

Asynchronous Nonlinear Sheaf Diffusion for Multi-Agent Coordination

Yichen Zhao*, Tyler Hanks*, Hans Riess*, Samuel Cohen, Matthew Hale, James Fairbanks

Abstract—Cellular sheaves and sheaf Laplacians provide a far-reaching generalization of graphs and graph Laplacians, resulting in a wide array of applications ranging from machine learning to multi-agent control. In the context of multi-agent systems, so called coordination sheaves provide a unifying formalism that models heterogeneous agents and coordination goals over undirected communication topologies, and applying sheaf diffusion drives agents to achieve their coordination goals. Existing literature on sheaf diffusion assumes that agents can communicate and compute updates synchronously, which is an unrealistic assumption in many scenarios where communication delays or heterogeneous agents with different compute capabilities cause disagreement among agents. To address these challenges, we introduce asynchronous nonlinear sheaf diffusion. Specifically, we show that under mild assumptions on the coordination sheaf and bounded delays in communication and computation, nonlinear sheaf diffusion converges to a minimizer of the Dirichlet energy of the coordination sheaf at a linear rate proportional to the delay bound. We further show that this linear convergence is attained from arbitrary initial conditions and the analysis depends on the spectrum of the sheaf Laplacian in a manner that generalizes the standard graph Laplacian case. We provide several numerical simulations to validate our theoretical results.

I. INTRODUCTION

Multi-agent coordination is a fundamental problem in control and optimization [1]. A key difficulty arises from the heterogeneity of systems, where agents may have different capabilities, state spaces, or assigned roles [2]. Further challenges arise in operational environments where communication is limited, such as wireless communication among low-flying UAVs [3] (unmanned aerial vehicles) or communication among UUVs [4] (unmanned undersea vehicles). In many of these environments, agents operate asynchronously, and they may both (a) generate information at different times and rates, say, due to different clock frequencies [5], and (b) communicate information at different times and rates,

*These authors contributed equally.

Tyler Hanks and Samuel Cohen are with the Department of Computer and Information Science and Engineering and James Fairbanks is with the Department of Mechanical and Aerospace Engineering, University of Florida, emails: {t.hanks, samuel.cohen, fairbanksj}@ufl.edu. Yichen Zhao, Hans Riess, and Matthew Hale are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, emails: {yzhao654, riess, mhale30}@gatech.edu

Riess was supported by DARPA under grant HR00112530235. Hanks was supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1842473. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. Hanks, Cohen, Hale, and Fairbanks were supported by DARPA grant HR00112220038. Hanks and Fairbanks were also partially supported by ONR grant N00014-23-1-2339. Hale and Zhao were also supported by AFOSR under grants FA9550-19-1-0169 and FA9550-23-1-0120.

due to environmental factors causing unexpected latency and throughput limitations [6].

There exists a wide range of networked multi-agent system models and applications, and this breadth has led to a large body of literature in which similar problems may require entirely new analyses to account for their particular features. To this end, the recently introduced unifying formalism of *coordination sheaves* [7] subsumes many existing graph-based coordination problems [8]. Roughly speaking, a coordination sheaf is a flexible data structure built upon a communication graph that assigns different spaces to both agents (nodes) and their interactions (edges). The spaces attached to nodes model the states or decision variables of the agents. The spaces attached to edges model measurements or communications between pairs of agents. This structure allows agents to have different goals and even for a single agent to pursue different goals with different neighbors capturing general forms of agreement and disagreement.

Existing algorithms for coordination sheaves rely on synchronous computation and communication to produce new iterates. This degree of synchronization is not always possible in practical applications as agents may compute new iterates and communicate them asynchronously leading to delays in when information is generated and shared among the agents [9]. Such delays cause agents to use outdated and inconsistent values of other agents' iterates in their computations. To address these challenges in the context of asynchronous multi-agent coordination, our primary contribution in this paper is to bring the coordination sheaves framework into the asynchronous setting.

Specifically, we develop an asynchronous multi-agent coordination algorithm for finding the minima of a Dirichlet energy function whose gradient is a nonlinear sheaf Laplacian [10], a generalization of the graph Laplacian to the sheaf-theoretic setting. We show that, under mild conditions on agents' coordination sheaf, a Laplacian-based sheaf diffusion update law can be applied asynchronously with assured convergence to the global minimizer of their Dirichlet energy. The form of asynchrony we consider allows agents to have bounded delays (where the bound can be arbitrarily large), which is sometimes called "partial asynchrony" in the literature [9].

Summary of Contributions

- We develop a partially asynchronous algorithm for sheaf diffusion (Algorithm 1).
- We prove that partially asynchronous nonlinear sheaf diffusion converges to the global minimizer with at least

a periodic linear convergence rate under mild conditions (Theorem 1).

- We provide numerical simulations demonstrating this convergence rate (Section V).

Related Work

Cellular sheaves originate in algebraic topology [11]–[13] and have recently emerged as a powerful modeling tool in systems engineering. Their ability to model local-to-global phenomena has found rich applications in opinion dynamics [10], gossip [14], mechanism design [15], circuit design [16], and graphic statics [17], just to name a few disciplines. In machine learning, sheaves and their Laplacians have served as effective message-passing operators for supervised learning on graphs [18]–[21]. Our definition of “coordination sheaves” is heavily influenced by seminal work on homological programming [7], [22], a paradigm integrating distributed optimization and homological algebra. Despite the interest in asynchronous algorithms for machine learning [23], existing literature on sheaf diffusion has been confined to the synchronous setting. To the best of our knowledge, this paper is the first to develop and analyze asynchronous sheaf diffusion, including over coordination sheaves.

Outline

The rest of this paper is organized as follows. Section II provides preliminaries on cellular sheaves and the problem statement. Section III develops an asynchronous sheaf diffusion algorithm. Section IV analyzes the convergence properties. Section V applies the sheaf diffusion algorithm in various asynchronous and synchronous settings. Section VI concludes the paper, after a brief discussion. (For notational conventions, please refer to Table I.)

Notation	Meaning
$\sigma_{\max}(\cdot)$	The largest singular value of an operator.
$\sigma_2(\cdot)$	The smallest nonzero singular value of an operator.
$\lambda_{\max}(\cdot)$	The largest eigenvalue of an operator.
$\lambda_2(\cdot)$	The smallest nonzero eigenvalue of an operator.
$ \cdot $	The cardinality of a set.
$\ \cdot\ $	The 2-norm (Euclidean norm or ℓ_2 -norm).
$A^\top$	The transpose of a matrix A .
A^+	The Moore-Penrose pseudoinverse of a matrix.
∇f	The gradient of a function f .

