse the convergence of our methods. SOBA is shown to achieve $\inf_{t \leq T} \|\nabla h(x^t)\|^2 = O(T^{-\frac{2}{5}})$ with decreasing step sizes and the ratio between the inner and outer step sizes going to 0. We

prove that SABA with fixed step sizes achieves $\frac{1}{T}\sum_{t=1}^{T}\|\nabla h(x^t)\|^2 = O(\frac{1}{T})$. SABA is therefore, to the best of our knowledge, the first **stochastic bilevel algorithm that matches the convergence rate of gradient descent** on h. We also prove that SABA achieves **linear convergence** under the assumption that h satisfies a Polyak-Łojasciewicz inequality. To the best of our knowledge, SABA is also the first stochastic bilevel algorithm to feature such a property. Finally, in Section 4, we provide an **extensive benchmark** of many stochastic bilevel methods on hyperparameters selection and data hyper-cleaning, and illustrate the usefulness of our approach.

Related work. The bilevel optimization problem has a strong history in the optimization community, taking root in game theory [von Stackelberg, 1952]. Gradient-based algorithms to solve (1) can be mainly classified in two different categories depending on how the gradient of h is computed, by *automatic differentiation* or *implicit differentiation*

Since the solution of the inner problem $z^*(x)$ is approximated by the output of an iterative algorithm, it is possible to use automatic differentiation [Wengert, 1964, Linnainmaa, 1976] to approximate $\nabla h(x)$. It consists in differentiating the different steps of the inner optimization algorithm – see [Baydin et al., 2018] for a review – and has been applied successfully to several bilevel problems arising in machine learning [Domke, 2012, Franceschi et al., 2017]. One of the main drawbacks of this approach is that it requires to store in memory each iterate of the inner optimization algorithm, although this problem can sometimes be overcome using invertible optimization algorithms [Maclaurin et al., 2015] or truncated backpropagation [Shaban et al., 2019].

The use of the implicit function theorem to obtain (2) and (3) is known as implicit differentiation [Bengio, 2000]. While the cost of computing exactly (2) can be prohibitive for large scale problems, Pedregosa [2016] showed that we can still converge to a stationary point of the problem by using approximate solutions of the inner problem and linear system (3), if the approximation error goes to 0 sufficiently quickly. The complexity of approximate implicit differentiation has been studied by Grazzi et al. [2020]. Ramzi et al. [2021] propose to reuse the computations done in the forward pass to approximate the solution of the linear system (3) when the inner problem is solved thanks to a quasi-Newton method.

In the last few years, several works have proposed different strategies to solve (1) in a stochastic fashion. A first set of methods relies on *two nested loops*: one inner loop to solve the inner problem with a stochastic method, and one outer loop to update the outer variable with an approximate gradient direction. In [Ghadimi and Wang, 2018, Ji et al., 2021, Chen et al., 2021b] the authors use several SGD iterations for the inner problem and then use stochastic Neumann approximations to get an estimate solution of the linear system, which provides them with an approximation of ∇h used to update x. The convergence of the hypergradient when using stochastic solvers for the inner problem and the linear system has been studied by Grazzi et al. [2021]. Arbel and Mairal [2021] replace the Neumann approximation by SGD steps to estimate (3).

Other authors have proposed *single loop* algorithms, alternating steps in the inner and the outer problem. Hong et al. [2021] propose to perform a single Neumann approximation of the Hessian and use a single SGD step for the inner problem. It was refined by [Guo et al., 2021b] and [Yang et al., 2021] where the optimization procedure uses a momentum acceleration. Other variations around this idea include [Huang and Huang, 2021, Khanduri et al., 2021, Chen et al., 2021a, Guo et al., 2021a]. We refer to Table 1 in appendix for a detailed comparison of these methods.

Notation. The set of integers between 1 and n (included) is denoted $[n]$. For $f : \mathbb{R}^p \times \mathbb{R}^d \to \mathbb{R}$, we denote $\nabla_1 f(z, x)$ its gradient w.r.t. the first variable and $\nabla_2 f(z, x)$ its gradient w.r.t. the second variable. The Hessian of f with respect to the first variable is denoted $\nabla^2_{11} f(z, x) \in \mathbb{R}^{p\times p}$, and the cross-derivatives matrix is $\nabla^2_{21} f(z, x) \in \mathbb{R}^{d\times p}$. If v is a vector, $\|v\|$ is its Euclidean norm. If M is a matrix, $\|M\|$ is its spectral norm. A function is said to be L-smooth, for $L > 0$, if it is differentiable, and its gradient is L-Lipschitz.

2 Proposed framework

In this section, we introduce our framework in which the solution of the inner problem, the solution of the linear system (3) and the outer variable all evolve at the same time, following directions that are written as a sum of derivatives of F_j and G_i. We define

$$D_z(z,v,x) = \nabla_1 G(z,x) \ , \tag{4}$$

$$D_v(z,v,x) = \nabla^2_{11} G(z,x)v + \nabla_1 F(z,x) \ , \tag{5}$$

$$D_x(z,v,x) = \nabla^2_{21} G(z,x)v + \nabla_2 F(z,x) \ . \tag{6}$$

These directions are motivated by the fact that the gradient of the value function is

$$\nabla h(x) = D_x(z^*(x), v^*(x), x) \ , \tag{7}$$

with $z^*(x)$ the minimizer of $G(\cdot, x)$ and $v^*(x)$ the solution of $\nabla^2_{11} G(z^*(x),x)v = -\nabla_1 F(z^*(x),x)$. When x is fixed, we approximate z^* by doing a gradient descent on G, following the direction $-D_z(z,v,x)$. Finally, when z and x are fixed, we find v^* by following the direction $-D_v(z,v,x)$, which corresponds to a gradient descent on $v \mapsto \frac{1}{2}\langle \nabla^2_{11} G(z,x)v, v\rangle + \langle \nabla_1 F(z,x), v\rangle$.

The rest of the paper is devoted to the study of the global dynamics where the three variables z, v and x evolve at the same time, following stochastic approximations of the directions D_z, D_v and D_x. The next proposition motivates the choice of these directions.

Proposition 2.1. *Assume that for all $x \in \mathbb{R}^d$, $G(\cdot, x)$ is strongly convex. If (z,v,x) is a zero of (D_z, D_v, D_x), then $z = z^*(x)$, $v = v^*(x)$ and $\nabla h(x) = 0$.*

We also note that the computation of these directions does *not* require to compute the matrices $\nabla^2_{11} G(z,x)$ and $\nabla^2_{21} G(z,x)$: we only need to compute their product with a vector, which can be computed at a cost similar to that of computing a gradient.

The framework we propose is summarized in Algorithm 1. It consists in following a joint update rule in (z,v,x) that follows directions D^t_z, D^t_v and D^t_x that are unbiased estimators of D_z, D_v, D_x. The first and most important remark is that whereas ∇h cannot be written as a sum over samples, the directions D_z, D_v and D_x involve only simple sums, since their expressions are "linear" in F and G:

$$D_z = \frac{1}{n}\sum_{i=1}^n \nabla_1 G_i(z,x) \ , \tag{8}$$

$$D_v = \frac{1}{n}\sum_{i=1}^n \nabla^2_{11} G_i(z,x)v + \frac{1}{m}\sum_{j=1}^m \nabla_1 F_j(z,x) \ , \tag{9}$$

$$D_x = \frac{1}{n}\sum_{i=1}^n \nabla^2_{21} G_i(z,x)v + \frac{1}{m}\sum_{j=1}^m \nabla_2 F_j(z,x) \ . \tag{10}$$

It is therefore straightforward to derive unbiased estimators of these directions.

2.1 First example: the SOBA algorithm

The simplest unbiased estimator is obtained by replacing each mean by one of its terms chosen uniformly at random, akin to what is done in classical single-level SGD. We call the resulting algorithm SOBA (StOchastic Bilevel Algorithm). To do so, we choose two independent random indices $i \in [n]$ and $j \in [m]$ uniformly and estimate each term coming from G using G_i and each term coming from F using F_j. This gives the unbiased **SOBA directions**

$$D^t_z = \nabla_1 G_i(z^t, x^t) \ , \tag{11a}$$

$$D^t_v = \nabla^2_{11} G_i(z^t, x^t)v^t + \nabla_1 F_j(z^t, x^t) \ , \tag{11b}$$

$$D^t_x = \nabla^2_{21} G_i(z^t, x^t)v^t + \nabla_2 F_j(z^t, x^t) \ . \tag{11c}$$

Algorithm 1 General framework

Input: initializations $z_0 \in \mathbb{R}^p$, $x_0 \in \mathbb{R}^d$, $v_0 \in \mathbb{R}^p$, number of iterations T, step size sequences $(\rho^t)_{t<T}$ and $(\gamma^t)_{t<T}$.
for $t = 0, \dots, T-1$ **do**
Update z: $z^{t+1} = z^t - \rho^t D_z^t$,
Update v: $v^{t+1} = v^t - \rho^t D_v^t$,
Update x: $x^{t+1} = x^t - \gamma^t D_x^t$,
where D_z^t, D_v^t and D_x^t are unbiased estimators of $D_z(z^t, v^t, x^t), D_v(z^t, v^t, v^t)$ and $D_x(z^t, v^t, x^t)$.
end for

This provides us with a first algorithm, SOBA, where we plug Equations (11a) to (11c) in Algorithm 1. We defer its analysis to the next section. Importantly, we use different step sizes for the update in (z, v) and for the update in x. We use the same step size in z and in v since the inner problem and the linear system have similar conditioning, which is that of $\nabla^2_{11} G(z^t, x^t)$. The need for a different step size for the outer and inner problem is clear: both problems can have a drastically different conditioning.

An important remark for SOBA is that all the stochastic directions used are computed at the same point z^t, v^t and x^t with the same indices (i, j). The update of z, v and x can thus be performed in parallel instead of sequentially, benefiting from hardware parallelism. Moreover, this enables to share the computations between the different directions. This is the case in hyperparameters selection where $G_i(z, x) = \ell_i(\langle z, d_i \rangle) + \frac{x}{2}\|z\|^2$, with d_i a training sample, and ℓ_i that measures how good is the prediction $\langle z, d_i \rangle$. In this setting, we have $\nabla_1 G_i(z, x) = \ell_i'(\langle z, d_i \rangle) d_i + xz$ and $\nabla^2_{11} G_i(z, x) v = \ell_i''(\langle z, d_i \rangle)\langle v, d_i \rangle d_i$. The prediction $\langle z, d_i \rangle$ can thus be computed only once to obtain both quantities. For more complicated models, where automatic differentiation is used to compute the different derivatives and Jacobian-vector products, we can store the computational graph only once to compute at the same time $\nabla_1 G_i(z, x), \nabla^2_{11} G_i(z, x) v$ and $\nabla^2_{21} G_i(z, x) v$, requiring only one backward pass, thanks to the $\mathcal{R}$ technique [Pearlmutter, 1994].

Although not obvious at first glance, we find that using the same indices (i, j) to compute all directions at the same time poses no theoretical or practical issue, compared to a method which would take new random indexes at each line in Equations (11a) to (11c).

Finally, like all single loop bilevel algorithms, our method updates at the same time the inner and outer variable, hereby avoiding unnecessary optimization of the inner problem when x is far from the optimum.

2.2 Global variance reduction with the SABA algorithm

In classical optimization, SGD fails to reach optimal rates because of the variance of the gradient estimator. Variance reduction algorithms aim at reducing this variance, in order to follow directions that are closer to the true gradient, and to achieve superior practical and theoretical convergence.

In our framework, since the directions D_z, D_v and D_x are all written as sums of derivatives of F_j and G_i, it is straightforward to adapt most classical variance reduction algorithms. We focus on the celebrated SAGA algorithm [Defazio et al., 2014]. The extension we propose is called SABA (Stochastic Average Bilevel Algorithm). The general idea is to replace each sum in the directions D by a sum over a memory, updating only one term at each iteration. To help the exposition, we denote $y = (z, x, v)$ the vector of joint variables. Since we have sums over i and over j, we have two memories for each variable: w_i^t for $i \in [n]$ and $\tilde{w}_j^t$ for $j \in [m]$, which keep track of the previous values of the variable w.

At each iteration t, we draw two random independent indices $i \in [n]$ and $j \in [m]$ uniformly and update the memories. To do so, we put $w_i^{t+1} = y^t$ and $w_{i'}^{t+1} = w_{i'}^t$ for $i' \neq i$, and $\tilde{w}_j^{t+1} = y^t$ and $\tilde{w}_{j'}^{t+1} = \tilde{w}_{j'}^t$ for $j' \neq j$. Each sum in the directions D is then approximated using SAGA-like rules:

given n functions $\phi_{i'}$ for $i' \in [n]$, we define

$$S[\phi]^t = \phi_i(w_i^{t+1}) - \phi_i(w_i^t) + \frac{1}{n}\sum_{i'=1}^{n} \phi_{i'}(w_{i'}^t) \ ,$$

and similarly, given m functions $\widetilde{\phi}_{j'}$ for $j' \in [m]$:

$$\tilde{S}[\widetilde{\phi}]^t = \widetilde{\phi}_j(\tilde{w}_j^{t+1}) - \widetilde{\phi}_j(\tilde{w}_j^t) + \frac{1}{m}\sum_{j'=1}^{m} \widetilde{\phi}_{j'}(\tilde{w}_{j'}^t) \ .$$

These are unbiased estimators of the average of the ϕ's:

$$\mathbb{E}_i\Big[S[\phi]^t\Big] = \frac{1}{n}\sum_{i=1}^{n} \phi_i(y^t), \ \ \mathbb{E}_j\left[\tilde{S}[\widetilde{\phi}]^t\right] = \frac{1}{m}\sum_{j=1}^{n} \widetilde{\phi}_j(y^t) \ .$$

With a slight abuse of notation, we call $\nabla^2_{11}Gv$ the sequence of functions $(y \mapsto \nabla^2_{11}G_i(z,x)v)_{i\in[n]}$ and $\nabla^2_{21}Gv$ the sequence of functions $(y \mapsto \nabla^2_{21}G_i(z,x)v)_{i\in[n]}$. We define the **SABA directions** as

$$\boxed{\begin{aligned} D_z^t &= S[\nabla_1 G]^t \ , && \text{(12a)}\\ D_v^t &= S[\nabla^2_{11}Gv]^t + \tilde{S}[\nabla_1 F] \ , && \text{(12b)}\\ D_x^t &= S[\nabla^2_{21}Gv]^t + \tilde{S}[\nabla_2 F] \ . && \text{(12c)} \end{aligned}}$$

These estimators are unbiased estimators of the directions D_z, D_v and D_x. The SABA algorithm corresponds to Algorithm 1 where we use Equations (12a) to (12c) as update directions. When taking a step size $\gamma^t = 0$ in the outer problem, hereby stopping progress in x, we recover the iterations of the SAGA algorithm on the inner problem. In practice, the sums in S and $\tilde{S}$ are computed by doing a rolling average, and the quantities $\phi_i(w_i^t)$ are stored rather than recomputed: the cost of computing the SABA directions is the same as that of SGD. It requires an additional memory for the five quantities, of total size $n \times p + (n+m) \times (p+d)$ floats. This memory load can be reduced in specific cases, for instance when G and F correspond to linear models, where the individual gradients and Hessian-vector products are proportional to the samples. In this case, we only store the proportionality ratio, reducing the memory load to $3n + 2m$ floats. Like for SOBA, the computations of the new quantities $\phi_i(w_i^{t+1})$ are done in parallel, thus benefiting from hardware acceleration and shared computations.

In the next section, we show that SABA is fast. It essentially has the same properties as SAGA: despite being stochastic, it converges with fixed step sizes, and reaches the same rate of convergence as gradient descent on h.

3 Theoretical analysis

In this section, we provide convergence rates of SOBA and SABA under some classical assumptions. The proofs and the constants in big-O are deferred in Appendix C.

3.1 Background and assumptions

We start by stating some regularity assumptions on the functions F and G.

