2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Diffusion of Information on Networked Lattices by Gossip

Hans Riess Robert Ghrist

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992539

***Abstract*— We study time-dependent dynamics on a network of order lattices, where structure-preserving lattice maps are used to fuse lattice-valued data over vertices and edges. The principal contribution is a novel asynchronous Laplacian, generalizing the usual graph Laplacian, adapted to a network of heterogeneous lattices. This resulting gossip algorithm is shown to converge asymptotically to stable "harmonic" distributions of lattice data. This general theorem is applicable to several general problems, including lattice-valued consensus, Kripke semantics, and threat detection, all using asynchronous local update rules.**

I. Introduction

The use of the graph Laplacian to diffuse information over networks is well-established in classical and contemporary work ranging from opinion dynamics [1] to distributed multi-agent consensus [2] and control [3], synchronization [4], [5], flocking [6], and much more. In the past decade, Laplacians that are adapted to handle vector-valued data, such as graph connection Laplacians [7], [8] or matrix-weighted Laplacians [9], have been revolutionary in signal processing processing [10], [11] and machine learning [12], [13].

While the ultimate form of a generalized Laplacian is as yet not present in applications, there are hints of a broader theory finding its way from algebraic topology to data science. The Laplacian from calculus class and the graph Laplacian are two extreme examples of a *Hodge Laplacian*. These are operators which act diffusively on data structures called *sheaves* [14]: see §II-B for brief details.

The present work is motivated by extending recent work on distributed consensus and data fusion from the setting of vector-valued data to that of data valued in more general partially-ordered sets and, specifically, lattices (in the algebraic as opposed to discrete sense): see §II-A. The fidelity with which lattices (including Boolean algebras) can model logical structures makes them appealing for representing distributed systems with complex logical behaviors, such as preference posets [15], robust linear temporal logic [16] and discrete signal processing [17]. Maps between lattices preserving structure and their residuals model the interchange of information between neighbors: see §IV-A.

University of Pennsylvania; Department of Electrical/Systems Engineering; hmr@seas.upenn.edu

University of Pennsylvania; Departments of Mathematics and Electrical/Systems Engineering; ghrist@math.upenn.edu

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering (Research Technology & Laboratory Directorate/Basic Research Office) under Grant No. HQ00342110001. The views expressed do not necessarily reflect the official policies of the Department of Defense nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Related work: The consensus literature is vast and includes Laplacian-based protocols [18] and (asynchronous) gossip-based protocols [19], [20]. Several works consider consensus on general functions [21], including max/min consensus [22], [23]. In the area of distributed computing, centralized branch-and-bound style algorithms for reaching agreement on lattices have been discovered [24]. Sheaves have shown promise in multi-agent systems, particularly from the perspective of concurrency [25], routing [26], [27], opinion dynamics [28] and sensor fusion [29].

The Bayesian approach to modeling knowledge/belief propagation via graphical models [30] is standard but fundamentally different than the approach here. Modal logics, particularly temporal logics, have seen numerous applications in model-checking [31] and control systems [32]–[35]. There are several use-cases of multimodal logics in the analysis of message-passing systems [36].

The authors previously defined a synchronous [Tarski] Laplacian and proved a Hodge-style fixed-point convergence result [37], which is extended here to the asynchronous setting. The *gossip algorithm* introduced in this paper generalizes an algorithm called the alternating algorithm [38, §3] introduced to synchronize event times of a pair of coupled discrete event systems, each described by a max-plus linear system [39]; our algorithm goes far beyond this in computing sections of (nearly) arbitrary lattice-valued network sheaves.

Outline: Background material (§II) and problem spcifications (§III) are followed by details of a novel Laplacian for asynchronous communication on networks of lattices (§IV). It is here that the main results on stability and convergence are proved. The subsequent section (§V) detail applications to multimodal logics, by using Kripke semantics, leading to a dual pair of *semantic* and *syntactic* Laplacians for diffusing knowledge and beliefs. This work ends (§VI) with some simulation.

II. Background

A. Lattices

Ordered sets model data types such as relations, concepts, rankings, matchings, concurrent events, as well as other taxonomies of information that are hierarchial, epistemic, or logical in nature: both in §V. Lattices double as ordered sets and algebraic structures consisting of two "merging" operations called *meet* and *join*.

Definition 1: A *lattice* is a tuple $\mathbf{Q} = (Q, \wedge, \vee, 0, 1)$. Q is a set with binary operations, $\wedge$ (meet) and $\vee$ (join), satisfying the following axioms: 1) $\wedge/\vee$ are commutative and associative; 2) for all $x \in Q$, $x \wedge x = x$ and $x \vee x = x$; 3) there exist elements $\bot, \top \in Q$ such that $\bot$ and $\top$ are

the identity of $\vee$ and $\wedge$ respectively; 4) for all $x, y \in Q$, $x = x \vee (x \wedge y) = x \wedge (x \vee y)$.

Equivalently, lattices can be viewed as (partially) ordered sets $(\mathbf{Q}, \preccurlyeq)$ with $x \preccurlyeq y \Leftrightarrow x \wedge y = x$ or, equivalently, $x \succcurlyeq y \Leftrightarrow x \vee y = x$. It will be useful to think of lattices as both partially-ordered sets and algebraic structures: the $\preccurlyeq$ notation will be crucial in proofs of all main results.