TABLE I: Nomenclature.

II. PRELIMINARIES & PROBLEM FORMULATION

In this section we first introduce coordination sheaves as a model for multi-agent systems and coordination goals. We then formulate the problem of achieving these coordination goals asynchronously.

A. Cellular Sheaf Background

We review the fundamental definitions of cellular sheaf theory (see [12], [24]–[26]) in order to specify general classes of coordination problems over the communication graph of a system. Recall an undirected graph consists of a pair $G = (V, E)$ where V is a set of nodes and $E \subseteq V \times V$ is a set of edges. An edge between nodes i and j is denoted

by an unordered pair of concatenated indices $ij = ji \in E$, and the neighbors of a given node is denoted $N_i := \{j \in V \mid ij \in E\}$.

Definition 1. Given an undirected graph $G = (V, E)$, a (**Euclidean**) **cellular sheaf** $\mathcal{F}$ over G is a data structure that assigns:

- A Euclidean space $\mathcal{F}(i)$ and inner product $\langle -, - \rangle_{\mathcal{F}(i)}$ to every node $i \in V$ (**vertex stalk**).
- A Euclidean space $\mathcal{F}(ij)$ and inner product $\langle -, - \rangle_{\mathcal{F}(ij)}$ to every edge $ij \in E$ (**edge stalk**).
- A linear map $\mathcal{F}_{i \leq ij} : \mathcal{F}(i) \rightarrow \mathcal{F}(ij)$ for every $i \in V$, $j \in N_i$ (**restriction map**).

Cellular sheaves, thus, attach spaces of various dimensions to a graph with restriction maps tethering these spaces together along edges as illustrated in Figure 1. Consider the following familiar example.

Example 1 (Constant Sheaf). Given a graph G and a dimension n , the constant sheaf, denoted $\mathbb{R}^n$, assigns the vector space $\mathbb{R}^n$ to every vertex and edge stalk, and identity maps $\mathbb{R}_{i \leq ij}^n = \text{id}_{\mathbb{R}^n}$ for every $i \in V$, $j \in N_i$.

We refer to a particular assignment $x_i \in \mathcal{F}(i)$ as a **local section**, and interpret this as a state for agent i . Edge stalks are interpreted as the communication or interaction space between agents. Collecting the data over all nodes with a direct sum of vector spaces yields the space of 0-cochains, $C^0(G; \mathcal{F}) = \bigoplus_{i \in V} \mathcal{F}(i)$, and data over edges is similarly gathered in the space of 1-cochains, $C^1(G; \mathcal{F}) = \bigoplus_{e \in E} \mathcal{F}(e)$. The total space $C^0(G; \mathcal{F})$ then corresponds to the global state space and $C^1(G; \mathcal{F})$ corresponds to the global communication space. These spaces, known in algebraic topology as cochain groups, are endowed with the inner products

$$\begin{aligned} \langle \mathbf{x}, \mathbf{x}' \rangle_{C^0(G; \mathcal{F})} &= \sum_{i \in V} \langle x_i, x'_i \rangle_{\mathcal{F}(i)} \\ \langle \mathbf{y}, \mathbf{y}' \rangle_{C^1(G; \mathcal{F})} &= \sum_{e \in E} \langle y_e, y'_e \rangle_{\mathcal{F}(e)} \end{aligned}$$

Norms are induced by these inner products in the usual way.

Remark 1. Our definition of Euclidean cellular sheaves permits arbitrary inner products to be assigned to stalks, rather than just the standard ℓ_2 inner product because of the interpretation of $\mathcal{F}(i)$ as a state space. For example, in a robotics system, it would be reasonable to choose the inner product on the state space $\mathcal{F}(i)$ so that $\|x_i\|_{\mathcal{F}(i)}^2$ is the Hamiltonian of the agent’s dynamics.

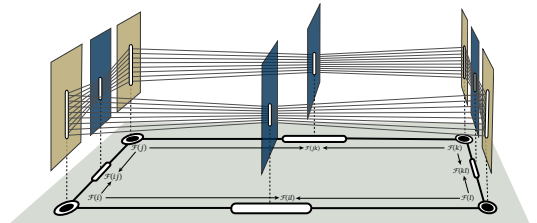


Fig. 1: Mental model of a cellular sheaf: over a cycle graph, agent’s stalks are tethered together with restriction maps.

A ubiquitous problem in cellular sheaf theory is to decide whether 0-cochains are coherent with respect to restriction maps. This consistency is expressed by the notion of a global section.

Definition 2. Suppose $\mathcal{F}$ is a cellular sheaf over G . A **global section** is a 0-cochain $\mathbf{x} \in C^0(G; \mathcal{F})$ such that $\mathcal{F}_i \lhd_{ij}(x_i) = \mathcal{F}_j \lhd_{ij}(x_j)$ for all $i \in V$ for all $j \in N_i$. The set of global sections is $\Gamma(G; \mathcal{F}) \subseteq C^0(G; \mathcal{F})$.

Observe that the global section condition is equivalent to minimizing a quadratic potential function $(1/2)\|\mathcal{F}_i \lhd_{ij}(x_i) - \mathcal{F}_j \lhd_{ij}(x_j)\|_{\mathcal{F}(ij)}^2$ over each edge $ij \in E$. In a multi-agent system, this can be interpreted as requiring that communicating agents reach consensus on their outputs to neighboring agents. Indeed, if the vertex stalks correspond to agents' spatial positions and the constant sheaf is chosen on a connected communication graph, global sections are precisely positional consensus vectors. To generalize beyond consensus, we can alter the edge potential functions to encode different coordination goals.

To make this precise, we need some notions from homological algebra. Define the **coboundary operator** $C^0(G; \mathcal{F}) \xrightarrow{\delta_{\mathcal{F}}} C^1(G; \mathcal{F})$ given by $(\delta_{\mathcal{F}}\mathbf{x})_{ij} = \mathcal{F}_i \lhd_{ij}(x_i) - \mathcal{F}_j \lhd_{ij}(x_j)$. The **degree-0** and **degree-1 cohomology** of a cellular sheaf is given by

$$H^0(G; \mathcal{F}) = \ker \delta_{\mathcal{F}}, \quad H^1(G; \mathcal{F}) = C^1(G; \mathcal{F}) / \text{im } \delta_{\mathcal{F}}.$$

It follows immediately that $H^0(G; \mathcal{F}) = \Gamma(G; \mathcal{F})$ because $x \in \ker \delta_{\mathcal{F}}$ precisely when $\mathcal{F}_i \lhd_{ij}(x_i) - \mathcal{F}_j \lhd_{ij}(x_j) = 0$ for all $ij \in E$. Global sections, then, can be computed as the kernel of the coboundary operator, where the relevant matrix grows in the number of nodes and edges of the graph and the dimensions of the stalks. To incorporate edge potential functions, we define the following notion of energy for a cellular sheaf.