Assumption 3.1. The function F is L^F-smooth in (z, x).

Note that the above assumption is typically verified in the machine learning context, *e.g.,* when F is the ordinary least squares (OLS) loss or the logistic loss.

Assumption 3.2. The function G is twice continuously differentiable on $\mathbb{R}^p \times \mathbb{R}^d$. For any $x \in \mathbb{R}^d$, $G(\,\cdot\,, x)$ is μ_G-strongly convex and L_1^G-smooth. For any $z \in \mathbb{R}^p$, $\nabla_1 G(z, \,\cdot\,)$ is L_2^G-Lipschitz continuous. Finally, $\nabla_{11}^2 G$ is L_{11}^G-Lipschitz and $\nabla_{21}^2 G$ is L_{21}^G-Lipschitz.

Strong convexity and smoothness with respect to z of G are verified when G is a regularized least-squares/logistic regression with a full rank design matrix, when the data is not separable for the logistic regression. Moreover, the strong convexity ensures the existence and uniqueness of the inner optimization problem for any $x \in \mathbb{R}^d$.

Assumption 3.3. There exists a constant $C_F > 0$ such that for any x we have $\|\nabla_1 F(z^*(x), x)\| \leq C_F$.

This assumption, combined with the strong convexity of $G(\,\cdot\,, x)$, shows boundedness of v^*. This assumption holds, for instance, in the case of hyperparameters selection for a Ridge regression problem. Note that Assumptions 3.1 to 3.3 are classical in stochastic bilevel optimization literature: they are found for instance in [Ghadimi and Wang, 2018, Hong et al., 2021, Ji et al., 2021, Arbel and Mairal, 2021].

The following lemma gives us some smoothness properties of the considered directions that will be useful to derive convergence rates of our methods.

Lemma 3.4. *Under the Assumptions 3.1 to 3.3, there exist L_z, L_v and L_x such that $\|D_z(z, v, x)\|^2 \leq L_z^2 \|z - z^*(x)\|^2$, $\|D_v(z, v, x)\|^2 \leq L_v^2(\|z - z^*(x)\|^2 + \|v - v^*(x)\|^2)$ and $\|D_x(z, v, x) - \nabla h(x)\|^2 \leq L_x^2(\|z - z^*(x)\|^2 + \|v - v^*(x)\|^2)$.*

In first order optimization, a fundamental assumption on the objective function is the smoothness assumption. In the case of vanilla gradient descent applied to a function f, it allows to get a convergence rate of $\|\nabla f(x^t)\|^2$ in $O(1/T)$, i.e. convergence to a stationnary point Nesterov [2004]. The following lemma proved by Ghadimi and Wang [2018, Lemma 2.2] ensures the smoothness of h under the previous assumptions.

Lemma 3.5. *Under the Assumptions 3.1 to 3.3, the function h is L^h-smooth for some constant $L^h > 0$.*

The constant L^h is specified in Appendix C.3. As usual with the analysis of stochastic methods, we define the expected norms of the directions

$$V_z^t = \mathbb{E}[\|D_z^t\|^2], V_v^t = \mathbb{E}[\|D_v^t\|^2], V_x^t = \mathbb{E}[\|D_x^t\|^2], \tag{13}$$

where the expectation is taken over the past. Thanks to variance-bias decomposition, they are the sum of the variance of the stochastic direction and the squared-norm of the unbiased direction. We use classical bounds on these quantities, to those found for instance in [Hong et al., 2021]:

Assumption 3.6. There exists B_z, B_v and B_x such that for all t, $\quad V_z^t \leq B_z^2(1 + \|D_z(z^t, v^t, x^t)\|^2)$, $V_v^t \leq B_v^2(1 + \|D_v(z^t, v^t, x^t)\|^2)$ and $\quad V_x^t \leq B_x^2$.

This assumption is verified for instance, if all the G_i and $\nabla_1 G_i$ have at most quadratic growth, and if F has bounded gradients. It is also verified if the iterates remain in a compact set. Note that we do not assume that G has bounded gradients, as this would contradict its strong-convexity.

Finally, for the analysis of SABA, we need regularity on each G_i and F_j:

Assumption 3.7. For all $i \in [n]$ and $j \in [m]$, the functions ∇G_i, ∇F_j, $\nabla_{11}^2 G_i$ and $\nabla_{21}^2 G_i$ are Lipschitz continuous in (z, x).

3.2 Fundamental descent lemmas

Our analysis for SOBA and SABA is based on the control of the two key quantities

$$\delta_z^t = \mathbb{E}[\|z^t - z^*(x^t)\|^2] \text{ and } \delta_v^t = \mathbb{E}[\|v^t - v^*(x^t)\|^2], \tag{14}$$

where the expectation is taken over the past.

Strong convexity of G and Lipschitz continuity of $z^*(x)$ allow to obtain the following lemma, where we drop the dependency of the step size ρ in t for clarity.

Lemma 3.8. *Assume that* $\rho \leq \frac{2}{\mu_G}$. *We have:*

$$\delta_z^{t+1} \leq \left(1 - \frac{\rho\mu_G}{2}\right)\delta_z^t + 2\rho^2 V_z^t + 4\frac{L_*^2}{\mu_G}\frac{\gamma^2}{\rho}V_x^t$$
$$\delta_v^{t+1} \leq \left(1 - \frac{\rho\mu_G}{4}\right)\delta_v^t + \rho\beta_{vz}\delta_z^t + 2\rho^2 V_v^t + 8\frac{L_*^2}{\mu_G}\frac{\gamma^2}{\rho}V_x^t$$

where L_* *is the maximum between the Lipschitz constants of* z^* *and* v^* *(see Lemma C.2) and* $\beta_{vz} = \frac{1}{\mu_G^3}(L^F\mu_G + L_{11}^G)^2$.

We insist that this result is obtained in general for Algorithm 1 with arbitrary unbiased directions. We can therefore invoke this lemma for the analysis of both SOBA and SABA. A similar inequality in z can for instance be found in [Hong et al., 2021]. The assumption on ρ is only here to simplify the bounds. It is not a strong assumption, since it is for instance implied by the classical assumption for convergence of the inner problem with x fixed, $\rho \leq (L_1^G)^{-1}$. We use the smoothness of h to get the following lemma.

Lemma 3.9. *Let* $h^t = \mathbb{E}[h(x^t)]$ *and* $g^t = \mathbb{E}[\|\nabla h(x^t)\|^2]$. *We have*

$$h^{t+1} \leq h^t - \frac{\gamma}{2}g^t + \frac{\gamma}{2}L_x^2(\delta_z^t + \delta_x^t) + \frac{L^h}{2}\gamma^2 V_x^t \quad .$$

If δ_z and δ_v both cancel, we get an inequality reminiscent of the smoothness inequality for SGD on h.

3.3 Analysis of SOBA

The analysis of SOBA is based on Lemmas 3.5 and 3.8. We have the following theorem, with fixed step sizes depending on the number of iterations:

Theorem 1 (Convergence of SOBA, fixed step size)**.** Fix an iteration $T > 1$. We consider fixed steps $\rho^t = T^{-\frac{2}{5}}$ and $\gamma^t = T^{-\frac{3}{5}}$. We assume that T is large enough so that $\rho^t \leq \min(\frac{\mu_G}{8L_z^2B_z^2}, \frac{\mu_G}{16L_v^2B_v^2}, \frac{2}{\mu_G})$. Let x^t the sequence of outer iterates for SOBA. Then, $\frac{1}{T}\sum_{t=1}^T \mathbb{E}[\|\nabla h(x^t)\|^2] = O(T^{-\frac{2}{5}})$.

This theorem is proven by plugging Assumption 3.6 in Lemma 3.8 and Lemma 3.9, and then summing all inequalities. This rate is the same as in [Hong et al., 2021]. In term of complexity, to get an ϵ-stationary solution, we need to do $O(\epsilon^{-\frac{5}{2}})$ calls to the oracles. In comparison with [Hong et al., 2021], we have no $\log\epsilon^{-1}$ factor in the complexity. Hence, adding the v variable does not change convergence speed but improves complexity. We obtain a similar rate using decreasing step sizes by assuming that h is bounded:

Theorem 2 (Convergence of SOBA, decreasing step size)**.** We consider steps $\rho^t = t^{-\frac{2}{5}}$ and $\gamma^t = t^{-\frac{3}{5}}$. We assume that h^t is bounded. Let x^t the sequence of outer iterates for SOBA. Then, $\inf_{t\leq T}\mathbb{E}[\|\nabla h(x^t)\|^2] = O(T^{-\frac{2}{5}})$.

The main technical difficulty consists in multiplying inequality in Lemma 3.9 by $\frac{\rho^t}{\gamma^t}$ and using an Abel transform to control the h^t term. The boundedness hypothesis happens for instance in hyperparameters selection or if the iterates remain in a compact set.

However, these results are slightly disappointing because we must assume that the outer step size γ^t becomes infinitely small in front of ρ^t to obtain convergence. In fact, in the proof, we first demonstrate that z^t and v^t converge, using only the hypothesis that V_x^t is bounded, and never exploiting any other link between (z, v) and x. Hence, x is treated as noise in the convergence of z and v. On the other hand, the analysis of SABA leverages the dynamic of all three variable, resulting in fast convergence with fixed step sizes.

3.4 SABA: a stochastic method with optimal rates

The following theorem shows $O(\frac{1}{T})$ convergence for the SABA algorithm in the general case where we only assume smoothness of h. Our analysis of SABA is inspired by the analysis of single-level SAGA by Reddi et al. [2016].

Theorem 3 (Convergence of SABA, smooth case)**.** Let z^t, v^t and x^t the iterates of SABA, with fixed step sizes ρ and γ. We suppose $\rho \leq \rho_*$ and $\gamma \leq \min(\rho\xi_*, \frac{1}{8L^h})$, where ρ_* and ξ_* depend only on F and G and are specified in appendix. Then, $\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|\nabla h(x^t)\|^2] = O(\frac{1}{T})$.

To prove the theorem, the idea is to control the distance from the memory to the current variables. We define

$$S^t = \frac{1}{n}\sum_{i=1}^{n} \|y^t - w_i^t\|^2 + \frac{1}{m}\sum_{j=1}^{m} \|y^t - \tilde{w}_j^t\|^2 \ .$$

In appendix, we show that, under the hypotheses of the theorem,we can find scalars $\phi_s, \phi_z, \phi_v, \zeta > 0$ such that the quantity $\mathcal{L}^t = h^t + \phi_s S^t + \phi_z \delta_z^t + \phi_v \delta_v^t$ satisfies $\mathcal{L}^{t+1} \leq \mathcal{L}^t - \zeta g^t$. Summing these inequalities for $t = 0 \dots T-1$ and using the fact that $\mathcal{L}^t$ is lower bounded demonstrates the theorem.

In this setting, the function h is not necessarily convex, hence we recover the same convergence rate as gradient descent on the function h. To the best of our knowledge, our method is the first stochastic bilevel optimization method to achieve this (see Table 1 in appendix). This also shows that the complexity of our method to get an ϵ-stationary solution is $O(\epsilon^{-1})$.

We note that contrary to most stochastic methods, the proposed algorithm converges with fixed step size, as usual for variance reduction methods. Our theorem indicates that the step size for the inner problem and the linear system resolution, ρ, should be taken small enough, and then the step size of the outer problem γ should be taken as a small enough fraction of ρ, while also being smaller than $\frac{1}{8L^h}$, which is 8 times less than the classical $\frac{1}{L^h}$ step size used for gradient descent on h. The fact that variance reduction methods cannot take steps as large as gradient descent is well established in single-level setting [Defazio et al., 2014].

Furthermore, if we assume that h satisfies a Polyak-Łojasiewicz (PL) inequality, we recover linear convergence. Recall that h has the PL property if there exists $\mu_h > 0$ such that for all $x \in \mathbb{R}^d$, $\frac{1}{2}\|\nabla h(x)\|^2 \geq \mu_h(h(x) - h^*)$ with h^* the minimum of h.

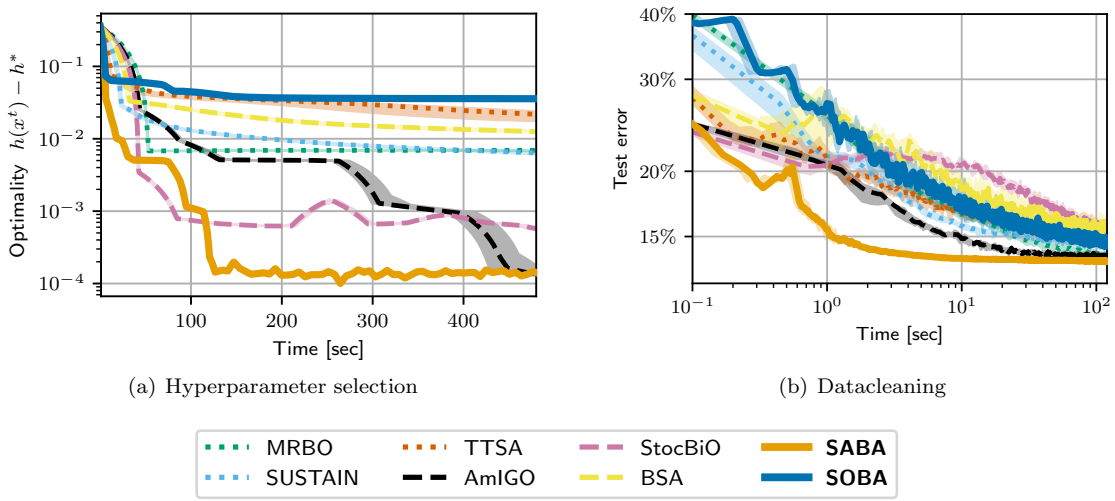
Theorem 4 (Convergence of SABA, PL case)**.** Assume that h satisfies the PL inequality. Let z^t, v^t and x^t the iterates of SABA, with fixed step sizes ρ and γ. We suppose $\rho \leq \rho'_*$ and $\gamma \leq \min(\rho\xi'_*, \frac{1}{8L^h})$, where ρ'_* and ξ'_* depend only on F and G and are specified in appendix. Then, $\mathbb{E}[h(x^T)] - h^* \leq (1 - \frac{\gamma\mu_h}{4})^T(h(x^0) - h^* + C^0)$, where C^0 is a constant specified in appendix that depends on the initialization of z, v, x and memory.

The proof is similar to that of the previous theorem: we find coefficients ϕ_s, ϕ_z, ϕ_v such that $\mathcal{L}^t = h^t + \phi_s S^t + \phi_z \delta_z^t + \phi_v \delta_v^t$ satisfies the inequality $\mathcal{L}^{t+1} \leq (1 - \frac{\gamma\mu_h}{4})\mathcal{L}^t$, which is then unrolled.

Note that in the case where we initialize z and v with $z^0 = z^*(x^0)$, $v^0 = v^*(x^0)$, and the memories $w_i^0 = w^0$, $\tilde{w}_j^0 = w^0$ for all i, j, the constant C^0 cancels and the bound simplifies to $\mathbb{E}[h(x^T)] - h^* \leq (1 - \frac{\gamma\mu_h}{4})^T(h(x^0) - h^*)$.

Just like classical variance reduction methods in single-level optimization, this theorem shows that our method achieves linear convergence under PL assumption on the value function. To the best of our knowledge, our method is the first stochastic bilevel optimization method that enjoys such property. We note that the PL hypothesis is more general than μ_h-strong convexity of h – it is a necessary condition for strong convexity.

