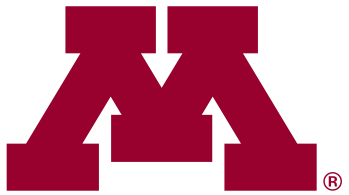
Sheaf Representation Learning:

Generalizing Graph Representation Learning with Cellular Sheaves

Thomas Gebhart

University of Minnesota, Department of Computer Science



Outline

- Motivation / Graph Representation Learning
- Cellular Sheaves
- Sheaf Representation Learning
- Applications

Graph Representation Learning

No universal definition, but for our purposes:

Given a graph $G = (V, E)$, possibly with node-valued signals $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, learn:

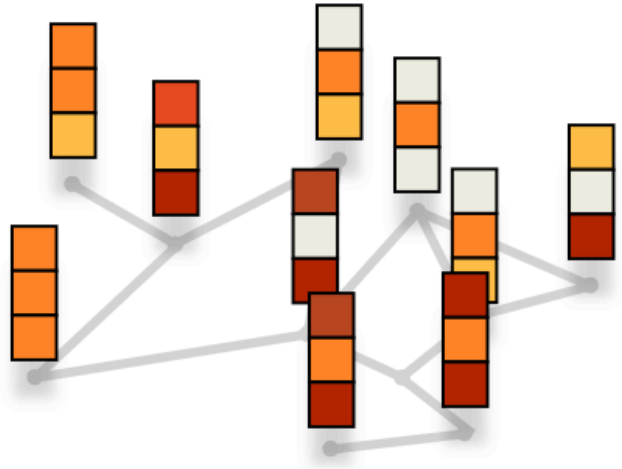
$$f : V \to \mathbb{R}^m \text{ (node embedding)}$$

or

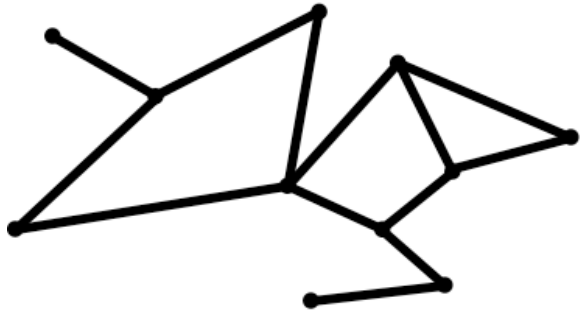
$$f : \mathcal{X}(G, \mathbb{R}^d) \to \mathcal{X}(G, \mathbb{R}^m) \text{ (graph signal processing)}$$

Graph Representation Learning

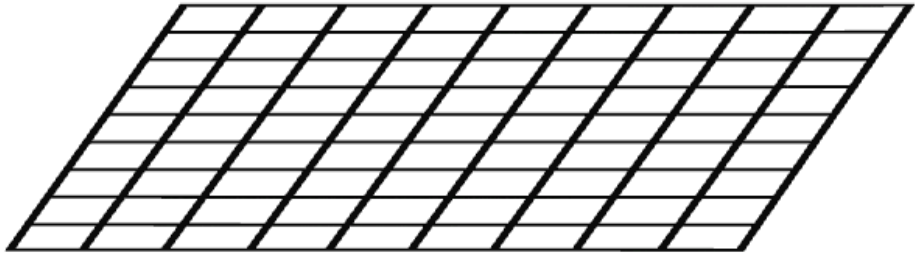
$\mathcal{X}(G, \mathbb{R}^d)$ as a Hilbert space of functions on $G = (V, E)$:



$\mathcal{X}(G, \mathbb{R}^d)$



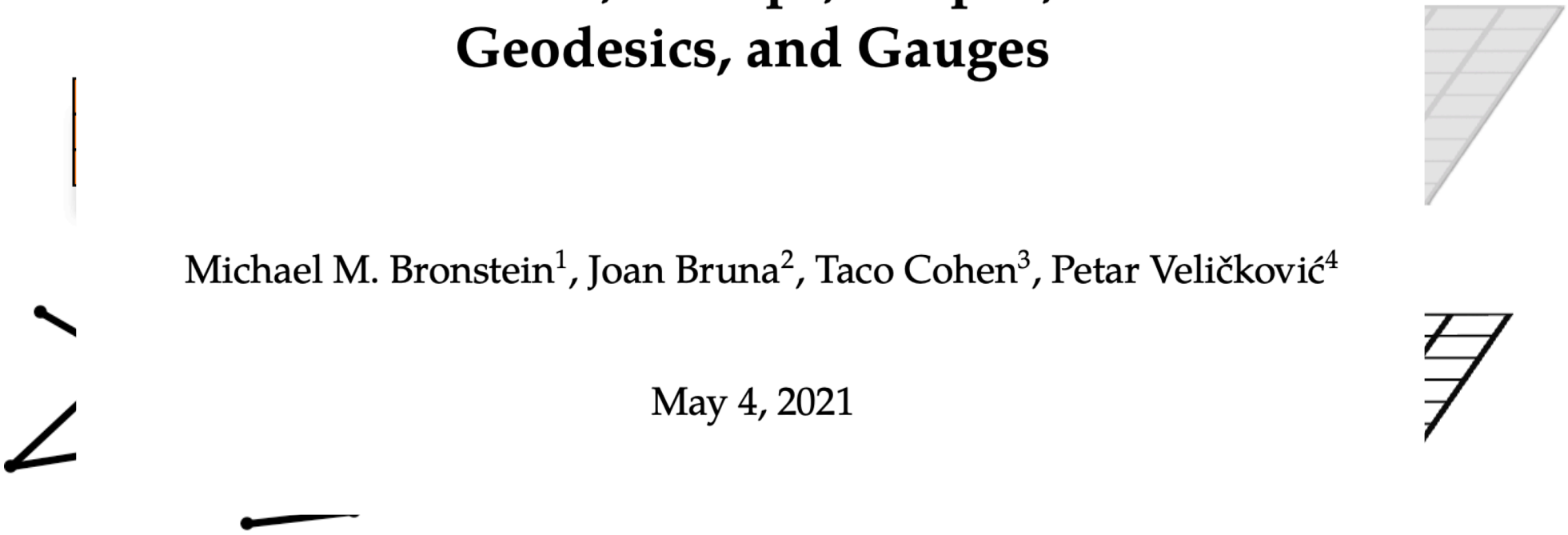
G



Graph Representation Learning

$\mathcal{X}(G, \mathbb{R}^d)$ as a Hilbert space of functions on $G = (V, E)$:

Geometric Deep Learning
Grids, Groups, Graphs,
Geodesics, and Gauges



Michael M. Bronstein[1], Joan Bruna[2], Taco Cohen[3], Petar Veličković[4]

May 4, 2021

Graph Representation Learning

Node embedding

Laplacian Eigenmaps

Node2Vec

DeepWalk

Knowledge Graph Embedding

Applications

Graph visualization, product discovery (clustering), document classification (node classification), Knowledge graph completion (link prediction)

Signal Processing

Spectral Clustering

Filtering / Shifting / Wavelets

Graph Neural Networks

Applications

Molecule classification (graph classification), document classification (node classification), Solubility prediction (graph regression)

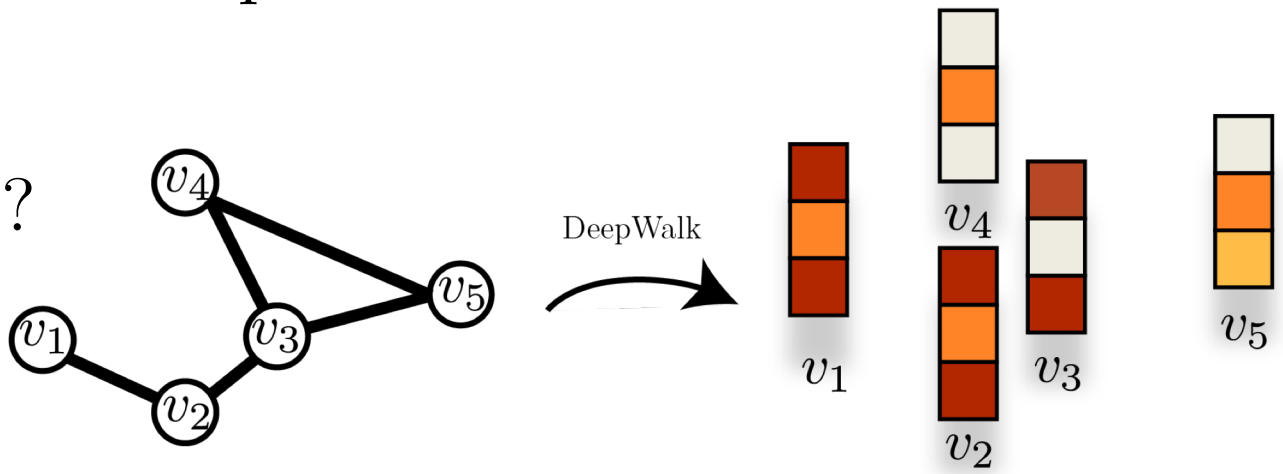
DeepWalk

Use the distributional hypothesis to embed nodes using Skip-gram with Negative Sampling (SGNS).