Definition 3. Given a cellular sheaf $\mathcal{F}$ on a graph G and potential functions $U_e: \mathcal{F}(e) \rightarrow \mathbb{R}$ for each edge $e \in E$, we let the global potential function $U: C^1(G; \mathcal{F}) \rightarrow \mathbb{R}$ be defined as $U(\mathbf{y}) := \sum_{e \in E} U_e(y_e)$. The **Dirichlet energy** function for $\mathcal{F}$ is then $f := U \circ \delta_{\mathcal{F}}: C^0(G; \mathcal{F}) \rightarrow \mathbb{R}$.

For the following special cases, the relationship between the minimizers of the Dirichlet energy and global sections are made explicit:

- (a) If all potential functions are chosen to have a unique global minimizer at 0, then minimizers of the sheaf energy function correspond to global sections:

$$\operatorname{argmin}_{\mathbf{x} \in C^0(G; \mathcal{F})} U(\delta_{\mathcal{F}}(\mathbf{x})) = \Gamma(G; \mathcal{F}).$$

- (b) If the global potential function U is minimized at some unique nonzero $b \in C^1(G; \mathcal{F})$ with $b \in \text{im } \delta_{\mathcal{F}}$, then the minimizers of the energy function correspond to an affine shift of the global sections, i.e.,

$$\operatorname{argmin}_{\mathbf{x} \in C^0(G; \mathcal{F})} U(\delta_{\mathcal{F}}(\mathbf{x})) = \delta_{\mathcal{F}}^+ b + \Gamma(G; \mathcal{F})$$

(see [7, Theorem 2]).

A sheaf together with a family of edge potential functions provides the data of a coordination sheaf.

Definition 4. A **coordination sheaf** consists of a choice of

- (a) communication graph $G = (V, E)$,
- (b) cellular sheaf $\mathcal{F}$ as in Definition 1,
- (c) family of edge potentials $\{U_e: \mathcal{F}(e) \rightarrow \mathbb{R}\}_{e \in E}$.

Every coordination sheaf has an associated Dirichlet energy function $f := U \circ \delta_{\mathcal{F}}$ where $U := \sum_{e \in E} U_e$ (Definition 3).

B. Problem Statement

The main goal in this paper is to develop an asynchronous distributed algorithm for achieving multi-agent goals specified by coordination sheaves. We formalize this as the problem of minimizing the Dirichlet energy of a sheaf.

Problem 1. Given a coordination sheaf $(G, \mathcal{F}, \{U_e\}_{e \in E})$, asynchronously solve

$$\operatorname{minimize}_{\mathbf{x} \in C^0(G; \mathcal{F})} f(\mathbf{x}) = U \circ \delta_{\mathcal{F}}(\mathbf{x}). \quad (1)$$

The key to the design of coordination sheaves is in choosing restriction maps and edge potentials such that minimizing the Dirichlet energy corresponds to achieving the desired coordination goal. The following example illustrates this philosophy with a representative application achievable by solving Problem 1.

Example 2 (Moving UAV formations). Consider the following coordination task for two teams of three unmanned aerial vehicles (UAVs), each with a designated leader. Each UAV is modeled by state spaces $\mathbb{R}^3 \oplus \mathbb{R}^3$ with concatenated state vectors $[p_i, v_i]^T$ for position p_i and velocity v_i . The task is for each team of three UAVs to maintain a fixed triangle formation by adjusting their position to achieve a specific 3D displacement vector $\hat{p}_{ij} \in \mathbb{R}^3$ relative to their leader. The formations need to move towards a target and, tracking the target, the leaders of each team align their velocity vectors together. The following coordination sheaf $\mathcal{F}$ encodes this task:

- The graph has nodes $V = \{1, 2, 3, 4, 5, 6\}$. There are edges between members of the same team as well as between the leaders: $E = \{12, 13, 23, 45, 46, 56, 14\}$.
- The node stalks are state spaces $\mathcal{F}(i) = \mathbb{R}^3 \oplus \mathbb{R}^3$. The edge stalks are $\mathcal{F}(ij) = \mathbb{R}^3$ (see Fig. 2).
- The restriction maps are projections (see Fig. 2).
- The edge potentials are $U_e(y_e) = (1/2)\|y_e - \hat{p}_{ij}\|^2$ between leader and followers and $U_e(y_e) = (1/2)\|y_e\|^2$ between leaders.

Minimizing the Dirichlet energy function of the sheaf

$$f(\mathbf{x}) = \frac{1}{2}\|v_1 - v_4\|^2 + \sum_{ij \in \{12, 13, 45, 46\}} \frac{1}{2}\|p_i - p_j - \hat{p}_{ij}\|^2$$

achieves the coordination goal. Solving Problem 1 would allow the agents to still achieve this goal even in the presence of uncertain communication and computation delays.

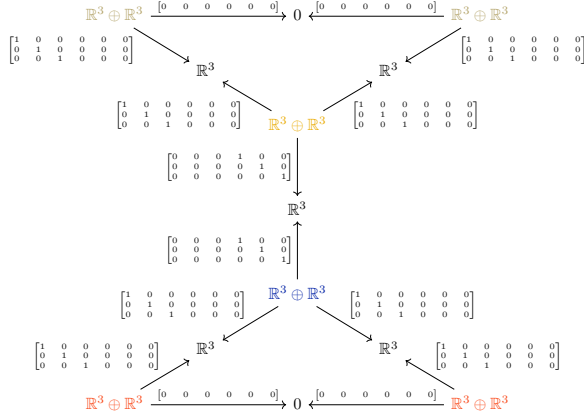


Fig. 2: In the above coordination sheaf, $\mathcal{F}_{i \triangleleft ij} = \mathcal{F}_{j \triangleleft ij}$ between leaders and leaders are projection onto velocity while the restriction maps between leaders and followers are projection onto position. The restriction maps between followers are the zero map. The diagram layout is evocative of the intended formation.

III. ALGORITHM DESIGN

We first introduce the synchronous gradient descent algorithm for minimizing the Dirichlet energy and discuss its relation to the sheaf Laplacian. We then introduce our model of asynchrony and bring this algorithm into the asynchronous setting.

A. Synchronous Algorithm

For agents to minimize Dirichlet energy in a distributed fashion, we use the nonlinear sheaf Laplacian (see [10]) defined as follows.