We see here the importance of *global* variance reduction. Indeed, using variance reduction only on z and SGD on x would lead to sub-linear convergence in x (indeed, this would be the case even with a perfect estimation of $z^*(x)$). Similarly, using variance reduction only on x and SGD on z would lead to sub-linear convergence in z, and hence in x. Using global variance reduction with respect to each



(a) Hyperparameter selection

(b) Datacleaning

Figure 2: Comparison of SOBA and SABA with other stochastic bilevel optimization methods. For each algorithm, we plot the median performance over 10 runs. In both experiments, SABA achieves the best performance. The dashed lines are for one loop competitor methods, the dotted lines are for two loops methods and the solid lines are the proposed methods. **Left**: hyperparameter selection for ℓ^2 penalized logistic regression on IJCNN1 dataset , **Right**: data hyper-cleaning on MNIST with $p = 0.5$ corruption rate.

variable as we propose here is the only way to achieve linear convergence. We now turn to experiments, where we find that our method is also promising from a practical point of view.

4 Experiments

Here we compare the performances of SOBA and SABA with competitor methods on different tasks. The different methods being compared are stocBiO [Ji et al., 2021], AmiGO [Arbel and Mairal, 2021], MRBO [Yang et al., 2021], TTSA [Hong et al., 2021], BSA [Ghadimi and Wang, 2018] and SUSTAIN [Khanduri et al., 2021].

Each method has an inner and an outer step size. In each experiment, for each algorithm, we perform an extensive grid search to identify the pair of step-sizes that leads to the fastest convergence. Moreover, each step size has the decrease rate provided by theory: for instance, for SOBA, the step sizes are $\rho^t = \alpha t^{-\frac{2}{5}}$ and $\gamma^t = \beta t^{-\frac{3}{5}}$, where α and β are chosen with a grid search. For SABA we use $\rho^t = \alpha$ and $\gamma^t = \beta$ where α and β are chosen with a grid search. This procedure is repeated for each algorithm. We use a large grid search: for the hyperparameter experiment, we loop through 49 pairs of step-sizes and for the data cleaning experiment we loop through 121 pairs of step-sizes for each algorithm. Each experiment is repeated with 10 random seeds and the median runs are displayed as well as $20\% - 80\%$ quantiles.

We use Python code with Numba [Lam et al., 2015] for fast implementation of stochastic methods, and the Python package `benchopt`[1] to perform the benchmark[2]. A detailed account of the experiments is provided in Appendix B.

[1] https://benchopt.github.io/

[2] Code will be released upon acceptance of the paper

4.1 Hyperparameters selection

The first task we perform is hyperparameters selection to choose regularization parameters on ℓ^2 logistic regression. Let us denote $((d_i^{\text{train}}, y_i^{\text{train}}))_{1\leq i\leq n}$ and $((d_i^{\text{val}}, y_i^{\text{val}}))_{1\leq i\leq m}$ the training and the validation sets. In this case, the inner variable θ corresponds to the parameters of the model, and the outer variable λ to the regularization. The functions F and G of the problem (1) are the logistic loss, with ℓ^2 penalty for G, that is to say

$$F(\theta, \lambda) = \frac{1}{m}\sum_{i=1}^{m} \varphi(y_i^{\text{val}} \langle d_i^{\text{val}}, \theta\rangle) \text{ , and} \tag{15}$$

$$G(\theta, \lambda) = \frac{1}{n}\sum_{i=1}^{n} \varphi(y_i^{\text{train}} \langle d_i^{\text{train}}, \theta\rangle) + \frac{1}{2} R(\theta, \lambda) \tag{16}$$

where $R(\theta, \lambda) = \theta^\top \mathbf{diag}(e^{\lambda_1}, \dots, e^{\lambda_p})\theta$ and $\varphi(u) = \log(1 + e^{-u})$. Note that the parametrization in e^λ of the penalty instead of λ can be surprising at first glance, but it is classical in the bilevel optimization literature [Pedregosa, 2016, Ji et al., 2021, Grazzi et al., 2021] because it avoids positivity constraints on λ. We fit a binary classification model on the `IJCNN1`[3] dataset. For this dataset, $n = 49\ 990$, $m = 91\ 701$ and $p = 22$.

The suboptimality gap $h(\lambda^t) - h^*$ is plotted in Figure 2 for each method. The lowest values are reached by AmIGO and SABA, but much quicker for SABA. Moreover, SABA is the only single-loop method that reaches a suboptimality below 10^{-3}. Among all methods, SOBA is the first to reach it best value, but this value is also the highest. The gap between SOBA and SABA highlights the benefits of variance reduction: it gives us a lower plateau and the fixed step sizes enable faster convergence.

4.2 Data hyper-cleaning

The second task we perform is data hyper-cleaning introduced in [Franceschi et al., 2017] on the `MNIST`[4] dataset. The data is paritioned into a training set $(d_i^{\text{train}}, y_i^{\text{train}})$, a validation set $(d_i^{\text{val}}, y_i^{\text{val}})$, and a test set. The training set contains 20000 samples, the validation set 5000 samples and the test set 10000 samples. The targets y take values in $\{0, \dots, 9\}$ and the samples x are in dimension 784. Each sample in the training set is *corrupted* with probability p: a sample is corrupted when we replace its label y_i by a random label in $\{0, \dots, 9\}$. Samples in the validation and test sets are not corrupted. The goal of datacleaning is to train a multinomial logistic regression on the train set and learn a weight per training sample, that should go to 0 for corrupted samples. This is formalized by the bilevel optimization problem (1) with

$$F(\theta, \lambda) = \frac{1}{m}\sum_{i=1}^{m} \ell(\theta d_i^{\text{val}}, y_i^{\text{val}}) \text{ , and}$$

$$G(\theta, \lambda) = \frac{1}{n}\sum_{i=1}^{n} \sigma(\lambda_i)\ell(\theta d_i^{\text{train}}, y_i^{\text{train}}) + C_r\|\theta\|^2$$

where ℓ is the cross entropy loss and σ is the sigmoid function. The inner variable θ is a matrix of size 10×784, and the outer variable λ is a vector in dimension $n_{\text{train}} = 20000$.

For the estimated parameters θ during optimization, we report in Figure 2 the test error, *i.e.*, the percent of wrong predictions on the testing data. We use for this experiment a corruption probability $p = 0.5$. In general, the error decreases quickly until it reaches a final value. We observe that our method SABA outperforms all the other methods by reaching faster its smallest error, which is smaller than the ones of the other methods. For SOBA, it reaches a lower final error than stocBiO and BSA. In appendix, we provide other convergence curves, and find that for higher values of p, SABA is still

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
[4] http://yann.lecun.com/exdb/mnist/

the fastest algorithm to reach its final accuracy, but the algorithm MRBO eventually reaches a lower test accuracy.

5 Conclusion

In this paper, we have presented a framework for bilevel optimization that enables the straightforward development of stochastic algorithms. The gist of our framework is that the directions in Equations (4) to (6) are all written as simple sums of samples derivatives. We leveraged this fact to propose SOBA, an extension of SGD to our framework, and SABA, an extension of SAGA to our framework, which achieves similar convergence rates as gradient descent on the value function. Finally, we think that our framework opens a large panel of potential methods for stochastic bilevel optimization involving techniques of extrapolation, variance reduction, momentum and so on.

Acknowledgements

We thank Othmane Sebbouh, Zaccharie Ramzi and Benoit Malézieux for their precious comments. The authors acknowledge the support of the ANER RAGA BFC. SV acknowledges the support of the ANR GraVa ANR-18-CE40-0005. This research was supported by DATAIA convergence institute as part of the " Programme d'Investissement d'Avenir ", (ANR-17-CONV-0003) operated by Inria.

References

Zeeshan Akhtar, Amrit Singh Bedi, Srujan Teja Thomdapu, and Ketan Rajawat. Projection-Free Algorithm for Stochastic Bi-level Optimization. *preprint ArXiv 2110.11721*, 2021.

Michael Arbel and Julien Mairal. Amortized Implicit Differentiation for Stochastic Bilevel Optimization. *preprint ArXiv 2111.14580*, 2021.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep Equilibrium Models. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2019.

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in Machine Learning: A survey. *Journal of Machine Learning Research*, 18 (153):1–43, 2018.

Yoshua Bengio. Gradient-Based Optimization of Hyperparameters. *Neural Computation*, 12(8): 1889–1900, 2000.

Quentin Bertrand, Quentin Klopfenstein, Mathieu Blondel, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon. Implicit differentiation of lasso-type models for hyperparameter optimization. In *International Conference on Machine Learning (ICML)*, pages 810–821. PMLR, 2020.

Léon Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In *Proceedings of COMPSTAT*, pages 177–186. Physica-Verlag HD, Heidelberg, 2010.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. A Single-Timescale Stochastic Bilevel Optimization Method. *preprint ArXiv 2102.04671*, 2021a.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the Gap: Tighter Analysis of Alternating Stochastic Gradient Methods for Bilevel Problems. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2021b.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies From Data. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123. IEEE, 2019.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2019.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 1646–1654, Montreal, QC, Canada, December 2014. Curran Associates, Inc.

Justin Domke. Generic methods for optimization-based modeling. In *International Conference on Artificial Intelligence and Statistics (AISTAT)*, volume 22, pages 318–326. PMLR, 2012.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning (ICML)*, pages 1165–1173. PMLR, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning (ICML)*, pages 1568–1577. PMLR, 2018.

Saeed Ghadimi and Mengdi Wang. Approximation Methods for Bilevel Programming. *preprint ArXiv 1802.02246*, 2018.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning (ICML)*, pages 3748–3758. PMLR, 2020.

Riccardo Grazzi, Massimiliano Pontil, and Saverio Salzo. Convergence properties of stochastic hypergradients. In *International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 3826–3834. PMLR, 2021.

Zhishuai Guo, Quanqi Hu, Lijun Zhang, and Tianbao Yang. Randomized Stochastic Variance-Reduced Methods for Multi-Task Stochastic Bilevel Optimization. *preprint ArXiv 2105.02266*, 2021a.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. On Stochastic Moving-Average Estimators for Non-Convex Optimization. *preprint ArXiv 2104.14840*, 2021b.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A Two-Timescale Framework for Bilevel Optimization: Complexity Analysis and Application to Actor-Critic. *preprint ArXiv 2007.05170*, 2021.

Feihu Huang and Heng Huang. BiAdam: Fast Adaptive Bilevel Optimization Methods. *preprint ArXiv 2106.11396*, 2021.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning (ICML)*, pages 4882–4892. PMLR, 2021.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. pages 315–323. Curran Associates, Inc., 2013.

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A Near-Optimal Algorithm for Stochastic Bilevel Optimization via Double-Momentum. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2021.

Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976.

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

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations (ICLR)*, 2018.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on machine learning (ICML)*, pages 2113–2122. PMLR, 2015.

IU E. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Number v. 87 in Applied Optimization. Kluwer Academic Publishers, Boston, 2004.

Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning (ICML)*, pages 737–746. PMLR, 2016.

Zaccharie Ramzi, Florian Mannel, Shaojie Bai, Jean-Luc Starck, Philippe Ciuciu, and Thomas Moreau. SHINE: SHaring the INverse Estimate from the forward pass for bi-level optimization and implicit models. *preprint ArXiv 2106.00553*, 2021.

Sashank J. Reddi, Suvrit Sra, Barnabas Poczos, and Alex Smola. Fast Incremental Method for Nonconvex Optimization. *preprint ArXiv 1603.06159*, 2016.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated Back-propagation for Bilevel Optimization. In *Artificial Intelligence and Statistics (AISTAT)*, pages 1723–1732, Okinawa, Japan, 2019.

Heinrich von Stackelberg. *Theory of the market economy*. Oxford University Press, 1952.

R. Wengert. A simple automatic derivative evaluation program. *Communications of the ACM*, 7(8): 463–464, 1964.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably Faster Algorithms for Bilevel Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2021.

A Extensive comparison between stochastic methods for bilevel optimization

We provide here tables summarizing other methods in stochastic bilevel optimization. They are grouped between methods that are based on two nested loops and methods that use only one loop.

In the following tables, the inner iterations are referred with the variable k and the outer iterations are referred with the variable t (or T for the total number of iterations).

In the litterature, there are three main ways to perform Hessian inversion. The HIA, first proposed in [Ghadimi and Wang, 2018], and SHIA, proposed in [Ji et al., 2021], procedures used for Hessian inversion are precised in Algorithm 2 and 3. These methods are based on Neumann approximation of the inverse of a matrix. SGD for Hessian inversion refers to Stochastic Gradient Descent on $v \mapsto \frac{1}{2}\langle \nabla^2_{11} G(z,x)v, v\rangle - \langle \nabla_1 F(z,x), v\rangle$. The complexity refers to the number of call to the oracles to get an ϵ-stationary solution. In these complexities, the notation $\tilde{O}$ hide polynomial factors in $\log \epsilon^{-1}$.

Algorithm 2 Hessian Inverse Approximation (HIA)

Input: variables $z \in \mathbb{R}^p$, $x \in \mathbb{R}^d$, gradient $\nabla_1 F(z,x) \in \mathbb{R}^p$, maximum number of iterations b, a parameter η.
Set $v^0 = \nabla_1 F(z,x)$
Choose $p \in \{0, \dots, b-1\}$ randomly.
for $k = 1, \dots, p$ **do**
  Sample $i \in [n]$
  Update v : $v^{k+1} = (I - \eta \nabla^2_{11} G(z,x))v^k$
end for
Return: $b\eta v_{p+1}$

Algorithm 3 Summed Hessian Inverse Approximation (SHIA)

Input: variables $z \in \mathbb{R}^p$, $x \in \mathbb{R}^d$, gradient $\nabla_1 F(z,x) \in \mathbb{R}^p$, maximum number of iterations b, a parameter η.
Set $v^0 = \nabla_1 F(z,x)$
Set $s^0 = v^0$
for $k = 0 \dots, b-1$ **do**
  Sample $i \in [n]$
  Update v: $v^{k+1} = (I - \eta \nabla^2_{11} G(z,x))v^k$
  Update s: $s^{k+1} = s^k + v^{k+1}$
end for
Return: ηs^b

The momentum column refers to the use of STORM [Cutkosky and Orabona, 2019] momentum in the inner loop or the outer loop. This momentum can be applied to either the inner or the implicit gradient estimate. If we consider the current estimate $y^t = (z^t, v^t, x^t)$ and the previous estimate $y^{t-1} = (z^{t-1}, v^{t-1}, x^{t-1})$, and we apply STORM to the quantity $\phi(y^t)$ with the memory $\hat{\phi}^t$, the momentum update rule reads

$$\hat{\phi}^{(t+1)} = \eta \phi(y^t) + (1-\eta)(\hat{\phi}^t + \phi(y^t) - \phi(y^{t-1})) \ .$$

Note that this update requires to evaluate the quantity ϕ twice per iteration, once in y^t and once in y^{t-1}. The memory is need to store the previous estimates y^{t-1} as well as the running estimate of the gradient $\hat{\phi}$.

Method (Two-loops)	Hessian inversion	Inner loop	Momentum	LR in	LR out	Complexity
BSA [Ghadimi and Wang, 2018]	HIA	SGD on inner	No	$O(k^{-1})$	$O(T^{-1/2})$	$O(\epsilon^{-3})$
stocBiO [Ji et al., 2021]	SHIA	SGD on inner	No	Constant	Constant	$\tilde{O}(\epsilon^{-2})$
VRBO [Yang et al., 2021]	SHIA	Gradient steps inner/outer with STORM	Yes (STORM)	Constant	Constant	$\tilde{O}(\epsilon^{-3/2})$
AmIGO [Arbel and Mairal, 2021]	SGD	SGD on inner	No	Constant	Constant	$O(\epsilon^{-2})$
Method (One-loop)	Hessian inversion	Inner step	Momentum	LR in	LR out	Complexity
TTSA [Hong et al., 2021]	HIA	SGD	No	$O(T^{-2/5})$	$O(T^{-3/5})$	$\tilde{O}(\epsilon^{-5/2})$
SMB [Guo et al., 2021b]	HIA	SGD with momentum	Yes	Constant	Constant	$\tilde{O}(\epsilon^{-4})$
MRBO [Yang et al., 2021]	SHIA	SGD with STORM	Yes (STORM)	$O(t^{-1/3})$	$O(t^{-1/3})$	$\tilde{O}(\epsilon^{-3/2})$
STABLE [Chen et al., 2021a]	Direct	SGD	No	$O(T^{-1/2})$	$O(T^{-1/2})$	$O(\epsilon^{-2})$
SUSTAIN [Khanduri et al., 2021]	HIA	SGD with STORM	Yes (STORM)	$O(t^{-1/3})$	$O(t^{-1/3})$	$O(\epsilon^{-3/2})$
SVRB [Guo et al., 2021a]	Direct + momentum	SGD with momentum	Yes	$O(t^{-1/3})$	$O(t^{-1/3})$	$\tilde{O}(\epsilon^{-3})$
SBFW [Akhtar et al., 2021]	HIA	SGD	No	$O(t^{-1/2})$	$O(T^{-3/4})$	$\tilde{O}(\epsilon^{-4})$
SOBA	SGD step	SGD	No	$O(t^{-2/5})$	$O(t^{-3/5})$	$O(\epsilon^{-5/2})$
SABA	SAGA step	SAGA	No	Constant	Constant	$\boldsymbol{O(\epsilon^{-1})}$

Table 1: Comparison of the stochastic bilevel optimization solvers in the literature. The complexity represents the number of oracle calls necessary to attain an ϵ accurate stationary point.

