Sheaf-based Positional Encodings for Graph Neural Networks

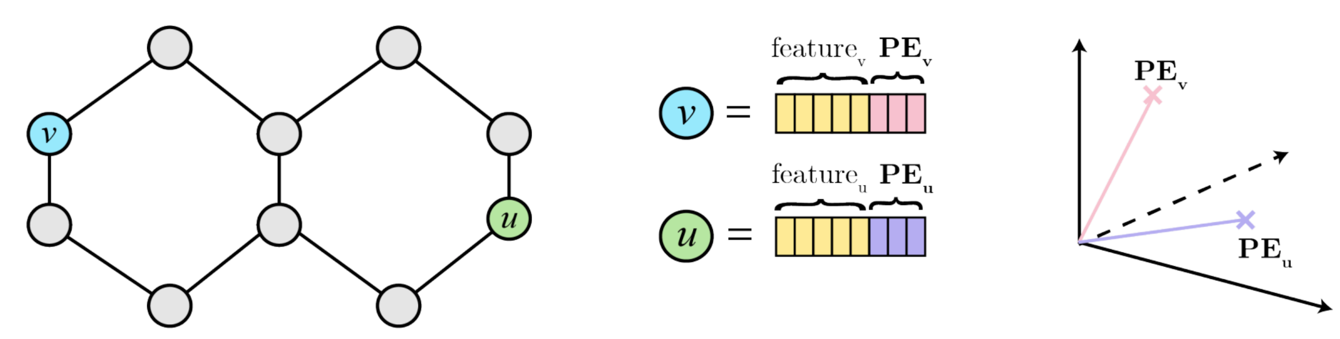
Yu He[1], Cristian Bodnar[2], Pietro Liò[3]
[1] Stanford University. Work done at University of Cambridge.
[2] Microsoft Research AI4Science [3] University of Cambridge

NEURAL INFORMATION PROCESSING SYSTEMS

NeurReps Workshop

TLDR: We propose to construct **positional encodings** for GNNs using **the sheaf Laplacian**, in the aim to encode both the **structural** and **semantic** information from the graph and its node data.

Positional Encodings for GNNs

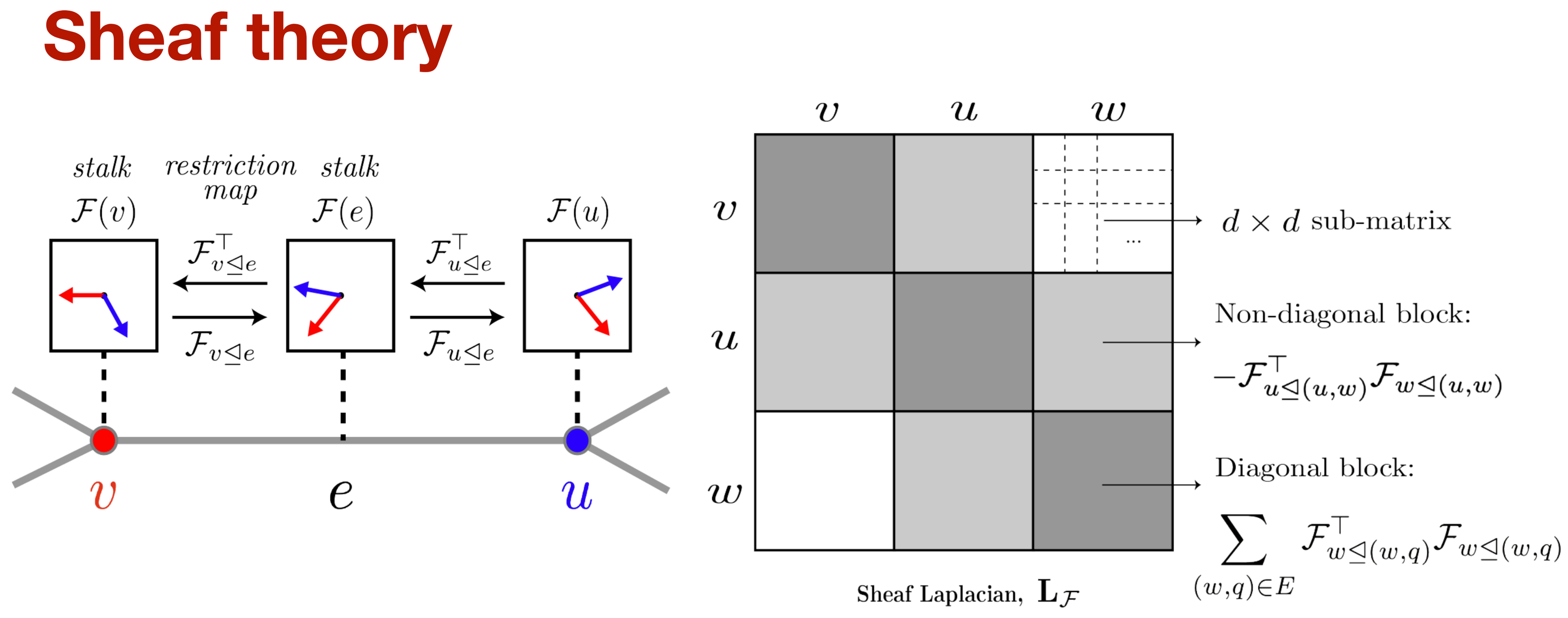


Positional encodings inform the nodes of their position in the graph, which helps to break the locality constraint from message-passing.

The graph Laplacian is a popular candidate for designing positional encodings for graphs, but it encodes purely the graph structure, without taking the node data into account.

However, **heterophilic graphs** have dissimilar nodes connected, reflected by their node features.