Intuitively, nodes sharing many common neighbors should be nearby in embedding space.

This embedding process performs an implicit matrix factorization (Levy and Goldberg, 2014).

Which matrix are we factorizing?



Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.

DeepWalk

SGNS implicitly factors matrix $\boldsymbol{M}^{\mathrm{W2V}}$ with entries given the point-wise mutual information between word w_i and its context c_j:

$$\begin{aligned}\boldsymbol{M}_{i,j}^{\mathrm{W2V}} &= \log\left(\frac{P(w_i, c_j)}{P(w_i)P(c_j)}\right) - \log b \\ &= \log\left(\frac{\#(w_i, c_j)|\mathcal{D}|}{\#(w_i)\#(c_j)}\right) - \log b\end{aligned}$$

Where $\mathcal{D}$ is the corpus of word-contexts (sentences) with a given window size and b is the negative sampling ratio.

How does this translate to graphs?

Levy, Omer and Goldberg, Yoav. "Neural word embedding as implicit matrix factorization". *Advances in Neural Information Processing Systems.* (2014).

DeepWalk

Let $G = (V, E, \boldsymbol{A})$ be non-bipartite, connected, and undirected.

Then $P(v) = \frac{d_v}{\nu(G)}$ is the unique stationary distribution of a random walk on G, where $\nu(G)$ is the volume.

Let $\mathcal{D}(T)$ contain the node pairs crossed within a T-sized window of an L-step random walk.

Further subdivide $\mathcal{D}(T)$ by r-step directed reachability:

$$\mathcal{D}(T) = \bigcup_{r=1}^{T} (\mathcal{D}_{\overrightarrow{r}} \cup \mathcal{D}_{\overleftarrow{r}})$$

$$\mathcal{D}_{\overrightarrow{r}} = \{(u, v) \mid (u, v) \in \mathcal{D}(T), u = v_j, v = v_{j+r}\}$$

$$\mathcal{D}_{\overleftarrow{r}} = \{(u, v) \mid (u, v) \in \mathcal{D}(T), u = v_{j+r}, v = v_j\}$$

DeepWalk

$$\mathcal{D}(T) = \bigcup_{r=1}^{T} (\mathcal{D}_{\overrightarrow{r}} \cup \mathcal{D}_{\overleftarrow{r}})$$

$$\mathcal{D}_{\overrightarrow{r}} = \{(u, v) \mid (u, v) \in \mathcal{D}(T), u = v_j, v = v_{j+r}\}$$

$$\mathcal{D}_{\overleftarrow{r}} = \{(u, v) \mid (u, v) \in \mathcal{D}(T), u = v_{j+r}, v = v_j\}$$

Let $\#(u, v)_{\overrightarrow{r}}$ count the number of times the pair (u, v) is seen in $\mathcal{D}_{\overrightarrow{r}}$ and $\#(u, v)_{\overleftarrow{r}}$ the number of times (u, v) appears in $\mathcal{D}_{\overleftarrow{r}}$.

Denote the transition matrix of this random walk by $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$.

Then as walk length $L \to \infty$:

$$\frac{\#(u, v)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|} \xrightarrow{p} \frac{d_u}{\nu(G)}(\boldsymbol{P}^r)_{u,v} \quad \text{and} \quad \frac{\#(u, v)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|} \xrightarrow{p} \frac{d_v}{\nu(G)}(\boldsymbol{P}^r)_{v,u}$$

DeepWalk

Similarly, the joint distribution of observations of pair (u, v) in the walks approaches:

$$\frac{\#(u,v)}{|\mathcal{D}|} \xrightarrow{p} \frac{1}{2T}\sum_{r=1}^{T}\left(\frac{d_u}{\nu(G)}(\boldsymbol{P}^r)_{u,v} + \frac{d_v}{\nu(G)}(\boldsymbol{P}^r)_{v,u}\right)$$

Combining this joint with the marginals, we see:

$$\boldsymbol{M}^{\mathrm{DW}} = \frac{\#(u,v)|\mathcal{D}|}{\#(u)\#(v)} \xrightarrow{p} \frac{\nu(G)}{2T}\left(\frac{1}{d_v}\sum_{r=1}^{T}(\boldsymbol{P}^r)_{u,v} + \frac{1}{d_u}\sum_{r=1}^{T}(\boldsymbol{P}^r)_{v,u}\right)$$

$$= \log\left(\frac{\nu(G)}{T}(\sum_{r=1}^{T}\boldsymbol{P}^r)\boldsymbol{D}^{-1}\right)$$

DeepWalk

$$\boldsymbol{M}^{\mathrm{DW}}(T) = \log\left(\frac{\nu(G)}{T}(\sum_{r=1}^{T}\boldsymbol{P}^r)\boldsymbol{D}^{-1}\right)$$

This looks like it should converge to something familiar...
What happens if we take window size $T \to \infty$?

Chanpuriya and Musco (2020) showed that this will indeed converge to a form determined by the (inverse) graph Laplacian:

$$\lim_{T\to\infty} \boldsymbol{M}^{\mathrm{DW}}(T) = \nu(G)\boldsymbol{D}^{-1/2}(\tilde{\boldsymbol{L}}^{\dagger} - \boldsymbol{I})\boldsymbol{D}^{-1/2} + \mathbf{1}$$

where

$$\tilde{\boldsymbol{L}} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$$

Chanpuriya, Sudhanshu, and Cameron Musco. "Infinitewalk: Deep network embeddings as Laplacian embeddings with a nonlinearity." *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020.

DeepWalk

Summary:

Asymptotic behavior of this "deep" node embedding method is driven by the graph Laplacian, mimicking spectral embedding.

Motivating Questions:

What if the initial distribution on nodes is $(d > 1)$-dimensional, resulting in vectors $\boldsymbol{x}_v$?

What if the passage between nodes (u, v) is mediated by some contextual transformation $\boldsymbol{\mathcal{F}}_{u \to v}$ from $\boldsymbol{x}_u$ to $\boldsymbol{x}_v$?

Graph Convolutional Neural Networks

The Graph Convolutional Neural Network (GCN) is a popular graph neural network architecture.

Used primarily for node classification/regression.

Assumes data comes in the form of a graph $G = (V, E, \boldsymbol{A})$, with node signals $\boldsymbol{X} \in \mathbb{R}^{n \times d_0}$, and $n = |V|$.

A GCN layer seeks to transform $\boldsymbol{X}^{(l-1)} \mapsto \boldsymbol{X}^{(l)}$ so that $\boldsymbol{X}^{(l)}$ contains node representations which better inform a task (classification, regression, etc.)

Summarily, a GCN layer is a function $f^{\mathrm{GCN}} : \mathcal{X}(G, \mathbb{R}^{d_{l-1}}) \to \mathcal{X}(G, \mathbb{R}^{d_l})$

GCN

$$f^{\mathrm{GCN}}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{W}) = \sigma\left(\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{X}\boldsymbol{W}\right)$$
$$= \sigma\left((\boldsymbol{I} - \tilde{\boldsymbol{L}})\boldsymbol{X}\boldsymbol{W}\right)$$

$\boldsymbol{I} - \tilde{\boldsymbol{L}}$ is a diffusion operator, akin to "heat diffusion" on G.

Repeated application $\boldsymbol{x}^l = (\boldsymbol{I} - \tilde{\boldsymbol{L}})\boldsymbol{x}^{l-1}$ minimizes the Dirichlet Energy $\epsilon(\boldsymbol{x}, \boldsymbol{A})$:

$$\epsilon(\boldsymbol{x}, \boldsymbol{A}) = \boldsymbol{x}^\top \tilde{\boldsymbol{L}}\boldsymbol{x}$$
$$= \frac{1}{2}\sum_{i,j} \tilde{a}_{ij}(x_i - x_j)^2$$