B Details on experiments

We provide here additional informations on the experiments.

B.1 Generalities

All the experiments are performed with Python code, using the package `Benchopt`. The API we developed consists in having the inner and the outer functions wrapped in python classes whose methods compute the different oracles. One important thing is that we incorporated a method `oracles` which, for a given function f defined on $\mathbb{R}^p \times \mathbb{R}^d$ and a vector $v \in \mathbb{R}^p$, returns the tuple $(f(z,x), \nabla_1 f(z,x), \nabla^2_{11} f(z,x)v, \nabla^2_{21} f(z,x)v)$ avoiding duplicate computation of intermediate results for these quantities.

We find that using mini-batches instead of individual samples to compute the stochastic estimates allowed for much faster computations, thanks to hardware acceleration and vectorization of the computations. We use continuous batches to avoid random memory access that slow down the computations. Concretely, if i_b is the index of the current batch and B is the batch-size, the indices of the corresponding samples are those in the set $\{i_b \times B, \dots, (i_b + 1) \times B - 1\}$. By doing so, the samples in a same batch are contiguous in memory, which facilitates the access. We use a batch-size of 64 in all experiments.

For the methods involving an inner loop (stocBiO, BSA, AmIGO), we perform 10 inner steps at each outer iteration. For the approximate Hessian vector product, we perform 10 steps per outer iteration for each methods using HIA (BSA, TTSA, SUSTAIN), SHIA (MRBO, stocBiO) or SGD (AmIGO) for the inversion of the linear system.

For the step sizes, they all have the form $\rho^t = \alpha/t^a$ and $\gamma^t = \beta/t^b$. For the pair of exponents (a, b), we choose the theoretical one from the original papers, that is $(1/2, 1/2)$ for BSA, $(1/3, 1/3)$ for MRBO and SUSTAIN, $(0, 0)$ for SABA, AmIGO and stocBiO, $(2/5, 3/5)$ for TTSA and SOBA. For (α, β), we perform a grid search (the grid is precised in the subsection dedicated to each experiment) and we keep for each method, the pair (α, β) that gives the lowest value of h (for the hyperparameters) or the lowest test accuracy (for the data cleaning task) in median over 10 runs for each possible pair. When we use HIA or SHIA for the Hessian inversion, we set $\eta = \alpha$ since the Hessian inversion problem has the same conditioning as the inner optimization problem.

For the STORM's momentum parameter in MRBO and SUSTAIN, we take $0.5/t^{2/3}$.

B.2 Hyperparameters selection on IJCNN1

In this experiment, we select the parameters regularization for a multiregularized logistic regression model precised in Equations (15) and (16) where we have one hyperparameter per feature.

In order to choose the select proper parameters (α, β) for each algorithm, we perform a grid search. We search α in a set of 9 values between 2^{-5} and 2^3 spaced on a log scale. For β, we choose r in a set of 7 values between 2^{-2} and 2 spaced on a logarithmic scale and we set $\beta = r\alpha$.

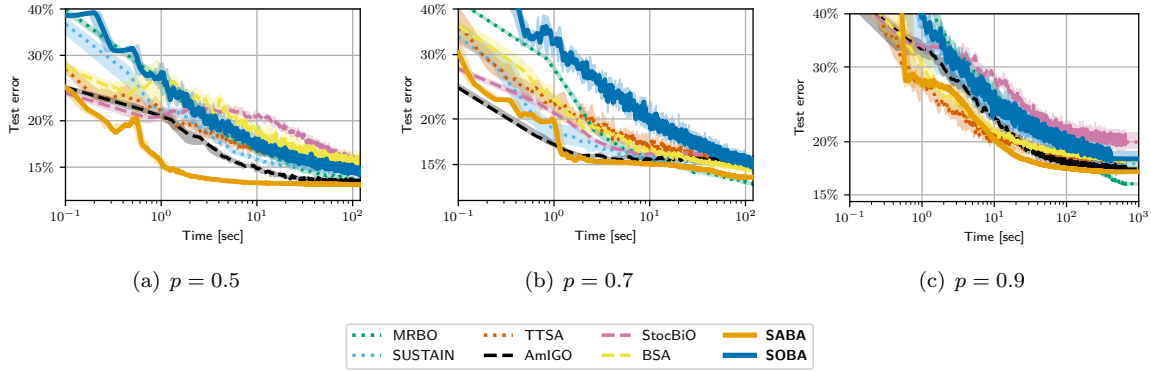
For this experiments, we use Just-In-Time (JIT) compilation thanks to the package Numba [Lam et al., 2015], to decrease the python overhead in the iteration loop.

To evaluate the value function h, we use L-BFGS [Liu and Nocedal, 1989] to solve compute $z^*(x^t)$ and then evaluate the function $h(x^t) = F(z^*(x^t), x^t)$.

B.3 Data hyper-cleaning

For the regularization parameter C_r, we choose $C_r = 0.1$ after a manual search in order to get the best final test accuracy.

In this experiment, the selection of the good pair (α, β) is also performed by grid search. The parameter α is picked in a set of 11 numbers between 2^{-4} and 4 spaced on a logarithmic scale. For β, we choose r in a set of 11 values between 2^{-5} and 2^0 spaced on a logarithmic scale and we set $\beta = r\alpha$.



(a) $p = 0.5$ (b) $p = 0.7$ (c) $p = 0.9$

Figure 3: Datacleaning experiment, with different corruption probability (higher means that more data are contamined).

Note that in this case, we could not use JIT from Numba since at the moment of the experiment, the softmax function coming from Scipy was not compatible with Numba.

We report in Figure 3 some additional convergence curves with different corruption probabilities $p \in \{0.5, 0.7, 0.9\}$ (the figure in the main text corresponds to $p = 0.5$). SABA is always the fastest algorithm to reach its final accuracy. For $p = 0.9$ and $p = 0.7$, the algorithm MRBO reaches an accuracy roughly 1% smaller than that of SABA, but takes roughly 10 times longer to get there.

C Proofs

C.1 Proof of Proposition 2.1

Proof. Let (z, v, x) a zero of (D_z, D_v, D_x). For D_z, this means that $\nabla_1 G(z, x) = 0$. Since $G(\,\cdot\,, x)$ is strongly convex, z is the minimizer of $G(\,\cdot\,, x)$, *i.e.* $z = z^*(x)$. The fact that (z, v, x) is a zero of D_v implies that $\nabla^2_{11} G(z, x)v = -\nabla_1 F(z, x)$. Replacing z by its value, we get $v = -\left[\nabla^2_{11} G(z^*(x), x)\right]^{-1} \nabla_1 F(z^*(x), x)$ which is $v^*(x)$ by definition. Putting all together and using the expression of $\nabla h(x)$ given by (2), we get

$$D_x(z, v, x) = \nabla_2 F(z^*(x), x) + \nabla_{21} G(z^*(x), x)v^*(x) = \nabla h(x) \ .$$

On the other hand, $D_x(z, v, x) = 0$ so $\nabla h(x) = 0$. □

C.2 Proof of Lemma 3.4

Proof. Let $(z, v, x) \in \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^d$. Using the fact that $\nabla_1 G(z^*(x), x) = 0$ and the L_1^G-smoothness of $G(\,\cdot\,, x)$, we have

$$\boxed{\|D_z(z, v, x)\|^2 = \|\nabla_1 G(z, x) - \nabla_1 G(z^*(x), x)\|^2 \leqslant L_G^2 \|z - z^*(x)\|^2 \ .}$$

For D_v, since $\nabla^2_{11} G(z^*(x), x)v^*(x) = -\nabla_1 F(z^*(x), x)$, we write

$$\begin{aligned} \|D_v\| &= \|(\nabla^2_{11} G(z, x)v + \nabla_1 F(z, x)) - (\nabla^2_{11} G(z^*(x), x)v^*(x) + \nabla_1 F(z^*(x), x))\| && (17) \\ &\leq \|[\nabla^2_{11} G(z, x) - \nabla^2_{11} G(z^*(x), x)]v^*(x)\| + \|\nabla^2_{11} G(z, x)[v - v^*(x)]\| && (18) \\ &\quad + \|\nabla_1 F(z, x) - \nabla_1 F(z^*(x), x)\| \ . \end{aligned}$$

For the first term, we use the Lipschitz continuity of $\nabla^2_{11} G$:

$$\|[\nabla^2_{11} G(z, x) - \nabla^2_{11} G(z^*(x), x)]v^*(x)\| \leq L_{11}^G \|z - z^*(x)\| \|v^*(x)\| \ .$$

Then, since G in μ_G-strongly convex w.r.t. z, $\nabla_1 F(z^*(\cdot), \cdot)$ is bounded and $v^*(x) = -[\nabla^2_{11} G(z^*(x), x)]^{-1} \nabla_1 F(z^*(x), x)$, we have

$$\|[\nabla^2_{11} G(z, x) - \nabla^2_{11} G(z^*(x), x)]v^*(x)\| \leq \frac{L_{11}^G C_F}{\mu_G} \|z - z^*(x)\| \ . \tag{19}$$

For the second term, we use the L_1^G-smoothness of $G(\,\cdot\,, x)$ and for the third term, we use the L^F-smoothness of F and we finally get

$$\|D_v\| \leq \left(\frac{L_{11}^G C_F}{\mu_G} + L^F \right) \|z - z^*(x)\| + L_1^G \|v - v^*(x)\| \ . \tag{20}$$

Then, taking $L_v = \sqrt{2} \max\left(\frac{L_{11}^G C_F}{\mu_G} + L^F, L_1^G \right)$, we get

$$\boxed{\|D_v(z, v, x)\|^2 \leq L_v^2 (\|z - z^*(x)\|^2 + \|v - v^*(x)\|^2) \ .} \tag{21}$$

For $D_x(z, v, x) - \nabla h(x)$ we start by writing

$$\begin{aligned} \|D_x(z, v, x) - \nabla h(x)\| &\leq \|\nabla_2 F(z, x) - \nabla_2 F(z^*(x), x)\| + \|\nabla^2_{21} G(z, x)v - \nabla^2_{21} G(z^*(x), x)v^*(x)\| && (22) \\ &\leq \|\nabla_2 F(z, x) - \nabla_2 F(z^*(x), x)\| + \|\nabla^2_{21} G(z, x)\| \|v - v^*(x)\| && (23) \\ &\quad + \|v^*(x)\| \|\nabla^2_{21} G(z, x) - \nabla^2_{21} G(z^*(x), x)\| \ . \end{aligned}$$

We bound the first term using the fact that $\nabla_2 F$ is L^F-Lipschitz continuous. For the second term, the fact that $\nabla^2_{21} G$ is bounded thanks to the Lipschitz continuity of $\nabla_1 G(z, \cdot)$. For the third term, we use that $\nabla^2_{21} G(\cdot, x)$ is L^G_{21}-Lipschitz continuous and the same derivation as Equation (19). We finally get

$$\|D_x - \nabla h(x)\| \leq \left(L^F + \frac{C_F L^G_{21}}{\mu_G} \right) \|z - z^*(x)\| + L^G_2 \|v - v^*(x)\| \ . \tag{24}$$

Taking $L_x = \sqrt{2} \max \left(L^F + \frac{C_F L^G_{21}}{\mu_G}, L^G_2 \right)$ yields

$$\boxed{\|D_v(z, v, x) - \nabla h(x)\|^2 \leq L_x^2 (\|z - z^*(x)\|^2 + \|v - v^*(x)\|^2) \ .} \tag{25}$$

□

C.3 Smoothness constant of h

From Ghadimi and Wang [2018, Lemma 2.2], we get the Lemma 3.9 which states the L^h-smoothness of h with

$$L^h = L^F + \frac{2L^F L^G_{21} + C_F^2 L^G_{21}}{\mu_G} + \frac{L^G_{11} L^G_2 C_F + L^G_2 L^G_{21} C_F + (L^G_2)^2 L^F}{\mu_G^2} + \frac{(L^G_2)^2 L^G_{11} C_F}{\mu_G^3} \ .$$

C.4 Proof of Lemma 3.8

We start by a technical lemma.

Lemma C.1. *Let $\mu > 0$ and $\mathcal{K} \triangleq \{M \in \mathcal{S}_p(\mathbb{R}), \mu I \preceq M\}$. Then, the application $f : \mathcal{K} \to \mathrm{GL}_p(\mathbb{R})$ given by $f(M) = M^{-1}$ is α-Lipschitz continuous with a constant $\alpha = \mu^{-2}$.*

Proof. The map f is differentiable at $M \in \mathcal{K}$ as an element of $\mathrm{GL}_p(\mathbb{R})$ and its derivative $\mathrm{d}f(M)$ is given by $\mathrm{d}f(M).H = -M^{-1} H M^{-1}$. Then we have

$$\|\mathrm{d}f(M).H\| \leqslant \left\|M^{-1}\right\|^2 \|H\| \leqslant \frac{1}{\mu^2} \|H\| \ .$$

Then the Lipschitz continuity follows. □

Then we show the Lipschitz continuity of z^* and v^*.

Lemma C.2. *There exists a constant $L_* > 0$ such that for any $x_1, x_2 \in \mathbb{R}^d$ we have*

$$\|z^*(x_1) - z^*(x_2)\| \leq L_* \|x_1 - x_2\|, \quad \|v^*(x_1) - v^*(x_2)\| \leq L_* \|x_1 - x_2\| \ .$$

Proof. Let $x \in \mathbb{R}^d$. The Jacobian of z^* is given by $\mathrm{d}z^*(x) = -[\nabla^2_{11} G(z^*(x), x)]^{-1} \nabla^2_{1,2} G(z^*(x), x)$. Thanks to the μ_G-strong convexity of G and the fact that $\nabla^2_{21} G$ is bounded, we have $\|\mathrm{d}z^*(x)\| \leq \frac{L^G_2}{\mu_G}$. Thus, z^* is Lipschitz continuous.

