Snake: a Stochastic Proximal Gradient Algorithm for Regularized Problems over Large Graphs

Adil Salim, Pascal Bianchi and Walid Hachem

Abstract—A regularized optimization problem over a large unstructured graph is studied, where the regularization term is tied to the graph geometry. Typical regularization examples include the total variation and the Laplacian regularizations over the graph. When applying the proximal gradient algorithm to solve this problem, there exist quite affordable methods to implement the proximity operator (backward step) in the special case where the graph is a simple path without loops. In this paper, an algorithm, referred to as “Snake”, is proposed to solve such regularized problems over general graphs, by taking benefit of these fast methods. The algorithm consists in properly selecting random simple paths in the graph and performing randomized walks in the graph, paving the way for a stochastic proximal gradient algorithm, whose convergence is proven. Applications to trend filtering and graph inpainting are provided among others. Numerical experiments are conducted over large graphs.

I. INTRODUCTION

Many applications in the fields of machine learning [1]–[3], signal and image restoration [4]–[6], or trend filtering [7]–[12] require the solution of the following optimization problem. On an undirected graph $G = (V,E)$ with no self loops, where $V = \{1, \ldots, N\}$ represents a set of $N$ nodes ($N \in \mathbb{N}^*$) and $E$ is the set of edges, find

$$\min_{x \in \mathbb{R}^V} F(x) + R(x, \phi), \quad (1)$$

where $F$ is a convex and differentiable function on $\mathbb{R}^V$ representing a data fitting term, and the function $x \mapsto R(x, \phi)$ represents a regularization term of the form

$$R(x, \phi) = \sum_{\{i,j\} \in E} \phi_{ij}(x(i), x(j)), \quad (2)$$

where $\phi = (\phi_{ij})_{ij \in E}$ is a family of convex and symmetric $\mathbb{R}^2 \rightarrow \mathbb{R}$ functions. The regularization term $R(x, \phi)$ will be called a $\phi$-regularization in the sequel. These $\phi$-regularizations often promote the sparsity or the smoothness of the solution. For instance, when $\phi_{ij}(x, x') = w_{ij}|x-x'|$ where $w = (w_{ij})_{ij \in E}$ is a vector of positive weights, the function $R(-, \phi)$ coincides with the weighted Total Variation (TV) norm. This kind of regularization is often used in programming problems over a graph which are intended to recover a piecewise constant signal across adjacent nodes [8]–[15]. Another example is the Laplacian regularization $\phi_{ij}(x, x') = (x - x')^2$, or its normalized version obtained by rescaling $x$ and $x'$ by the degrees of each node in $e$ respectively. Laplacian regularization tends to smoothen the solution in accordance with the graph geometry [1].

The Forward-Backward (or proximal gradient) algorithm is one of the most popular approaches towards solving Problem (1). This algorithm produces the sequence of iterates

$$x_{n+1} = \operatorname{prox}_{\gamma R(-, \phi)}(x_n - \gamma \nabla F(x_n)), \quad (2)$$

where $\gamma > 0$ is a fixed step, and where

$$\operatorname{prox}_{\gamma}(y) = \arg \min_x \left( g(x) + \frac{1}{2\gamma} \|x - y\|^2 \right)$$

is the well-known proximity operator applied to the proper, lower semicontinuous (lsc), and convex function $g$ (here $\| \cdot \|$ is the standard Euclidean norm). When $F$ satisfies a smoothness assumption, and when $\gamma$ is small enough, it is indeed well-known that the sequence $(x_n)$ converges to a minimizer of (1), assuming this minimizer exists.

Implementing the proximal gradient algorithm requires the computation of the proximity operator applied to $R(-, \phi)$ at each iteration. When $N$ is large, this computation is in general affordable only when the graph exhibits a simple structure. For instance, when $R(-, \phi)$ is the TV norm, the so-called taut-string algorithm is an efficient algorithm for computing the proximity operator when the graph is one-dimensional (1D) [16] (see Figure 1) or when it is a two-dimensional (2D) regular grid [15]. Similar observations can be made for the Laplacian regularization [17], where, e.g., the discrete cosine transform can be implemented. When the graph is large and unstructured, these algorithms cannot be used, and the computation of the proximity operator is more difficult ([8], [18]).

This problem is addressed in this paper. Towards obtaining a simple algorithm, we first express the functions $F(\cdot)$ and $R(-, \phi)$ as the expectations of functions defined on a random walk in the graph, paving the way for a randomized version of the proximal gradient algorithm. Stochastic online algorithms in the spirit of this algorithm are often considered as simple and reliable procedures for solving high dimensional machine learning problems, including in the situations where the randomness is not inherent to these problems [19], [20]. One specificity of the algorithm developed here lies in that it reconciles two requirements: on the one hand, the random versions of $R(-, \phi)$ should be defined on simple paths, i.e., on walks without loops (see Figure 1), in a way to make benefit of the power of the existing fast algorithms for computing the proximity operator. Owing to the existence of a procedure for
selecting these simple paths, we term our algorithm as the “Snake” algorithm. On the other hand, the expectations of the functions handled by the optimization algorithm coincide with $F(\cdot)$ and $R(\cdot, \phi)$ respectively (up to a multiplicative constant), in such a way that the algorithm does not introduce any bias on the estimates.

The problem of computing the TV-proximity operator over a 1D-graph with a complexity of order $O(N)$ is observed that, empirically, this methods performs better than other concurrent approaches. As a matter of fact, this last general stochastic approximation algorithm that we develop still needs a global knowledge of the graph topology during the iterations. On the contrary, the Snake algorithm only needs a local knowledge. Finally, in [9], the authors propose to replace the computation of the TV-proximity operator over the graph $G$ by the computation of the TV-proximity operator over an 1D-subgraph of $G$ well chosen. This produces an approximation of the solution whereas the Snake algorithm is proven to converge to the exact solution.

The snake algorithm obtains this decomposition online. In [11], the authors propose to compute a decomposition of the graph in 1D-graphs and then solve Problem (1) by means of the TV-proximity operators over these 1D-graphs. Although the decomposition of the graph is fast in many applications, the algorithm [12] relies on an offline decomposition of the whole graph that needs a global knowledge of the graph topology. The Snake algorithm obtains this decomposition online. In [11], the authors propose a working set strategy to compute the TV-proximity operator. At each iteration, the graph is cut in two well-chosen subgraphs and a reduced problem of (1) is deduced from this cut. The reduced problem is then solved efficiently. This method has shown speed-ups when $G$ is an image (i.e. a two dimensional grid). Although the decomposition of the graph is not done during the preprocessing time, the algorithm [11] still needs a global knowledge of the graph topology during the iterations.

Before presenting our approach, we provide an overview of the literature dealing with our problem. First consider the case where $R(\cdot, \phi)$ coincides with the TV norm. As said above, fast methods exist when the graph has a simple structure. We refer the reader to [13] for an overview of iterative solvers of Problem (1) in these cases. In [12], the author introduces a dynamical programming method to compute the proximity operator on a 1D-graph with a complexity of order $O(N)$. Still in the 1D case, Condat [16] revisited recently an algorithm that is due to Mammen and Van De Geer [23] referred to as the taut-string algorithm. The complexity of this algorithm is $O(N^2)$ in the worst-case scenario, and $O(N)$ in the most realistic cases. The taut-string algorithm is linked to a total variation regularized problem in [24]. This algorithm is generalized to 2D-grids, weighted TV norms and $\ell^p$ TV norms by Barbero and Sra in [13]. To generalize to 2D-grids, the TV regularization can be written as a sum of two terms on which one can apply 1D methods, according to [25] and [26]. Over general graphs, there is no immediate way to generalize the taut string method. The problem of computing the TV-proximity operator over a general graph is addressed in [8].

The authors of [8] suggest to solve the problem using a projected Newton algorithm applied to the dual problem. It is observed that, empirically, this methods performs better than other concurrent approaches. As a matter of fact, this statement holds when the graph has a moderate size. As far as large graphs are concerned, the iteration complexity of the projected Newton method can be a bottleneck. To address this problem, the authors of [14] and [8] propose to solve the problem distributively over the nodes using the Alternating Direction Method of Multipliers (ADMM).

In [12] the authors propose to compute a decomposition of the graph in 1D-graphs and then solve Problem (1) by means of the TV-proximity operators over these 1D-graphs. Although the decomposition of the graph is fast in many applications, the algorithm [12] relies on an offline decomposition of the whole graph that needs a global knowledge of the graph topology. The Snake algorithm obtains this decomposition online. In [11], the authors propose a working set strategy to compute the TV-proximity operator. At each iteration, the graph is cut in two well-chosen subgraphs and a reduced problem of (1) is deduced from this cut. The reduced problem is then solved efficiently. This method has shown speed-ups when $G$ is an image (i.e. a two dimensional grid). Although the decomposition of the graph is not done during the preprocessing time, the algorithm [11] still needs a global knowledge of the graph topology during the iterations. On the contrary, the Snake algorithm only needs a local knowledge. Finally, in [9], the authors propose to replace the computation of the TV-proximity operator over the graph $G$ by the computation of the TV-proximity operator over an 1D-subgraph of $G$ well chosen. This produces an approximation of the solution whereas the Snake algorithm is proven to converge to the exact solution.

In the case where $R(\cdot, \phi)$ is the Laplacian regularization, the computation of the proximity operator of $R$ reduces to the resolution of a linear system $(\mathcal{L} + \alpha I)x = b$ where $\mathcal{L}$ is the Laplacian matrix of the graph $G$ and $I$ the identity matrix. On an 1D-graph, the latter resolution can be done efficiently and relies on an explicit diagonalization of $\mathcal{L}$ ([17]) by means of the discrete cosine transform, which take $O(N \log(N))$ operations. Over general graphs, the problem of computing the proximity operator of the Laplacian regularization is introduced in [2]. There exist fast algorithms to solve it due to Spielman and Teng ([27]). They are based on recursively preconditioning the conjugate gradient method using graph theoretical results ([18]). Nevertheless, the preconditionning phase which may be demanding over very large graphs. Compared to [18], our online method Snake requires no preprocessing step.