Example 1: Suppose S is a set. The powerset 2^S is a (Boolean) lattice $(2^S, \cap, \cup, \emptyset, S)$. The truth values $\mathbf{2} = (\{0,1\}, \wedge, \vee, 0, 1)$ is a lattice. Other important lattices are embedded in 2^S such as lattices representing ontologies [40], partitions [41], rankings, preferences [15], [42], and information-theoretic content [43]. The extended real line $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ is a lattice with min and max. Cartesian products of lattices are lattices with the component-wise meet and join operations; the lattice $\bar{\mathbb{R}}^n$ and its matrix algebra is integral to the study of discrete event systems [39], $\mathbf{2}^n$ to logic gates [44].

In order to work with systems of lattices, we will work with *lattice maps* $\varphi : \mathbf{P} \to \mathbf{Q}$. Such a map is *order preserving* if $x \preccurlyeq y \Rightarrow \varphi(x) \preccurlyeq \varphi(y)$ and is *join preserving* if $\varphi(x \vee y) = \varphi(x) \vee \varphi(y)$ and $\varphi(\bot_{\mathbf{P}}) = \bot_{\mathbf{Q}}$. Join preserving maps are automatically order preserving. A dual definition of *meet preserving* maps holds with similar consequences.

B. Network Sheaves

Suppose $G = (V, E)$ is an undirected graph (possibly with loops). An edge between (not necessarily distinct) nodes i and j in V is denoted by an unordered concatenated pair of indices $ij = ji \in E$. The set $N_i = \{j : ij \in E\}$ are the *neighbors* of i.

Given such a fixed network G, we define a data structure over G taking values in lattices. Such a structure is an example of a *cellular sheaf* [45], [46] (though the details of sheaf theory are not needed here).

Definition 2: A *finite lattice-valued network sheaf* $\mathcal{F}$ over G is a data structure that assigns:

1) A finite lattice $(\mathcal{F}(i), \wedge_i, \vee_j)$ to every node $i \in V$.
2) A finite lattice $(\mathcal{F}(ij), \wedge_{ij}, \vee_{ij})$ to every edge $ij \in E$.
3) Join-preserving maps

$$\mathcal{F}(i) \xrightarrow{\mathcal{F}_{i \trianglelefteq ij}} \mathcal{F}(ij) \xleftarrow{\mathcal{F}_{j \trianglelefteq ij}} \mathcal{F}(j) \tag{1}$$

for every $ij \in E$.

In most applications we imagine, the maps from node data to edge data will be join-preserving. Thinking of a sheaf as a distributed system, the state of the system is given by an *assignment* of vertex data: a tuple $\mathbf{x} \in \prod_{i \in V} \mathcal{F}(i)$ of choices of data $x_i \in \mathcal{F}(i)$ for each i. The data over the edges and the maps $\mathcal{F}_{i \trianglelefteq ij}$ are used to compare the compatibility of vertex data in an assignment. The following definition is crucial.

Definition 3: The *sections* of $\mathcal{F}$ are assignments $\mathbf{x}$ that are compatible: for every $ij \in E$,

$$\mathcal{F}_{i \trianglelefteq ij}(x_i) = \mathcal{F}_{j \trianglelefteq ij}(x_j). \tag{2}$$

The set of sections of $\mathcal{F}$ is denoted $\Gamma(\mathcal{F})$. These are the "globally compatible" states. In the simplest example of a *constant sheaf* (which assigns a fixed lattice to each vertex and edge, with identity maps between vertex and edge data), sections are precisely assignments of an identical element to each vertex (and edge): consensus over the network.

Example 2: The true utility of a sheaf lies in heterogeneity. For example, assign to each vertex $i \in V$ a finite set S_i and to each edge $ij \in E$ a finite set S_{ij} and set-maps $S_i \to S_{ij} \leftarrow S_j$. This induces several interesting sheaves of lattices. The *powerset sheaf* assigns the powersets (lattices of all subsets with union and intersection) to vertices and edges, with induced maps. The *partition sheaf* assigns the partition lattices (partitions of set elements with partition union and refinement) to vertex and edge sets, again with induced maps. More examples from formal concept analysis are less well-known [40], but very general.

III. Problem Formulation

An abstract formulation is sufficient, but, for concreteness, consider a scenario in which a collection of geographically dispersed agents collect, process, and communicate data based on local sensing. Proximity gives rise to a communications network, modeled as an undirected graph G. The data are lattice-valued, but each agent $i \in V$ works within its lattice $\mathcal{F}(i)$, assumed to be finite. In order for two proximate agents i and j to communicate their individualized data (residing in $\mathcal{F}(i)$ and $\mathcal{F}(j)$ respectively), they must "fuse" their observiations into a common lattice $\mathcal{F}(ij)$ by means of structure-preserving lattice maps. Together, the system forms a network sheaf.

The problem envisioned is distributed consensus by means of asynchronous communication and updates. By consensus, we do not mean that everyone agrees on a particular fixed lattice value; rather, each node agrees upon choices of *local* data that, when translated and compared to all neighbors' data over communication edges, agree. In the context of a sheaf of lattices, this is precisely the condition of an assignment being a *section*. The synchronous version of this problem – everyone communicates simultaneously with neighbors and updates immediately – is solvable via the Tarski Laplacian as per [37]. The asynchronous problem is our focus here. Select sensors broadcast their data to neighboring nodes according to some *firing sequence*.

Denote by $\tau : \{0, 1, 2, \dots\} \to 2^V$ the firing sequence of selected nodes which broadcast as a function of (ordered, discrete) time. At $t \in \{0, 1, 2, \dots\}$, active nodes $i \in \tau_t$ broadcast to each agent $j \in N_i$.[1] No other nodes broadcast.