For $\|v^*(x_1) - v^*(x_2)\|$, we start from the definition v^*:

$$\|v^*(x_1) - v^*(x_2)\| = \|[\nabla^2_{11} G(z^*(x_1), x_1]^{-1} \nabla_1 F(z^*(x_1), x_1) - [\nabla^2_{11} G(z^*(x_2), x_2]^{-1} \nabla_1 F(z^*(x_2), x_2)\| \tag{26}$$

$$\begin{aligned} &\leqslant \|([\nabla^2_{11} G(z^*(x_1), x_1)]^{-1} - [\nabla^2_{11} G(z^*(x_2), x_2)]^{-1} \nabla_1 F(z^*(x_1), x_1)\| \\ &\quad + \|[\nabla^2_{11} G(z^*(x_2), x_2)]^{-1} (\nabla_1 F(z^*(x_2), x_2) - \nabla_1 F(z^*(x_1), x_1))\| \ . \end{aligned} \tag{27}$$

For the first term, we use Lemma C.1 and the fact that $\nabla^2_{11} G$ is Lipschitz to get

$$\begin{aligned}\|[\nabla^2_{11}G(z^*(x_1),x_1)]^{-1}-\nabla^2_{11}G(z^*(x_2),x_2)]^{-1}\| &\leq \frac{1}{\mu_G^2}\|\nabla^2_{11}G(z^*(x_1),x_1)-\nabla^2_{11}G(z^*(x_2),x_2)\| \\ &\leq \frac{L^G_{11}}{\mu_G^2}\|(z^*(x_1),x_1)-(z^*(x_2),x_2)\| \\ &\leq \frac{L^G_{11}}{\mu_G^2}[\|z^*(x_1)-z^*(x_2)\|+\|x_1-x_2\|] \\ &\leq \frac{L^G_{11}}{\mu_G^2}\left[1+\frac{L^G_2}{\mu_G}\right]\|x_1-x_2\| \ .\end{aligned}$$

And then, since $\nabla_1 F(z^*(\,\cdot\,),\,\cdot\,)$ is bounded:

$$\left\|([\nabla^2_{11}G(z^*(x_1),x_1)]^{-1}-[\nabla^2_{11}G(z^*(x_2),x_2)]^{-1}\nabla_1 F(z^*(x_1),x_1)\right\| \leq \frac{C_F L^G_{11}}{\mu_G^2}\left[1+\frac{L^G_2}{\mu_G}\right]\|x_1-x_2\| \ .$$

For the second term, the strong convexity of $G(\,\cdot\,,x)$ and the fact that $\nabla_1 F$ is Lipschitz continuous lead to

$$\begin{aligned}\|[\nabla^2_{11}G(z^*(x_2),x_2)]^{-1}(\nabla_1 F(z^*(x_2),x_2)-\nabla_1 F(z^*(x_1),x_1))\| &\leq \frac{1}{\mu_G}\|\nabla_1 F(z^*(x_2),x_2)-\nabla_1 F(z^*(x_1),x_1)\| && (28)\\ &\leq \frac{L^F}{\mu_F}\|(z^*(x_1),x_1)-(z^*(x_2),x_2)\| && (29)\\ &\leq \frac{L^F}{\mu_G}[\|z^*(x_1)-z^*(x_2)\|+\|x_1-x_2\|] && (30)\\ &\leq \frac{L^F}{\mu_G}\left[1+\frac{L^G_2}{\mu_G}\right]\|x_1-x_2\| \ . && (31)\end{aligned}$$

Then we get

$$\|v^*(x_1)-v^*(x_2)\| \leq \left[\frac{C_F L^G_{11}}{\mu_G^2}\left[1+\frac{L^G_2}{\mu_G}\right]+\frac{L^F}{\mu_G}\left[1+\frac{L^G_2}{\mu_G}\right]\right]\|x_1-x_2\| \ . \tag{32}$$

We conclude by setting

$$L_* = \max\left(\frac{L^G_2}{\mu_G}, \frac{C_F L^G_{11}}{\mu_G^2}\left[1+\frac{L^G_2}{\mu_G}\right]+\frac{L^F}{\mu_G}\left[1+\frac{L^G_2}{\mu_G}\right]\right) \ .$$

□

In what follows, we denote by $\mathbb{E}_t[\,\cdot\,]$ the expectation conditionally on z^t, v^t and x^t.
We now provide the proof of Lemma 3.8.

Proof. **Inequality for δ_z.**
We find for $a>0$, using Young's inequality

$$\delta_z^{t+1} \leq (1+a)\mathbb{E}[\|z^{t+1}-z^*(x^t)\|^2]+(1+a^{-1})\mathbb{E}[\|z^*(x^{t+1})-z^*(x^t)\|^2] \ . \tag{33}$$

We study each member, using the unbiasedness of D_z^t and the μ_G-strong convexity of $G(\,\cdot\,,x^t)$:

$$\begin{aligned}\mathbb{E}_t[\|z^{t+1}-z^*(x^t)\|^2] &= \mathbb{E}_t[\|z^t-z^*(x^t)\|^2]-\rho\mathbb{E}_t[\langle D_z^t, z^t-z^*(x^t)\rangle]+\rho^2\mathbb{E}_t[\|D_z^t\|^2] && (34)\\ &= \|z^t-z^*(x^t)\|^2-\rho\langle\nabla_z G(z^t,x^t), z^t-z^*(x^t)\rangle+\rho^2\mathbb{E}_t[\|D_z^t\|^2] && (35)\\ &\leq (1-\rho\mu_G)\|z^t-z^*(x^t)\|^2+\rho^2\mathbb{E}_t[\|D_z^t\|^2] \ . && (36)\end{aligned}$$

Taking the total expectation yields

$$\mathbb{E}[\|z^{t+1} - z^*(x^t)\|^2] \leq (1 - \rho\mu_G)\delta_z^t + \rho^2 V_z^t \ . \tag{37}$$

The second member is bounded using Lipschitz continuity of z^*:

$$\mathbb{E}[\|z^*(x^{t+1}) - z^*(x^t)\|^2] \leq L_*^2 \mathbb{E}[\|x^{t+1} - x^t\|^2] = L_*^2 \gamma^2 V_x^t$$

which gives overall

$$\delta_z^{t+1} \leq (1+a)\left[(1 - \rho\mu_G)\delta_z^t + \rho^2 V_z^t\right] + (1 + a^{-1}) L_*^2 \gamma^2 V_x^t \tag{38}$$

In order to keep a decrease in δ_z, we might want to use $a = \frac{1}{2}\rho\mu_G$, which gives the bound

$$\boxed{\delta_z^{t+1} \leq \left(1 - \frac{\rho\mu_G}{2}\right)\delta_z^t + 2\rho^2 V_z^t + \beta_{zx}\frac{\gamma^2}{\rho} V_x^t} \tag{39}$$

with $\beta_{zx} = 4\frac{L_*^2}{\mu_G}$. Indeed, this gives $(1 + \frac{1}{2}\rho\mu_G)(1 - \rho\mu_G) \leq 1 - \frac{1}{2}\rho\mu_G$. We have $a \leq 1$ since $\rho \leq \frac{2}{\mu_G}$, so $(1+a)\rho^2 \leq 2\rho^2$. Finally, we also have $1 + a^{-1} \leq 2a^{-1} = \frac{4}{\rho\mu_G}$.

Inequality for δ_v. We use a similar technique to get for $b > 0$

$$\delta_v^{t+1} \leq (1+b)\mathbb{E}[\|v^{t+1} - v^*(x^t)\|^2] + (1 + b^{-1})\mathbb{E}[\|v^*(x^{t+1}) - v^*(x^t)\|^2] \ . \tag{40}$$

For the first term, we have

$$\mathbb{E}_t[\|v^{t+1} - v^*(x^t)\|^2] = \|v^t - v^*(x^t)\|^2 - \rho\langle D_v(z^t, v^t, x^t), v^t - v^*(x^t)\rangle + \rho^2 \mathbb{E}_t[\|D_v^t\|^2] \tag{41}$$

Now, using that $D_v(z^*(x^t), v^*(x^t), x^t) = 0$:

$$\begin{aligned}
\langle D_v(z^t, v^t, x^t), v^t - v^*(x^t)\rangle &= \langle D_v(z^t, v^t, x^t) - D_v(z^*(x^t), v^*(x^t), x^t), v^t - v^*(x^t)\rangle & (42)\\
&= \langle \nabla_{11}^2 G(z^t, x^t)(v^t - v^*(x^t)), v^t - v^*(x^t)\rangle & (43)\\
&\quad + \langle (\nabla_{11}^2 G(z^t, x^t) - \nabla_{11}^2 G(z^*(x^t), x^t))v^*(x^t), v^t - v^*(x^t)\rangle\\
&\quad + \langle (\nabla_1 F(z^t, x^t) - \nabla_1 F(z^*(x^t), x^t)), v^t - v^*(x^t)\rangle\\
&\geq \mu_G\|v^t - v^*(x^t)\|^2 - \frac{L_{11}^G C_F}{\mu_G}\|z^t - z^*(x^t)\|\|v^t - v^*(x^t)\| & (44)\\
&\quad - L^F\|z^t - z^*(x^t)\|\|v^t - v^*(x^t)\|\\
&\geq \mu_G\|v^t - v^*(x^t)\|^2 - \omega\|z^t - z^*(x^t)\|\|v^t - v^*(x^t)\| & (45)
\end{aligned}$$

where $\omega = L^F + \frac{L_{11}^G C_F}{\mu_G}$. We then use $\sqrt{\|z^t - z^*(x^t)\|\|v^t - v^*(x^t)\|} \leq \frac{1}{2}c\|v^t - v^*(x^t)\|^2 + \frac{1}{2}c^{-1}\|z^t - z^*(x^t)\|^2$ with $c = \frac{\mu_G}{\omega}$ to get

$$-\langle D_v(z^t, v^t, x^t), v^t - v^*(x^t)\rangle \leq -\frac{1}{2}\mu_G\delta_v^t + \frac{\omega^2}{2\mu_G}\delta_z^t \ .$$

We get the overall inequality by taking the total expectation

$$\mathbb{E}[\|v^{t+1} - v^*(x^t)\|^2] \leq \left(1 - \frac{\rho\mu_G}{2}\right)\delta_v^t + \rho\frac{\omega^2}{2\mu_G}\delta_z^t + \rho^2 V_v^t$$

We also use Lipschitz on v^* to bound the other term

$$\mathbb{E}[\|v^*(x^{t+1}) - v^*(x^t)\|^2] \leq L_*^2\gamma^2 V_x^t$$

and we obtain the full inequality by taking $b = \frac{\rho\mu_G}{4}$ in Equation (40)

$$\boxed{\delta_v^{t+1} \leq \left(1 - \frac{\rho\mu_G}{4}\right)\delta_v^t + \rho\beta_{vz}\delta_z^t + 2\rho^2 V_v^t + \beta_{vx}\frac{\gamma^2}{\rho} V_x^t} \tag{46}$$

with $\beta_{zv} = \frac{\omega^2}{\mu_G}$ and $\beta_{vx} = 8\frac{L_*^2}{\mu_G}$. □

C.5 Proof of Lemma 3.9

Proof. We use smoothness of h to get

$$\mathbb{E}_t[h(x^{t+1})] \leq h(x^t) - \gamma\langle D_x(z^t, v^t, x^t), \nabla h(x^t)\rangle + \frac{L^h}{2}\gamma^2\mathbb{E}_t[\|D_x^t\|^2]$$

We use the previous Lemma 3.4 to get

$$-\langle D_x(z^t, v^t, x^t), \nabla h(x^t)\rangle \leq -\|\nabla h(x^t)\|^2 + L_x\sqrt{\|z^t - z^*(x^t)\|^2 + \|v^t - v^*(x^t)\|^2}\|\nabla h(x^t)\| \tag{47}$$

$$\leq -\frac{1}{2}\|\nabla h(x^t)\|^2 + \frac{L_x^2}{2}(\|z^t - z^*(x^t)\|^2 + \|v^t - v^*(x^t)\|^2) \tag{48}$$

where the last inequality comes from $ab \leq \frac{L_x}{2}a^2 + \frac{1}{2L_x}b^2$ with $a = \sqrt{\|z^t - z^*(x^t)\|^2 + \|v^t - v^*(x^t)\|^2}$ and $b = \|\nabla h(x^t)\|$. Finally, taking the total expectation, we get

$$\boxed{h^{t+1} \leq h^t - \frac{\gamma}{2}g^t + \frac{\gamma L_x^2}{2}(\delta_z^t + \delta_v^t) + \frac{L^h}{2}\gamma^2 V_x^t \ .} \tag{49}$$

□

C.6 Proof of Theorem 1

This section is devoted to the proof of Theorem 1 that we recall here.

Theorem 1 (Convergence of SOBA, fixed step size)**.** Fix an iteration $T > 1$. We consider fixed steps $\rho^t = T^{-\frac{2}{5}}$ and $\gamma^t = T^{-\frac{3}{5}}$. We assume that T is large enough so that $\rho^t \leq \min(\frac{\mu_G}{8L_z^2B_z^2}, \frac{\mu_G}{16L_v^2B_v^2}, \frac{2}{\mu_G})$. Let x^t the sequence of outer iterates for SOBA. Then, $\frac{1}{T}\sum_{t=1}^T \mathbb{E}[\|\nabla h(x^t)\|^2] = O(T^{-\frac{2}{5}})$.

We start by some manipulations that will be useful to prove both SOBA theorems.

Lemma C.3. *Under the assumption that* $\rho^t \leq \min\left(\frac{\mu_G}{8L_z^2B_z^2}, \frac{\mu_G}{16L_v^2B_z^2}, \frac{2}{\mu_G}\right)$*, it holds*

$$\frac{\mu_G}{4}\sum_{t=0}^T \rho^t\delta_z^t \leq \delta_z^0 + 2B_z^2\sum_{t=0}^T(\rho^t)^2 + \beta_{zx}B_x^2\sum_{t=0}^T\frac{(\gamma^t)^2}{\rho^t} \tag{50}$$

$$\frac{\mu_G}{8}\sum_{t=0}^T \rho^t\delta_v^t \leq \delta_v^0 + 2B_v^2\sum_{t=0}^T(\rho^t)^2 + \beta_{vz}\sum_{t=0}^T\rho^t\delta_z^t + \beta_{vx}B_x^2\sum_{t=0}^T\frac{(\gamma^t)^2}{\rho^t} \tag{51}$$

Proof. Assumption 3.6 and Lemma 3.4 give

$$V_z^t \leq B_z^2(1 + L_z^2\delta_z^t)$$

and

$$V_v^t \leq B_v^2(1 + L_v^2(\delta_z^t + \delta_v^t)) \ .$$

Plugging these inequalities in Lemma 3.8 gives

$$\delta_z^{t+1} \leq \left(1 - \frac{\rho^t\mu_G}{2} + 2(\rho^t)^2L_z^2B_z^2\right)\delta_z^t + 2(\rho^t)^2B_z^2 + \beta_{zx}\frac{(\gamma^t)^2}{\rho^t}B_x^2 \tag{52}$$

and

$$\delta_v^{t+1} \leq \left(1 - \frac{\rho^t\mu_G}{4} + 2(\rho^t)^2L_v^2B_v^2\right)\delta_v^t + 2(\rho^t)^2B_v^2 + \rho^t\beta_{vz}\delta_z^t + \beta_{vx}\frac{(\gamma^t)^2}{\rho^t}B_x^2 \ . \tag{53}$$

Under the hypothesis that $\rho^t \leq \frac{\mu_G}{8L_z^2 B_z^2}$ and $\rho^t \leq \frac{\mu_G}{16L_v^2 B_v^2}$, these equations simplify to

$$\delta_z^{t+1} \leq \left(1 - \frac{\rho^t \mu_G}{4}\right) \delta_z^t + 2(\rho^t)^2 B_z^2 + \beta_{zx} \frac{(\gamma^t)^2}{\rho^t} B_x^2 \tag{54}$$

and

$$\delta_v^{t+1} \leq \left(1 - \frac{\rho^t \mu_G}{8}\right) \delta_v^t + 2(\rho^t)^2 B_v^2 + \rho^t \beta_{vz} \delta_z^t + \beta_{vx} \frac{(\gamma^t)^2}{\rho^t} B_x^2 \ . \tag{55}$$

Next, summing these equations for $t = 0 \dots T$ gives

$$\boxed{\frac{\mu_G}{4} \sum_{t=0}^{T} \rho^t \delta_z^t \leq \delta_z^0 + 2B_z^2 \sum_{t=0}^{T} (\rho^t)^2 + \beta_{zx} B_x^2 \sum_{t=0}^{T} \frac{(\gamma^t)^2}{\rho^t}} \tag{56}$$

and

$$\boxed{\frac{\mu_G}{8} \sum_{t=0}^{T} \rho^t \delta_v^t \leq \delta_v^0 + 2B_v^2 \sum_{t=0}^{T} (\rho^t)^2 + \beta_{vz} \sum_{t=0}^{T} \rho^t \delta_z^t + \beta_{vx} B_x^2 \sum_{t=0}^{T} \frac{(\gamma^t)^2}{\rho^t} \ .} \tag{57}$$

□

We are now ready to prove Theorem 1.