Definition 5. Given a cellular sheaf $\mathcal{F}$ on a graph G with global potential $U: C^1(G; \mathcal{F}) \rightarrow \mathbb{R}$, the **nonlinear sheaf Laplacian** is a map $L_{\mathcal{F}}^{\nabla U}: C^0(G; \mathcal{F}) \rightarrow C^0(G; \mathcal{F})$ defined by the gradient of the energy function, i.e.,

$$L_{\mathcal{F}}^{\nabla U} := \delta_{\mathcal{F}}^{\top} \circ \nabla U \circ \delta_{\mathcal{F}} = \nabla(U \circ \delta_{\mathcal{F}}).$$

Example 3. Suppose $U(\mathbf{y}) = \sum_{e \in E} (1/2) \|y_e\|^2$. Then, $L_{\mathcal{F}}^{\nabla U}$ is the linear sheaf Laplacian $L_{\mathcal{F}} = \delta_{\mathcal{F}}^{\top} \delta_{\mathcal{F}}$ (as defined in [27]). Furthermore, for these same quadratic potentials, $L_{\mathbb{R}}^{\nabla U}$ specializes to the standard graph Laplacian [28].

In the synchronous setting, a natural choice for minimizing the Dirichlet energy function of a given sheaf $\mathcal{F}$ is gradient descent, which can be given in terms of the nonlinear sheaf Laplacian (see Definition 5) as

$$\mathbf{x}(t+1) := \mathbf{x}(t) - \gamma L_{\mathcal{F}}^{\nabla U} \mathbf{x}(t)$$

for a given step-size $\gamma > 0$, which we view as a discrete-time dynamical system called **sheaf diffusion** (see [19]). These iterates can be computed locally by having agent i execute

$$x_i(t+1) = x_i(t) - \gamma [L_{\mathcal{F}}^{\nabla U} \mathbf{x}(t)]_i$$

for all $i \in V$, where

$$[L_{\mathcal{F}}^{\nabla U} \mathbf{x}(t)]_i = \sum_{j \in N_i} \mathcal{F}_{i \triangleleft ij}^{\top} \nabla U_{ij}(\mathcal{F}_{i \triangleleft ij} x_i(t) - \mathcal{F}_{j \triangleleft ij} x_j(t)).$$

These dynamics are local as each agent i need only know its own state $x_i(t)$ and messages $\mathcal{F}_{j \triangleleft ij} x_j(t)$ from its immediate neighbors $j \in N_i$. Under strong-convexity assumptions on the edge potentials, the sheaf diffusion dynamics drive any initial condition $\mathbf{x}(0)$ to the orthogonal projection of $\mathbf{x}(0)$ onto $\delta_{\mathcal{F}}^{\perp} b + H^0(G; \mathcal{F})$, where b is the (necessarily) unique minimizer of the global potential function $U = \sum_{e \in E} U_e$, provided $b \in \text{im } \delta_{\mathcal{F}}$ [7].

B. Asynchronous Algorithm

We use $t \in \mathbb{N}$ as a discrete-time iteration counter, and let $T_i \subseteq \mathbb{N}$ denote the iterations at which agent i performs a computation. For each $t \in \mathbb{N}$, we define $\tau_j^i(t)$ to be the block model x_j that is received by agent i at its t -th iteration. Note that agents do not need to know T_i or τ_j^i for any i or j , they are only used for analysis. To solve Problem 1, we employ a partially asynchronous sheaf diffusion algorithm.

Algorithm 1. Given a coordination sheaf (Definition 4), let $x_1(0), \dots, x_N(0)$, and $\gamma > 0$ be given. For all $i \in V$, asynchronously execute

$$x_i(t+1) = \begin{cases} x_i(t), & t \notin T_i, \\ x_i(t) - \gamma [L_{\mathcal{F}}^{\nabla U} \mathbf{x}^i(t)]_i, & t \in T_i, \end{cases} \quad (2)$$

where the local updates in (2) are computed with the 0-cochain $\mathbf{x}^i(t) = (x_1(\tau_1^i(t)), x_2(\tau_2^i(t)), \dots, x_N(\tau_N^i(t)))$.

Two standard assumptions about asynchrony appear in the distributed computing literature (see, e.g. [9]). The first of these is the *totally asynchronous* setting where delays can be unbounded. In this setting, the iterates of each agent's local state generated by (2) may not converge to a minimizer for consensus-type problems whose Hessians are not block diagonally dominant [29, Theorem 4.1]. An example is given in the standard reference [9, Section 7.1, Example 1.3]. For a constant sheaf, the linear sheaf Laplacian reduces to a graph Laplacian (see Example 3), which is only weakly diagonally dominant and thus may not converge under total asynchrony. Therefore we only consider the *partially asynchronous* setting where maximum delay bounds are assumed.

Assumption 1 (Partial Asynchrony).

- (a) There exists an integer $B \geq 0$ such that

$$\{t, t+1, \dots, t+B\} \cap T_i \neq \emptyset$$

for every $i \in V$ and for every $t \in \mathbb{N}$.

- (b) For all $i \in V$, $t - B \leq \tau_j^i(t) \leq t$ for all $j \in N_i$ and for all $t \in T_i$.

- (c) For all $i \in V$ and for all $t \in T_i$, $\tau_i^i(t) = t$.

Part (a) ensures that each agent updates at least once every $B+1$ iterations. Part (b) ensures that communication delays are bounded above by B . Part (c) ensures that each agent always has access to the value of its latest iterate. In short, all delays in communications and computations are bounded by B , but B is not necessarily small.

IV. CONVERGENCE ANALYSIS

We show that given the sheaf Dirichlet energy function $f := U \circ \delta_{\mathcal{F}}$ on a coordination sheaf, the iterates generated by Algorithm 1 converge to the global minimizer. We start by stating assumptions on each edge potential U_e as follows.

Assumption 2 (m_e -Strongly Convex). The edge potential U_e is m_e strongly convex for each $e \in E$, i.e.,

$$\langle \nabla U_e(x_e) - \nabla U_e(y_e), x_e - y_e \rangle \geq m_e \|x_e - y_e\|^2$$

for some $m_e > 0$ and all $x_e, y_e \in \mathcal{F}(e)$.

Assumption 3 (K_e -Smooth). The edge potential U_e is K_e -smooth for each $e \in E$, i.e.,

$$\|\nabla U_e(x_e) - \nabla U_e(y_e)\| \leq K_e \|x_e - y_e\|$$

for some $K_e > 0$ and all $x_e, y_e \in \mathcal{F}(e)$.

We then provide the following results for f .