II. OUTLINE OF THE APPROACH AND PAPER ORGANIZATION

The starting point of our approach is a new stochastic optimization algorithm that has its own interest. This algorithm will be presented succinctly here, and more rigorously in Sec. III below. Given an integer $L > 0$, let $\xi = (\xi^1, \ldots, \xi^L)$ be a random vector where the $\xi^i$ are valued in some measurable space. Consider the problem

$$\min_x \sum_{i=1}^{L} \mathbb{E}_\xi \left[ f_i(x, \xi^i) + g_i(x, \xi^i) \right]$$

where the $f_i(\cdot, \xi^i)$ are convex and differentiable, and the $g_i(\cdot, \xi^i)$ are convex. Given $\gamma > 0$, define the operator $T_{\gamma,i}(x, s) = \text{prox}_{\gamma g_i(\cdot, s)}(x - \gamma \nabla f_i(x, s))$. Given a sequence $(\xi_n)$ of independent copies of $\xi$, and a sequence of positive steps $(\gamma_n) \in \ell^2 \setminus \ell^1$, we consider the algorithm

$$x_{n+1} = T_{\gamma_{n+1}}(x_n; \xi_{n+1})$$

Fig. 1. Left: General graph on which is colored the simple path 3-1-0-6-7. Right: 1D-graph.
where
\[ T_\gamma(\cdot, (s^1, \ldots, s^L)) = T_{\gamma,L}(\cdot, s^L) \circ \cdots \circ T_{\gamma,1}(\cdot, s^1) \]
and where \( \circ \) stands for the composition of functions: \( f \circ g(\cdot) = f(g(\cdot)) \). In other words, an iteration of this algorithm consists in the composition of \( L \) random proximal gradient iterations. The case where \( L = 1 \) was treated in [21].

Assuming that the set of minimizers of the problem is non empty, Th. [1] below states that the sequence \( (x_n) \) converges almost surely to a (possibly random) point of this set. The proof of this theorem which is rather technical is deferred to Sec. VII. It follows the same canvas as the approach of [21], with the difference that we are now dealing with possibly different functions \( f_i, g_i \) and non-independent noises \( \xi \) for \( i \in \{1, \ldots, L\} \).

We now want to exploit this stochastic algorithm to develop a simple procedure leading to a solution of Problem [1]. This will be done in Sec. IV and will lead to the Snake algorithm. The first step is to express the function \( R(\cdot, \phi) \) as the expectation of a function with respect to a finite random walk. Given an integer \( M > 0 \) and a finite walk \( s = (v_0, v_1, \ldots, v_M) \) of length \( M \) on the graph \( G \), where \( v_i \in V \) and \( \{v_i, v_{i+1}\} \in E \), write
\[
R(x, \phi_x) = \sum_{i=1}^{M} \phi_{\{v_{i-1}, v_i\}}(x(v_{i-1}), x(v_i)).
\]

Now, pick a node at random with a probability proportional to the degree (i.e., the number of neighbors) of this node. Once this node has been chosen, pick another one at random uniformly among the neighbors of the first node. Repeat the process of choosing neighbors \( M \) times, and denote as \( \xi \in V^{M+1} \) the random walk thus obtained. With this construction, we get that
\[
\frac{1}{|E|}R(x, \phi) = \frac{1}{|E|} \mathbb{E}_\xi [R(x, \phi_\xi)]
\]
using some elementary Markov chain formalism (see Prop. 2 below).

In these conditions, a first attempt of the use of Algorithm (4) is to consider Problem (1) as an instance of Problem (3) with \( L = 1 \), \( f_1(x, \xi) = \frac{1}{|E|} F(x) \), and \( g_1(x, \xi) = \frac{1}{|E|} R(x, \phi_\xi) \). Given an independent sequence \( (\xi_n) \) of walks having the same law as \( \xi \) and a sequence \( (\gamma_n) \) of steps in \( \ell^2 \setminus \ell^1 \), Algorithm (4) boils down to the stochastic version of the proximal gradient algorithm
\[
x_{n+1} = \text{prox}_{\gamma_{n+1}} R(x, \phi_{\xi_{n+1}})(x_n - \gamma_{n+1} \frac{1}{|E|} \nabla F(x_n)). \tag{5}
\]

By Th. 1 (or by [21]), the iterates \( x_n \) converge almost surely to a solution of Problem (1).

However, although simpler than the deterministic algorithm (2), this algorithm is still difficult to implement for many regularization functions. As said in the introduction, the walk \( \xi \) is often required to be a simple path. Obviously, the walk generation mechanism described above does not prevent \( \xi \) from having repeated nodes. A first way to circumvent this problem would be to generate \( \xi \) as a loop-erased walk on the graph. Unfortunately, the evaluation of the corresponding distribution is notoriously difficult. The generalization of Prop. 2 to loop-erased walks is far from being immediate.

As an alternative, we identify the walk \( \xi \) with the concatenation of at most \( M \) simple paths of maximal length that we denote as \( \xi^1, \ldots, \xi^M \), these random variables being valued in the space of all walks in \( G \) of length at most \( M \):
\[
\xi = (\xi^1, \xi^2, \ldots, \xi^M).
\]

Here, in the most frequent case where the number of simple paths is strictly less than \( M \), the last \( \xi^i \)'s are conventionally set to a trivial walk, i.e. a walk with one node and no edge. We also denote as \( \ell(\xi^i) \) the length of the simple path \( \xi^i \), i.e., the number of edges in \( \xi^i \). We now choose \( L = M \), and for \( i = 1, \ldots, L \), we set \( f_i(x, \xi^i) = \frac{\ell(\xi^i)}{|E|} F(x) \) and \( g_i(x, \xi^i) = \frac{1}{|E|} R(x, \phi_{\xi^i}) \). If \( \ell(\xi^i) > 0 \), and \( f_i(x, \xi^i) = g_i(x, \xi^i) = 0 \) otherwise. With this construction, we show in Sec. IV that \( \frac{1}{|E|}(F(x) + R(x, \phi)) = \sum_{i=1}^{L} \mathbb{E}_\xi [f_i(x, \xi^i) + g_i(x, \xi^i)] \) and that the functions \( f_i, g_i \) fulfill the general assumptions required for the Algorithm (4) to converge to a solution of Problem (1). In summary, at each iteration, we pick up a random walk of length \( L \) according to the procedure described above, we split it into simple paths of maximal length, then we successively apply the proximal gradient algorithm to these simple paths.

After recalling the contexts of the taut-string and the Laplacian regularization algorithms (Sec. V), we simulate Algorithm (4) in several application contexts. First, we study the so called graph trend filtering [8], with the parameter \( k \) defined in [8] set to one. Then, we consider the graph inpainting problem [1], [2], [15]. These contexts are the purpose of Sec. VI.

Finally, a conclusion and some future research directions are provided in Sec. VII.

III. A General Stochastic Proximal Gradient Algorithm

Notations. We denote by \( (\Omega, \mathcal{F}, \mathbb{P}) \) a probability space and by \( \mathbb{E}[\cdot] \) the corresponding expectation. We let \( (\Xi, \mathcal{B}) \) be an arbitrary measurable space. We denote \( X \) some Euclidean space and by \( \mathcal{B}(X) \) its Borel \( \sigma \)-field. A mapping \( f : X \times \Xi \to \mathbb{R} \) is called a normal convex integrand if \( f \) is \( \mathcal{B}(X) \otimes \mathcal{B} \)-measurable and if \( f(\cdot, s) \) is convex for all \( s \in \Xi \) [28].

A. Problem and General Algorithm

In this section, we consider the general problem
\[
\min_{x \in X} \sum_{i=1}^{L} \mathbb{E} \left[ f_i(x, \xi^i) + g_i(x, \xi^i) \right] \tag{6}
\]
where \( L \) is a positive integer, the \( \xi^i : \Omega \to \Xi \) are random variables (r.v.), and the functions \( f_i : X \times \Xi \to \mathbb{R} \) and \( g_i : X \times \Xi \to \mathbb{R} \) satisfy the following assumption:

Assumption 1. The following holds for all \( i \in \{1, \ldots, L\} \):
1) The \( f_i \) and \( g_i \) are normal convex integrands.
2) For every \( x \in X \), \( \mathbb{E}[|f_i(x, \xi^i)|] < \infty \) and \( \mathbb{E}[|g_i(x, \xi^i)|] < \infty \).
3) For every \( s \in \Xi \), \( f_i(\cdot, s) \) is differentiable. We denote as \( \nabla f_i(\cdot, s) \) its gradient w.r.t. the first variable.