Proof. With a fixed step size $\rho^t = \rho = \frac{1}{T^{\frac{2}{5}}}$ and $\gamma^t = \gamma = \frac{1}{T^{\frac{3}{5}}}$, the first equation gives

$$\frac{1}{T} \sum_{t=0}^{T} \delta_z^t \leq \frac{4}{\mu_G} \left[\frac{\delta_z^0}{T\rho} + 2B_z^2 \rho + \beta_{zx} B_x^2 \frac{\gamma^2}{\rho^2} \right] \leq \frac{K_z}{T^{\frac{2}{5}}}$$

where $K_z = \frac{4}{\mu_G}(\delta_z^0 + 2B_z^2 + B_x^2 \beta_{zx})$ does not depend on T. Note that here we used $T^{\frac{3}{5}} \geq T^{\frac{2}{5}}$.

Next, using Equation (51), we get

$$\frac{1}{T} \sum_{t=0}^{T} \delta_v^t \leq \frac{8}{\mu_G} \left[\frac{\delta_v^0}{T\rho} + 2B_v^2 \rho + \frac{\beta_{vz} K_z}{T^{\frac{2}{5}}} + \beta_{vx} B_x^2 \frac{(\gamma)^2}{\rho^2} \right] \leq \frac{K_v}{T^{\frac{2}{5}}}$$

with $K_v = \frac{8}{\mu}(\delta_v^0 + 2B_v^2 + \beta_{vz} K_z + \beta_{vx} B_x^2)$.

Finally, summing the equations in x in Lemma 3.9 we obtain

$$\frac{1}{T} \sum_{t=0}^{T} \mathbb{E} \|\nabla h(x^t)\|^2 \leq 2 \left(\frac{h^0 - h^*}{T\gamma} + \frac{L_x^2 (K_v + K_z)}{2} \frac{1}{T^{\frac{2}{5}}} + L^h \gamma B_x^2 \right) \leq \frac{K_x}{T^{\frac{2}{5}}}$$

with $K_x = 2(h^0 - h^*) + L_x^2 (K_v + K_z) + 2L^h B_x^2$, which shows the advertised result. □

C.7 Proof of Theorem 2

Proof. In the decreasing step size case, after enough iterations we have automatically $\rho^t \leq \frac{\mu_G}{8L_z^2 B_z^2}$ and $\rho^t \leq \frac{\mu_G}{16L_v^2 B_v^2}$, which allows us to use Equations (50) and (51). WLOG, we assume that this happends at iteration 0. Taking $\rho^t = t^{-\frac{2}{5}}$, $\gamma^t = t^{-\frac{3}{5}}$, we recall the integral majorization for $\beta < 1$:

$$\sum_{t=1}^{T} t^{-\beta} \leq \int_0^T t^{-\beta} dt = \frac{T^{1-\beta}}{1-\beta}$$

and

$$\sum_{t=1}^{T} t^{-1} \leq 1 + \int_{1}^{T} t^{-1} dt = 1 + \log(T) \ .$$

We use this in Equation (54) to get

$$\sum_{t=0}^{T} \rho^t \delta_z^t \leq \frac{4}{\mu_G} \left[\delta_z^0 + \left(10 B_z^2 + 5 \beta_{zx} B_x^2 \right) T^{\frac{1}{5}} \right] \leq K_z T^{\frac{1}{5}}$$

where $K_z = \frac{4}{\mu_G}(\delta_z^0 + (10B_z^2 + 5\beta_{zx}B_x^2))$.

Then, Equation (55) yields

$$\sum_{t=0}^{T} \rho^t \delta_v^t \leq \frac{8}{\mu_G} \left[\delta_v^0 + (\beta_{vz} K_z + 10 B_v^2 + 5 \beta_{vx} B_x^2) T^{\frac{1}{5}} \right] \leq K_v T^{\frac{1}{5}}$$

with $K_v = \frac{8}{\mu_G}(\delta_v^0 + \beta_{vz}K_z + 10B_v^2 + 5\beta_{vx}B_x^2)$.

Finally, taking the equation in Lemma 3.9 and multiplying it by $\frac{\rho^t}{\gamma^t}$ gives

$$\rho^t g^t \leq \frac{\rho^t}{\gamma^t}(h^t - h^{t+1}) + \frac{L_x^2}{2} \rho^t (\delta_z^t + \delta_v^t) + L^h \gamma^t \rho^t B_x^2 \ .$$

Summing these equations gives

$$\sum_{t=0}^{T} \rho^t g^t \leq \sum_{t=0}^{T} t^{\frac{1}{5}} (h^t - h^{t+1}) + \frac{L_x^2}{2}(K_v + K_z) T^{\frac{1}{5}} + L^h B_x^2 (1 + \log(T)) \ .$$

We now use an Abel transform to deal with the first sum:

$$\sum_{t=0}^{T} t^{\frac{1}{5}} (h^t - h^{t+1}) = \sum_{t=0}^{T} t^{\frac{1}{5}} h^t - \sum_{t=0}^{T} t^{\frac{1}{5}} h^{t+1} \tag{58}$$

$$= \sum_{t=1}^{T} t^{\frac{1}{5}} h^t - \sum_{t=1}^{T} (t-1)^{\frac{1}{5}} h^t - T^{\frac{1}{5}} h^{T+1} \tag{59}$$

$$\leq \sum_{t=1}^{T} (t^{\frac{1}{5}} - (t-1)^{\frac{1}{5}}) h^t \ . \tag{60}$$

Here, we use the hypothesis that h^t is bounded by h^∞ to get

$$\sum_{t=0}^{T} t^{\frac{1}{5}} (h^t - h^{t+1}) \leq h^\infty T^{\frac{1}{5}} \ .$$

Finally, we obtain the bound

$$\sum_{t=0}^{T} t^{-\frac{2}{5}} g^t \leq \left[h^\infty + \frac{L_x^2}{2}(K_v + K_z) \right] T^{\frac{1}{5}} + L^h B_x^2 (1 + \log(T)) \leq K_x T^{\frac{1}{5}}$$

with $K_x = h^\infty + \frac{L_x^2}{2}(K_v + K_z) + L^h B_x^2 (1 + e^5)$ (because $e^5 = \sup_x \log(x) x^{-1/5}$).

We therefore obtain

$$\inf_{t \leq T} g^t \leq \left(\sum_{t=1}^{T} t^{-\frac{2}{5}} \right)^{-1} \sum_{t=1}^{T} t^{-\frac{2}{5}} g^t \leq \frac{3}{5(T+1)^{\frac{3}{5}}} K_x T^{\frac{1}{5}}$$

and

$$\inf_{t\leq T} g^t \leq \frac{3K_x}{5(T+1)^{\frac{2}{5}}}$$

which shows the advertised result.

□

C.8 Proof of Theorem 3

In this section, we prove Theorem 3 that we recall here

Theorem 3 (Convergence of SABA, smooth case)**.** Let z^t, v^t and x^t the iterates of SABA, with fixed step sizes ρ and γ. We suppose $\rho \leq \rho_*$ and $\gamma \leq \min(\rho\xi_*, \frac{1}{8L^h})$, where ρ_* and ξ_* depend only on F and G and are specified in appendix. Then, $\frac{1}{T}\sum_{t=1}^T \mathbb{E}[\|\nabla h(x^t)\|^2] = O(\frac{1}{T})$.

The constants ρ_* and ξ_* are given by

$$\rho^* = \min\left(\frac{4}{\mu_G}, \sqrt{\frac{\kappa}{8(L'_z+L'_v)}}, \frac{\mu_G}{32L_z^2}, \frac{\mu_G}{64L_v^2}, \frac{\beta_{vz}}{8L_v^2}, \sqrt{\frac{L_x^2+2\psi_v\beta_{vz}}{\beta_{sz}}}, \frac{L_x}{\sqrt{2\beta_{sv}}}, \frac{\kappa}{8(\psi_z L'_z+\psi_v L'_v)}\right)$$

and

$$\xi^* = \min\left(\sqrt{\frac{L'_z+L'_v}{L'_x}}, \frac{L_v}{\sqrt{2}L_x}, \sqrt{\frac{\mu_G}{32\beta_{zx}L_x^2}}, \sqrt{\frac{\mu_G}{64\beta_{vx}L_x^2}}, \sqrt{\frac{\beta_{vz}}{8\beta_{vx}L_x^2}}, \right.$$
$$\left. \frac{1}{\sqrt{8\beta_{zx}\psi_z}}, \frac{L^h L'_x\tilde{\rho}}{2\beta_{zx}L'_x\psi_z+2\beta_{vx}L'_x\psi_v}, \frac{\kappa}{8L^hL'_x\tilde{\rho}}, \frac{1}{\sqrt{4(2P\tilde{\rho}^3+4\beta_{zx}\psi_z+4\psi_v\beta_{vx})}}\right)$$

where

$$\tilde{\xi} = \min\left(\frac{1}{\sqrt{8\beta_{zx}\psi_z}}, \frac{L^hL'_x\tilde{\rho}}{2\beta_{zx}L'_x\psi_z+2\beta_{vx}L'_x\psi_v}, \frac{\kappa}{8L^hL'_x\tilde{\rho}}, \frac{1}{\sqrt{4(2P\tilde{\rho}^3+4\beta_{zx}\psi_z+4\psi_v\beta_{vx})}}\right) ,$$

$$\tilde{\rho} = \min\left(\sqrt{\frac{L_x^2+2\psi_v\beta_{vz}}{\beta_{sz}}}, \frac{L_x}{\sqrt{2\beta_{sv}}}, \frac{\kappa}{8(\psi_zL'_z+\psi_vL'_v)}\right) ,$$

$$\psi_v = \frac{16L_x^2}{\mu_G}, \quad \psi_z = \frac{8}{\mu_G}(L_x^2+2\psi_v\beta_{vz}) \quad \text{and} \quad \kappa = \min\left(\frac{1}{n}, \frac{1}{m}\right) .$$

C.8.1 Control of distance from memory to iterates

We can view our method has having two "parallel" memories for each variable (z_i^t, v_i^t, x_i^t) for $i \in 1[n]$ corresponding to calls in G and $(z_j'^t, v_j'^t, x_j'^t)$ for $j \in [m]$ corresponding to calls to F. At each iteration, we sample i at random uniformly and do $(z_i^{t+1}, v_i^{t+1}, x_i^{t+1}) = (z^t, v^t, x^t)$ and $(z_{i'}^{t+1}, v_{i'}^{t+1}, x_{i'}^{t+1}) = (z_{i'}^t, v_{i'}^t, x_{i'}^t)$ for $i' \neq i$, and similarly for the other memory.

In what follows, we focus on controlling the error between the iterates and the memories. We define to make things simpler

$$E_z^t = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\|z^t - z_i^t\|^2] , \quad E_v^t = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\|v^t - v_i^t\|^2] , \quad E_x^t = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\|x^t - x_i^t\|^2] ,$$

and similarly $E_x'^t, E_v'^t$ and $E_x'^t$.

Lemma C.4. *We have the following inequalities:*

$$E_z^{t+1} \leq \left(1 - \frac{1}{2n}\right) E_z^t + \rho^2 \mathbb{E}\|D_z^t\|^2 + 2n\rho^2 \mathbb{E}[\|D_z(z^t, v^t, x^t)\|^2] \ ,$$

$$E_v^{t+1} \leq \left(1 - \frac{1}{2n}\right) E_v^t + \rho^2 \mathbb{E}\|D_v^t\|^2 + 2n\rho^2 \mathbb{E}[\|D_v(z^t, v^t, x^t)\|^2] \ ,$$

$$E_x^{t+1} \leq \left(1 - \frac{1}{2n}\right) E_x^t + \gamma^2 \mathbb{E}\|D_x^t\|^2 + 2n\gamma^2 \mathbb{E}[\|D_x(z^t, v^t, x^t)\|^2] \ ,$$

$$E_z'^{t+1} \leq \left(1 - \frac{1}{2m}\right) E_z^t + \rho^2 \mathbb{E}\|D_z^t\|^2 + 2m\rho^2 \mathbb{E}[\|D_z(z^t, v^t, x^t)\|^2] \ ,$$

$$E_v'^{t+1} \leq \left(1 - \frac{1}{2m}\right) E_v^t + \rho^2 \mathbb{E}\|D_v^t\|^2 + 2m\rho^2 \mathbb{E}[\|D_v(z^t, v^t, x^t)\|^2] \ ,$$

and

$$E_x'^{t+1} \leq \left(1 - \frac{1}{2m}\right) E_x^t + \gamma^2 \mathbb{E}\|D_x^t\|^2 + 2m\gamma^2 \mathbb{E}[\|D_x(z^t, v^t, x^t)\|^2] \ .$$

Proof. We provide the detailed proof for E_z^t. The approach for the five others is similar.

Let $i \in [n]$. Taking the expectation of $\|z^{t+1} - z_i^{t+1}\|^2$ conditionaly to z^t, v^t, x^t yields

$$\mathbb{E}_t[\|z^{t+1} - z_i^{t+1}\|^2] = \frac{1}{n}\mathbb{E}_t[\|z^{t+1} - z^t\|^2] + \frac{n-1}{n}\mathbb{E}_t[\|z^{t+1} - z_i^t\|^2] \ .$$

Then, using the fact that $\mathbb{E}_t[D_z^t(z^t, v^t, x^t)] = D_z(z^t, v^t, x^t)$, we have

$$\mathbb{E}_t[\|z^{t+1} - z_i^t\|^2] = \mathbb{E}_t[\|z^{t+1} - z^t\|^2] + \|z^t - z_i^t\|^2 - 2\rho\langle D_z(z^t, v^t, x^t), z^t - z_i^t\rangle \ . \tag{61}$$

We then upper-bound crudely the scalar product by Cauchy-Schwarz and Young inequalities with parameter β:

$$\mathbb{E}_t[\|z^{t+1} - z_i^t\|^2] \leq \mathbb{E}_t[\|z^{t+1} - z^t\|^2] + \rho\beta^{-1}\|D_z(z^t, v^t, x^t)\|^2 + (1 + \rho\beta)\|z^t - z_i^t\|^2$$

As a consequence, by taking the total expectation and summing for all $i \in [n]$, we find

$$E_z^{t+1} \leq \rho^2 \mathbb{E}[\|D_z^t\|^2] + \eta\beta^{-1}\left(1 - \frac{1}{n}\right)\mathbb{E}[\|D_z(z^t, v^t, x^t)\|^2] + (1 + \rho\beta)\left(1 - \frac{1}{n}\right)E_z^t \ .$$

Finally, we take $\beta = \frac{1}{2n\rho}$ to obtain

$$\boxed{E_z^{t+1} \leq \left(1 - \frac{1}{2n}\right) E_z^t + \rho^2 \mathbb{E}\|D_z^t(z^t, v^t, x^t)\|^2 + 2n\rho^2 \mathbb{E}[\|D_z(z^t, v^t, x^t)\|^2] \ .} \tag{62}$$

□

C.8.2 Bounds on the variances

We begin by showing an important boundedness result:

Lemma C.5. *Assume that for all t, $\rho^t = \rho < \min(\frac{\mu_G}{8L_z^2 B_z^2}, \frac{\mu_G}{16L_v^2 B_v^2}, \frac{2}{\mu_G})$ and $\gamma^t = \gamma$. Then, the sequence $(\mathbb{E}[\|v^t\|^2])_t$ is bounded.*

Proof. The assumption on ρ ensures that Equations (54) and (55) hold. Since ρ is supposed to be constant, by unrolling Equation (54), we have

$$\delta_z^t \leq \left(1 - \frac{\rho\mu_G}{4}\right)^t \delta_z^0 + K$$

for $K = \frac{4}{\mu_G}\left(2\rho B_z^2 + \beta_{zx}\frac{\gamma}{\rho}B_x^2\right)$. Since $0 < \rho < \frac{2}{\mu_G} < \frac{4}{\mu_G}$, $0 < \left(1 - \frac{\rho\mu_G}{4}\right) < 1$ and then the sequence $(\delta_z^t)_t$ is bounded.