Lemma 1. *The sheaf Dirichlet energy f (Definition 3) is bounded below.*

Proof. By Assumption 2, the edge potential U_e is m_e -strongly convex for each $e \in E$, meaning U_e is bounded below. Thus the sum $U(\mathbf{y}) := \sum_{e \in E} U_e(y_e)$ is also bounded below, which implies the sheaf Dirichlet energy f is bounded below. $\square$

We show that the Lipschitz constant for f can be obtained from the spectral properties of the linear sheaf Laplacian $L_{\mathcal{F}}$, and the Lipschitz constants K_e for each edge potential function U_e .

Lemma 2. *The function f (Definition 3) is K -smooth, i.e.,*

$$\|L_{\mathcal{F}}^{\nabla U} \mathbf{x} - L_{\mathcal{F}}^{\nabla U} \mathbf{y}\| \leq K \|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in C^0(G; \mathcal{F}),$$

where

$$K = \left(\max_{e \in E} K_e\right) \cdot \lambda_{\max}(L_{\mathcal{F}}) > 0 \quad (3)$$

is the Lipschitz constant.

Proof. For any $\mathbf{x}, \mathbf{y} \in C^0(G; \mathcal{F})$, we obtain

$$\begin{aligned} \|L_{\mathcal{F}}^{\nabla U} \mathbf{x} - L_{\mathcal{F}}^{\nabla U} \mathbf{y}\| &= \|\delta_{\mathcal{F}}^{\top} \nabla U(\delta_{\mathcal{F}} \mathbf{x}) - \delta_{\mathcal{F}}^{\top} \nabla U(\delta_{\mathcal{F}} \mathbf{y})\| \\ &= \|\delta_{\mathcal{F}}^{\top} (\nabla U(\delta_{\mathcal{F}} \mathbf{x}) - \nabla U(\delta_{\mathcal{F}} \mathbf{y}))\| \\ &\leq \|\delta_{\mathcal{F}}^{\top}\| \|\nabla U(\delta_{\mathcal{F}} \mathbf{x}) - \nabla U(\delta_{\mathcal{F}} \mathbf{y})\|, \end{aligned}$$

by Assumption 3, the Lipschitz constant for the global potential is $\|\nabla U(\delta_{\mathcal{F}} \mathbf{x}) - \nabla U(\delta_{\mathcal{F}} \mathbf{y})\| \leq (\max_{e \in E} K_e) \|\delta_{\mathcal{F}} \mathbf{x} - \delta_{\mathcal{F}} \mathbf{y}\|$, where $\max_{e \in E} K_e$ is the largest Lipschitz constant over all the edge potentials. Thus we obtain

$$\begin{aligned} \|L_{\mathcal{F}}^{\nabla U} \mathbf{x} - L_{\mathcal{F}}^{\nabla U} \mathbf{y}\| &\leq \left(\max_{e \in E} K_e\right) \|\delta_{\mathcal{F}}^{\top}\| \|\delta_{\mathcal{F}} \mathbf{x} - \delta_{\mathcal{F}} \mathbf{y}\| \\ &\leq \left(\max_{e \in E} K_e\right) \|\delta_{\mathcal{F}}^{\top}\| \|\delta_{\mathcal{F}}\| \|\mathbf{x} - \mathbf{y}\| \\ &= \left(\max_{e \in E} K_e\right) \sigma_{\max}^2(\delta_{\mathcal{F}}) \|\mathbf{x} - \mathbf{y}\|, \end{aligned}$$

where $\|\delta_{\mathcal{F}}^{\top}\| = \|\delta_{\mathcal{F}}\| = \sigma_{\max}(\delta_{\mathcal{F}})$. Since

$$\sigma_{\max}(\delta_{\mathcal{F}}) = \sqrt{\lambda_{\max}(\delta_{\mathcal{F}}^{\top} \delta_{\mathcal{F}})} = \sqrt{\lambda_{\max}(L_{\mathcal{F}})},$$

therefore the Lipschitz constant is (3). $\square$

Remark 2. The Lipschitz constant for the linear sheaf Laplacian reduces to $K = \lambda_{\max}(L_{\mathcal{F}})$ because all the edge potential functions are $U_e(y_e) = (1/2)\|y_e\|^2$, which is 1-Lipschitz. This result also agrees with the relevant bound for the graph Laplacian $L_{\mathbb{R}}^{\nabla U}$ (see [8, Section 5.3.1]).

Lemma 3. *The function f (Definition 3) is convex.*

Proof. By Assumption 2, the edge potential U_e is m_e -strongly convex for each $e \in E$, which implies the global potential $U(\mathbf{y}) = \sum_{e \in E} U_e(y_e)$ is m -strongly convex where $m = \min\{m_1, \dots, m_{|E|}\}$, and since $\delta_{\mathcal{F}}$ is linear, the composition $f = U \circ \delta_{\mathcal{F}}$ is then convex. $\square$

We denote $\mathcal{X}^* = \{\mathbf{x} \in C^0(G; \mathcal{F}) \mid L_{\mathcal{F}}^{\nabla U} \mathbf{x} = 0\}$ as the minimizer set and assume it is nonempty. We then denote f^* as the minimum value of f , i.e., $f^* := \inf_{\mathbf{x}} f(\mathbf{x})$. We define the following global error bound (EB) condition, which will be used in our convergence analysis to bound the distance from agents' iterates to the nearest minimizer of their Dirichlet energy. Note that in general, the solution set $\mathcal{X}^*$ is a linear subspace of $C^0(G; \mathcal{F})$, meaning the nearest minimizer is always given by orthogonal projection onto this subspace.

Definition 6. The global error bound (EB) inequality holds if there exists a $\kappa > 0$ such that

$$\min_{\mathbf{x}^* \in \mathcal{X}^*} \|\mathbf{x} - \mathbf{x}^*\| \leq \kappa \|L_{\mathcal{F}}^{\nabla U} \mathbf{x}\| \quad (4)$$

holds for all $\mathbf{x} \in C^0(G; \mathcal{F})$.

We next show that the Dirichlet energy function satisfies the global EB inequality, and we give an explicit form for κ .