Remark. In this paper, we assume that the functions \( g_i(\cdot; \xi) \) have a full domain for almost all \( \xi \). This assumption can be relaxed with some effort, along the ideas developed in [21].
For every $i = 1, \ldots, L$ and every $\gamma > 0$, we introduce the mapping $T_{\gamma,i} : X \times \Xi \to X$ defined by
$$T_{\gamma,i}(x,s) = \text{prox}_{\gamma g_i(\cdot,s)}(x - \gamma \nabla f_i(x,s)).$$
We define $T_\gamma : X \times \Xi^L \to X$ by
$$T_\gamma(\cdot,(s^1,\ldots,s^L)) = T_{\gamma,L}(\cdot,s^L) \circ \cdots \circ T_{\gamma,1}(\cdot,s^1).$$
Let $\xi$ be the random vector $\xi = (\xi^1,\ldots,\xi^L)$ with values in $\Xi^L$ and let $(\xi_n : n \in \mathbb{N}^*)$ be a sequence of i.i.d. copies of $\xi$, defined on the same probability space $(\Omega,\mathcal{F},\mathbb{P})$. For all $n \in \mathbb{N}^*$, $\xi_n = (\xi_n^1,\ldots,\xi_n^L)$. Finally, let $(\gamma_n)$ be a positive sequence. Our aim is to analyze the convergence of the iterates $(x_n)$ recursively defined by:
$$x_{n+1} = T_{\gamma_n+1}(x_n,\xi_{n+1}),$$
(7)
as well as the intermediate variables $\bar{x}_{n+1}^i = x_n$ and $\bar{x}_{n+1}^i = T_{\gamma_n+1,i}(\bar{x}_{n+1}^{i-1},\xi_{n+1}^i)$, $i = 1,\ldots, L$. (8)
In particular, $x_{n+1} = x_{n+1}^L = T_{\gamma_n+1,L}(\bar{x}_{n+1}^{L-1},\xi_{n+1}^L)$. For every $i = 1,\ldots, L$

$$x_{n+1} = \text{prox}_{\gamma_{n+1}g_i(\cdot,\xi_{n+1})}(x_n - \gamma_{n+1}\nabla f_i(x_n,\xi_{n+1})),$$
(9)

and converge a.s. to a minimizer of $E[f_1(x,\xi) + g_1(x,\xi)]$ under the convenient hypotheses.

It is worth noting that the present algorithm (7) cannot be written as an instance of (9). Indeed, the operator $T_{\gamma}$ is a composition of $L$ (random) operators, whereas the stochastic forward backward algorithm (2) has a simpler structure. This composition raises technical difficulties that need to be specifically addressed.

**B. Almost sure convergence**

We make the following assumptions.

**Assumption 2.** The positive sequence $(\gamma_n)$ satisfies the conditions
\[ \sum \gamma_n = +\infty \quad \text{and} \quad \sum \gamma_n^2 < \infty, \]
(i.e., $(\gamma_n) \in l^2 \setminus l^1$). Moreover, $\frac{\gamma_{n+1}}{\gamma_n} \to 1$.

**Assumption 3.** The following holds for all $i \in \{1,\ldots,L\}$:
1) There exists a measurable map $K_i : \Xi \to \mathbb{R}_+$ s.t. the following holds $\mathbb{P}$-a.e.: for all $x, y \in X$,
\[ \|\nabla f_i(x,\xi) - \nabla f_i(y,\xi)\| \leq K_i(\xi_i)\|x - y\|. \]
2) For all $\alpha > 0$, $E[K_i(\xi_i)^\alpha] < \infty$.

We denote by $Z$ the set of minimizers of Problem (6). Thanks to Ass. [1] the qualification conditions hold, ensuring that a point $x^*$ belongs to $Z$ iff
\[ 0 \in \sum_{i=1}^L \nabla E[f_i(x,\xi^i)] + \partial E[g_i(x,\xi^i)]. \]
The (sub)differential and the expectation operators can be interchanged [32], and the above optimality condition also reads
\[ 0 \in \sum_{i=1}^L E[\nabla f_i(x,\xi^i)] + E[\partial g_i(x,\xi^i)], \]
where $E[\partial g_i(x,\xi^i)]$ is the Aumann expectation of the random set $\partial g_i(x,\xi^i)$, defined as the set of expectations of the form $E[\varphi_i(\xi^i)]$, where $\varphi_i : \Xi \to X$ is a measurable map s.t. $\varphi_i(\xi^i)$ is integrable and
\[ \varphi_i(\xi^i) \in \partial g_i(x,\xi^i) \quad \text{P-a.e.}, \quad \forall i. \]
Therefore, the optimality condition (10) means that there exist $L$ integrable mappings $\varphi_1,\ldots,\varphi_L$ satisfying (11) and s.t.
\[ 0 = \sum_{i=1}^L E[\nabla f_i(x,\xi^i)] + E[\varphi_i(\xi^i)]. \]

When (11)-(12) hold, we say that the family $(\nabla f_i(x^*,\xi^i),\varphi_i(\xi^i))_{i=1,\ldots,L}$ is a representation of the minimizer $x^*$. In addition, if for some $\alpha \geq 1$ and every $i = 1,\ldots,L$, $E[\|\nabla f_i(x^*,\xi^i)\|^\alpha] < \infty$ and $E[\|\varphi_i(\xi^i)\|^\alpha] < \infty$, we say that the minimizer $x^*$ admits a $\alpha$-integrable representation.

**Assumption 4.** 1) The set $Z$ is not empty.
2) For every $x^* \in Z$, there exists $\varepsilon > 0$ s.t. $x^*$ admits a $(2 + \varepsilon)$-integrable representation $(\nabla f_i(x^*,\xi^i),\varphi_i(\xi^i))_{i=1,\ldots,L}$.

We denote by $\partial g_0(x,\xi^i)$ the least norm element in $\partial g_i(x,\xi^i)$.

**Assumption 5.** For every compact set $K \subset X$, there exists $\eta > 0$ such that for all $i = 1,\ldots,L$,
\[ \sup_{x \in K} E[\|\partial g_i(x,\xi^i)\|^{1+\eta}] < \infty. \]

We can now state the main result of this section, which will be proven in Sec. VII.

**Theorem 1.** Let Ass. [7],[5] hold true. There exists a r.v. $X_n$ s.t. $\mathbb{P}(X_n \in Z) = 1$ and s.t. $(x_n)$ converges a.s. to $X_n$ as $n \to \infty$. Moreover, for every $i = 0,\ldots,L - 1$, $\bar{x}_n^i$ converges a.s. to $x^*_i$.

**IV. THE SNAKE ALGORITHM**

**A. Notations**

Let $L \geq 1$ be an integer. We refer to a walk of length $\ell$ over the graph $G$ as a sequence $s = (v_0, v_1, \ldots, v\ell)$ in $V^{\ell+1}$ such that for every $i = 1,\ldots,\ell$, the pair $(v_{i-1}, v_i)$ is an edge of the graph. A walk of length zero is a single vertex.

We shall often identify $s$ with the graph $G(s)$ whose vertices and edges are respectively given by the sets $V(s) = \{v_0,\ldots,v\ell\}$ and $E(s) = \{(v_0,v_1),\ldots,(v\ell-1,v\ell)\}$.
Let $L \geq 1$. We denote by $\Xi$ the set of all walks over $G$ with length $\leq L$. This is a finite set. Let $\mathcal{X}$ be the set of all subsets of $\Xi$. We consider the measurable space $(\Xi, \mathcal{X})$.

Let $s = (v_0, v_1, \ldots, v_t) \in \Xi$ with $0 < t \leq L$. We abusively denote by $\phi_s$ the family of functions $(\phi_{v_{i-1}, v_i})_{i=1,\ldots,t}$. We refer to the $\phi_s$-regularization of $x$ as the $\phi_s$-regularization on the graph $s$ of the restriction of $x$ to $s$ that is
\[
R(x, \phi_s) = \sum_{i=1}^{t} \phi_{v_{i-1}, v_i}(x(v_{i-1}), x(v_i)).
\]

Besides, $R(x, \phi_s)$ is defined to be 0 if $s$ is a single vertex (that is $\ell = 0$).

We say that a walk is a simple path if there is no repeated node $i.e.$ all elements in $s$ are different or if $s$ is a single vertex. Throughout the paper, we assume that when $s$ is a simple path, the computation of $\text{prox}_{R(\cdot, \phi_s)}$ can be done easily.

B. Writing the Regularization Function as an Expectation

One key idea of this paper is to write the function $R(x, \phi)$ as an expectation in order to use a stochastic approximation algorithm, as described in Sec. III

Denote by $\deg(v)$ the degree of the node $v \in V$, i.e., the number of neighbors of $v$ in $G$. Let $\pi$ be the probability measure on $V$ defined as
\[
\pi(v) = \frac{\deg(v)}{2|E|}, \quad v \in V.
\]

Define the probability transition kernel $P$ on $V^2$ as $P(v, w) = 1_{\{v, w\} \in E} / \deg(v)$ if $\deg(v) > 0$, and $P(v, w) = 1_{v = w}$ otherwise, where $1$ is the indicator function.

We refer to a Markov chain (indexed by $\mathbb{N}$) over $V$ with initial distribution $\pi$ and transition kernel $P$ as an infinite random walk over $G$. Let $(v_k)_{k \in \mathbb{N}}$ be a infinite random walk over $G$ defined on the canonical probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with $\omega = V^\mathbb{N}$. The first node $v_0$ of this walk is randomly chosen in $V$ according to the distribution $\pi$. The other nodes are drawn recursively according to the conditional probability
\[
\mathbb{P}(v_{k+1} = w | v_k = v) = P(v_k, w).
\]

In other words, conditionally to $v_k$, the node $v_{k+1}$ is drawn uniformly from the neighborhood of $v_k$. Setting an integer $L \geq 1$, we define the random variable $\xi$ from $(v_k)_{k \in \mathbb{N}}$ as $\xi = (v_0, v_1, \ldots, v_L)$.