This notion of a firing sequence suffices to cover asynchronous updating. The regularization of time to $\{0, 1, 2, \dots\}$ is a convenience and does not impact results. The firing sequence is, in practice, not known *a priori*. This is of no consequence since our results will hold independent of the choice of firing sequence. The following assumptions will hold throughout.

[1] If desired, one may choose a selection of egdes incident to the node and broadcast only to those neighbors. This results in more bookkeeping and a refined version of liveness, below, but does not substantially change the results or proofs.

Assumption 1 (Latency-Free): Within a single time instance t, nodes may broadcast (if firing), receive data (always), and compute (always).

Assumption 2 (Liveness): For all $i \in V$ and for every $t \in \{0, 1, 2, \dots\}$ there is a $t' \geqslant t$ such that $i \in \tau_{t'}$. As such, agents neither die nor are removed from the system.

Assumption 3 (Cross-Talk): Suppose i is an agent and $j, j' \in N_i \cap \tau_t$ are active neighbors. Then, j and j' can simultaneously broadcast to i without resulting in a fault.

Under these assumptions, we wish to solve the following distributed asynchronous constrained agreement problem. Assume 1) a network $G = (V, E)$; 2) a sheaf $\mathcal{F}$ of lattices over G; 3) an firing sequence $\tau : \{0, 1, 2, \dots\} \to 2^V$ of broadcasts; and 4) an initial condition $\mathbf{x}[0]$, being an assignment of an element $x_v \in \mathcal{F}(v)$ to each agent $v \in V$. Using only local communication subordinate to the firing sequence τ, evolve the initial condition $\mathbf{x}[0]$ to a section $\mathbf{x} \in \Gamma(\mathcal{F})$.

This problem has elements of consensus (because of the local agreement implied in a section) as well as data fusion (due to the lattice maps merging data from vertex lattices to edge lattices).

IV. An Asynchronous Laplacian

Our method for solving this asynchronous constrained agreement problem is to define an asynchronous harmonic flow on the sheaf by localizing the Tarski Laplacian of [37].

A. The Tarski Laplacian

Throughout, $\mathcal{F}$ is a finite lattice-valued sheaf over a network G. Our first step towards a Laplacian involves preliminaries on *residuals* [47], also known as *Galois connections* [48]. These are a type of *adjoint* or lattice-theoretic analogue of the familiar *Moore-Penrose pseudoinverse* in matrix algebra.

Definition 4: Given a join-preserving lattice map $\varphi : \mathbf{P} \to \mathbf{Q}$, its *residual* is the map $\varphi^+ \colon \mathbf{Q} \to \mathbf{P}$ given by

$$\varphi^+(p) = \bigvee \{q \mid \varphi(q) \preccurlyeq p\}.$$

Like an adjoint, it reverses the direction of the map, resembling a pseudoinverse more closely in some cases. The following two lemmas have straightforward proofs via definitions.

Lemma 1: Suppose $\varphi : \mathbf{P} \to \mathbf{Q}$ is join-preserving and injective. Then $\varphi^+ \circ \varphi = id$.

Lemma 2: For $\varphi : \mathbf{P} \to \mathbf{Q}$ join-preserving, the following identities hold:

1) for all $p \in \mathbf{P}$, $\varphi^+ \circ \varphi(p) \succcurlyeq p$; and
2) for all $p \in \mathbf{P}$ and $q \in \mathbf{Q}$,

$$\varphi(p) \preccurlyeq q \Leftrightarrow p \preccurlyeq \varphi^+(q). \tag{3}$$

A lattice-theoretic analogue of the graph Laplacian – the *Tarski Laplacian* – was introduced in [37]. For $\mathcal{F}$ a network sheaf of lattices over G and $\mathbf{x}$ an assignment of vertex data, the Tarski Laplacian L acts as:

$$(L\mathbf{x})_i = \bigwedge_{j \in N_i} \mathcal{F}^+_{i \trianglelefteq ij} \mathcal{F}_{j \trianglelefteq ij}(x_j). \tag{4}$$

The key construct is to localize this operator subordinate to a firing sequence τ.

Definition 5: The *asynchronous Tarski Laplacian* is the map

$$L : \{0, 1, 2, \dots\} \times \prod_{i \in V} \mathcal{F}(i) \to \prod_{i \in V} \mathcal{F}(i)$$

which acts on an assignment $\mathbf{x}$ as

$$(L_t\mathbf{x})_i = \bigwedge_{j \in N_i \cap \tau_t} \mathcal{F}^+_{i \trianglelefteq ij} \mathcal{F}_{j \trianglelefteq ij}(x_j). \tag{5}$$

This is a restriction of the Tarski Laplacian (4) in that at time t, only immediate neighbors to broadcasting nodes are updated; all other nodes are unchanged. In the extreme of a firing sequence where all nodes broadcast at all times, the full Tarski Laplacian ensues.

B. Gossip and harmonic states

The rationale for the *Laplacian* moniker lies in the efficacy of L as a diffusion operator on states. Given an initial state $\mathbf{x}[0]$, heat flow is defined by the following discrete-time dynamical system:

$$\mathbf{x}[t+1] = (\mathrm{id} \wedge L_t)\, \mathbf{x}[t] \tag{6}$$

with initial condition $\mathbf{x}[0] \in \prod_{i \in V} \mathcal{F}(i)$.