GCN

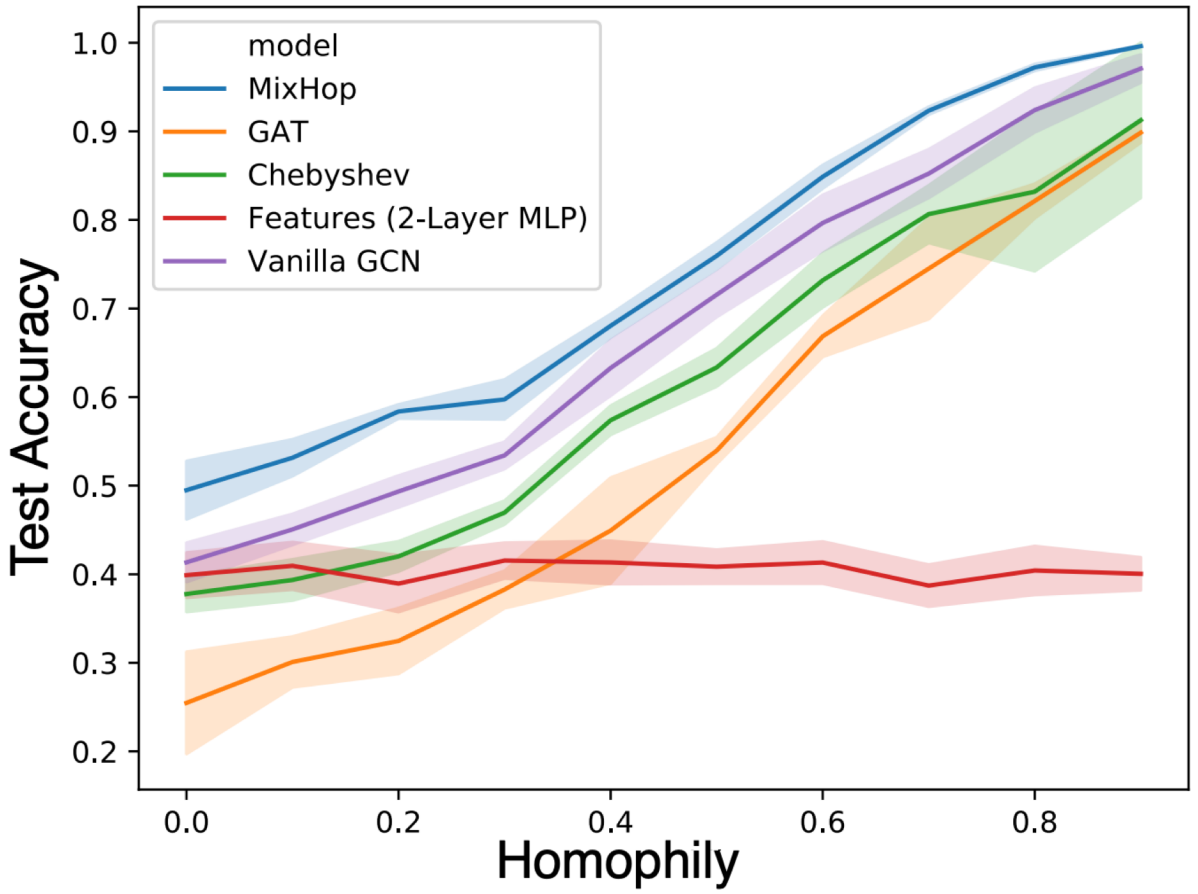
$$\epsilon(\boldsymbol{x}, \boldsymbol{A}) = \frac{1}{2}\sum_{i,j}\tilde{a}_{ij}(x_i - x_j)^2$$

If we fix $\boldsymbol{W} = \boldsymbol{I}$, stacking $\cdots \circ f^{\mathrm{GCN}} \circ f^{\mathrm{GCN}}$ layers minimizes $\epsilon(\boldsymbol{x}, \boldsymbol{A})$.

Thus a "deep" GCN of this type will learn smooth node features: a degree-weighted local averaging dictated by the graph topology.

These features will also have a homophily bias.

Same issues when $\boldsymbol{W}$ learnable.



Abu-El-Haija, Sami, et al. "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing." *international conference on machine learning*. PMLR, 2019.

Graph Laplacian

A gradient operator differencing values between adjacent nodes.

Many definitions:

$$\tilde{\boldsymbol{L}} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$$

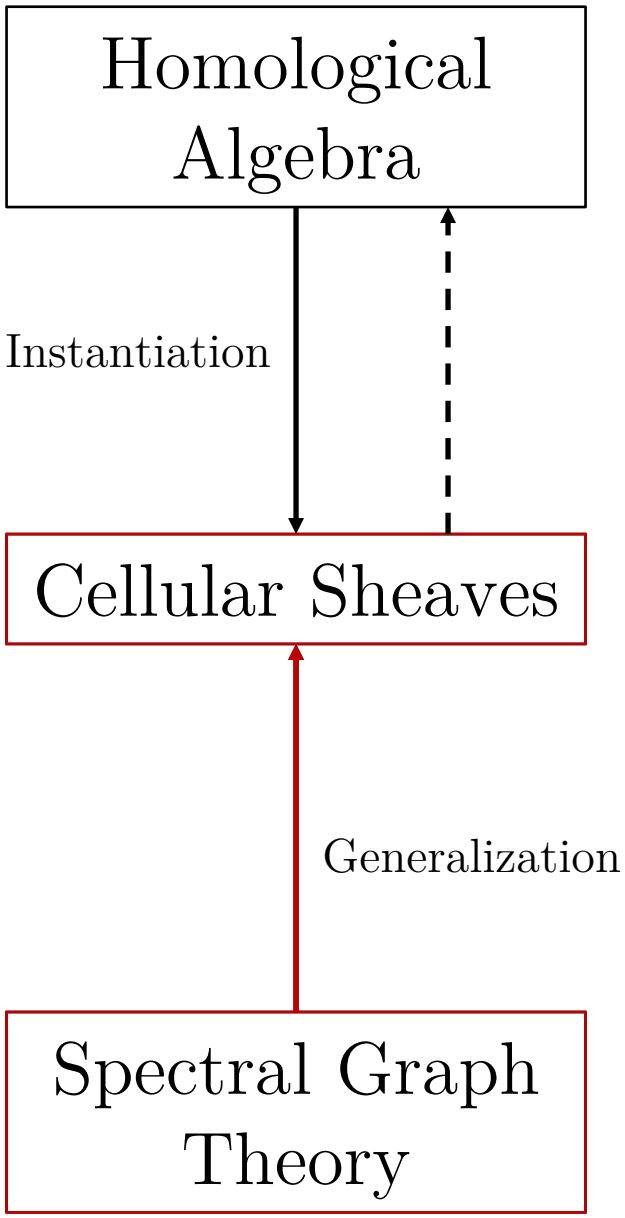
$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}^{\top}$$

coboundary operator

$$\boldsymbol{L}\boldsymbol{x} = \frac{1}{2}\sum_{u,v} a_{u,v}(x_u - x_v)^2$$

Smoothness attainable only through edge-wise scaling of node values: lack of expressibility and control.

Similarly, edge “typing” is implementable only through choice of edge weight.


Homological Algebra
Instantiation
Cellular Sheaves
Generalization
Spectral Graph Theory

Sheaves on Graphs

A cellular sheaf $\mathcal{F}$ on a graph $G = (V, E)$ consists of the following data:

- a vector space $\mathcal{F}(v)$ for each vertex $v \in V$,
- a vector space $\mathcal{F}(e)$ for each edge $e \in E$,

Stalks

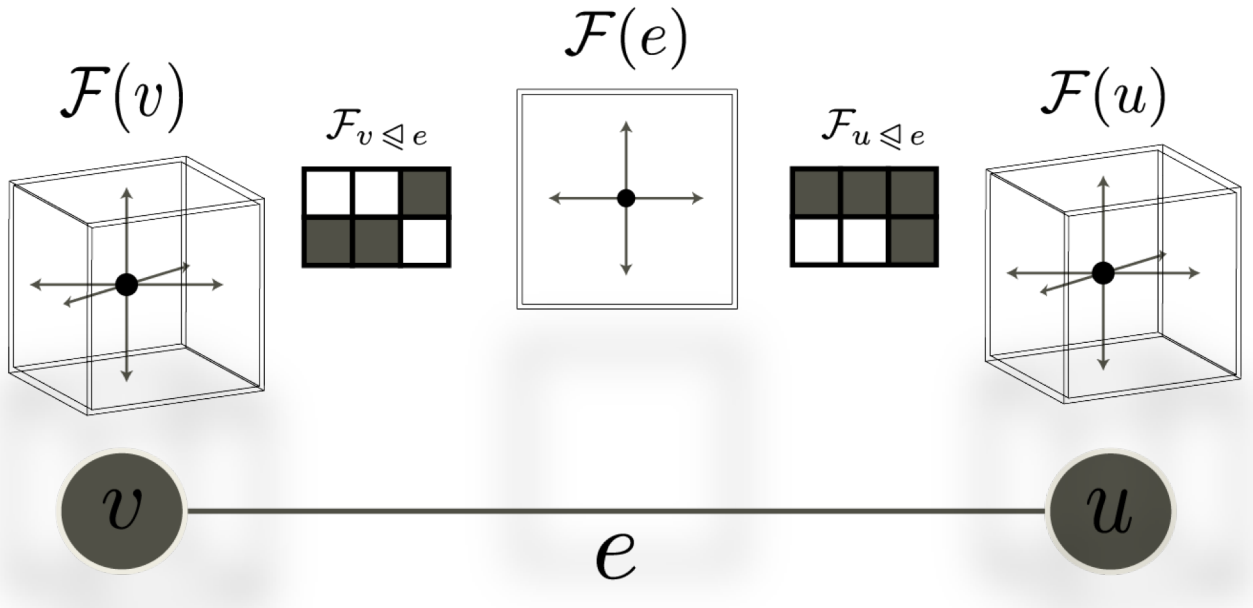
- a linear map $\boldsymbol{\mathcal{F}}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident vertex-edge pair $v \trianglelefteq e$ of G.