**Proposition 2.** For every $x \in \mathbb{R}^V$,
\[
\frac{1}{|E|} R(x, \phi) = \frac{1}{L} \mathbb{E}[R(x, \phi_{\xi})].
\]

**Proof.** It is straightforward to show that $\pi$ is an invariant measure of the Markov chain $(v_k)_{k \in \mathbb{N}}$. Moreover, $\mathbb{P}(v_k = w, v_{k-1} = v) = \pi(v)P(v, w) = 1_{\{v, w\} \in E} / (2|E|)$, leading to the identity
\[
\mathbb{E}[\phi_{v_{k-1}, v_k}(x(v_{k-1}), x(v_k))] = \frac{1}{|E|} R(x, \phi),
\]
which completes the proof by symmetry of $\phi_e, \forall e \in E$. $\Box$

This proposition shows that Problem (1) is written equivalently
\[
\min_{x \in \mathbb{R}^V} \frac{1}{|E|} F(x) + \mathbb{E} \frac{1}{L} R(x, \phi_{\xi}).
\]

Hence, applying the stochastic proximal gradient algorithm to solve (14) leads to a new algorithm to solve (1), which was mentioned in Sec. [1] Eq. (5):
\[
x_{n+1} = \text{prox}_{\gamma_n \frac{1}{|E|} \nabla F(x_n)}(x_n - \gamma_n \frac{1}{|E|} \nabla F(x_n)).
\]

Although the iteration complexity is reduced in (15) compared to (2), the computation of the proximity operator of the $\phi$-regularization over the random subgraph $\xi_{n+1}$ in the algorithm (15) can be difficult to implement. This is due to the possible presence of loops in the random walk $\xi$. As an alternative, we split $\xi$ into several simple paths. We will then replace the proximity operator over $\xi$ by the series of the proximity operators over the simple paths induced by $\xi$, which are efficiently computable.

C. Splitting $\xi$ into Simple Paths

Let $(v_k)_{k \in \mathbb{N}}$ be an infinite random walk on $(\Omega, \mathcal{F}, \mathbb{P})$. We recursively define a sequence of stopping time $(\tau_i)_{i \in \mathbb{N}}$ as $\tau_0 = 1$ and for all $i \geq 0$,
\[
\tau_{i+1} = \min\{k \geq \tau_i : v_k \in \{v_{\tau_i}, \ldots, v_{\tau_i+1}\}\}
\]
if the above set is nonempty, and $\tau_{i+1} = +\infty$ otherwise. We now define the stopping times $t_i$ for all $i \in \mathbb{N}$ as $t_i = \min(\tau_i, L + 1)$. Finally, for all $i \in \mathbb{N}^*$ we can consider the random variable $\xi^i$ on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in $(\Xi, \mathcal{X})$ defined by
\[
\xi^i = (v_{t_i-1}, v_{t_i-2}, \ldots, v_0).
\]

We denote by $N$ the smallest integer $n$ such that $t_n = L + 1$. We denote by $\ell(\xi^i)$ the length of the simple path $\xi^i$.

**Example.** Given a graph with vertices $V = \{a, b, c, \ldots, z\}$ and a given edge set that is not useful to describe here, consider $\omega = \Omega$ and the walk $\xi(\omega) = (c, a, e, g, a, f, a, b, h)$ with length $L = 8$. Then, $t_0(\omega) = 1$, $t_1(\omega) = 4$, $t_2(\omega) = 6$, $t_3(\omega) = 8$, $t_4(\omega) = \ldots = 9$, and $\xi(\omega) = 3$ simple paths and we have $\xi^1(\omega) = (c, a, e, g), \xi^2(\omega) = (g, a, f), \xi^3(\omega) = (f, a, b, h)$ and $\xi^4(\omega) = \ldots = \xi^8(\omega) = (h)$. Their respective lengths are $\ell(\xi^1(\omega)) = 3, \ell(\xi^2(\omega)) = 2, \ell(\xi^3(\omega)) = 3$ and $\ell(\xi^4(\omega)) = 0$ for all $i = 4, \ldots, 8$. We identify $\xi(\omega)$ with $(\xi^1(\omega), \ldots, \xi^8(\omega))$.

It is worth noting that, by construction, $\xi^i$ is a simple path. Moreover, the following statements hold:

- We have $1 \leq N \leq L$ a.s.
- These three events are equivalent for all $i$: $\{\xi^i$ is a single vertex\}, $\{(\ell(\xi^i) = 0\}$ and $\{i \geq N + 1\}$
- The last element of $\xi^N$ is a.s. $v_L$
- $\sum_{i=1}^L \ell(\xi^i) = L$ a.s.

In the sequel, we identify the random vector $(\xi^1, \ldots, \xi^L)$ with the random variable $\xi = (v_0, \ldots, v_L)$. As a result, $\xi$ is seen as a r.v with values in $\Xi^L$.

Our notations are summarized in Table I. For every $i = 1, \ldots, L$, define the functions $f_i, g_i$ on $\mathbb{R}^V \times \Xi$ in such a way that
\[
f_i(x, \xi^i) = \frac{\ell(\xi^i)}{L}\frac{F(x)}{|E|},
\]
\[
g_i(x, \xi^i) = \frac{1}{L} R(x, \phi_{\xi^i}).
\]
Note that when \( i > N(\omega) \) then \( f_i(x, \xi^i(\omega)) = g_i(x, \xi^i(\omega)) = 0 \).

**Proposition 3.** For every \( x \in \mathbb{R}^V \), we have
\[
\frac{1}{|E|} (F(x) + R(x, \phi)) = \sum_{i=1}^{L} \left[ f_i(x, \xi^i) + g_i(x, \xi^i) \right].
\]

**Proof.** For every \( \omega \in \Omega \) and every \( x \in \mathbb{R}^V \),
\[
\frac{1}{|E|} R(x, \phi_\xi(\omega)) \leq \frac{1}{|E|} F(x).
\]
Integrating, and using Prop. 2, it follows that
\[
\sum_{i=1}^{L} f_i(x, \xi^i) + g_i(x, \xi^i) = \frac{1}{|E|} F(x). \]
This completes the proof. \( \square \)

### D. Main Algorithm

Prop. 3 suggests that minimizers of Problem (1) can be found by minimizing the right-hand side of (18). This can be achieved by means of the stochastic approximation algorithm provided in Sec. III. The corresponding iterations (16) read as
\[
x_{n+1} = \text{prox}_{\gamma_n g_i} (x_n - \gamma_n \nabla f_i(x_n, \xi^i)).
\]

**Theorem 4.** Let Ass. 2 hold true. Assume that the convex function \( F \) is differentiable and that \( \nabla F \) is Lipschitz continuous. Assume that Problem (1) admits a minimizer. Then, there exists a r.v. \( X_\omega \) s.t. \( X_\omega(\omega) \) is a minimizer of (1) for all \( \omega \) \( \mathbb{P} \)-a.e., and s.t. the sequence \( (x_n) \) defined above converges a.s. to \( X_\omega \) as \( n \to \infty \). Moreover, for every \( i = 0, \ldots, L-1 \), \( \bar{x}^i_n \) converges a.s. to \( X_i \).

**Proof.** It is sufficient to verify that the mappings \( f_i, g_i \) defined by (16) and (17) respectively fulfill Ass. 1 of Th. 1. Then, Th. 1 gives the conclusion. Ass. 1 and 2 are trivially satisfied. It remains to show that for every minimizer \( x_\omega \), the existence of a \((2+\epsilon)\)-representation, for some \( \epsilon > 0 \). Any such \( x_\omega \) satisfies Eq. (12), where \( \phi_i \) satisfies (11). By definition of \( f_i \) and \( g_i \), it is straightforward to show that there exists a deterministic constant \( C_i \), depending only on \( x_\omega \), such that \( \|\nabla f_i(x_i, \xi^i)\| < C_i \), and \( \|\phi_i(\xi^i)\| < C_i \). This proves Ass. 1 of Ass. 3 can be easily checked by the same arguments. \( \square \)

Consider the general \( \phi \)-regularized problem (1), and assume that an efficient procedure to compute the proximity operator of the \( \phi \)-regularization over an 1D-graph is available. The sequence \( (x_n) \) is generated by minimizing the algorithm \( \text{Snake}(x_0, \ell) \).
that the procedure stopped because the maximum length was achieved. In that case, \( c \) represents the last simple path generated, and the algorithm doesn’t use the pivot node \( w \).

**Remark.** Although Snake converges for every value of the hyperparameter \( L \), a natural question is about the influence of \( L \) on the behavior of the algorithm. In the case where \( R(\cdot, \phi) \) is the TV regularization, \( \| x - x' \|_1 \) reduces to the Laplacian regularization that is

\[
\| x \|_L^2 = \| \nabla x \|_2^2 = x^* L x.
\]

Its unweighted version is

\[
\sum_{\{i,j\} \in E} (x(i) - x(j))^2 = \| \nabla x \|_2^2 = x^* L x.
\]

In the case where \( \phi(\{i,j\}, x, x') = w_{\{i,j\}} (x'/\sqrt{\text{deg}(i)} - x'/\sqrt{\text{deg}(j)})^2 \),

\[
R(x, \phi) = \sum_{\{i,j\} \in E} w_{\{i,j\}} \left( \frac{x(i)}{\sqrt{\text{deg}(i)}} - \frac{x'(i)}{\sqrt{\text{deg}(j)}} \right)^2
\]

is the normalized Laplacian regularization.

We now explain one method to compute the proximity operator of the unweighted Laplacian regularization over an 1D-graph. The computation of the proximity operator of the normalized Laplacian regularization can be done similarly. The computation of the proximity operator of the weighted Laplacian regularization over an 1D-graph is as fast as the computation the proximity operator of the unweighted Laplacian regularization over an 1D-graph, using for example Thomas’ algorithm.