Heat flow is the analogue of iterating the *random-walk* or *Perron* operator $I - \epsilon L$ (where I is the identity matrix, L the graph Laplacian matrix, $\epsilon > 0$) on scalar-valued data on a graph [18]. The principal result of this work is a type of "Hodge Theorem:" the *harmonic* states (the equilibria of $\mathrm{id} \wedge L$ is a proxy for the kernel of the Laplacian) are exactly the globally consistent solutions to the sheaf.

Theorem 1 (Main Theorem): For any finite lattice-valued network sheaf $\mathcal{F}$ with join-preserving structure maps and any firing sequence τ satisfying liveness, the sections of $\mathcal{F}$, $\Gamma(\mathcal{F})$, are precisely the time-independent solutions to heat flow (6).

The argument in the proof of Theorem 1 in Appendix A immediately implies an iterative protocol we call *gossip* for an agent $i \in V$ to converge to a harmonic state by firing a finite number of times. The protocol is exactly the localized heat flow (6).

Local assignments $x_i[0] \in \mathcal{F}(i)$ are initialized by all nodes, then, in a series of rounds $t = 0, 1, 2, \dots$, proceed as follows. Each node i listens for adjacent nodes broadcasting their "fused observation" $\mathcal{F}_{j \trianglelefteq ij}(x_j[t])$, $j \in N_i$. Upon receiving the "encoded" message, i applies the residual map $\mathcal{F}^+_{i \trianglelefteq ij}$ to $\mathcal{F}_{j \trianglelefteq ij}(x_j[t])$. At the end of each round, i aggregates all of the "decoded" messages, including the original local assignment $x_i[t]$, taking meets. The resulting element of $\mathcal{F}(i)$ is the local assignment for the new round, $x_i[t+1]$.
Finiteness of $\mathcal{F}(i)$ implies the gossip algorithm terminates in finitely many iterations since $\mathbf{x}[t+1] \preccurlyeq \mathbf{x}[t]$, or else $\mathbf{x}[t]$ is a section.[2]

[2]In fact, a weaker condition, that $\prod_{i \in V}$ satisfies a descending chain condition, is possible if we require $\mathcal{F}_{i \trianglelefteq ij}$ to preserve joins of an arbitrary subset.

V. Semantics

Examples of the Tarski Laplacian and heat flow are especially well-suited to distributed multimodal logic, where: vertices of a network correspond to agents; edges are communications between agents; assignments are sets of states associated to each vertex; and sections are assignments that express a consensus of knowledge across the network. For this we require the basic theory of Kripke semantics [36].

A. Kripke semantics

Definition 6: A *Kripke frame* $\mathbf{F} = (S, \mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_n)$ consists of a set of *states*, S, and a sequence of binary relations $\mathcal{K}_i \subseteq S \times S$ used to encode modal operators. A *Kripke model* over a set Φ of atomic propositions consists of the data $\mathcal{M} = (\mathbf{F}, \pi)$, where $\pi : S \to 2^\Phi$ validates whether or not a state $s \in S$ satisfies an atomic proposition $p \in \Phi$.

Suppose φ is a formula; then one writes $(\mathcal{M}, s) \models \varphi$ if s *satisfies* the formula φ in the model $\mathcal{M}$. The semantics of a Kripke model is defined inductively using the symbols `true`, $\wedge$, and $\neg$ in their typical usage:

1) $(\mathcal{M}, s) \models \texttt{true}$ for all $s \in S$.
2) For $p \in \Phi$ atomic, $(\mathcal{M}, s) \models p$ if and only if $p \in \pi(s)$.
3) For φ an formula, $(\mathcal{M}, s) \models \neg\varphi$ if and only if $(\mathcal{M}, s) \not\models \varphi$.
4) For φ, ψ formulae, $(\mathcal{M}, s) \models \varphi \wedge \psi$ if and only if $(\mathcal{M}, s) \models \varphi$ and $(\mathcal{M}, s) \models \psi$.

There are additional (dual) modal operators on formulae, K_i and P_i (typically corresponding to knowledge and possibility), based on the binary relations $\mathcal{K}_i$. One writes $(\mathcal{M}, s) \models K_i\varphi$ if and only if $(\mathcal{M}, t) \models \varphi$ for all $t \in S$ such that $(s, t) \in \mathcal{K}_i$. The dual operators P_i are defined via $P_i\varphi = \neg K_i \neg\varphi$. Standard operations in propositional logic such as $\varphi \to \psi$ are derived in the usual way [49]. By abuse of notation, we write $\models \varphi$ if $(\mathcal{M}, s) \models \varphi$ for all $s \in S$.

Depending on a number of axioms placed on the modal operators K_i and P_i, one has rich interpretations for K_i and P_i. For instance, the Knowledge Axiom [36]

$$\models K_i\varphi \to \varphi \tag{7}$$

and the Introspection Axiom [36]

$$\models K_i\varphi \to K_iK_i\varphi \tag{8}$$

together suggest the interpretation of $K_i\varphi$ and $P_i\varphi$: *agent i knows φ* and *agent j considers φ possible*, respectively. On the other hand, if (7) not hold, but the Consistency Axiom [36] does hold,

$$\models \neg K_i(\texttt{false}), \tag{9}$$

$K_i\varphi$ and $P_i\varphi$ could be interpreted as: *agent i believes φ* and *agent i does not disbelieve (i.e., is undecided about) φ.*[3]

The set of all finite formulae inductively obtained by Φ is called the *language* denoted $\mathcal{L}(\Phi)$ (omitting the Φ where understood). For $\varphi \in \mathcal{L}$, and model $\mathcal{M}$, the *intent* of φ is the subset

$$\varphi^{\mathcal{M}} = \{s \in S \mid (\mathcal{M}, s) \models \varphi\}.$$

Formulae $\varphi, \psi \in \mathcal{L}$ are *semantically equivalent* if $\varphi^{\mathcal{M}} = \psi^{\mathcal{M}}$. Semantic equivalence is an equivalence relation, written $\varphi \equiv \psi$, with $\mathcal{L}_{\equiv}(\mathcal{M})$ denoting the set of equivalence classes in $\mathcal{L}$ up to semantic equivalence.