Lemma 4. *Given a coordination sheaf in the sense of Definition 4, the associated Dirichlet energy $f := U \circ \delta_{\mathcal{F}}$ satisfies the global error bound in the sense of Definition 6, with constant $\kappa = \frac{1}{m\sigma_2(\delta_{\mathcal{F}})}$, where $m = \min\{m_1, \dots, m_{|E|}\}$.*

Proof. It is shown in [30, Appendix B] that for a ϱ -strongly convex function g and a linear map $x \mapsto Ax$, the function $g(Ax)$ is convex and it satisfies the Polyak-Łojasiewicz (PL) inequality with parameter $= \varrho\sigma_2(A)$. That is, $g(Ax)$ satisfies

$$\frac{1}{2} \|\nabla g(Ax)\|^2 \geq \varrho \sigma_2(A) (g(Ax) - g^*), \quad (5)$$

where g^* is the global minimum value of $g \circ A$, which exists because $g \circ A$ is convex. The Dirichlet energy f is itself the composition of an m -strongly convex function U with a linear map $\delta_{\mathcal{F}}$ by definition, and hence (5) gives

$$\frac{1}{2} \|L_{\mathcal{F}}^{\nabla U} \mathbf{x}\|^2 \geq m\sigma_2(\delta_{\mathcal{F}}) (f(\mathbf{x}) - f^*).$$

The result follows from observing that if a function satisfies the PL inequality with parameter μ , then it satisfies the EB condition with parameter $\kappa = \frac{1}{\mu}$ [30, Appendix A]. $\square$

Next we define the following nonnegative quantities that measure the progress of the algorithm.

$$\alpha(t) := f(\mathbf{x}(t)) - f^*, \quad (6)$$

$$\beta(t) := \sum_{\tau=t-B-1}^{t-1} \|\mathbf{x}(\tau+1) - \mathbf{x}(\tau)\|^2. \quad (7)$$

We then obtain the following global periodic convergence result for sequences $\{\alpha(t), \beta(t)\}$ defined as in (6) and (7).

Theorem 1. *Let Assumptions 1 (i.e., the partial asynchrony setting), 2, and 3 hold. Given the cost (1) defined in Problem 1, there exists some $\gamma_0 > 0$ such that if the step size satisfies $\gamma \in (0, \gamma_0)$, the sequences $\{\alpha(t), \beta(t)\}$ generated by (2) satisfy*

$$\alpha(r(B+1)) \leq a(1-\gamma c)^r, \quad (8)$$

$$\beta(r(B+1)) \leq b(1-\gamma c)^r \quad (9)$$

for all $r \in \mathbb{N}$, where a, b , and $c < \frac{1}{\gamma}$ are positive constants.

Proof. Since Assumptions 1 to 3 hold, by Lemmas 1 to 4, the result follows by [31, Theorem 9]. $\square$

Remark 3. A smaller global EB constant κ from (4) provides a tighter upper bound in [31, Lemma 8], which makes c in (8) and (9) larger. This allows for larger contraction between iterations, i.e., faster convergence. The global EB constant κ also depends on the spectrum of the (linear) sheaf Laplacian $L_{\mathcal{F}}$, which we will discuss in detail later (see Section VI).

Theorem 1 implies that $\{\alpha(t), \beta(t)\}$ converges at least linearly with every $B+1$ step. Also by Lemma 4, the global EB inequality holds for any $\mathbf{x} \in C^0(G; \mathcal{F})$; this implies that the linear convergence rate can be guaranteed regardless of the initial condition, i.e., the global state converges to the optimum at least linearly with every $B+1$ step, and the convergence result does not depend on initial conditions $x_1(0), \dots, x_N(0)$ for any agent.

V. NUMERICAL RESULTS

We implemented several different experiments to validate our theoretical results. Throughout this section, we use $\text{rand}(n, m)$ to denote the probability distribution of $n \times m$ matrices whose entries are drawn uniformly at random from the range $[0, 1]$.

A. Experimental Setup

Primitives for constructing cellular sheaves and sheaf Laplacians are implemented in the `AlgebraicOptimization.jl` Julia package¹. For our experiments, we implemented a simulation of the partially asynchronous distributed computing environment as follows.

Each agent tracks its local state as well as the (potentially stale) states of its neighboring agents. To initialize the simulation, each agent i generates a local upper bound $b_i \in [B]$

on how frequently it will update its state. It also generates a time $p_i \in [b_i]$ to update its state. There is a global iteration counter t , and when $t \bmod b_i = p_i$, agent i updates its local state according to the iteration in (2). In other words, the set T_i of times when agent i computes a local update is

$$T_i = \{t_i \in \mathbb{N} \mid t_i \bmod b_i = p_i\}.$$

Similarly, each agent generates an upper bound $b'_i \in [B]$ and a time $p'_i \in [b'_i]$ for how frequently it will broadcast its state to neighbors. When $t \bmod b'_i = p'_i$, agent i sends its most recent local state $x_i(t)$ to all neighboring agents $j \sim i$ who immediately use this value to update $\mathbf{x}^j(t)$. To make the update and broadcast schedules more random, each agent also resamples its p_i and p'_i values after every update and broadcast respectively.

The update bounds b_i were drawn from an evenly weighted mixture of normal distributions centered at $0.05B$ and $0.5B$. Similarly, the broadcast bounds b'_i were drawn from an evenly weighted mixture of normals centered at $0.1B$ and $0.8B$. This models a heterogeneous system with a mixture of fast and slow agents in terms of both update and broadcast times with more frequent computation than communication.

B. Results

Experiment 1: We tested asynchronous sheaf diffusion for different types of sheaves and different values of the global delay bound B . In particular, we used a random 20 node 4-regular graph G as a fixed communication topology and varied the types of restriction maps to produce different test cases. Each run of asynchronous sheaf diffusion was initialized from a fixed value of $\mathbf{x}(0)$ drawn from a Gaussian distribution centered at the origin with a variance of 10. Figure 3 shows the sheaf Dirichlet energy and relative error in global state over time for various values of B ranging from 0 to 200. In particular, Figure 3.III relies on the following example to construct restriction maps.

Example 4 (Matrix-Weighted Sheaf). A **matrix-weighted graph** is an undirected graph G with positive semi-definite $n \times n$ matrices W_{ij} associated to each edge ij . These have an associated matrix-weighted Laplacian with diagonal blocks $L_{i,i} = \sum_{j \in N_i} W_{ij}$ and off-diagonal blocks $L_{i,j} = -W_{ij}$. Matrix-weighted graphs and Laplacians have found applications in formation control [32]. To any matrix-weighted graph G , we can construct an associated **matrix-weighted sheaf** $\mathcal{F}$ over G as follows. For each matrix weight W_{ij} , perform a rank-revealing QR decomposition $W_{ij} = Q_{ij}R_{ij}$ and set each restriction map $\mathcal{F}_{i \triangleleft ij} = \mathcal{F}_{j \triangleleft ij} = R_{ij}$. The sheaf Laplacian of $\mathcal{F}$ then corresponds to the matrix-weighted Laplacian of G [24].