The proximity operator of a fixed point \( y \in \mathbb{R}^{\ell+1} \) is obtained as a solution to a quadratic programming problem of the form:

\[
\min_{x \in \mathbb{R}^{\ell+1}} \frac{1}{2} \| x - y \|^2 + \lambda \sum_{k=1}^{\ell} (x(k) - y(k))^2,
\]

where \( \lambda > 0 \) is a scaling parameter. Writing the first order conditions, the solution \( x \) satisfies

\[
(I + 2 \lambda \mathcal{L}) x = y
\]

where \( \mathcal{L} \) is the Laplacian matrix of the 1D-graph with \( \ell + 1 \) nodes and \( I \) is the identity matrix in \( \mathbb{R}^{\ell+1} \). By \( \mathcal{L} \) can be diagonalized explicitly. In particular, \( I + 2 \lambda \mathcal{L} \) has eigenvalues

\[
1 + 4 \lambda \left( 1 - \cos \left( \frac{\pi k}{\ell + 1} \right) \right),
\]

and eigenvectors \( e_k \in \mathbb{R}^{\ell+1} \)

\[
e_k(j) = \frac{1}{2(\ell + 1)} \cos \left( \frac{k j}{\ell + 1} - \frac{\pi k}{2(\ell + 1)} \right),
\]

for \( 0 \leq k < n \). Hence, \( x = C^* \Lambda^{-1} C y \), where \( \Lambda \) gathers the eigenvalues of \( I + 2 \lambda \mathcal{L} \) and the operators \( C \) and \( C^* \) are the discrete cosine transform operator and the inverse discrete cosine transform. The practical computation of \( x \) can be found in \( O(\ell \log(\ell)) \) operations.

**VI. Examples**

We now give some practical instances of Problem (1) by particularizing \( F \) and the \( \phi \)-regularization in (1). We also provide some simulations to compare our method to existing algorithms.
A. Trend Filtering on Graphs

Consider a vector \( y \in \mathbb{R}^V \). The Graph Trend Filtering (GTF) estimate on \( V \) with parameter \( k \) set to one is defined in [8] by

\[
\hat{y} = \arg \min_{x \in \mathbb{R}^V} \frac{1}{2} \|x - y\|^2 + \lambda \sum_{(i,j) \in E} |x(i) - x(j)|.
\]

(20)

where \( \lambda > 0 \) is a scaling parameter. In the GTF context, the vector \( y \) represents a sample of noisy data over the graph \( G \) and the GTF estimate represents a denoised version of \( y \). When \( G \) is an 1D or a 2D-graph, the GTF boils down to a well known context [4], [7]. When \( G \) is a general graph, the GTF estimate is studied in [8] and [10]. The estimate \( \hat{y} \) is obtained as the solution of a TV-regularized risk minimization with \( F(x) = \frac{1}{2} \|x - y\|^2 \) where \( y \) is fixed. We address the problem of computing the GTF estimate on two real life graphs from [34] and one sampled graph. The first one is the Facebook graph which is a network of 4039 nodes and 88234 edges extracted from the Facebook social network. The second one is the Orkut graph with 3072441 nodes and 117185083 edges. Orkut was also an on-line social network. The third graph is sampled according to a Stochastic Block Model (SBM). Namely we generate a graph of 4000 nodes with four well-separated clusters of 1000 nodes (also called “communities”) as depicted in Fig. (2). Then we draw independently \( N^2 \) Bernoulli r.v. \( E(i,j) \), encoding the edges of the graph (an edge between nodes \( i \) and \( j \) is present if \( f(i,j) = 1 \)), such that \( P\{E(i,j) = 1\} = P(c_i, c_j) \) where \( c_i \) denotes the community of the node \( i \) and where

\[
\begin{align*}
P(c, c') &= .1 \text{ if } c = c' \\
P(c, c') &= .005 \text{ otherwise}
\end{align*}
\]

This model is called the stochastic block model for the matrix \( P \) [34]. It amounts to a blockwise Erdös-Rényi model with parameters depending only on the blocks. It leads to 81117 edges.

We assume that every node is provided with an unknown value in \( \mathbb{R} \) (the set of all these values being referred to as the signal in the sequel). In our example, the value \( y(i) \) at node \( i \) is generated as \( y(i) = l(c_i) + \sigma \epsilon_i \) where \( l \) is a mapping from the communities to a set of levels (in Fig. 2, \( l(i) \) is an integer in \( \{0, 255\} \)), and \( \epsilon \) denotes a standard Gaussian white noise with \( \sigma > 0 \) as its standard deviation. In Figure 2 we represent an example of the signal \( y \) (left figure) along with the “initial” values \( l(c_i) \) represented in grayscale at every node.

Over the two real life graphs, the vector \( y \) is sampled according to a standard Gaussian distribution of dimension \( |V| \). The parameter \( \lambda \) is set such that \( \mathbb{E}[\frac{1}{2} \|x - y\|^2] = \mathbb{E}[\lambda \sum_{(i,j) \in E} |x(i) - x(j)|] \) if \( x, y \) are two independent r.v with standardized Gaussian distribution. The initial guess \( x_0 \) is set equal to \( y \). The step size \( \gamma_n \) set equal to \( |V|/(10n) \) for the two real life graphs and \( |V|/(5n) \) for the SBM realization graph. We ran the Snake algorithm for different values of \( L \), except over the Orkut graph where \( L = |V| \).

The dual problem of (20) is quadratic with a box constraint. The Snake algorithm is compared to the well-known projected gradient (PG) algorithm for the dual problem. To solve the dual problem of (20), we use L-BFGS-B [35] as suggested in [8]. Note that, while running on the Orkut graph, the algorithm L-BFGS-B leads to a memory error from the solver [35] in SciPy (using one thread of a 2800 MHz CPU and 256GB RAM).

Figures 3 and 4 show the objective function as a function of time for each algorithm.
In the case of the TV regularization, we observe that Snake takes advantage of being an online method, which is known to be twofold (\cite{19}, \cite{20}). First, the iteration complexity is controlled even over large general graphs: the complexity of the computation of the proximity operator is empirically linear \cite{16}. On the contrary, the projected gradient algorithm involves a matrix-vector product with complexity $O(|E|)$. Hence, e.g., the projected gradient algorithm has an iteration complexity of at least $O(|E|)$. The iteration complexity of Snake can be set to be moderate in order to frequently get iterates while running the algorithm. Then, Snake is faster than L-BFGS-B and the projected gradient algorithms for the dual problem in the first iterations of the algorithms.

Moreover, for the TV regularization, Snake seems perform globally better than L-BFGS-B and the projected gradient algorithms for the dual problem in the first iterations of the algorithms.

The parameter $L$ seems to have a minor influence on the performance of the algorithm since, in Figure 4, the curves corresponding to different values of $L$ are closely superposed. The log scale in Figure 5 allows us to see that the curve corresponding Snake with $L = 1000$ performs slightly better that the others. Figure 6 shows supplementary curves in log scale where Snake is run over the Facebook graph with different values of $L$.

In Figure 6, the best performing value is $L = 500$.

Over the three graphs, the value $L = O(|V|)$ is a good value, if not the best value to use the Snake algorithm. One can show that, while sampling the first steps of the infinite random walk over $G$ from the node, say $v$, the expected time of return to the random node $v$ is $|V|$. Hence, the value $L = |V|$ allow Snake to significantly explore the graph during one iteration.

B. Graph Inpainting

The problem of graph inpainting has been studied in \cite{1}, \cite{2}, \cite{15} and can be expressed as follows. Consider a vector $y \in \mathbb{R}^{|V|}$, a subset $O \subset V$. Let $\bar{O}$ be its complementary in $V$.

The harmonic energy minimization problem is defined in \cite{2} by

$$
\min_{x \in \mathbb{R}^{|V|}} \sum_{\{i,j\} \in E} (x(i) - x(j))^2
$$

subject to $x(i) = y(i), \forall i \in O$.

This problem is interpreted as follows. The signal $y \in \mathbb{R}^{|V|}$ is partially observed over the nodes and the aim is to recover $y$ over the non observed nodes. The subset $O \subset V$ is the set of the observed nodes and $\bar{O}$ the set of unobserved nodes. An example is shown in Figure 7.

Denote by $G_\bar{O} = (\bar{O}, E_\bar{O})$ the subgraph of $G$ induced by $\bar{O}$. Namely, $\bar{O}$ is the set of vertices, and the set $E_\bar{O}$ is formed by the edges $\{i,j\} \in E$ s.t. $i \in \bar{O}$ and $j \in \bar{O}$. The harmonic energy minimization is equivalent to the following Laplacian regularized problem over the graph $G_\bar{O}$:

$$
\min_{x \in \mathbb{R}^{|\bar{O}|}} F(x) + \sum_{\{i,j\} \in E_\bar{O}} (x(i) - x(j))^2
$$

(21)

where

$$
F(x) = \sum_{i \in O, j \in O} (x(i) - y(j))^2.
$$

Fig. 5. Snake applied to the TV regularization over the SBM realization graph, in log scale

Fig. 6. Snake applied to the TV regularization over the Facebook graph, in log scale

Fig. 7. Left: Partially observed data (unobserved nodes are black, data is the color of nodes). Right: Fully observed data (the color is observed for all nodes).
The signal $y$ is sampled according to a standardized Gaussian distribution of dimension $|V|$. We compared the Snake algorithm to existing algorithm over the Orkut graph. The set $V$ is divided in two parts of equal size to define $O$ and $\bar{O}$. The initial guess is set equal to zero over the set of unobserved nodes $\bar{O}$, and to the restriction of $y$ to $O$ over the set of observed nodes $O$. We compare our algorithm with the conjugate gradient.

Figures 8 and 9 represent the objective function $\sum_{(i,j) \in E} (x(i) - x(j))^2$ as a function of time. Over the Facebook graph, the parameter $L$ is set equal to $|V|/10$. The step size $\gamma_n$ are set equal to $|V|/(10n)$. Over the Orkut graph, $L$ is set equal to $|V|/50$. The step size are set equal to $|V|/(5\sqrt{n})$ on the range displayed in Figure 8. Even if the sequence $\{5\sqrt{n}\}_{n \in \mathbb{N}}$ does not satisfy the Ass. 2, it is a standard trick in stochastic approximation to take a slow decreasing step size in the first iterations of the algorithm (37). It allows the iterates to be quickly close to the set of solutions without converging to the set of solutions. Then, one can continue the iterations using a step size satisfying Ass. 2 to make the algorithm converging. There is a trade-off between speed and precision while choosing the step-size.