B. Semantic diffusion

Our goal is to adapt the technology of sheaves of lattices and Laplacians to Kripke semantics over a network of agents. One simple approach is to use powerset lattices 2^S of states S of a frame. Let $G = (V, E)$ be a network and $\mathbf{F}$ a frame. Define the *semantic sheaf* $\mathcal{I}$ over G so that the data over each vertex and edge is precisely 2^S. The following is crucial to define the structure-preserving maps:

Definition 7: For $\mathbf{F} = (S, \mathcal{K}_1, \dots, \mathcal{K}_n)$ a frame and for each $i \in \{0, 1, \dots, n\}$, there is residual pair

$$2^S \underset{\mathcal{K}_i^{\forall}}{\overset{\mathcal{K}_i^{\exists}}{\rightleftarrows}} 2^S \tag{10}$$

given by the formulae

$$\mathcal{K}_i^{\exists}(\sigma) = \{t \in S \mid \exists s \in \sigma,\ (s, t) \in \mathcal{K}_i\},$$
$$\mathcal{K}_i^{\forall}(\sigma) = \{s \in S \mid \forall t \in S, (s, t) \in \mathcal{K}_i \Rightarrow t \in \sigma\}.$$

Lemma 3: Each map $\mathcal{K}_i^{\exists} : 2^S \to 2^S$ is join-preserving and $\left(\mathcal{K}_i^{\exists}\right)^+ = \mathcal{K}_i^{\forall}$.

Proof: First,

$$\mathcal{K}_i^{\exists}(\sigma \cup \sigma') = \{t \in S \mid \exists\ s \text{ or } s' \in \sigma,\ (s, t) \text{ or } (s', t) \in \mathcal{K}_i\}.$$

Second,

$$\begin{aligned}\left(\mathcal{K}_i^{\exists}\right)^+(\sigma) &= \bigcup\{\alpha \in 2^S \mid \mathcal{K}_i^{\exists}(\alpha) \subseteq \sigma\} \\ &= \{s \in S \mid \forall t \in S, (s, t) \in \mathcal{K}_i \Rightarrow t \in \sigma\} \\ &= \mathcal{K}_i^{\forall}(\sigma).\end{aligned}$$

■

Such a sheaf of powerset lattices has an asynchronous Tarski Laplacian and a corresponding heat flow. The following definitions are straight translations from §IV.

Definition 8: Suppose $\mathbf{F} = (S, \mathcal{K}_1, \dots, \mathcal{K}_n)$ is a frame and $\mathcal{I}$ a sheaf of powersets 2^S over a network $G = (V, E)$ where $V = \{1, 2, \dots, n\}$. Let $\tau : \{0, 1, 2, \dots\} \to V$ be a firing sequence. The (asynchronous) *semantic Laplacian* is the operator acting on $\boldsymbol{\sigma} \in \prod_{i \in V} 2^S$ via:

$$(L_t\boldsymbol{\sigma})_i = \bigcap_{j \in N_i \cap \tau_t} \mathcal{K}_i^{\forall}\mathcal{K}_j^{\exists}(\sigma_j). \tag{11}$$

The associated *heat flow* is the dynamical system

$$\boldsymbol{\sigma}[t+1] = (\mathrm{id} \wedge L_t)\boldsymbol{\sigma}[t] \tag{12}$$

Our main result follows directly from Theorem 1, interpreted in the language of this section.

Theorem 2: Suppose $G = (V, E)$ is a network with firing sequence τ satisfying liveness. Let $\mathcal{M} =$

[3]Belief is assumed to satisfy the law of excluded middle: believing something does not make it true.

$(S, \mathcal{K}_1, \ldots, \mathcal{K}_N, \Phi, \pi)$ be a Kripke model. Then, the sections of $\mathcal{I}$ are exactly time-independent solutions to the heat flow (6).

These sections are interpretable as *possibility consensus* assignments of the model $\mathcal{M}$ on G. The assignment of formulae $(\varphi_i)_{i\in V}$ satisfies, for each edge $ij \in E$, $P_i\varphi_i \equiv P_j\varphi_j$. This does not mean that the formulae are in consensus as identical formulae; rather, they are semantically equivalent in the language $\mathcal{L}$.

C. Syntactic diffusion

The following lemma motivates why local residuations $\mathcal{K}_i^{\exists} \dashv \mathcal{K}_i^{\forall}$ induce a flow of knowledge.