Restriction Maps

Sheaves on Graphs

A cellular sheaf $\mathcal{F}$ on a graph $G = (V, E)$ consists of the following data:

- a vector space $\mathcal{F}(v)$ for each vertex $v \in V$,
- a vector space $\mathcal{F}(e)$ for each edge $e \in E$,
- a linear map $\boldsymbol{\mathcal{F}}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident vertex-edge pair $v \trianglelefteq e$ of G.



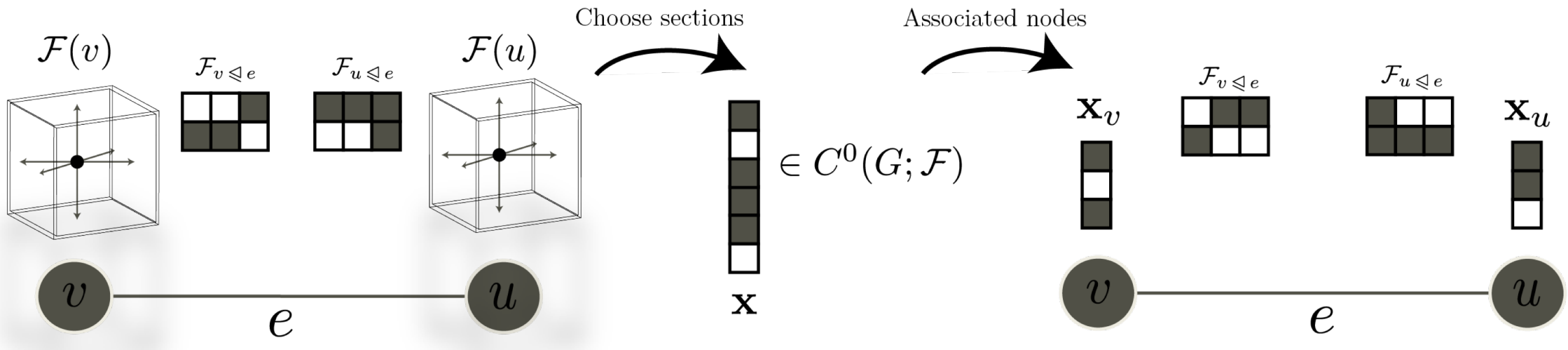
Sheaves on Graphs

Define the $\mathcal{F}$-valued space of signals on vertices and edges, respectively, by:

$$C^0(G;\mathcal{F}) = \bigoplus_{v \in V} \mathcal{F}(v) \qquad \text{Space of 0-cochains}$$

$$C^1(G;\mathcal{F}) = \bigoplus_{e \in E} \mathcal{F}(e) \qquad \text{Space of 1-cochains}$$

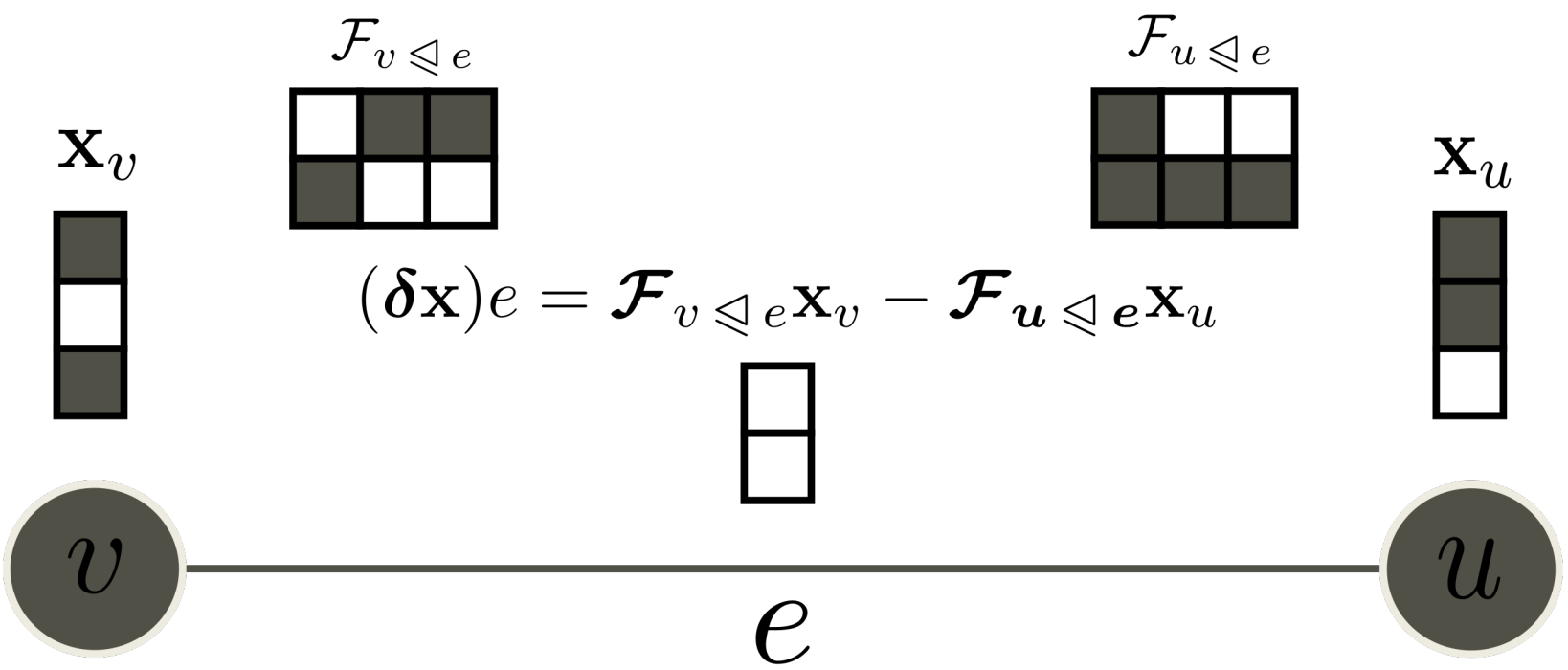
For $\mathcal{F}$ vector-valued, we may view $\mathbf{x} \in C^0(G;\mathcal{F})$ as a concatenation of vectors.



Sheaves on Graphs

Define the coboundary map $\boldsymbol{\delta} : C^0(G;\mathcal{F}) \to C^1(G;\mathcal{F})$ which computes value along edge $e = (u, v)$:

$$(\boldsymbol{\delta}\mathbf{x})e = \boldsymbol{\mathcal{F}}_{v \trianglelefteq e}\mathbf{x}_v - \boldsymbol{\mathcal{F}}_{\boldsymbol{u} \trianglelefteq \boldsymbol{e}}\mathbf{x}_u$$



Sheaves on Graphs

Given coboundary $\boldsymbol{\delta}$, we define the *sheaf Laplacian* $\boldsymbol{L}_{\mathcal{F}}$ as:

$$\boldsymbol{L}_{\mathcal{F}} = \boldsymbol{\delta}^{\top}\boldsymbol{\delta}$$

$$(\boldsymbol{L}_{\mathcal{F}})_{u,v} = \begin{pmatrix} \boldsymbol{\mathcal{F}}^{\top}_{v \trianglelefteq e}\boldsymbol{\mathcal{F}}_{v \trianglelefteq e} & -\boldsymbol{\mathcal{F}}^{\top}_{v \trianglelefteq e}\boldsymbol{\mathcal{F}}_{u \trianglelefteq e} \\ -\boldsymbol{\mathcal{F}}^{\top}_{u \trianglelefteq e}\boldsymbol{\mathcal{F}}_{v \trianglelefteq e} & \boldsymbol{\mathcal{F}}^{\top}_{u \trianglelefteq e}\boldsymbol{\mathcal{F}}_{u \trianglelefteq e} \end{pmatrix}$$

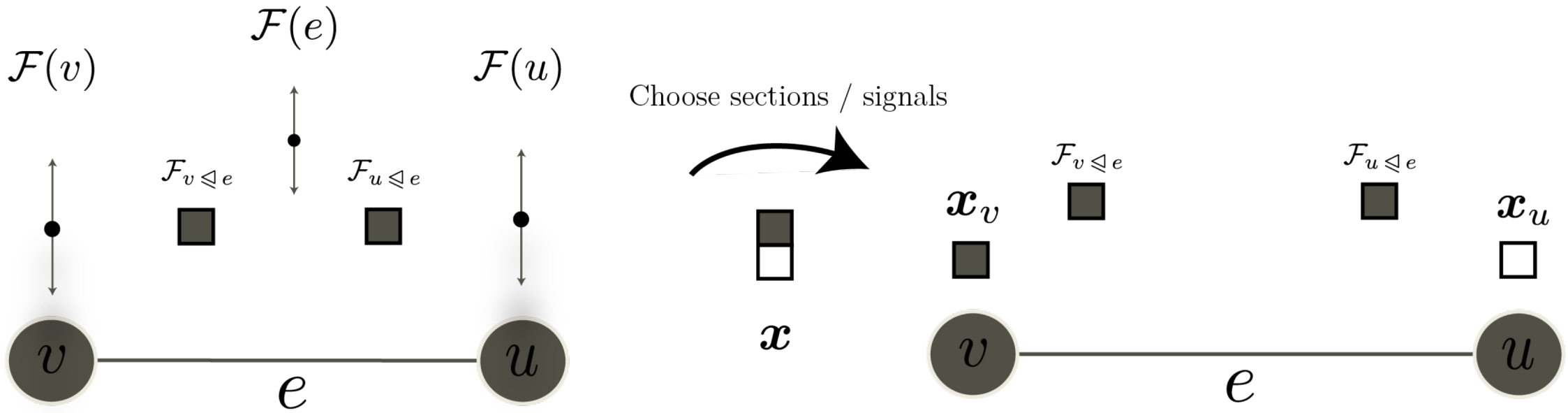
$$(\boldsymbol{L}_{\mathcal{F}}\mathbf{x})_{v} = \sum_{u,v \trianglelefteq e} \boldsymbol{\mathcal{F}}^{\top}_{v \trianglelefteq e}(\boldsymbol{\mathcal{F}}_{v \trianglelefteq e}\mathbf{x}_{v} - \boldsymbol{\mathcal{F}}_{u \trianglelefteq e}\mathbf{x}_{u})$$