From this and Equation (55), we get with the same technique that the sequence $(\delta_v^t)_t$ is bounded.

We conclude by writing

$$\mathbb{E}[\|v^t\|^2] \leq 2(\delta_v^t + \mathbb{E}[\|v^*(x^t)\|^2])$$

which is the sum of two bounded terms. □

The following lemma gives us upper-bounds for $\mathbb{E}[\|D_z^t(z^t, v^t, x^t)\|^2]$, $\mathbb{E}[\|D_v^t(z^t, v^t, x^t)\|^2]$, and $\mathbb{E}[\|D_x^t(z^t, v^t, x^t)\|^2]$.

Lemma C.6. *For SABA, there are constants $L_z', L_v', L_x' > 0$ such that*

$$\mathbb{E}[\|D_z^t(z^t, v^t, x^t)\|]^2 \leq 2\mathbb{E}[\|D_z(z^t, v^t, x^t)\|^2] + 2L_z'(E_z^t + E_x^t) \ ,$$

$$\mathbb{E}[\|D_v^t(z^t, v^t, x^t)\|^2] \leq 2\mathbb{E}[\|D_v(z^t, v^t, x^t)\|^2] + 2L_v'(E_z^t + E_x^t + E_v^t + E_z'^t + E_x'^t)$$

and

$$\mathbb{E}[\|D_x^t(z^t, v^t, x^t)\|^2] \leq 2\mathbb{E}[\|D_x(z^t, v^t, x^t)\|^2] + 2L_x'(E_z^t + E_x^t + E_v^t + E_z'^t + E_x'^t) \ .$$

Proof. For SABA, if we consider i sampled from $[n]$ at iteration t, we have

$$D_z^t = \nabla_z G_i(z^t, x^t) - \nabla_1 G_i(z_i^t, x_i^t) + \frac{1}{n}\sum_{i'=1}^n \nabla_1 G(z_{i'}^t, x_{i'}^t) \ .$$

Hence we get

$$\begin{aligned}\mathbb{E}_t[\|D_z^t(z^t, v^t, x^t)\|^2] &= \mathbb{E}_t[\|\nabla_1 G_i(z^t, x^t) - \nabla_1 G_i(z_i^t, x_i^t) + \frac{1}{n}\sum_{i'=1}^N \nabla_1 G_{i'}(z_{i'}^t, x_{i'}^t) - \nabla_1 G(z^t, x^t) + \nabla_1 G(z^t, x^t)\|^2] \\ &\leq 2\|\nabla_1 G(z^t, x^t)\|^2 + 2\mathbb{E}_t[\|\nabla_1 G_i(z^t, x^t) - \nabla_1 G_i(z_i^t, x_i^t) + \frac{1}{n}\sum_{i'=1}^N \nabla_1 G(z_{i'}^t, x_{i'}^t) - \nabla_1 G(z^t, x^t)\|^2] \ . \end{aligned} \tag{63}$$

The second term is the variance of $\nabla_z G_i(z^t, x^t) - \nabla_z G_i(z_i^t, x_i^t)$, which is therefore upper-bounded by

$$\begin{aligned}\mathbb{E}_t[\|[\nabla_1 G_i(z^t, x^t) - \nabla_1 G_i(z_i^t, x_i^t)\|^2] &= \frac{1}{n}\sum_{i=1}^n \|[\nabla_1 G_i(z^t, x^t) - \nabla_1 G_i(z_i^t, x_i^t)\|^2 \\ &\leq \frac{L_z'}{n}\sum_{i=1}^n (\|z^t - z_i^t\|^2 + \|x^t - x_i^t\|^2) \end{aligned} \tag{64}$$

where the inequality comes from the Lipschitz continuity of each $\nabla_1 G_i$ with $L_z' = \max_{i\in[n]} L_{G_i}$.

Then, by plugging (64) into (63) and taking the total expectation, we get

$$\boxed{\mathbb{E}[\|D_z^t(z^t, v^t, x^t)\|]^2 \leq 2\mathbb{E}[\|D_z(z^t, v^t, x^t)\|^2] + 2L_z'(E_z^t + E_x^t) \ .} \tag{65}$$

Things are quite similar for the other variables, albeit a bit more difficult.

In v, it holds

$$\begin{aligned}\mathbb{E}_t[\|D_v^t(z^t,v^t,x^t)\|^2] =&\mathbb{E}_t[\|\nabla_1 F_j(z^t,x^t)-\nabla_1 F_j(z_j'^t,x_j'^t)+\frac{1}{m}\sum_{j'=1}^m \nabla_1 F_{j'}(z_{j'}'^t,x_{j'}'^t) \qquad (66)\\ &+\nabla_{11}^2 G_i(z^t,x^t)v^t-\nabla_{11}^2 G_i(z_i^t,x_i^t)v_i^t+\frac{1}{n}\sum_{i'=1}^n \nabla_1 G_{i'}(z_{i'}^t,x_{i'}^t)\\ &-D_v(z^t,v^t,x^t)+D_v(z^t,v^t,x^t)\|^2]\\ \leqslant& 2\mathbb{E}_t[\|D_v(z^t,v^t,x^t)\|^2] \qquad (67)\\ &+2\mathbb{E}_t[\|\nabla_1 F_j(z^t,x^t)-\nabla_1 F_j(z_j'^t,x_j'^t)+\frac{1}{m}\sum_{j'=1}^m \nabla_1 F_{j'}(z_{j'}'^t,x_{j'}'^t)\\ &+\nabla_{11}^2 G_i(z^t,x^t)v^t-\nabla_{11}^2 G_i(z_i^t,x_i^t)v_i^t+\frac{1}{n}\sum_{i'=1}^n \nabla_1 G_{i'}(z_{i'}^t,x_{i'}^t)\\ &-D_v(z^t,v^t,x^t)\|^2]\end{aligned}$$

Here, we see that we need to control the variance of $\nabla_1 F_j(z^t,x^t)-\nabla_1 F_j(z_j'^t,x_j'^t)+\nabla_{11}^2 G_i(z^t,x^t)v^t-\nabla_{11}^2 G_i(z_i^t,x_i^t)v_i^t$. Since i and j are independent, this is a sum of two independent random variables, hence its variance is the sum of the variances, which is upper-bounded by

$$\mathbb{E}_t[\|\nabla_1 F_j(z^t,x^t)-\nabla_1 F_j(z_j'^t,x_j'^t)\|^2]+\mathbb{E}_t[\|\nabla_{11}^2 G_i(z^t,x^t)v^t-\nabla_{11}^2 G_i(z_i^t,x_i^t)v_i^t\|^2] \ .$$

The Lipschitz continuity of all the ∇G_i, $\nabla_{11}^2 G_i$ and $\nabla_1 F_j$, the boundedness of $\mathbb{E}[\|v\|^2]$ (Lemma C.5) and taking the total expectation in (63) lead to

$$\boxed{\mathbb{E}[\|D_v^t(z^t,v^t,x^t)\|^2]\leq 2\mathbb{E}[\|D_v(z^t,v^t,x^t)\|^2]+2L_v'(E_z^t+E_x^t+E_v^t+E_z'^t+E_x'^t) \ .} \qquad (68)$$

In x we have similarly

$$\boxed{\mathbb{E}[\|D_x^t(z^t,v^t,x^t)\|^2]\leq 2\mathbb{E}[\|D_x(z^t,v^t,x^t)\|^2]+2L_x'(E_z^t+E_x^t+E_v^t+E_z'^t+E_x'^t) \ .} \qquad (69)$$

□

We now form $S^t=E_z^t+E_x^t+E_v^t+E_z'^t+E_x'^t$, and letting $\kappa=\min(\frac{1}{m},\frac{1}{n})$. We have the following lemma

Lemma C.7. *If $2\rho^2(L_z'+L_v')+2\gamma^2 L_x'\leq\frac{\kappa}{2}$ and $2L_x^2\rho^2\leq\rho^2 L_v^2$, it holds*

$$S^{t+1}\leq(1-\frac{\kappa}{2})S^t+\beta_{sz}\rho^2\delta_z^t+\beta_{sv}\rho^2\delta_v^t+2P\gamma^2\mathbb{E}[\|\nabla h(x^t)\|^2]$$

for some $L_s,\beta_{sz},P>0$.

Proof. We now form $S^t=E_z^t+E_x^t+E_v^t+E_z'^t+E_x'^t$, and letting $\kappa=\min(\frac{1}{m},\frac{1}{n})$, it holds following eq. (62) (and omitting the dependencies in (z^t,v^t,x^t) in the direction for simplicity)

$$S^{t+1}\leq(1-\kappa)S^t+\mathbb{E}\left[2\rho^2(\|D_z^t\|^2+\|D_v^t\|^2)+2\gamma^2\|D_x^t\|^2+2(m+n)[\rho^2(\|D_z\|^2+\|D_v\|^2)+\gamma^2\|D_x\|^2]\right] \ .$$

Using the previous bounds (65), (68) and (69), we get

$$S^{t+1}\leq(1-\kappa+2\rho^2(L_z'+L_v')+2\gamma^2L_x'))S^t+(2(m+n)+4)\mathbb{E}[\rho^2(\|D_z\|^2+\|D_v\|^2)+\gamma^2\|D_x\|^2] \ .$$

Next, since $\xi^2 L'_x \leq L'_z + L'_v$ and $8\rho^2(L'_z + L'_v) \leq \kappa$, we have $2\rho^2(L'_z + L'_v) + 2\gamma^2 L'_x \leq \frac{\kappa}{2}$ and so, letting $P = (2(m+n)+4)$ we get

$$S^{t+1} \leq \left(1 - \frac{\kappa}{2}\right) S^t + P\mathbb{E}[\rho^2(\|D_z\|^2 + \|D_v\|^2) + \gamma^2\|D_x\|^2] \ .$$

To finish, we use Lemma 3.4 to get

$$S^{t+1} \leq \left(1 - \frac{\kappa}{2}\right) S^t + P[\rho^2((L_z^2 + L_v^2)\delta_z^t + L_v^2\delta_v^t) + 2\gamma^2(\mathbb{E}[\|\nabla h(x^t)\|^2] + L_x^2(\delta_z^t + \delta_v^t))] \ .$$

Then, using that $2L_x^2\gamma^2 \leq \rho^2 L_v^2$, we get the bound, letting $L_s = L_z^2 + L_v^2$:

$$\boxed{S^{t+1} \leq \left(1 - \frac{\kappa}{2}\right) S^t + \beta_{sz}\rho^2\delta_z^t + \beta_{sv}\rho^2\delta_v^t + 2P\gamma^2\mathbb{E}[\|\nabla h(x^t)\|^2]}$$

with $\beta_{sz} = 2PL_s$, $\beta_{sv} = 2PL_v^2$ □

Note that by definition, each quantity E_z^t is smaller than S^t.
We will therefore use the cruder bounds on the $\mathbb{E}[\|D_z^t\|^2]$ as follows:

$$\mathbb{E}[\|D_z^t(z^t, v^t, x^t)\|^2] \leq 2L_z^2\delta_z^t + 2L'_z S^t \ ,$$
$$\mathbb{E}[\|D_v^t(z^t, v^t, x^t)\|^2] \leq 2L_v^2(\delta_z^t + \delta_v^t) + 2L'_v S^t$$

and

$$\mathbb{E}[\|D_x^t(z^t, v^t, x^t)\|^2] \leq 4(\|\nabla h(x^t)\|^2 + L_x^2(\delta_z^t + \delta_v^t)) + 2L'_x S^t \ .$$

C.8.3 Putting it all together

Recall that we denote $g^t = \mathbb{E}[\|\nabla h(x^t)\|^2]$ and $h^t = \mathbb{E}[h(x^t)]$. In the following lemma, we adapt Lemma 3.8 and Lemma 3.9 to the SABA algorithm.

Lemma C.8. *Under the conditions of Theorem 3, it holds*

$$\delta_z^{t+1} \leq \left(1 - \frac{1}{4}\rho\mu_G\right)\delta_z^t + \left[4\rho^2 L'_z + 2\beta_{zx}\frac{\gamma^2}{\rho}L'_x\right] S^t + 4\beta_{zx}\frac{\gamma^2}{\rho}[g^t + L_x^2\delta_v^t] \ ,$$

$$\delta_v^{t+1} \leq \left(1 - \frac{1}{8}\eta\mu_G\right)\delta_v^t + 2\rho\beta_{vz}\delta_z^t + \left[4\rho^2 L'_v + 2\beta_{vx}\frac{\gamma^2}{\rho}L'_x\right] S^t + 4\beta_{vx}\frac{\gamma^2}{\rho}g^t$$

and

$$\frac{\gamma}{4}g^t \leq h^t - h^{t+1} + \alpha L_x^2(\delta_z^t + \delta_v^t) + L^h L'_x\gamma^2 S^t \ .$$

Proof. We plug the previous inequalities in the bounds obtained before on the δ quantities (eq. (39)):

$$\delta_z^{t+1} \leq \left(1 - \frac{1}{2}\rho\mu_G + 4\rho^2 L_z^2 + 4\beta_{zx}\frac{\gamma^2}{\rho}L_x^2\right)\delta_z^t + \left[4\rho^2 L'_z + 2\beta_{zx}\frac{\gamma^2}{\rho}L'_x\right] S^n + 4\beta_{zx}\frac{\gamma^2}{\rho}\left[g^t + L_x^2\delta_v^t\right]$$

where $\beta_{xz} = 4\frac{4L_*^2}{\mu_G}$.

Using that $\rho \leq \min\left(\frac{\mu_G}{64L_v^2}, \frac{\beta_{vz}}{8L_v^2}\right)$ and $\gamma^2 \leq \rho^2 \min\left(\frac{\mu_G}{64\beta_{vx}L_x^2}, \frac{\beta_{vz}}{8\beta_{vx}L_x^2}\right)$, we get $4\rho^2 L_z^2 + 4\beta_{zx}\frac{\alpha^2}{\rho}L_x^2 \leq \frac{1}{4}\rho\mu_G$ and therefore

$$\boxed{\delta_z^{t+1} \leq \left(1 - \frac{1}{4}\rho\mu_G\right)\delta_z^t + \left[4\rho^2 L'_z + 2\beta_{zx}\frac{\gamma^2}{\rho}L'_x\right] S^t + 4\beta_{zx}\frac{\gamma^2}{\rho}[g^t + L_x^2\delta_v^t] \ .} \tag{70}$$

In v, we have

$$\delta_v^{t+1} \leq \left(1 - \frac{\rho\mu_G}{4} + 4L_v^2\rho^2 + 4\beta_{vx}L_x^2\frac{\gamma^2}{\rho}\right)\delta_v^t + \left(\beta_{vz}\rho + 4L_v^2\rho^2 + 4\beta_{vx}L_x^2\frac{\gamma^2}{\rho}\right)\delta_z^t \tag{71}$$

$$+ \left[4\rho^2 L_v' + 2\beta_{vx}\frac{\gamma^2}{\rho}L_x'\right]S^t + 4\beta_{vx}\frac{\gamma^2}{\rho}g^t \tag{72}$$

Then, $\rho \leq \frac{\mu_G}{32L_z^2}$ and $\gamma^2 \leq \rho^2\frac{\mu_G}{32\beta_{zx}L_x^2}$ imply $4L_v^2\rho^2 + 4\beta_{vx}L_x^2\frac{\gamma^2}{\eta} \leq \frac{\rho\mu}{8}$ and $\beta_{vz}\eta \geq 4L_v^2\rho^2 + 4\beta_{vx}L_x^2\frac{\gamma^2}{\rho}$. So we have

$$\boxed{\delta_v^{t+1} \leq \left(1 - \frac{1}{8}\eta\mu_G\right)\delta_v^t + 2\rho\beta_{vz}\delta_z^t + \left[4\rho^2 L_v' + 2\beta_{vx}\frac{\gamma^2}{\rho}L_x'\right]S^t + 4\beta_{vx}\frac{\gamma^2}{\rho}g^t \ .} \tag{73}$$

Finally, in x, using Lemma 3.9, we get

$$\left(\frac{\gamma}{2} - 2L^h\gamma^2\right)g^t \leq h^t - h^{t+1} + \left(\frac{\gamma L_x^2}{2} + 2\gamma^2 L^h L_x^2\right)(\delta_z^t + \delta_v^t) + L^h L_x'\gamma^2 S^t \ .$$

Since $4L^h\gamma \leq 8L^h\gamma \leq 1$, we have

$$\boxed{\frac{\gamma}{4}g^t \leq h^t - h^{t+1} + \gamma L_x^2(\delta_z^t + \delta_v^t) + L^h L_x'\gamma^2 S^t} \tag{74}$$

□

We are now ready to prove Theorem 3.