Lemma 4: Suppose $\mathcal{M} = (\mathbf{F}, \pi)$ is a model and $\varphi \in \mathcal{L}(\mathcal{M})$. Then, the following hold

$$\mathcal{K}_i^{\forall}\left(\varphi^{\mathcal{M}}\right) = (K_i\varphi)^{\mathcal{M}}, \tag{13}$$

$$\mathcal{K}_i^{\exists}\left(\varphi^{\mathcal{M}}\right) = (P_i\varphi)^{\mathcal{M}}. \tag{14}$$

Proof: Writing

$$\mathcal{K}_i^{\forall}\left(\varphi^{\mathcal{M}}\right) = \{s \in S \mid \forall t \in S, (s,t) \in \mathcal{K}_i \Rightarrow (\mathcal{M}, t) \models \varphi\},$$

we prove (13). For (14), evaluating $\mathcal{K}_i^{\exists}(\varphi^{\mathcal{M}})$ yields

$$\{t \in S \mid \exists s \text{ such that } (\mathcal{M}, s) \models \varphi \text{ with } (s,t) \in \mathcal{K}_i\}.$$

On the other hand, evaluating $(P_i\varphi)^{\mathcal{M}}$ yields

$$\begin{aligned}(\neg K_i \neg\varphi)^{\mathcal{M}} &= \left((K_i\neg\varphi)^{\mathcal{M}}\right)^c \\ &= \{s \mid \forall\ t \text{ such that } (s,t) \in \mathcal{K}_i,\ (\mathcal{M}, t) \not\models \varphi\}^c \quad .\end{aligned}$$

■

The following simple observation together with Lemma 4 allows us to freely go back and forth between syntax and semantics, opening the way for syntactic diffusion dynamics.

Lemma 5: Suppose $\{\varphi\}_i$ is a finite set of formula in $\mathcal{L}(\mathcal{M})$. Then, $\left(\bigvee_{i\in I}\varphi_i\right)^{\mathcal{M}} = \bigcup_i \varphi_i^{\mathcal{M}}$ and $\left(\bigwedge_i \varphi_i\right)^{\mathcal{M}} = \bigcap_i \varphi_i^{\mathcal{M}}$.

Definition 9: Suppose $\boldsymbol{\varphi} = (\varphi_i)_{i\in V}$ is a tuple of formulae in $\mathcal{M}$. The (asynchronous) *syntactic Laplacian* acts on assignments as:

$$(L_t\boldsymbol{\varphi})_i = \bigwedge_{j\in N_i \cap \tau_t} K_i P_j \varphi_j. \tag{15}$$

This is a straight translation of the semantic Tarski Laplacian, using Lemmas 4 and 5.

D. Dual Laplacians

The classical meet-join duality in lattices pushes through to all other structures built therefrom. In particular, the Tarski (and thus semantic and syntactic) Laplacians come in dual variants, implicating how syntactic and semantic consensus is interpreted.

Definition 10: The *dual* Tarski, semantic, and syntactic Laplacians are given by, respectively:

$$(L^*\mathbf{x})_i = \bigvee_{j\in N_i} \mathcal{F}^+_{i\trianglelefteqslant j}\mathcal{F}_{j\trianglelefteqslant ij}(x_j) \tag{16}$$

$$(L^*\boldsymbol{\sigma})_i = \bigcup_{j\in N_i} \mathcal{K}_i^{\exists}\mathcal{K}_j^{\forall}(\sigma_j) \tag{17}$$

$$(L^*\boldsymbol{\varphi})_i = \bigvee_{j\in N_i} P_i K_j \varphi_j. \tag{18}$$

The corresponding asynchronous dual Laplacians L_t^* intersect neighboring vertices with those in a firing sequence τ.

The *dual heat flow* iterates the operator $\mathrm{id} \vee L_t^*$. Convergence results to sections remain. These sections are interpretable as *knowledge consensus* assignments of the model $\mathcal{M}$ on G. The assignment of formulae $(\varphi_i)_{i\in V}$ satisfies, for each edge $ij \in E$, $K_i\varphi_i \equiv K_j\varphi_j$.

VI. Examples

In this section, we supply an example of a semantic sheaf modeling knowledge gain; we also provide a numerical experiment demonstrating the correctness of asynchronous heat flow (6) as well as convergence behavior. The *Lyapunov function* acts on assignments in (6) via

$$V(\mathbf{x}) = \sum_{ij\in E} d\left(\mathcal{F}_{i\trianglelefteqslant ij}(x_i), \mathcal{F}_{j\trianglelefteqslant ij}(x_j)\right); \tag{19}$$

each d is a distance function on $\mathcal{F}(ij)$. If d is a metric, then $V(\mathbf{x}) \geqslant 0$ and $V(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ is a section. Below, the metric is taken to be the well-known *Jaccard distance* between subsets.

A. Threat detection

In this example, we model the knowledge of N "smart" sensors tasked with detecting M possible targets $\Phi = \{A_1, A_2, \ldots, A_M\}$. Sensors are equipped with two-way links forming a communication pattern modeled by an undirected graph $G = (V, E)$. Each sensor i has a local state s_i and a set of possible local states S_i which may vary from sensor to sensor. For instance, s_i could represent a risk posed at a particular location. The atomic propositions Φ are each interpreted as *threat A is present*. A global state $s \in \prod_{i=1}^N S_i$ is a tuple of all local states. The ground truth on whether or not threat A is present is determined by a map $\pi : \prod_{i=1}^N S_i \to 2^{\Phi}$. To model the knowledge of individual sensors, the right choice of Kripke relation $\mathcal{K}_i$ is the equivalence relation on $S = \prod_{i=1}^N S_i$ given by $s \sim_i t$ if and only if $s_i = t_i$, reflecting the intuition that a sensor i would know A is present – or, in general, any formula in $\mathcal{L}(\Phi)$ – precisely when, given a current the local state $s_i \in S_i$, A was present for every possible global state that had s_i as a local state.