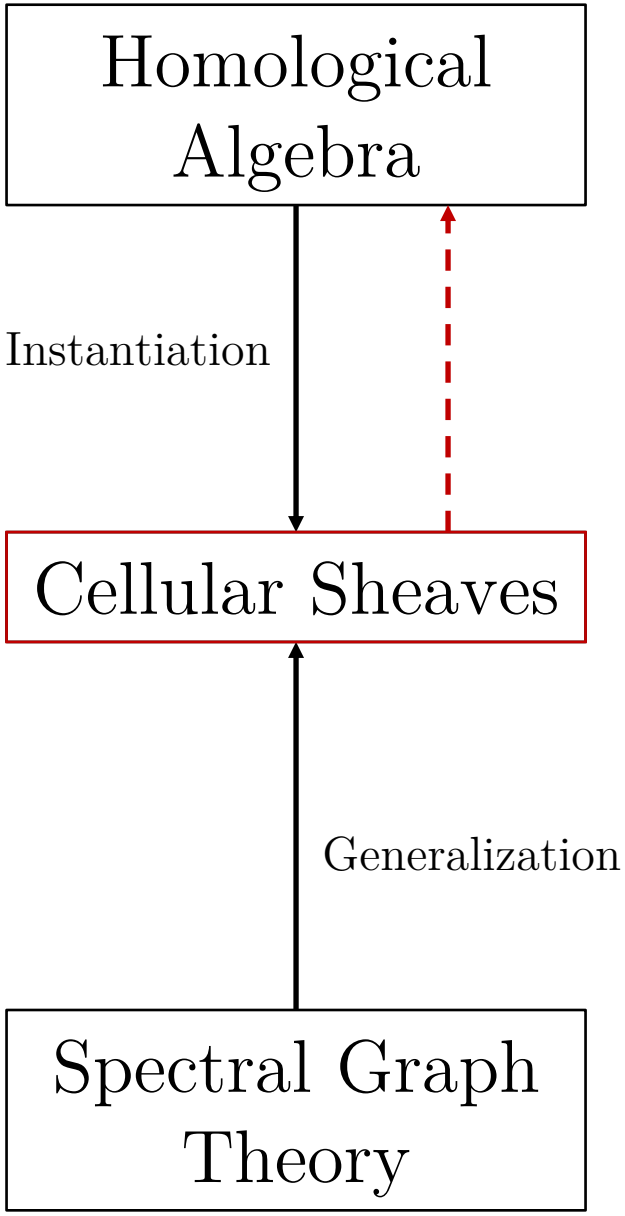
$$\mathbf{x}^{\top}\boldsymbol{L}_{\mathcal{F}}\mathbf{x} = \sum_{u,v \trianglelefteq e} \left\|\boldsymbol{\mathcal{F}}_{v \trianglelefteq e}\mathbf{x}_{v} - \boldsymbol{\mathcal{F}}_{u \trianglelefteq e}\mathbf{x}_{u}\right\|^{2} = E(\mathbf{x}, \mathcal{F})$$

Graph Laplacian (Redux)

We can recover $\boldsymbol{L}_G$ in this sheaf-theoretic language.

Set $\mathcal{F}(v) = \mathbb{R}$ for all vertices $v \in V$ and choose restriction maps such that $\boldsymbol{\mathcal{F}}_{v \trianglelefteq e}^{\top} \boldsymbol{\mathcal{F}}_{u \trianglelefteq e} = a_{u,v}$ for all edges $e = (u, v)$.




Homological Algebra
Instantiation
Cellular Sheaves
Generalization
Spectral Graph Theory

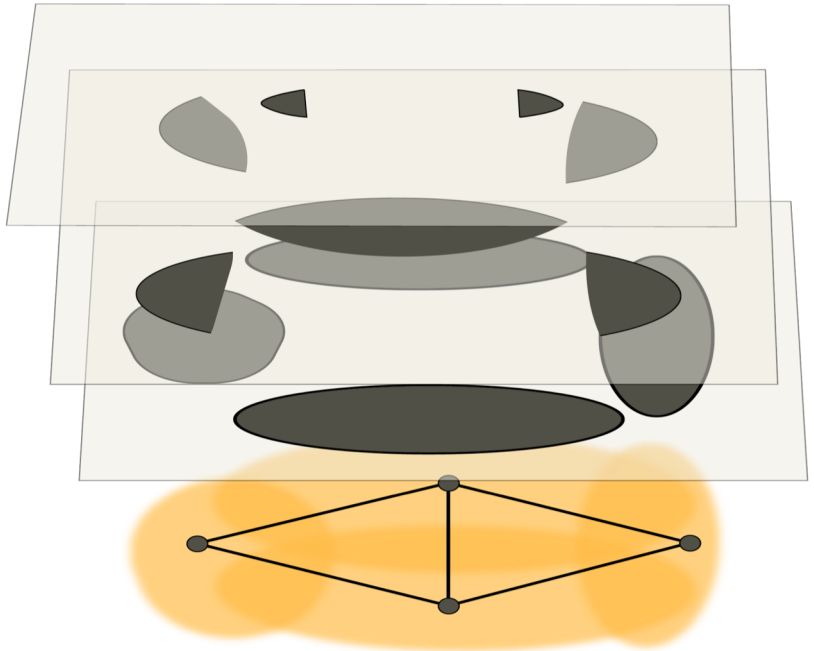
Cellular Sheaves

Sheaves are categorical objects, and were central in the development of algebraic geometry in the mid- 20$^{\text{th}}$ century.

Their usage in applied math is a much more recent development (Curry 2014, Hansen 2019).

Sheaves associate "data" to the open sets of a topological space.

A limit-preserving contravariant functor from the category of open sets of a topological space.



Cellular sheaves associate "data" to the cells of a cell complex.

A limit-preserving functor from the Alexandrov topology on the incidence poset.

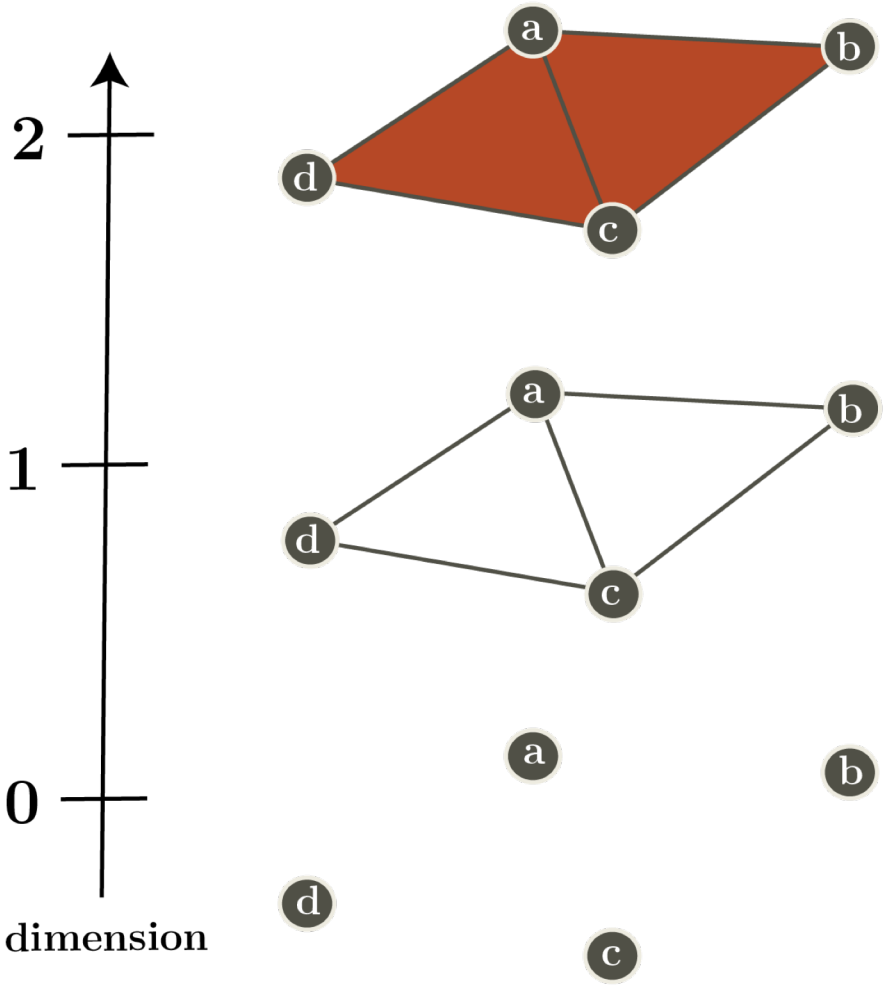
Curry, Justin Michael. *Sheaves, cosheaves and applications*. University of Pennsylvania, (2014).
Hansen, Jakob, and Robert Ghrist. "Toward a spectral theory of cellular sheaves." *Journal of Applied and Computational Topology* 3.4 (2019).