In our experiment, the vector $b$ is sampled according to a standardized Gaussian distribution of dimension $|V|$. We compare our algorithm with the conjugate gradient over the Orkut graph. The set $O$ is used, the complexity of the computation of the proximity operator is $O(L \log(L))$. In contrast, the iteration complexity of the conjugate gradient algorithm can be a bottleneck (at least $O(|E|)$) as far as very large graphs are concerned.

Besides, Snake for the Laplacian regularization does not perform globally better than the conjugate gradient. This is because the conjugate gradient is designed to fully take advantage of the quadratic structure. On the contrary, Snake is not specific to quadratic problems.

C. Online Laplacian solver

Let $L$ the Laplacian of a graph $G = (V, E)$. The resolution of the equation $Lx = b$, where $b$ is a zero mean vector, has numerous applications ([18], [38]). It can be found by minimizing the Laplacian regularized problem

$$\min_{x \in \mathbb{R}^V} \frac{1}{2} x^* L x - b^* x + \frac{1}{2} \sigma \|L x - b\|^2.$$ 

In our experiment, the vector $b$ is sampled according to a standardized Gaussian distribution of dimension $|V|$. We compare our algorithm with the conjugate gradient over the Orkut graph.

Figure 10 represents the quantity $\|L x_n - b\|$ as a function of time, where $x_n$ is the iterate provided either by Snake or by the conjugate gradient method. The parameter $L$ is set equal to $|V|$. The step size $\gamma_n$ are set equal to $|V|/(2n)$. Snake appears to be more stable than the conjugate gradient method, has a better performance at start up.

VII. PROOF OF TH. [1]

We start with some notations. We endow the probability space $(\Omega, \mathcal{F}, P)$ with the filtration $(\mathcal{F}_n)$ defined as $\mathcal{F}_n = \sigma(\xi_1, \ldots, \xi_n)$, and we write $E_n = E[\cdot | \mathcal{F}_n]$. In particular, $E_0 = E$. We also define $G_i(x) = E[g_i(x, \xi)]$ and $F_i(x) = E[f_i(x, \xi)]$ for every $x \in X$. We denote by $\mu_i$ and $\mu$ the probability laws of $\xi$ and $\xi$ respectively. Finally, $C$ and $\eta$ will refer to positive constants whose values can change from an
equation to another. The constant $\eta$ can be chosen arbitrarily small.

In [21], the case $L = 1$ is studied (algorithm 3). Here we shall reproduce the main steps of the approach of [21], only treating in detail the specificities of the case $L \geq 1$. We also note that in [21], we considered the so-called differential inclusions (DI) defined on the set of absolutely continuous functions from $\mathbb{R}_+ = [0, \infty)$ to $X$ as follows:

$$
\begin{cases}
  \dot{z}(t) & \in - \sum_{i=1}^L (\nabla F_i(z(t)) + \partial G_i(z(t))) \\
  z(0) & = a.
\end{cases}
$$

It is well known that this DI has a unique solution, i.e., a unique absolutely continuous mapping $z : \mathbb{R}_+ \to X$ such that $z(0) = a$, and $\dot{z}(t) \in - \sum(\nabla F_i(z(t)) + \partial G_i(z(t)))$ for almost all $t > 0$.

Consider now the map $\Phi : X \times \mathbb{R}_+ \to X$, $(a, t) \mapsto z(t)$, where $z(t)$ is the DI solution with the initial value $z(0) = a$. Then, $\Phi$ is a semi-flow [39, 40].

Let us introduce the following function $f$ from $X^N$ to the space of $\mathbb{R}_+ \to X$ continuous functions. For $u = (u_n) \in X^N$, the function $f$ is the continuous interpolated process obtained from $u$ as

$$u(t) = u_n + \frac{u_{n+1} - u_n}{\gamma_{n+1}} (t - \tau_n)$$

for $t \in \left[\tau_n, \tau_{n+1}\right)$, where $\tau_n = \sum_n^{n+1} \gamma_k$. Consider the interpolated function $x = f((x_n))$. We shall prove the following facts:

- The sequence $(\|x_n - x_s\|)$ is almost sure convergent for each $x_s \in Z$ (Prop. 3);
- The process $x(t)$ is an almost sure Asymptotic Pseudo Trajectory (APT) of the semi-flow $\Phi$, a concept introduced by Benaim and Hirsch in the field of dynamical systems [11]. Namely, for each $t > 0$,

$$\sup_{u \in [0, T]} \|x(t + u) - \Phi(x(t), u)\| \xrightarrow{a.s.} 0,$$

(23)

Taken together, these two results lead to the a.s. convergence of $(x_n)$ to some r.v. $X^*$ supported by the set $Z$, as is shown by [21, Cor. 3.2]. The convergence of the $(\bar{x}_n)$ stated by Th. [1] will be shown in the course of the proof.

Before entering the proof, we recall some well known facts relative to the so called Moreau envelopes. For more details, the reader is referred to e.g. [39, Ch. 2], or [36, Ch. 12]. The Moreau envelope of parameter $\gamma$ of a convex function $h$ with domain $X$ is the function

$$h^\gamma(x) = \min_{w \in X} h(w) + (2\gamma)^{-1} \|w - x\|^2.$$

The function $h^\gamma$ is a differentiable function on $X$, and its gradient is given by the equation

$$\nabla h^\gamma(x) = \gamma^{-1}(x - \text{prox}_{\gamma h}(x)).$$

(24)

This gradient is a $\gamma^{-1}$-Lipschitz continuous function satisfying the inequality $\|\nabla h^\gamma(x)\| \leq \|\partial h^0(x)\|$, where $\partial h^0(x)$ is the least-norm element of $\partial h(x)$. Finally, for all $(x, u) \in X \times X$ and for all $v \in \partial h(u)$, the inequality

$$\langle \nabla h^\gamma(x) - v, \text{prox}_{\gamma h}(x) - u \rangle \geq 0$$

(25)

holds true. With the formalism of the Moreau envelopes, the mapping $T_{\gamma, i}$ can be rewritten as

$$T_{\gamma, i}(x, s) = x - \gamma \nabla f_i(x, s) - \gamma \nabla g_i^\gamma(x - \gamma \nabla f_i(x, s), s)$$

(26)

thanks to (24), where $\nabla g_i^\gamma(\cdot, s)$ is the gradient of the Moreau envelope $g_i^\gamma(\cdot, s)$. We shall adopt this form in the reminder of the proof.

The following lemma is proven in Appendix A-A

**Lemma 5.** For $i = 1, \ldots, L$, let

$$x_i = (T_{\gamma, i}(\cdot, s^i) \circ \cdots \circ T_{\gamma, 1}(\cdot, s^1))(x).$$

Then, with Ass. 3, there exists a measurable map $\kappa : \Xi^L \to \mathbb{R}_+$ s.t. $\mathbb{E}[\kappa(\xi)^{\alpha}] < \infty$ for all $\alpha \geq 1$ and s.t. for all $s = (s^1, \ldots, s^L) \in \Xi^L$,

$$\|\nabla f_i(x_i^{-1}, s^i)\| \leq \kappa(s) \sum_{k=1}^i \|\nabla f_k(x, s^k)\| + \|\nabla g_i^\gamma(x_i^{-1}, s^i, s^i)\|$$

$$\leq \kappa(s) \sum_{k=1}^i \|\nabla f_k(x, s^k)\| + \|\nabla g_i^\gamma(x, s^k)\|.$$

Recall that we are studying the iterations $\bar{x}_{n+1} = T_{\gamma, n+1}(\bar{x}_{n+1}, \xi_{n+1})$, for $i = 1, \ldots, L$, $n \in \mathbb{N}^*$, with $\bar{x}_n = x_n$ and $x_n+1 = \bar{x}_{n+1}$. In this section and in Appendix A, we shall write for conciseness, for any $x_n \in Z$,

$$\nabla g_i^\gamma = \nabla g_i^{\gamma_{n+1}}(\bar{x}_{n+1}^{-1} - \gamma_{n+1} \nabla f_i(\bar{x}_{n+1}^{-1} + \xi_{n+1}^{-1} + \xi_{n+1}^{-1}), \xi_{n+1}^{-1}),$$

$$\text{prox}_{\gamma_{n+1}}(\cdot, \xi_{n+1}^{-1})$$

$$\nabla f_i = \nabla f_i(\bar{x}_{n+1}^{-1} + \xi_{n+1}^{-1}),$$

$$\varphi_i = \varphi_i(\xi_{n+1}^{-1}),$$

(see Ass. 4) and

$$\gamma = \gamma_{n+1}.$$
and let us also define the function
\[ H_{\gamma}(x, (s_1, \ldots, s^k)) = \sum_{i=1}^{L} \left[ \nabla f_i(x^{i-1}, s^i) + \nabla g_i^{\gamma}(x^{i-1}) - \gamma \nabla f_i(x^{i-1}, s^i), s^i \right] \]
where we recall the notation \( x^i = (T_{\gamma_1}(s^i) \circ \cdots \circ T_{\gamma_1}(s^1))(x) \). By Lem. 5 and Hölder inequality, and using the fact that \( \gamma(x_\gamma + \cdot) \) is equicontinuous and bounded with probability one.