B. Numerical experiments

In the following simulation, we demonstrate the validity of the gossip algorithm by: 1) generating a random geometric graph ($N = 40, r = 0.08$); 2) assigning (Kripke) relation on S (with10 states), one for each node, by wiring each $(x, y) \in \mathcal{K}_i$ randomly ($p = 0.9$ if $x = y$, $p = 0.1$ otherwise); 3) selecting several random firing sequence $\tau : \{0, 1, \ldots\} \to 2^V$ which determine which nodes broadcast to their neighbors at time t. For a random initial assignment $\boldsymbol{\sigma}[0]$, we run the

heat flow dynamics using the dual asynchronous semantic Laplacian (17) for each firing sequence (Fig. 1).

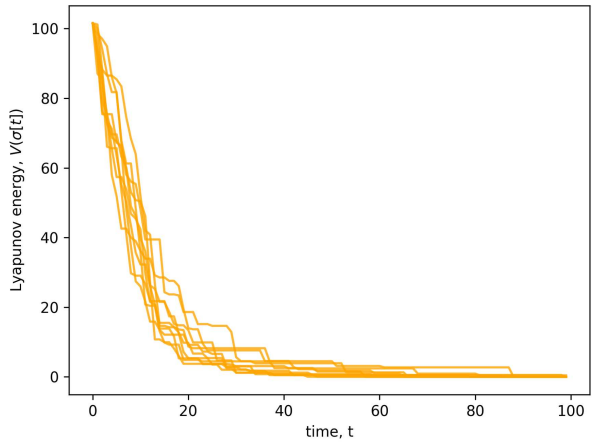

Fig. 1. The Lyapunov energies of each iteration of the gossip algorithm computed from the dual asynchronous semantic Laplacian of the Kripke frame defined above. Each trial is a different firing sequence τ.

REFERENCES

[1] M. Taylor, "Towards a mathematical theory of influence and attitude change," *Human Relations*, vol. 21, no. 2, pp. 121–139, 1968.

[2] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[3] V. M. Preciado, M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Distributed control of the laplacian spectral moments of a network," in *Proceedings of the 2010 American Control Conference*, pp. 4462–4467, IEEE, 2010.

[4] V. Preciado and G. Verghese, "Synchronization in generalized Erdös Rényi networks of nonlinear oscillators," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 4628–4633, 2005.

[5] R. Sepulchre, D. Paley, and N. Leonard, "Collective motion and oscillator synchronization," in *Cooperative control*, pp. 189–205, Springer, 2005.

[6] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic control*, vol. 52, no. 5, pp. 863–868, 2007.

[7] A. Singer and H.-T. Wu, "Vector Diffusion Maps and the Connection Laplacian," *Communications in Pure and Applied Mathematics*, vol. 65, no. 8, 2012.

[8] A. S. Bandeira, A. Singer, and D. A. Spielman, "A cheeger inequality for the graph connection laplacian.," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 4, pp. 1611–1630, 2013.

[9] S. E. Tuna, "Synchronization under matrix-weighted laplacian.," *Automatica*, vol. 73, pp. 76–81, 2016.

[10] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[11] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[12] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[13] L. Ruiz, F. Gama, and A. Ribeiro, "Graph neural networks: architectures, stability, and transferability," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 660–682, 2021.

[14] G. E. Bredon, *Sheaf Theory*. No. 170 in Graduate Texts in Mathematics, Springer, 2 ed., 1997.

[15] V. Janis, S. Montes, B. Seselja, and A. Tepavcevic, "Poset-valued preference relations," *Kybernetika*, vol. 51, no. 5, pp. 747–764, 2015.

[16] T. Anevlavis, M. Philippe, D. Neider, and P. Tabuada, "Being correct is not enough: efficient verification using robust linear temporal logic," *ACM Transactions on Computational Logic (TOCL)*, vol. 23, no. 2, pp. 1–39, 2022.

[17] M. Püschel, B. Seifert, and C. Wendler, "Discrete signal processing on meet/join lattices," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3571–3584, 2021.

[18] 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.

[19] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," 2003.

[20] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[21] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[22] B. M. Nejad, S. A. Attia, and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies," in *2009 XXII International Symposium on Information, Communication and Automation Technologies*, pp. 1–7, IEEE, 2009.

[23] A. Tahbaz-Salehi and A. Jadbabaie, "A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 4664–4669, IEEE, 2006.

[24] X. Zheng, *The lattice agreement problem in distributed systems*. PhD thesis, 2021.

[25] J. A. Goguen, "Sheaf semantics for concurrent interacting objects," *Mathematical Structures in Computer Science*, vol. 2, no. 2, pp. 159–191, 1992.

[26] R. Ghrist and S. Krishnan, "A topological max-flow-min-cut theorem," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 815–818, IEEE, 2013.

[27] M. Moy, R. Cardona, R. Green, J. Cleveland, A. Hylton, and R. Short, "Path optimization sheaves," *arXiv preprint arXiv:2012.05974*, 2020.

[28] J. Hansen and R. Ghrist, "Opinion dynamics on discourse sheaves," *SIAM Journal on Applied Mathematics*, vol. 81, no. 5, pp. 2033–2060, 2021.

[29] M. Robinson, "Sheaves are the canonical data structure for sensor integration," *Information Fusion*, vol. 36, pp. 208–224, 2017.

[30] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[31] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[32] Y. Kantaros, M. Guo, and M. M. Zavlanos, "Temporal logic task planning and intermittent connectivity control of mobile robot networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4105–4120, 2019.

[33] A. Rodionova, L. Lindemann, M. Morari, and G. Pappas, "Time-robust control for stl specifications," pp. 572–579, Institute of Electrical and Electronics Engineers,, 2021.