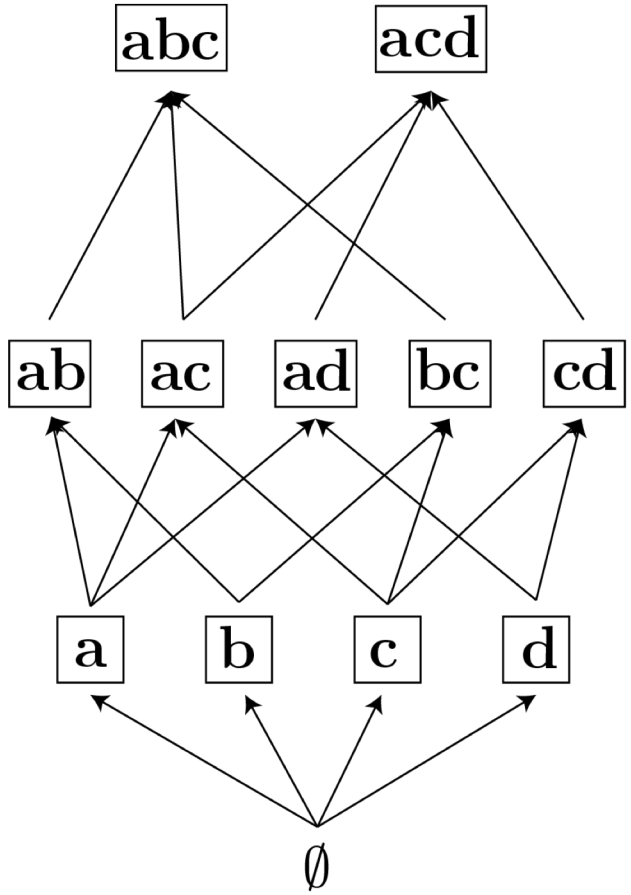
Cellular Sheaves

A *regular cell complex* (Ω, P_Ω) is a space Ω partitioned into a set of *cells* $\{\Omega_\sigma\}_{\sigma \in P_\Omega}$ satisfying the following conditions:

- For each $\omega \in \Omega$, each neighborhood of ω intersects finitely many Ω_σ.
- For all $\sigma, \tau \in P_\Omega$, if $\overline{\Omega}_\tau \cap \Omega_\sigma \neq \emptyset$ then $\Omega_\sigma \subseteq \overline{\Omega}_\tau$. If this is the case, we say σ is a *face* of τ and declare the pair to be *incident*, denoted $\sigma \trianglelefteq \tau$ when σ is a face of τ.
- For each Ω_σ, there is a homeomorphism from the closed d_σ-dimensional ball to the closure $\overline{\Omega}_\sigma$ which additionally maps the interior of the ball onto the open set Ω_σ.

Cellular Sheaves



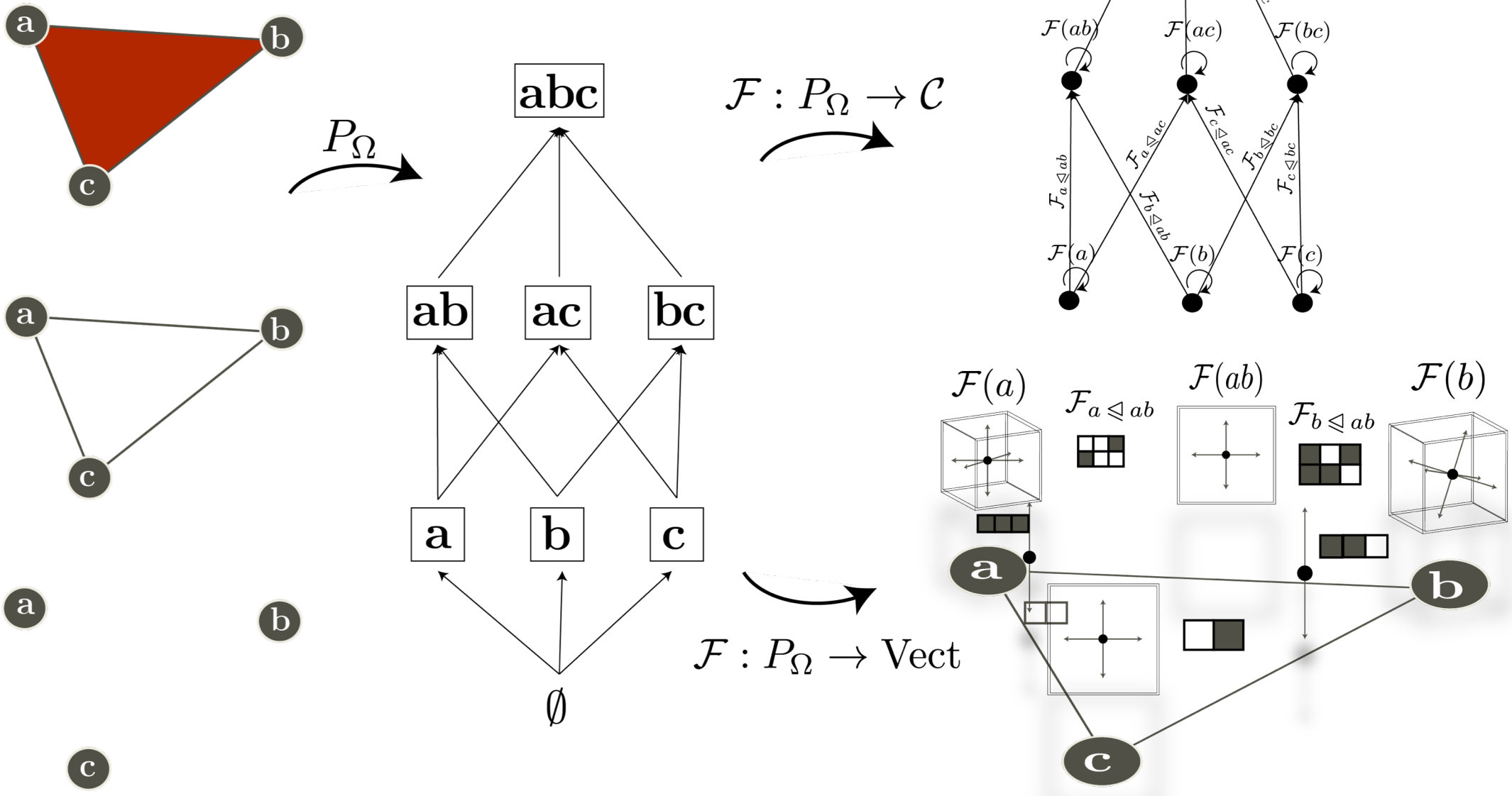


Cellular Sheaves

A cellular sheaf valued in category $\mathcal{C}$ on regular cell complex (Ω, P_Ω) is a covariant functor $\mathcal{F} : P_\Omega \to \mathcal{C}$:

- for each $\sigma \in P_\Omega$, there is an associated object $\mathcal{F}(\sigma)$ in $\mathcal{C}$.
- if $\sigma \trianglelefteq \tau$, there is a morphism $\mathcal{F}_{\sigma \trianglelefteq \tau} : \mathcal{F}(\sigma) \to \mathcal{F}(\tau)$ in $\mathcal{C}$.
- the morphism $\mathcal{F}_{\sigma \trianglelefteq \tau}$ satisfies $\mathcal{F}_{\tau \trianglelefteq \omega} \circ \mathcal{F}_{\sigma \trianglelefteq \tau} = \mathcal{F}_{\sigma \trianglelefteq \omega}$ for $\sigma \trianglelefteq \tau \trianglelefteq \omega$ in P_Ω.

Cellular Sheaves



Sheaf Representation Learning

What does this categorical perspective buy us?

Any results pertaining to sheaves on graphs or cellular sheaves have a direct categorical interpretation.

Highlights the mathematical/categorical assumptions being made in graph representation learning.

Clarifies how one might repurpose these ideas for other types of data or scenarios, and provides a route for proving which results will translate.

Homological operations (sheaf cohomology) are immediate.