In order to characterize its cluster points, choose \( T > 0 \), and consider an elementary event on the probability one set where \( x \) is equicontinuous and bounded on \([0, T]\). With a small notational abuse, let \( n \) be a subsequence along which \( (x(\tau_n + \cdot))_n \) converges on \([0, T]\) to some continuous function \( z(t) \). This function then is written as
\[
z(t) - z(0) = \lim_{n \to \infty} \int_0^t \mu(ds) H_{\gamma(x_\gamma + \cdot)}(x(\tau_n + u), s).
\]
By the boundedness of \((x_n)\) (Prop. 6–1), Lem. 5, and Ass. 3 and 5 the sequence of functions \( (H_{\gamma(x_\gamma + \cdot)}(x(\tau_n + u), s))_n \) in the parameters \((u, s)\) is bounded in the Banach space \( L^{1+\varepsilon}(du \otimes \mu) \), for some \( \varepsilon > 0 \), where \( du \) is the Lebesgue measure on \([0, T]\). Since the unit ball of \( L^{1+\varepsilon}(du \otimes \mu) \) is weakly compact in this space by the Banach-Alaoglu theorem, since this space is reflexive, we can extract a subsequence (still denoted as \((n)\)) such that \( H_{\gamma(x_\gamma + \cdot)}(x(\tau_n + u), s) \) converges weakly in \( L^{1+\varepsilon}(du \otimes \mu) \), as \( n \to \infty \), to a function \( Q(u, s) \).

The remainder of the proof consists in showing that \( Q \) can be written as
\[
Q(u, s) = \sum_{i=1}^{L} (b_i(u, s^i) + p_i(u, s^i)),
\]
where \( b_i(u, s^i) = \nabla f_i(z(u), s^i) \) and \( p_i(u, s^i) \in \partial g_i(z(u), s^i) \) for \( du \otimes \mu_i \)-almost all \((u, s^i)\). Indeed, once this result is established, it becomes clear that \( z(t) \) is an absolutely continuous function whose derivative satisfies almost everywhere the inclusion \( \dot{z}(t) \in -\sum_i (\nabla F_i(z(t)) + \partial G_i(z(t))) \), noting that we can exchange the integration and the differentiation (resp. the subdifferential) in the expression of \( \nabla F \) (resp. of \( \partial G \)).

We just provide here the main argument of this part of the proof, since it is similar to its analogue in [21]. Let \( i \in \{1, \ldots, L\} \). Let us focus on the sequence of functions \((u, s) \in [0, T] \times \mathbb{X} \) defined by
\[
\nabla g_{\gamma(x_\gamma + \cdot)}(x(\tau_n + u) + \gamma) \nabla f_i(x(\tau_n + u) + \gamma, s),
\]
and indexed by \( n \). This sequence is bounded in \( L^{1+\varepsilon}(du \otimes \mu_i) \) on a probability one set, as a function of \((u, s)\), for the same reasons as those explained above for \((H_{\gamma(x_\gamma + \cdot)}(x(\tau_n + u), s))_n \).

We need to show that any weak limit point \( p_i(u, s) \) of this sequence satisfies \( p_i(u, s) \in \partial g_i(z(u), s) \) for \( du \otimes \mu_i \)-almost all \((u, s)\). Using the fact that \( x(\tau_n + \cdot) \to z(\cdot) \) almost surely, along with the inequality \( \langle \nabla g_i^{\gamma}(x, s) \rangle (x) - w, \gamma_{\gamma_i}(\cdot, s) \rangle (x) \geq 0 \), valid for all \( x, v \in \mathbb{X} \) and \( w \in \partial g_i(v, s) \), we show that \( p_i(u, s) - w, z(u) - v \rangle = 0 \) for \( du \otimes \mu_i \)-almost all \((u, s)\).

Thus by Lem. 5 and Hölder inequality, and using the fact that the sequence \((x_n)\) is almost surely bounded by Prop. 6–1, it can be shown that
\[
\sup_n \| h_{\gamma_\gamma + \cdot} (x_n) \| < \infty \quad \text{w. p. 1}.
\]

\section{Conclusion}
A fast regularized optimization algorithm over large unstructured graphs was introduced in this paper. This algorithm is a variant of the proximal gradient algorithm that operates on randomly chosen simple paths. It belongs to the family of stochastic approximation algorithms with a decreasing step size.
Appendix A
Proofs for Sec. VII

A. Proof of Lem. 3

We start by writing $\|\nabla f_i(\bar{x}^{i-1}, s^i)\| \leq \|\nabla f_i(\bar{x}^{i-2}, s^i)\| + K_i(s^i)\|\bar{x}^{i-1} - \bar{x}^{i-2}\|$, where $K_i(s^i)$ is provided by Ass. 3.

Using the identity $\bar{x}^{i-1} = \Theta_{\gamma, i-1}(\bar{x}^{i-2})$, where $\Theta_{\gamma, i}$ is given by (26), and recalling that $\nabla g_i(\cdot, s^i)$ is $\gamma^{-1}$-Lipschitz, we get

$$\|\nabla f_i(\bar{x}^{i-1}, s^i)\| \leq \|\nabla f_i(\bar{x}^{i-2}, s^i)\| + \gamma K_i(s^i)(2\|\nabla f_{i-1}(\bar{x}^{i-2}, s^{i-1})\| + \|\nabla g_{i-1}^2(\bar{x}^{i-2}, s^{i-1})\|).$$

Similarly,

$$\|\nabla g_i^2(\bar{x}^{i-1} - \gamma \nabla f_i(\bar{x}^{i-1}, s^i), s^i)\| \leq \|\nabla f_i(\bar{x}^{i-1}, s^i)\| + 2\|\nabla f_{i-1}(\bar{x}^{i-2}, s^{i-1})\| + \|\nabla g_{i-1}^2(\bar{x}^{i-2}, s^{i-1})\|.$$

Iterating down to $\bar{x}^0 = x$, we get the result since for every $i$, $K_i(\cdot)$ admits all their moments.

B. Proof of Prop. 4

Let $x_n$ be an arbitrary element of $Z$. Let $i \in \{1, \ldots, L\}$.

We start by writing

$$\|x_{n+1}^i - x_s\|^2 = \|x_{n+1}^i - \bar{x}^{i-1}_{n+1}\|^2 + \|\bar{x}^{i-1}_{n+1} - x_s\|^2 + 2\langle \bar{x}^{i-1}_{n+1} - x^{i-1}_{n+1}, x^i_{n+1} - x_s\rangle$$

$$= \|x_{n+1}^i - x_s\|^2 + \gamma^2 \|\nabla f_i + \nabla g_i\|^2 - 2\gamma \langle \nabla f_i, \nabla g_i, x_{n+1}^i - x_s\rangle - 2\gamma \langle \nabla g_i^2, -\varphi_i, x_{n+1}^i - x_s\rangle$$

$$= \|x_{n+1}^i - x_s\|^2 + A_1 + A_2 + A_3 + A_4.$$

Most of the proof consists in bounding the $A_i$’s. For an arbitrary $\beta > 0$, it holds that $\langle a, b \rangle \leq (\beta/2)\|a\|^2 + \|b\|^2/(2\beta)$. We rewrite this inequality as $\langle a, b \rangle \leq \eta\|a\|^2 + C\|b\|^2$, where $\eta > 0$ is a constant chosen as small as desired, and $C > 0$ is fixed accordingly. Similarly, $\|a + b\|^2 \leq (1 + \eta)\|a\|^2 + C\|b\|^2$ where $\eta > 0$ is as small as desired. We shall repeatedly use these inequalities without mention. Similarly to $C$, the value of $\eta$ can change from a line of calculation to another.

Starting with $A_1$, we have

$$A_1 \leq \gamma^2(1 + \eta)\|\nabla g_i^2\|^2 + C\gamma^2\|\nabla f_i\|^2.$$

We have $A_2 \leq 0$ by the convexity of $f_L$. We can write

$$A_3 = -2\gamma\langle \nabla g_i^2, -\varphi_i, \text{prox}_{\gamma g_i} - x_s\rangle - 2\gamma\langle \nabla g_i^2, -\varphi_i, \bar{x}^{i-1}_{n+1} - \gamma \nabla f_i - \text{prox}_{\gamma g_i}\rangle - 2\gamma\langle \nabla g_i^2, -\varphi_i, \nabla f_i \rangle.$$

By (25), the first term at the right hand side is $\leq 0$. By (24),

$$A_3 \leq -2\gamma^2\|\nabla g_i^2\|^2 + 2\gamma \langle \varphi_i, \nabla g_i^2 + \nabla f_i \rangle - 2\gamma \langle \nabla g_i^2, \nabla f_i \rangle \leq -(2 - \eta)\gamma^2\|\nabla g_i^2\|^2 + C\gamma^2\|\nabla f_i\|^2 + C\gamma^2\|\varphi_i\|^2.$$

As regards $A_4$, we have

$$A_4 = -2\gamma\langle \nabla f_i^* + \varphi_i, x_n - x_s\rangle - 2\gamma\langle \nabla f_i^* + \varphi_i, \bar{x}_{n+1}^{i-1} - x_n\rangle.$$

Gathering these inequalities, we get

$$\|x_{n+1}^i - x_s\|^2 \leq \|x_n - x_s\|^2 - (1 - \eta)\gamma^2\|\nabla g_i^2\|^2 + C\gamma^2\|\nabla f_i\|^2 + C\gamma^2\|\varphi_i\|^2 - 2\gamma\langle \nabla f_i^* + \varphi_i, x_n - x_s\rangle - 2\gamma\langle \nabla f_i^* + \varphi_i, \bar{x}_{n+1}^{i-1} - x_n\rangle.$$

Iterating over $i$, we get

$$\|x_{n+1}^i - x_s\|^2 \leq \|x_n - x_s\|^2 - (1 - \eta)\gamma^2\sum_{k=1}^i \|\nabla g_k^2\|^2 + C\gamma^2\sum_{k=1}^i \|\varphi_k\|^2 - 2\gamma\sum_{k=1}^i \langle \nabla f_k^* + \varphi_k, x_n - x_s\rangle - 2\gamma\sum_{k=1}^i \langle \nabla f_k^* + \varphi_k, \bar{x}_{n+1}^{k-1} - x_n\rangle.$$