Proof. We consider the Lyapunov function

$$\mathcal{L}^t = h^t + \phi_s S^t + \phi_z\delta_z^t + \phi_v\delta_v^t \tag{75}$$

for some constants ϕ_s, ϕ_z and ϕ_v.

We have

$$\begin{aligned}
\mathcal{L}^{t+1} - \mathcal{L}^t \leq &\left(\gamma L_x^2 + \beta_{sz}\phi_s\rho^2 - \phi_z\frac{\rho\mu_G}{4} + 2\phi_v\rho\beta_{vz}\right)\delta_z^t \\
&+ \left(\gamma L_x^2 + \beta_{sv}\rho^2\phi_s + 4L_x^2\beta_{zx}\phi_z\frac{\gamma^2}{\rho} - \phi_v\frac{\rho\mu_G}{8}\right)\delta_v^t \\
+ \left(L^h L_x'\gamma^2 - \frac{\kappa}{2}\phi_s + 4\rho^2\phi_z L_z' + 2\beta_{zx}\frac{\gamma^2}{\rho}L_x'\phi_z + 4\rho^2 L_v'\phi_v + 2\beta_{vx}\frac{\gamma^2}{\rho}L_x'\phi_v\right)&S^t \\
&+ \left(-\frac{\gamma}{4} + 2\phi_s P\gamma^2 + 4\beta_{zx}\phi_z\frac{\gamma^2}{\rho} + 4\phi_v\beta_{vx}\frac{\gamma^2}{\rho}\right)g^t \ .
\end{aligned}$$

We want to find ϕ_z, ϕ_v, and ϕ_s such that there is a decrease, so we want to satisfy at the same time, denoting $\xi = \frac{\gamma}{\rho}$:

$$\begin{aligned}
\phi_z\frac{\mu_G}{4} &\geq \xi L_x^2 + \beta_{sz}\phi_s\rho + 2\phi_v\beta_{vz} \\
\phi_v\frac{\mu_G}{8} &\geq \xi L_x^2 + \beta_{sv}\rho\phi_s + 4L_x^2\beta_{zx}\phi_z\xi^2 \\
\frac{\kappa}{2}\phi_s &\geq L^h L_x'\xi^2\rho^2 + 4\rho^2\phi_z L_z' + 4\rho^2 L_v'\phi_v + [2\beta_{zx}L_x'\phi_z + 2\beta_{vx}L_x'\phi_v]\xi^2\rho \\
\frac{1}{4} &\geq 2\phi_s P\gamma + 4\beta_{zx}\phi_z\xi + 4\phi_v\beta_{vx}\xi
\end{aligned}$$

Let us take $\phi_s = \psi_s \rho \xi$, $\phi_z = \psi_z \xi$ and $\phi_v = \psi_v \xi$. The equations become

$$\begin{aligned}
\psi_z \frac{\mu_G}{4} &\geq L_x^2 + \beta_{sz}\psi_s\rho^2 + 2\psi_v\beta_{vz} \\
\psi_v \frac{\mu_G}{8} &\geq L_x^2 + \beta_{sv}\rho^2\psi_s + 4L_x^2\beta_{zx}\psi_z\xi^2 \\
\frac{\kappa}{2}\psi_s &\geq L^h L_x'\xi\rho + 4\rho\psi_z L_z' + 4\rho L_v'\psi_v + [2\beta_{zx}L_x'\psi_z + 2\beta_{vx}L_x'\psi_v]\xi^2 \\
\frac{1}{4} &\geq 2\psi_s\rho^2\xi^2 P + 4\beta_{zx}\psi_z\xi^2 + 4\psi_v\beta_{vx}\xi^2
\end{aligned}$$

Let us take $\psi_v = \frac{16L_x^2}{\mu_G}$, $\psi_z = \frac{8}{\mu_G}(L_x^2 + 2\psi_v\beta_{vz})$ and $\psi_s = 1$. Note that importantly, all these values are independent of the step sizes. The equations become

$$L_x^2 + 2\psi_v\beta_{vz} \geq \beta_{sz}\rho^2 \tag{76}$$

$$L_x^2 \geq \beta_{sv}\rho^2 + 4L_x^2\beta_{zx}\psi_z\xi^2 \tag{77}$$

$$\frac{\kappa}{2} \geq L^h L_x'\xi\rho + 4\rho\psi_z L_z' + 4\rho L_v'\psi_v + [2\beta_{zx}L_x'\psi_z + 2\beta_{vx}L_x'\psi_v]\xi^2 \tag{78}$$

$$\frac{1}{4} \geq 2P\rho^3\xi^2 + 4\beta_{zx}\psi_z\xi^2 + 4\psi_v\beta_{vx}\xi^2 \tag{79}$$

These equations are verified since by assumptions

$$\rho \leq \tilde{\rho} = \min\left(\sqrt{\frac{L_x^2 + 2\psi_v\beta_{vz}}{\beta_{sz}}}, \frac{L_x}{\sqrt{2\beta_{sv}}}, \frac{\kappa}{8(\psi_z L_z' + \psi_v L_v')}\right)$$

and

$$\xi \leq \tilde{\xi} = \min\left(\frac{1}{\sqrt{8\beta_{zx}\psi_z}}, \frac{L^h L_x'\tilde{\eta}}{2\beta_{zx}L_x'\psi_z + 2\beta_{vx}L_x'\psi_v}, \frac{\kappa}{8L^h L_x'\tilde{\eta}}, \frac{1}{\sqrt{4(2P\tilde{\eta}^3 + 4\beta_{zx}\psi_z + 4\psi_v\beta_{vx})}}\right) .$$

Now, let us define

$$\zeta = \frac{\gamma}{4} - 2\phi_s P\gamma^2 - 4\beta_{zx}\phi_z\frac{\gamma^2}{\rho} - 4\phi_v\beta_{vx}\frac{\gamma^2}{\rho} .$$

This quantity is positive thanks to the last inequality (79). We have using the Lyapunov inequality

$$\mathcal{L}^{t+1} - \mathcal{L}^t \leq -\zeta g^t$$

which gives, by summation:

$$\sum_{t=1}^{+\infty} g^t \leq \frac{1}{\zeta}\mathcal{L}^0 < +\infty .$$

Therefore

$$\boxed{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla h(x^t)\|^2] = O\left(\frac{1}{T}\right) .}$$

□

C.9 Proof of Theorem 4

We are now going to prove Theorem 4 that we recall here:

Theorem 4 (Convergence of SABA, PL case)**.** Assume that h satisfies the PL inequality. Let z^t, v^t and x^t the iterates of SABA, with fixed step sizes ρ and γ. We suppose $\rho \leq \rho'_*$ and $\gamma \leq \min(\rho\xi'_*, \frac{1}{8L^h})$, where ρ'_* and ξ'_* depend only on F and G and are specified in appendix. Then, $\mathbb{E}[h(x^T)] - h^* \leq (1 - \frac{\gamma\mu_h}{4})^T(h(x^0) - h^* + C^0)$, where C^0 is a constant specified in appendix that depends on the initialization of z, v, x and memory.

Here, we have

$$\rho'_* = \min\left(\sqrt[3]{\frac{L_x^2 + 2\psi_v\beta_{vz}}{2\beta_{sz}}}, \sqrt[3]{\frac{L_x^2}{3\beta_{sv}}}, \sqrt{\frac{8\psi_v L'_z + 8L'_v\psi_v}{2c'}}, \sqrt[5]{\frac{1}{48\psi_s P}}\right) \ ,$$

$$\xi'_* = \min\left(\sqrt[3]{\frac{L_x^2 + 2\psi_v\beta_{vz}}{2c'\psi_z}}, \sqrt{\frac{L_x^2}{12L_x^2\beta_{zx}\psi_z}}, \sqrt{\frac{L_x^2}{3c'\psi_v}}, \sqrt{\frac{8\psi_v L'_z + 8L'_v\psi_v}{2L^h L'_x}}, \sqrt[5]{\frac{1}{48\psi_s P}}, \sqrt{\frac{1}{96\beta_{zx}\psi_z}}, \sqrt{\frac{1}{96\beta_{vx}\psi_v}}\right) \ ,$$

$$c' = \frac{\mu_h}{4}, \quad \psi_v = \frac{16L_x^2}{\mu_G}, \quad \text{and} \quad \psi_z = \frac{8}{\mu_G}(L_x^2 + 2\psi_v\beta_{vz})$$

Proof. For simplicity, we assume that $h^* = 0$ and so for any $x \in \mathbb{R}^d$ the PL inequality reads:

$$\frac{1}{2}\|\nabla h(x)\|^2 \geq \mu_h h(x) \ . \tag{80}$$

Then, eq. (74) gives

$$h^{t+1} \leq \left(1 - \frac{\gamma\mu_h}{2}\right) h^t + \gamma L_x^2(\delta_z^n + \delta_v^n) + L^h L'_x\gamma^2 S^t \ .$$

We take $\mathcal{L}^t$ the Lyapunov function given in Equation (75). We find

$$\begin{aligned}
\mathcal{L}^{t+1} - \mathcal{L}^t \leq &\left(\gamma L_x^2 + \beta_{sz}\phi_s\rho^2 - \phi_z\frac{\rho\mu_G}{4} + 2\phi_v\rho\beta_{vz}\right)\delta_z^t \\
&+ \left(\gamma L_x^2 + \beta_{sv}\rho^2\phi_s + 4L_x^2\beta_{zx}\phi_z\frac{\gamma^2}{\rho} - \phi_v\frac{\rho\mu_G}{8}\right)\delta_v^t \\
+ \left(L^h L'_x\gamma^2 - \frac{\kappa}{2}\phi_s + 4\rho^2\phi_z L'_z + 2\beta_{zx}\frac{\gamma^2}{\rho}L'_x\phi_z + 4\rho^2 L'_v\phi_v + 2\beta_{vx}\frac{\gamma^2}{\rho}L'_x\phi_v\right)&S^t \\
&+ \left(-\frac{\gamma}{4} + 2\phi_s P\gamma^2 + 4\beta_{zx}\phi_z\frac{\gamma^2}{\rho} + 4\phi_v\beta_{vx}\frac{\gamma^2}{\eta}\right) \times 2\mu_h h^t
\end{aligned}$$

provided that $\zeta = \frac{\gamma}{4} - 2\phi_s P\gamma^2 - 4\beta_{zx}\phi_z\frac{\gamma^2}{\rho} - 4\phi_v\beta_{vx}\frac{\gamma^2}{\rho} > 0$.

We now try to find linear convergence, hence we subtract to this $c\mathcal{L}^t$ to get

$$\begin{aligned}
\mathcal{L}^{t+1} - (1-c)\mathcal{L}^t \leq &\left(\gamma L_x^2 + \beta_{sz}\phi_s\rho^2 - \phi_z\frac{\rho\mu_G}{4} + 2\phi_v\rho\beta_{vz} + c\phi_z\right)\delta_z^t \\
&+ \left(\gamma L_x^2 + \beta_{sv}\rho^2\phi_s + 4L_x^2\beta_{zx}\phi_z\frac{\gamma^2}{\rho} - \phi_v\frac{\rho\mu_G}{8} + c\phi_v\right)\delta_v^t \\
+ \left(L^h L'_x\gamma^2 - \frac{\kappa}{2}\phi_s + 4\rho^2\phi_z L'_z + 2\beta_{zx}\frac{\gamma^2}{\rho}L'_x\phi_z + 4\rho^2 L'_v\phi_v + 2\beta_{vx}\frac{\gamma^2}{\rho}L'_x\phi_v + c\phi_s\right)&S^t \\
&+ \left(-\frac{\gamma}{4} + 2\phi_s P\gamma^2 + 4\beta_{zx}\phi_z\frac{\gamma^2}{\rho} + 4\phi_v\beta_{vx}\frac{\gamma^2}{\eta} + \frac{c}{2\mu_h}\right) \times 2\mu_h h^t \ .
\end{aligned}$$

Hence, the set of inequations for decrease becomes

$$\phi_z \frac{\mu_G}{4} \geq \frac{c}{\rho}\phi_z + \xi L_x^2 + \beta_{sz}\phi_s\rho + 2\phi_v\beta_{vz} \tag{81}$$

$$\phi_v \frac{\mu_G}{8} \geq \frac{c}{\rho}\phi_v + \xi L_x^2 + \beta_{sv}\rho\phi_s + 4L_x^2\beta_{zx}\phi_z\xi^2 \tag{82}$$

$$\frac{\kappa}{2}\phi_s \geq c\phi_s + L^h L'_x\gamma^2 + 4\rho^2\phi_z L'_z + 2\beta_{zx}\frac{\gamma^2}{\rho}L'_x\phi_z + 4\rho^2 L'_v\phi_v + 2\beta_{vx}\frac{\gamma^2}{\rho}L'_x\phi_v \tag{83}$$

$$\frac{1}{4} \geq \frac{c}{2\mu_h} + 2\phi_s P\gamma + 4\beta_{zx}\phi_z\xi + 4\phi_v\beta_{vx}\xi \ . \tag{84}$$

We see that it is more convenient to write $c = \gamma c'$, and we take $\phi_s = \psi_s\rho\xi$, $\phi_z = \psi_z\xi$ and $\phi_v = \psi_v\xi$, with $\psi_v = \frac{16L_x^2}{\mu_G}$, $\psi_z = \frac{8}{\mu_G}(L_x^2 + 2\psi_v\beta_{vz})$ and $\psi_s = 1$, giving:

$$L_x^2 + 2\psi_v\beta_{vz} \geq \beta_{sz}\rho^3 + c'\xi\psi_z \tag{85}$$

$$L_x^2 \geq \beta_{sv}\rho^3 + 4L_x^2\beta_{zx}\psi_z\xi^2 + c'\xi\psi_v \tag{86}$$

$$8\psi_z L'_z + 8L'_v\psi_v \geq L^h L'_x\xi + c'\xi\rho \tag{87}$$

$$\frac{1}{4} \geq \frac{c'}{2\mu_h} + 2P\rho^3\xi^2 + 4\beta_{zx}\psi_z\xi^2 + 4\psi_v\beta_{vx}\xi^2 \ . \tag{88}$$

We take $c' = \frac{\mu_h}{4}$, so that the last inequality becomes

$$\frac{1}{8} \geq 2\psi_s P\rho^3\xi^2 + 4\beta_{zx}\psi_z\xi^2 + 4\psi_v\beta_{vx}\xi^2 \ .$$

As a consequence, for $\rho < \rho'_*$ and $\xi < \xi'_*$, we get

$$\mathcal{L}^{t+1} \leq \left(1 - \frac{\gamma\mu_h}{4}\right)\mathcal{L}^t$$

and we get linear convergence

$$\boxed{h^t - h^* \leq \left(1 - \frac{\gamma\mu_h}{4}\right)^t \mathcal{L}^0 \ .}$$

□