Sheaf Cohomology

Denote the space of $\mathcal{F}$-valued k-cochains on Ω as the vector space

$$C^k(\Omega;\mathcal{F}) = \bigoplus_{\dim\sigma=k} \mathcal{F}(\sigma)$$

The coboundary operator extends to $\delta^k : C^k(\Omega;\mathcal{F}) \to C^{k+1}(\Omega;\mathcal{F})$ in the usual way, producing the complex

$$C^0(\Omega,\mathcal{F}) \xrightarrow{\delta^0} C^1(\Omega;\mathcal{F}) \xrightarrow{\delta^1} C^2(\Omega;\mathcal{F}) \xrightarrow{\delta^2} \cdots$$

The cohomology of this complex, denoted $H^k(\Omega;\mathcal{F})$ is the sheaf cohomology of Ω with coefficients in $\mathcal{F}$.

Sheaf Cohomology

$H^0(\Omega;\mathcal{F})$ is isomorphic to the space of global sections of $\mathcal{F}$.

A 0-cochain $x \in C^0(\Omega;\mathcal{F})$ is in $H^0(\Omega;\mathcal{F})$ if $\delta^0 x = 0$. Therefore, $H^0(\Omega;\mathcal{F})$ consists of all $x \in C^0(\Omega;\mathcal{F})$ such that

$$\mathcal{F}_{\sigma_i \trianglelefteq \tau} x_{\sigma_i} = \mathcal{F}_{\sigma_j \trianglelefteq \tau} x_{\sigma_j} \text{ for all } \tau, \dim \tau = 1$$

If $\mathcal{F}$ is Vect-valued over G, this becomes easy to compute:

$$H^0(G;\mathcal{F}) = \ker \boldsymbol{\delta} \subset C^0(G;\mathcal{F})$$

$$\text{and } E(\mathbf{x},\mathcal{F}) = \mathbf{x}^\top \boldsymbol{L}_{\mathcal{F}} \mathbf{x} = 0 \text{ implies } \mathbf{x} \in H^0(G;\mathcal{F})$$

Big Picture

Assume you are given a base space Ω with relational / open set structure defining a cell complex P_Ω.

Assume signals are associated to this base space which interact locally according to the incidence structure of this cell complex.

Choose a Set-like category $\mathcal{C}$ within which you would like to represent these signals/interactions.

If $\mathcal{C} = \text{Vect}$, we can implement global diffusion processes via linear algebra, via projection onto the kernel of the boundary operator.

Extensions to other data categories and higher cohomology approximations are exciting avenues for future work which bridge theory and application.

Applications

Sheaf Neural Networks

Can we learn functions of sheaf-valued signals to, for example, classify 0-cells (vertices) of a cell complex (graph)?

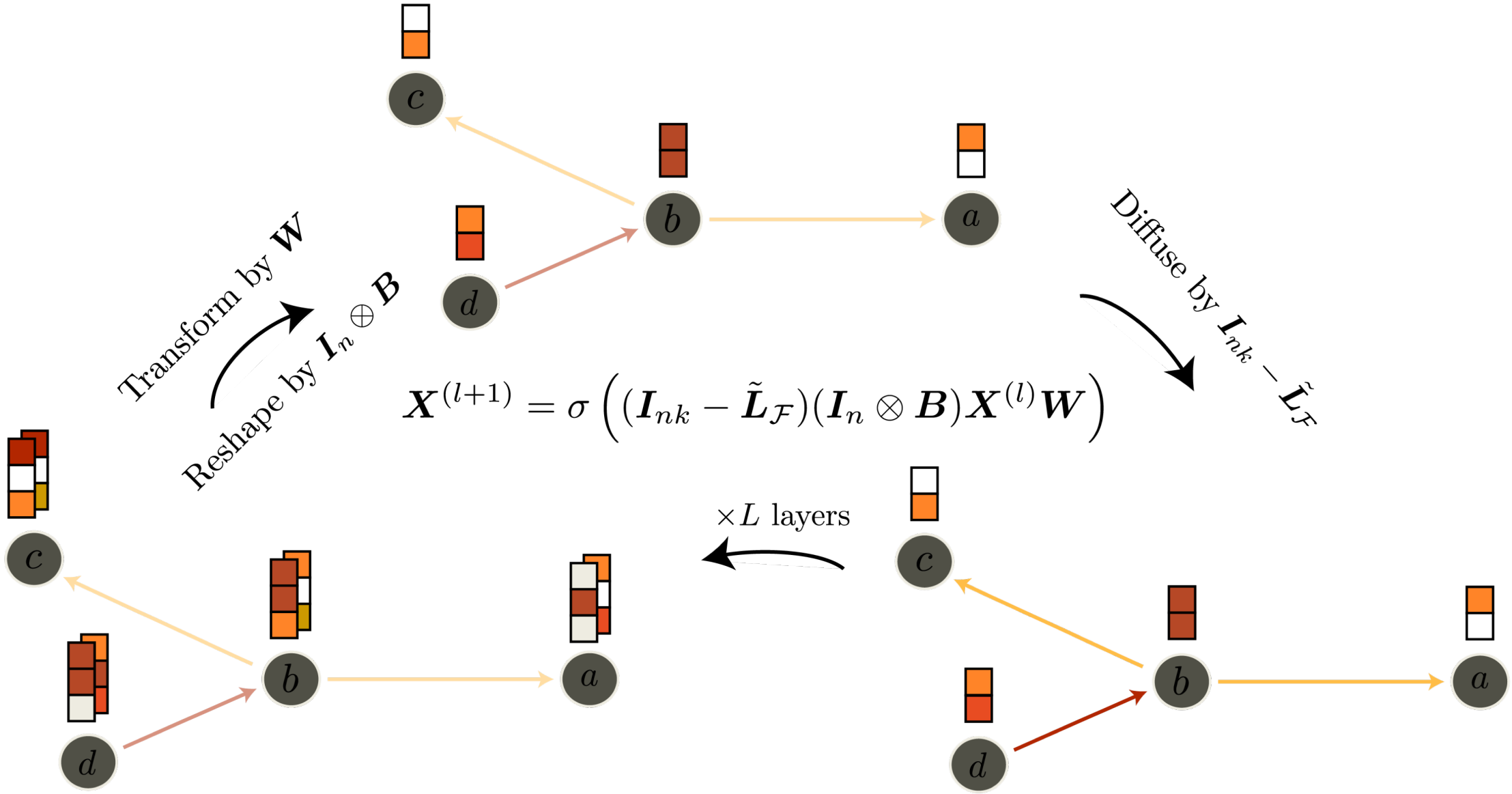
$$f^{\mathrm{SCN}} : C^0(G;\mathcal{F}) \to C^0(G;\mathcal{F}')$$

For simplicity, consider $\mathcal{F} = \mathcal{F}'$. Let $\boldsymbol{X} \in \mathbb{R}^{nk \times d}$ be the collection of d k-dimensional 0-cochains. Then f^{SCN} is given by:

$$f^{\mathrm{SCN}}(\boldsymbol{X}) = \sigma\left((\boldsymbol{I}_{nk} - \tilde{\boldsymbol{L}}_{\mathcal{F}})(\boldsymbol{I}_n \otimes \boldsymbol{B})\boldsymbol{X}\boldsymbol{W}\right)$$

where σ is an element-wise activation, $\boldsymbol{I}_{nk}$ is a $(nk \times nk)$ identity matrix, $\boldsymbol{I}_n$ is an $(n \times n)$ identity matrix, $\tilde{\boldsymbol{L}}_{\mathcal{F}}$ is the (normalized) sheaf Laplacian, and $\boldsymbol{B} \in \mathbb{R}^{k \times k}$ and $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ are learnable.

Sheaf Neural Networks



Sheaf Neural Networks

Repeated application of $\boldsymbol{I}_{nk} - \tilde{\boldsymbol{L}}_{\mathcal{F}}$ "smoothes" each co-chain channel $\mathbf{x}$ *with respect to the sheaf structure* $\mathcal{F}$ so that $\mathbf{x} \to H^0(G;\mathcal{F})$.

Stacking f^{SCN} layers biases towards minimizing $E(\mathbf{x},\mathcal{F}) = \mathbf{x}^\top \tilde{\boldsymbol{L}}_{\mathcal{F}}\mathbf{x}$ along each cochain channel.

In other words, we are learning $f^{\text{SCN}} : C^0(G;\mathcal{F}) \to C^0(G;\mathcal{F})$ by locally augmenting message passing on a sheaf.