[34] H. Riess, Y. Kantaros, G. Pappas, and R. Ghrist, "A temporal logic-based hierarchical network connectivity controller," in *2021 Proceedings of the Conference on Control and its Applications*, pp. 17–24, SIAM, 2021.

[35] Y. Kantaros, M. Malencia, V. Kumar, and G. J. Pappas, "Reactive temporal logic planning for multiple robots in unknown environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11479–11485, IEEE, 2020.

[36] R. Fagin, J. Y. Halpern, Y. Moses, and M. Vardi, *Reasoning about knowledge*. MIT press, 2004.

[37] R. Ghrist and H. Riess, "Cellular sheaves of lattices and the tarski laplacian," *Homology, Homotopy & Applications*, 2022.

[38] R. A. Cuninghame-Green and P. Butkovic, "The equation $A \otimes x = B \otimes y$ over (max,+)," *Theoretical Computer Science*, vol. 293, no. 1, pp. 3–12, 2003.

[39] R. Cuninghame-Green, "Minimax algebra and applications," *Fuzzy Sets and Systems*, vol. 41, no. 3, pp. 251–267, 1991.

[40] R. Wille, "Restructuring lattice theory: An approach based on hierarchies of concepts," in *Ordered Sets*, pp. 445–470, Springer, 1982.

[41] B. A. Davey and H. A. Priestley, *Introduction to lattices and order*. Cambridge university press, 2002.

[42] G. Curello and L. Sinander, "The preference lattice," *arXiv preprint arXiv:1902.07260*, 2019.
[43] C. Shannon, "The lattice theory of information," *Transactions of the IRE professional Group on Information Theory*, vol. 1, no. 1, pp. 105–107, 1953.
[44] E. Gilbert, "Lattice theoretic properties of frontal switching functions," *Journal of Mathematics and Physics*, vol. 33, no. 1-4, pp. 57–67, 1954.
[45] A. D. Shepard, *A cellular description of the derived category of a stratified space*. PhD thesis, Brown University, 1985.
[46] J. M. Curry, *Sheaves, cosheaves and applications*. PhD thesis, University of Pennsylvania, 2014.
[47] T. S. Blyth and M. F. Janowitz, *Residuation theory*. Elsevier, 2014.
[48] O. Ore, "Galois connexions," *Transactions of the American Mathematical Society*, vol. 55, no. 3, pp. 493–513, 1944.
[49] E. Mendelson, *Introduction to mathematical logic*. Chapman and Hall/CRC, 2009.

APPENDIX

A. Proof of Theorem 1

Suppose first that $\mathbf{x}[t] \in \Gamma(\mathcal{F})$ is a section. Then, for all $ij \in E$,

$$\mathcal{F}_{i \trianglelefteq ij}(x_i[t]) = \mathcal{F}_{j \trianglelefteq ij}(x_j[t]).$$

Hence, by Lemma 2,

$$\begin{aligned}(L_t\mathbf{x}[t])_i &= \bigwedge_{j \in N_i \cap \tau_t} \mathcal{F}^+_{i \trianglelefteq j} \mathcal{F}_{j \trianglelefteq ij}(x_j[t]) \\ &= \bigwedge_{j \in N_i \cap \tau_t} \mathcal{F}^+_{i \trianglelefteq j} \mathcal{F}_{i \trianglelefteq ij}(x_i[t]) \\ &\succcurlyeq x_i[t].\end{aligned}$$

Then, $x_i[t+1] = (L_t\mathbf{x}[t])_i \wedge x_i[t] = x_i[t]$. Hence, $x_i[t+1] = x_i[t]$ for all $i \in V$. Inducting in t, sections are time-independent.

Conversely, suppose $\mathbf{x}[t] = \mathbf{x}[0]$ is a time-independent solution. Then, by (6),

$$(L_t\mathbf{x}[0])_i \succcurlyeq x_i[0] \tag{20}$$

for all $i \in V$ and all $t \in \{0, 1, 2, \dots\}$. Thus, for all $i \in V$ and $j \in N_i \cap \tau_t$,

$$\begin{aligned}&\mathcal{F}^+_{i \trianglelefteq j} \mathcal{F}_{j \trianglelefteq ij}(x_j[0]) \\ &\quad\succcurlyeq \bigwedge_{j \in N_i \cap \tau_t} \mathcal{F}^+_{i \trianglelefteq j} \mathcal{F}_{j \trianglelefteq ij}(x_j[0]) \\ &\quad\succcurlyeq x_i[0],\end{aligned}$$

using the definition of the asynchronous Tarski Laplacian. Applying Lemma 2 yields:

$$\mathcal{F}_{j \trianglelefteq ij}(x_j[0]) \succcurlyeq \mathcal{F}_{i \trianglelefteq ij}(x_i[0]). \tag{21}$$

Suppose (21) holds for a particular $i \in V$ and a $j \in N_i \cap \tau_t$. By liveness, there exist $t' \geqslant t$ such that $i \in \tau_{t'}$ so that $i \in N_j \cap \tau_t$. In particular, there is a smallest $t' > t$ with

$$\mathcal{F}_{j \trianglelefteq ij}(x_j[t']) \preccurlyeq \mathcal{F}_{i \trianglelefteq ij}(x_i[t']). \tag{22}$$

By hypothesis, $x_i[0] = x_i[t']$ for all $i \in V$. Hence, equations (21) and (22) imply $\mathcal{F}_{i \trianglelefteq ij}(x_i[t]) = \mathcal{F}_{j \trianglelefteq ij}(x_j[t])$ for all t. Therefore, $\mathbf{x}[t]$ is a section.