The summand in the last term can be written as

$$-2\gamma\sum_{k=1}^{i-1} \langle \nabla f_k^* + \varphi_k, x_{n+1}^{k-1} - x_{n+1}^i\rangle = -2\gamma\sum_{k=1}^{i-1} \langle \nabla f_k^* + \varphi_k, x_{n+1}^{k-1} - x_{n+1}^i\rangle = -2\gamma \sum_{k=1}^{i-1} \langle \nabla f_k^* + \varphi_k, \nabla f_k + \nabla g_k^2\rangle \leq \gamma^2 C\|\nabla f_k^*\|^2 + \gamma^2 C\|\varphi_k\|^2 + \gamma^2 C\sum_{k=1}^{i-1} \|\nabla f_k^*\|^2 + \gamma^2 C\sum_{k=1}^{i-1} \|\nabla g_k^2\|^2.$$

where we used $\langle a, b \rangle \leq \eta\|a\|^2 + C\|b\|^2$ as above. Therefore, for all $i = 1, \ldots, L$,

$$\|x_{n+1}^i - x_s\|^2 \leq \|x_n - x_s\|^2 - (1 - \eta)\gamma^2\sum_{k=1}^i \|\nabla g_k^2\|^2 + C\gamma^2\sum_{k=1}^i \|\varphi_k\|^2 + C\gamma^2\sum_{k=1}^i \|\nabla f_k^*\|^2 - 2\gamma\sum_{k=1}^i \langle \nabla f_k^* + \varphi_k, x_n - x_s\rangle - 2\gamma\sum_{k=1}^i \langle \nabla f_k^* + \varphi_k, \bar{x}_{n+1}^{k-1} - x_n\rangle.$$

(28)
We consider the case \( i = L \). Using Ass. 4
\[
\mathbb{E}_n \left[ \| x_{n+1}^L - x_* \|^2 \right] \leq \| x_n - x_* \|^2 - (1 - \eta) \gamma^2 \mathbb{E}_n \left[ \sum_{k=1}^L \| \nabla g_k \|^2 \right]
\]
\[
+ C \gamma^2 \sum_{k=1}^L \mathbb{E}_n[\| \nabla f_k \|^2] + 2 \mathbb{E}_n \left[ \sum_{k=1}^L \nabla f_k^\prime + \varphi_k, x_n - x_* \right].
\]

The last term at the right hand side is zero since
\[
\mathbb{E}_n \left[ \sum_{k=1}^L \nabla f_k^\prime + \varphi_k, x_n - x_* \right] = 0
\]
by definition of \( \nabla f_k^\prime \) and \( \varphi_k \). Besides, using Ass. 3 for all \( k \) we have
\[
\mathbb{E}_n[\| \nabla f_k \|^2] \leq C \mathbb{E}_n[\| \nabla f_k^\prime \|^2] + C \mathbb{E}_n[K^2_k(\xi_{n+1})]\| x_{n+1}^k - x_* \|^2].
\]

Then,
\[
\mathbb{E}_n[\| x_{n+1} - x_* \|^2] \leq \| x_n - x_* \|^2 + C \gamma^2
\]
\[
- (1 - \eta) \gamma^2 \mathbb{E}_n \left[ \sum_{k=1}^L \| \nabla g_k \|^2 \right] + C \gamma^2 \sum_{k=1}^L \mathbb{E}_n[\| K^2_k(\xi_{n+1}) \| x_{n+1}^k - x_* \|^2].
\]

We shall prove by induction that for all r.v \( P_k \) which is a monomial expression of the r.v \( K^2_k(\xi_{n+1}) \), there exists \( C > 0 \) such that
\[
\mathbb{E}_n \left[ P_k \| x_{n+1}^k - x_* \|^2 \right] \leq C(1 + \| x_n - x_* \|^2),
\]
for all \( k = 1, \ldots, L \). Note that such a r.v \( P_k \) is independent of \( \mathcal{F}_n \), non-negative and for all \( \alpha > 0 \), \( \mathbb{E}[P_k^\alpha] < \infty \) by Ass. 5.

Using Ass. 3 the induction hypothesis (30) is satisfied if \( k = 1 \). Assume that it holds true until the step \( k \) - 1 for some \( k \leq L \). Using (28) and Ass. 3
\[
\mathbb{E}_n \left[ P_k \| x_{n+1}^k - x_* \|^2 \right] \leq C \| x_n - x_* \|^2
\]
\[
+ C \gamma^2 \mathbb{E}_n \left[ P_k \sum_{\ell=1}^{k-1} \| \nabla f_\ell \|^2 \right] + C \sum_{\ell=1}^{k-1} \mathbb{E}_n[\| \nabla f_\ell \|^2] + 2 \mathbb{E}_n \left[ \sum_{\ell=1}^{k-1} \nabla f_\ell^\prime + \varphi_\ell, x_n - x_* \right].
\]

The last term at the right hand side can be bounded as
\[
- 2\gamma \mathbb{E}_n P_k \left( \sum_{\ell=1}^{k-1} \nabla f_\ell^\prime + \varphi_\ell, x_n - x_* \right)
\]
\[
\leq C \| x_n - x_* \|^2 + C \mathbb{E}_n \left[ P_k \sum_{\ell=1}^{k-1} \| \nabla f_\ell \|^2 \right] + C \sum_{\ell=1}^{k-1} \| \varphi_\ell \|^2
\]
\[
\leq C \| x_n - x_* \|^2 + C
\]
(32)

using Hölder inequality and Ass. 4.

For all \( \ell = 1, \ldots, k - 1 \),
\[
\mathbb{E}_n [P_k \| \nabla f_\ell \|^2] \leq C \mathbb{E}_n[P_k \| \nabla f_\ell \|^2]
\]
\[
+ C \mathbb{E}_n \left[ P_k K^2_k(\xi_{n+1}) \| x_{n+1}^k - x_* \|^2 \right] \leq C(1 + \| x_n - x_* \|^2)
\]

where we used Hölder inequality and Ass. 4.

For the first term at the right hand side and the induction hypothesis (30) at the step \( \ell \) with the r.v \( P_k := P_k K^2_k(\xi_{n+1}) \) for the second term.

Plugging (32) and (33) into (31) and using again Hölder inequality and Ass. 4, we find that (30) holds true at the step \( k \). Hence (30) holds true for all \( k = 1, \ldots, L \). Finally, plugging (30) into (29) with \( P_k = K^2_k(\xi_{n+1}) \) for all \( k = 1, \ldots, L \) we get
\[
\mathbb{E}_n[\| x_{n+1} - x_* \|^2] \leq (1 + C \gamma^2) \| x_n - x_* \|^2 + C \gamma^2
\]
\[
- (1 - \eta) \gamma^2 \mathbb{E}_n \left[ \sum_{k=1}^L \| \nabla g_k \|^2 \right].
\]

By the Robbins-Siegmund lemma [42], used along with \( (\gamma_n) \in \ell^2 \), we get that \( (\| x_n - x_* \|) \) converges almost surely, showing the first point.

By taking the expectations at both sides of this inequality, we also obtain that \( (\mathbb{E}[\| x_n - x_* \|^2]) \) converges, \( \sup_n \mathbb{E}[\| x_n - x_* \|^2] < \infty \), and \( \mathbb{E}\sum_{n=1}^\infty \sum_{k=1}^L \| \nabla g_k \|^2 < \infty \).

As \( \sup_n \mathbb{E}[\| x_n - x_* \|^2] < \infty \), we have by Ass. 3 that \( \sup_n \mathbb{E}[\| \nabla f_\ell \|^2] < \infty \). Using Lem. 3 and iterating, we easily get that \( \mathbb{E}\sum_{n=1}^\infty \gamma_n \sum_{k=1}^L \| \nabla f_\ell \|^2 < \infty \) for all \( i \).

Since \( \| x_{n+1}^i - x_n \| \leq \gamma \| \nabla f_\ell \| + \gamma \| \nabla g_i \| \), we get that \( \sum_{n=1}^\infty \mathbb{E}[\| x_{n+1}^i - x_n \|^2] < \infty \). By Borel-Cantelli’s lemma, we get that \( x_{n+1}^i - x_n \rightarrow 0 \) almost surely. The almost sure convergence of \( x_{n+1} - x_n \) to zero is shown similarly, and the proof of Prop. 6 is concluded.

### References

[1] A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan, “Asymptotic behavior of \( \ell_p \)-based Laplacian regularization in semi-supervised learning,” in COLT, 2016, pp. 879–906.

[2] X. Zhu, Z. Ghahramani, and J. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in ICML, 2003, pp. 912–919.

[3] D. Hallac, J. Leskovec, and S. Boyd, “Network lasso: Clustering and optimization in large graphs,” in SIGKDD, 2015, pp. 387–396.

[4] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, “An introduction to total variation for image analysis,” Theoretical foundations and numerical methods for sparse recovery, vol. 9, pp. 263–340, 2010.

[5] W. Hinterberger, M. Hintermüller, K. Kunisch, M. Von Oehsen, and O. Scherzer, “Tube methods for bv regularization,” Journal of Mathematical Imaging and Vision, vol. 19, no. 3, pp. 219–235, 2003.

[6] Z. Harchaoui and C. Lévy-Leduc, “Multiple change-point estimation with a total variation penalty,” Journal of the American Statistical Association, 2012.

[7] R. J. Tibshirani, “Adaptive piecewise polynomial estimation via trend filtering,” The Annals of Statistics, vol. 42, no. 1, pp. 285–323, 2014.