Sheaf Neural Networks

GCN

$$f^{\mathrm{GCN}} : \mathcal{X}(G, \mathbb{R}^{d_{l-1}}) \to \mathcal{X}(G, \mathbb{R}^{d_l})$$

$$f^{\mathrm{GCN}}(\boldsymbol{X}) = \sigma\left((\boldsymbol{I} - \tilde{\boldsymbol{L}}_G)\boldsymbol{X}\boldsymbol{W}\right)$$

$$\dot{\boldsymbol{X}}(t+1) = -\tilde{\boldsymbol{L}}_G\boldsymbol{X}(t)$$

$$\begin{aligned}\epsilon(\boldsymbol{x}, \boldsymbol{A}) &= \boldsymbol{x}^\top \tilde{\boldsymbol{L}}_G \boldsymbol{x} \\ &= \frac{1}{2}\sum_{i,j} \tilde{a}_{ij}(x_i - x_j)^2\end{aligned}$$

SCN

$$f^{\mathrm{SCN}} : C^0(G; \mathcal{F}) \to C^0(G; \mathcal{F}')$$

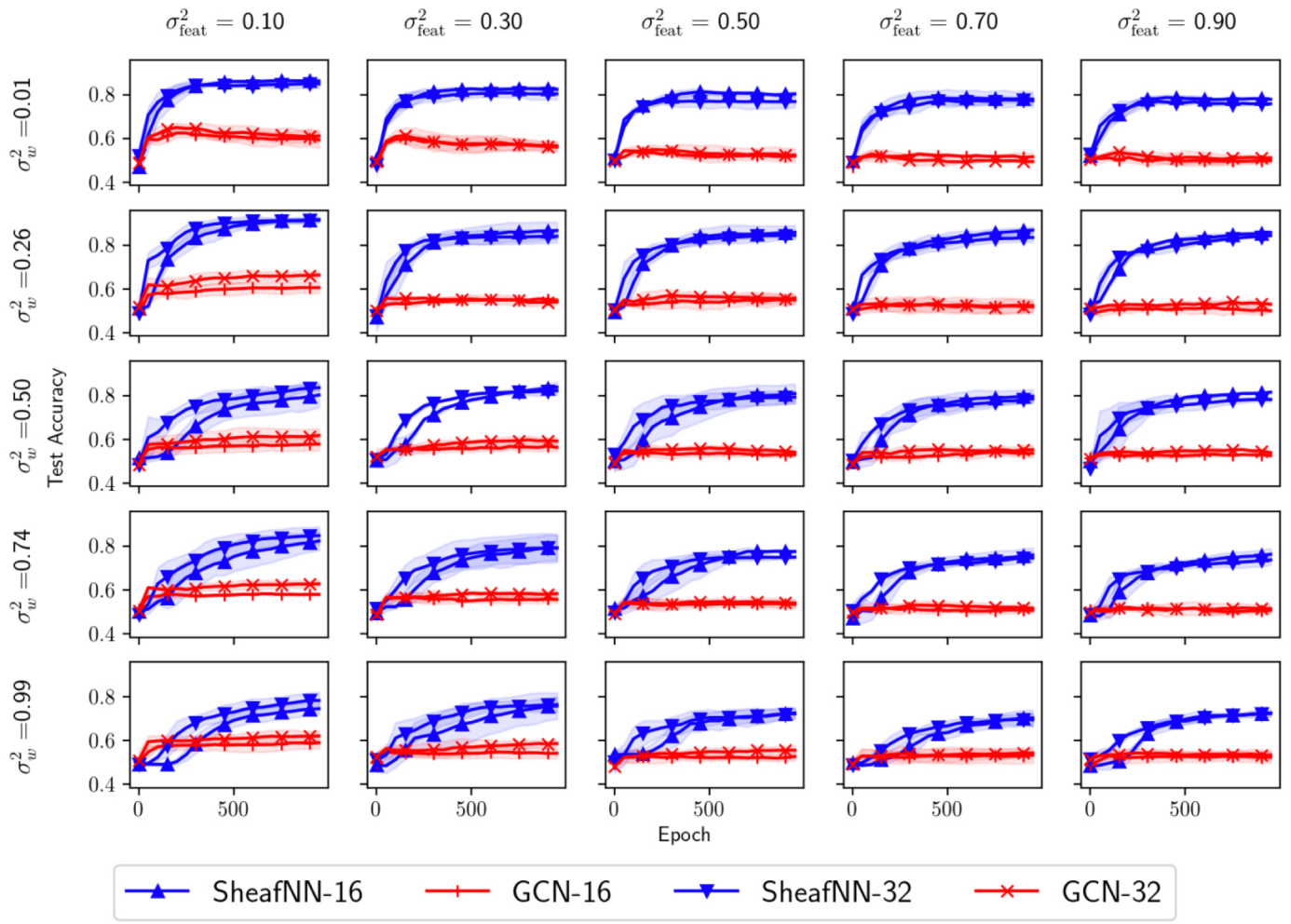
$$f^{\mathrm{SCN}}(\boldsymbol{X}) = \sigma\left((\boldsymbol{I}_{nk} - \tilde{\boldsymbol{L}}_{\mathcal{F}})(\boldsymbol{I}_n \otimes \boldsymbol{B})\boldsymbol{X}\boldsymbol{W}\right)$$

$$\dot{\boldsymbol{X}}(t+1) = -\tilde{\boldsymbol{L}}_{\mathcal{F}}\boldsymbol{X}(t)$$

$$\begin{aligned}E(\mathbf{x}, \mathcal{F}) &= \mathbf{x}^\top \tilde{\boldsymbol{L}}_{\mathcal{F}} \mathbf{x} \\ &= \sum_{u,v \trianglelefteq e} \left\| \boldsymbol{\mathcal{F}}_{v \trianglelefteq e} \boldsymbol{D}_v^{-1/2} \mathbf{x}_v - \boldsymbol{\mathcal{F}}_{u \trianglelefteq e} \boldsymbol{D}_u^{-1/2} \mathbf{x}_u \right\|^2\end{aligned}$$

Sheaf Neural Networks

Synthetic dataset of signed graphs illustrates the limitations of GCNs



Hansen, Jakob, and Thomas Gebhart. 'Sheaf neural networks.' *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond* (2020).

Sheaf Neural Networks

Recent work (Bodnar et al., 2022) provides further proof of the limitations of GCNs and how SCNs can overcome these problems.

A hypothesis class of sheaves with k-dimensional stalks $\mathcal{H}^k$ has *linear separation power* over a family of graph $\mathcal{G}$ if for any $G = (V, E) \in \mathcal{G}$ there is a sheaf $(\mathcal{F}, G) \in \mathcal{H}^k$ that can linearly separate the classes of G in the limit of $\dot{\boldsymbol{X}}(t+1) = -\tilde{\boldsymbol{L}}_{\mathcal{F}}\boldsymbol{X}(t)$ for non-degenerate initial conditions.

Summary: By varying the subspace from which we choose the restriction maps, the separation power of the corresponding sheaf diffusion can be changed.

Bodnar, Cristian, et al. "Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns." *Advances in Neural Information Processing Systems* 35 (2022).

Sheaf Neural Networks

Let G be a connected graph with $C \geq 3$ classes. Then $\mathcal{H}^1$ cannot linearly separate the classes of G for any initial conditions $\boldsymbol{X}(0) \in \mathbb{R}^{n \times d}$.

This implies the diffusion underlying standard GCNs is ill-suited to solve this classification task at large depths.

Bodnar, Cristian, et al. "Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns." *Advances in Neural Information Processing Systems* 35 (2022).

Sheaf Neural Networks

Let $\mathcal{H}^k_{\text{orth}} = \{(\mathcal{F}, G) : \boldsymbol{\mathcal{F}}_{v \trianglelefteq e} \in O(k)\}$ be the hypothesis class constructed from $O(k)$ bundles (orthogonal restriction maps).

Let $\mathcal{G}$ be the class of connected graphs with $C \leq 2k$ classes. Then for all $k \in \{2, 4\}, \mathcal{H}^k_{\text{orth}}$ has linear separation power over $\mathcal{G}$.

Depending on how these restriction maps are chosen, one has more "space" within which to separate classes.

Bodnar, Cristian, et al. "Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns." *Advances in Neural Information Processing Systems* 35 (2022).

Summary

- Many graph representation learning problems are implicitly driven by the diffusion behavior of signals on a graph, encoded by the Laplacian.
- This is a powerful and intuitive idea, but it has limitations.
  - Signals are assumed 1-dimensional.
  - Incident interactions are simple scaling.
  - Unable to describe heterogeneous interactions.
  - Bias towards homophilous representations.
- Cellular sheaves provide a natural framework for the generalization of many graph representation learning processes.
- Categorical and cohomological underpinnings of sheaves provide a wealth of modeling choices while retaining consistency requirements.

References

Bronstein, Michael M., et al. “Geometric deep learning: Grids groups, graphs, geodesics, and gauges”. *arXiv preprint arXiv:2104.13478* (2021).

Levy, Omer and Goldberg, Yoav. “Neural word embedding as implicit matrix factorization”. *Advances in Neural Information Processing Systems.* (2014).

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* 2014.

Abu-El-Haija, Sami, et al. "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing." *international conference on machine learning.* PMLR, 2019.

Chanpuriya, Sudhanshu, and Cameron Musco. "Infinitewalk: Deep network embeddings as Laplacian embeddings with a nonlinearity." *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2020.

Curry, Justin Michael. *Sheaves, cosheaves and applications.* University of Pennsylvania, (2014).

Hansen, Jakob, and Robert Ghrist. "Toward a spectral theory of cellular sheaves." *Journal of Applied and Computational Topology* 3.4 (2019).

Hansen, Jakob, and Thomas Gebhart. "Sheaf neural networks." *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond* (2020).

Bodnar, Cristian, et al. "Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns." *Advances in Neural Information Processing Systems* 35 (2022).