Experiment 2: To test the global convergence properties of asynchronous sheaf diffusion, we fixed a sheaf $\mathcal{F}$ over a random 20 node 4-regular graph with randomly sampled restriction maps $\mathcal{F}_{i \triangleleft ij} \sim \text{rand}(1, 4)$. We then ran asynchronous sheaf diffusion with a fixed delay bound of $B = 50$ from 100 different initial conditions sampled from a Gaussian distribution centered at the origin with a variance

¹Code is available at <https://github.com/AlgebraicJulia/AlgebraicOptimization.jl/tree/acc-experiments>

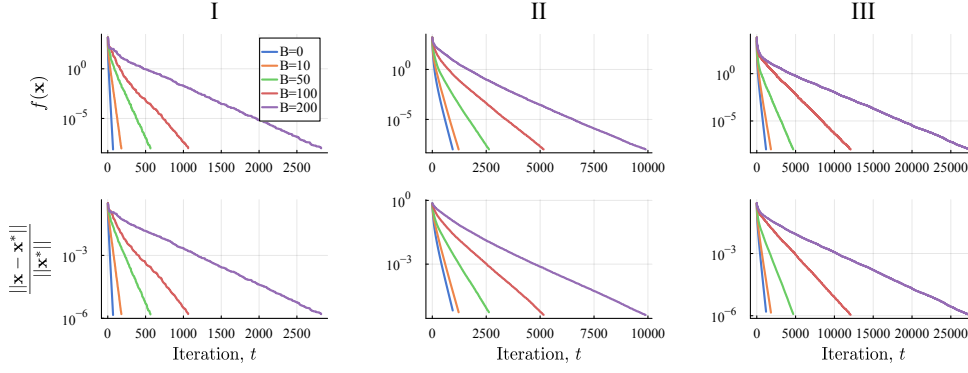
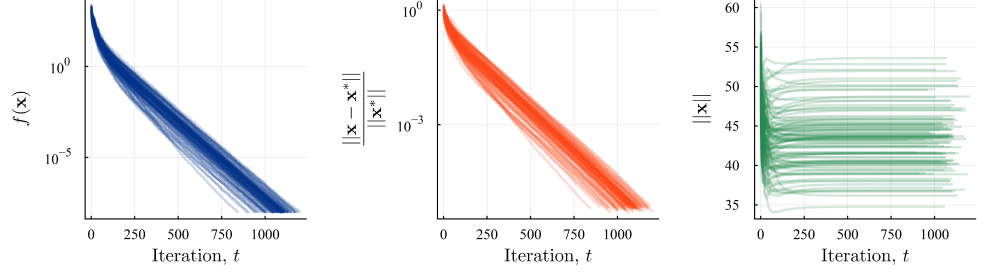


Fig. 3: Asynchronous convergence for various values of B . The top row plots energy while the bottom row plots relative error. The experiments were conducted on different sheaves over the same random 4-regular graph G . I) the constant sheaf $\mathbb{R}^4$ over G . II) a sheaf with random restriction maps $\mathcal{F}_i \leq_{ij} \sim \text{rand}(1, 4)$. III) a sheaf with random matrix-weighted edges. The positive semi-definite matrix weights had a 0.2 probability of being strictly positive definite.

Fig. 4: Running sheaf diffusion over the same sheaf $\mathcal{F}$ from 100 different initial conditions with $B = 50$. The sheaf is over a random 4-regular graph with random restriction maps $\mathcal{F}_i \leq_{ij} \sim \text{rand}(1, 4)$. The initial conditions are sampled randomly from a Gaussian centered at the origin with variance 10.



of 10. The energy value, relative error, and iterate norm over time are shown in Figure 4.

Experiment 3: In the case of synchronous sheaf diffusion, the iterates converge to the orthogonal projection of $x(0)$ onto the space of global sections, which we denote as $x(0)^\perp$ [24]. In this experiment, we investigate the effect of asynchrony on this result. We used the same sheaf $\mathcal{F}$ from Experiment 2, and ran asynchronous sheaf diffusion to convergence with various delay bounds starting from $B = 0$ up to $B = 2^{15}$. To mitigate the effects of random update schedules, we conducted three trials for each B value and averaged the results. We then measured the distance between the average of the final iterates x^* reached for each B and orthogonal projection of the initial condition onto $\Gamma(G; \mathcal{F})$. This is shown in Figure 5a.

Experiment 4: Finally, we investigated the impact of the smallest non-zero eigenvalue of the sheaf Laplacian $\lambda_2(L_{\mathcal{F}})$ on convergence of asynchronous sheaf diffusion. For this experiment, we generated sheaves over Erdos-Renyi random graphs with 20 nodes and 0.3 connection probability whose restriction maps were sampled from $\text{rand}(1, 4)$ and computed the smallest non-zero eigenvalue of their Laplacian. We then ran asynchronous sheaf diffusion to convergence for each sheaf with a fixed delay bound of $B = 50$ and recorded the number of iterations required for convergence. The scatterplot of λ_2 versus number of iterations for convergence is shown in Figure 5b.

VI. DISCUSSION AND FUTURE WORK

Experimental observations are consistent with the main claim of Theorem 1, namely $B+1$ step global linear convergence of asynchronous sheaf diffusion from arbitrary initial conditions. Experiment 1 demonstrates the scaling of the linear convergence rate as B increases for various different

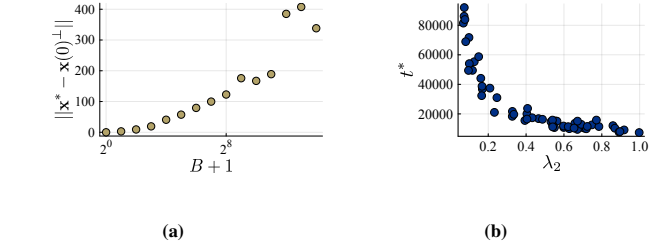


Fig. 5: (a) Distance between solution obtained via asynchronous sheaf diffusion and orthogonal projection of the initial condition $x(0)$ onto $\Gamma(G; \mathcal{F})$ for various values of B . (b) Smallest non-zero eigenvalue of the sheaf Laplacian (λ_2) versus number of iterations for asynchronous sheaf diffusion to reach convergence (t^*) for Erdos-Renyi random graphs with 20 nodes and 0.3 connection probability. In all cases, the communication bound was fixed to $B = 50$.

types of sheaves. Importantly, testing on the constant sheaf (Figure 3.I) shows that our results specialize to the case of standard consensus on graphs in an asynchronous setting. Additionally, the case $B = 0$ corresponds to the synchronous case, which as expected converges the most quickly. Overall, this experiment shows a consistent relationship between convergence rate and B -value, with larger B 's resulting in a slower linear convergence rate for all types of sheaves tested.

As can be seen from Experiment 2, all trajectories converge at a linear rate regardless of initial condition. This backs up the theoretical claim of global linear convergence, rather than linear convergence only after some $\hat{t}$ as is sometimes seen in the literature [33]. One interesting observation from Figure 4 is that the norm of the iterates quickly becomes constant. This suggests that our algorithm quickly converges to a sphere in $C^0(G; \mathcal{F})$ and then moves around in this sphere to reach a global section. In addition, Figure 5a clearly shows a positive correlation between B and the distance between

$\mathbf{x}^*$ and $\mathbf{x}(0)^\perp$, meaning that for larger communication and computation delays, solutions drift further from the solution obtained via synchronous computation.

The convergence of the algorithm is related to the smallest nonzero eigenvalue of the linear sheaf Laplacian $\lambda_2(L_{\mathcal{F}})$, as shown in Figure 5b. This is also related to the smallest nonzero singular value of the coboundary operator

$$\sigma_2(\delta_{\mathcal{F}}) = \sqrt{\lambda_2(\delta_{\mathcal{F}}^\top \delta_{\mathcal{F}})} = \sqrt{\lambda_2(L_{\mathcal{F}})}. \quad (10)$$

From a theoretical perspective, the EB constant κ is inversely proportional to $\sigma_2(\delta_{\mathcal{F}})$ (Lemma 4), hence κ is also inversely proportional to $\lambda_2(L_{\mathcal{F}})$ by (10), which makes the convergence rate proportional to $\lambda_2(L_{\mathcal{F}})$ (Remark 3). This is clearly supported by Experiment 4. It is therefore desirable to design sheaves such that $\lambda_2(L_{\mathcal{F}})$ is as large as possible. An interesting area of future work would be to investigate how the geometric properties of the restriction maps of a cellular sheaf combine with the topological properties of the graph over which it is defined to determine how well conditioned its Laplacian is for asynchronous diffusion. The study of the spectral theory of cellular sheaves was initiated in [27]. Other areas for future work include incorporating directed or time varying communication topologies.

REFERENCES

- [1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [2] G. Bao, L. Ma, and X. Yi, "Recent advances on cooperative control of heterogeneous multi-agent systems subject to constraints: a survey," *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 539–551, Dec. 2022.
- [3] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, May 2015.
- [4] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, May 2005.
- [5] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. H. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, Sept. 2008.
- [6] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.
- [7] T. Hanks, H. Riess, S. Cohen, T. Gross, M. Hale, and J. Fairbanks, "Distributed multi-agent coordination over cellular sheaves," *arXiv preprint arXiv:2504.02049*, 2025.
- [8] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.
- [9] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.
- [10] J. Hansen and R. Ghrist, "Opinion dynamics on discourse sheaves," *SIAM Journal on Applied Mathematics*, vol. 81, no. 5, pp. 2033–2060, 2021.
- [11] A. D. Shepard, "A Cellular Description of the Derived Category of a Stratified Space," Ph.D., Brown University, 1985.
- [12] J. M. Curry, "Sheaves, cosheaves and applications," Ph.D. dissertation, University of Pennsylvania, 2014.
- [13] G. E. Bredon, *Sheaf Theory*. Springer Science & Business Media, Dec. 2012.
- [14] H. Riess and R. Ghrist, "Diffusion of Information on Networked Lattices by Gossip," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 5946–5952.
- [15] H. Riess, M. Munger, and M. M. Zavlanos, "Max-plus synchronization in decentralized trading systems," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 221–227.
- [16] M. Robinson, "Asynchronous logic circuits and sheaf obstructions," *Electronic Notes in Theoretical Computer Science*, vol. 283, pp. 159–177, 2012.
- [17] Z. Cooperband and R. Ghrist, "Towards homological methods in graphic statics," *Journal of the International Association for Shell and Spatial Structures*, vol. 64, no. 4, pp. 266–277, 2023.
- [18] J. Hansen and T. Gebhart, "Sheaf neural networks," in *TDA & Beyond*, 2020.
- [19] C. Bodnar, F. Di Giovanni, B. Chamberlain, P. Lio, and M. Bronstein, "Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 527–18 541, 2022.
- [20] C. Battiloro, Z. Wang, H. Riess, P. Di Lorenzo, and A. Ribeiro, "Tangent Bundle Convolutional Learning: From Manifolds to Cellular Sheaves and Back," *IEEE Transactions on Signal Processing*, vol. 72, pp. 1892–1909, 2024.
- [21] O. Zaghen, A. Longa, S. Azzolin, L. Telyatnikov, A. Passerini, and P. Liò, "Sheaf Diffusion Goes Nonlinear: Enhancing GNNs with Adaptive Sheaf Laplacians," in *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*. PMLR, Oct. 2024, pp. 264–276.
- [22] J. Hansen and R. Ghrist, "Distributed Optimization with Sheaf Homological Constraints," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 565–571.
- [23] A. J. Dudzik, T. von Glehn, R. Pascanu, and P. Veličković, "Asynchronous algorithmic alignment with cocycles," in *Learning on Graphs Conference*. PMLR, 2024, pp. 3–1.
- [24] J. Hansen, "Laplacians of cellular sheaves: Theory and applications," Ph.D. dissertation, University of Pennsylvania, 2020.
- [25] D. Rosiak, *Sheaf theory through examples*. MIT Press, 2022.
- [26] A. Ayzenberg, T. Gebhart, G. Magai, and G. Solomadin, "Sheaf theory: from deep geometry to deep learning," *arXiv preprint arXiv:2502.15476*, 2025.
- [27] J. Hansen and R. Ghrist, "Toward a spectral theory of cellular sheaves," *Journal of Applied and Computational Topology*, vol. 3, no. 4, p. 315–358, Aug. 2019.
- [28] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Soc., 1997.
- [29] A. Frommer and D. B. Szyld, "On asynchronous iterations," *Journal of computational and applied mathematics*, vol. 123, no. 1-2, pp. 201–216, 2000.
- [30] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2016, pp. 795–811.
- [31] Y. Zhou, Y. Liang, Y. Yu, W. Dai, and E. P. Xing, "Distributed proximal gradient algorithm for partially asynchronous computer clusters," *Journal of Machine Learning Research*, vol. 19, no. 19, pp. 1–32, 2018.
- [32] M. H. Trinh, C. Van Nguyen, Y.-H. Lim, and H.-S. Ahn, "Matrix-weighted consensus and its applications," *Automatica*, vol. 89, pp. 415–419, Mar. 2018.
- [33] P. Tseng, "On the rate of convergence of a partially asynchronous gradient projection algorithm," *SIAM Journal on Optimization*, vol. 1, no. 4, pp. 603–619, 1991.