Sheaf theory



Cellular sheaf

A cellular sheaf $(G, \mathcal{F})$ on an undirected graph $G = (V, E)$ consists of:

- *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 $\mathcal{F}_{v\trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident node-edge pair $v \trianglelefteq e$.*

We call the vector spaces of the nodes and edges as **stalks**, and the linear maps as **restriction maps**.

0-cochain

The space of 0-cochains $C^0(G;\mathcal{F}) := \bigoplus_{v\in V}\mathcal{F}(v)$ is the space formed by all the stalks associated with the nodes of the graph, where $\bigoplus$ denotes the direct sum of vector spaces.

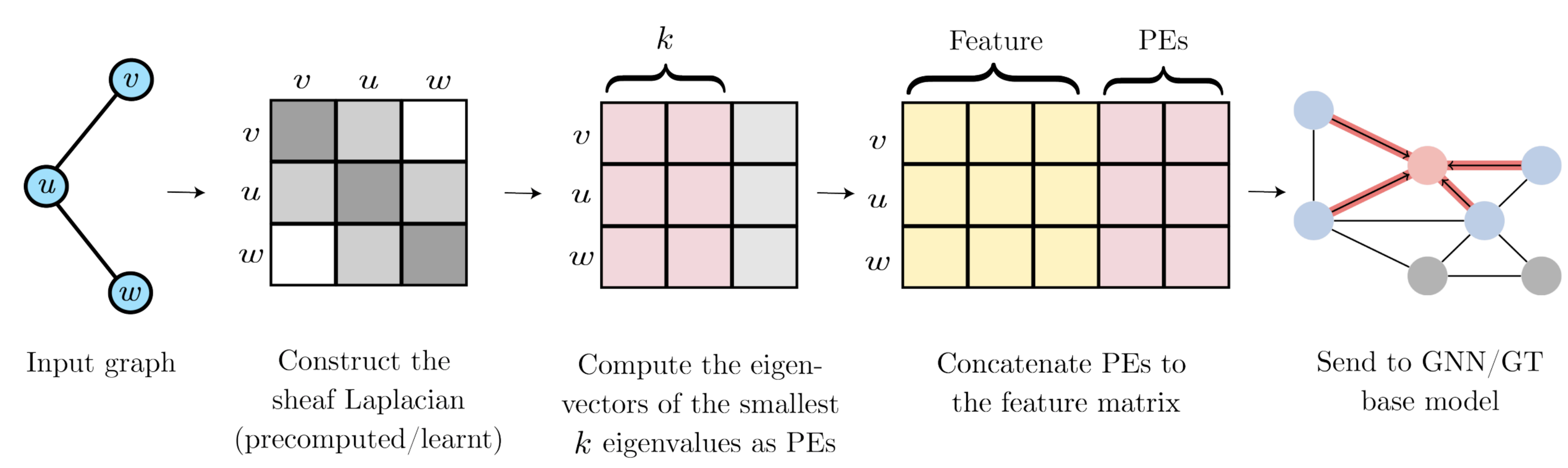
The sheaf Laplacian operator for a given cellular sheaf measures the aggregated "disagreement of opinions" at each node.

Sheaf Laplacian

Given a cellular sheaf $(G;\mathcal{F})$, the sheaf Laplacian is a linear map $\mathbf{L}_\mathcal{F} : C^0(G,\mathcal{F}) \to C^0(G,\mathcal{F})$, which can be defined node-wise as $\mathbf{L}_\mathcal{F}(\mathbf{x})_v = \sum_{v,u\trianglelefteq e}(\mathcal{F}_{v\trianglelefteq e}\mathbf{x}_v - \mathcal{F}_{u\trianglelefteq e}\mathbf{x}_u)$. Here, $\mathbf{x} \in C^0(G;\mathcal{F})$ is a 0-cochain, and $\mathbf{x}_v$ is the vector in $\mathcal{F}(v)$ of node v.

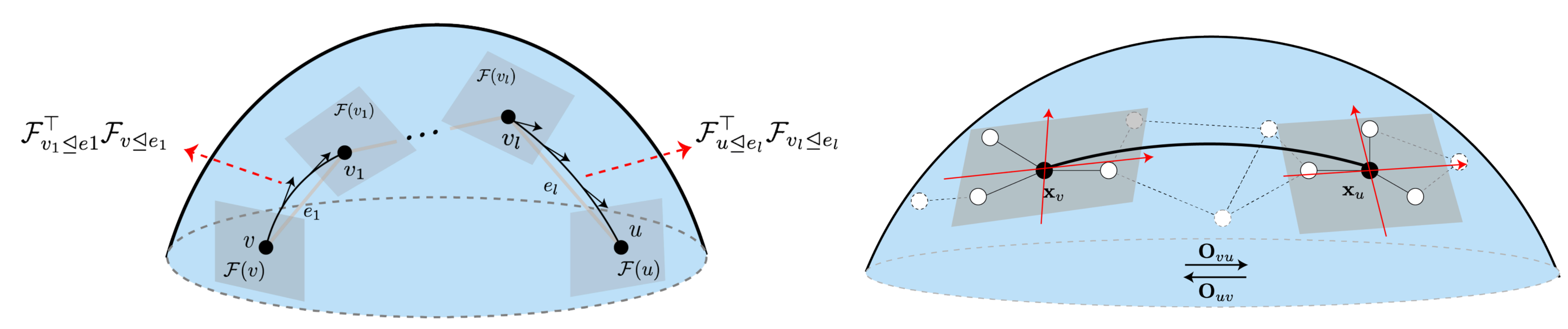
The sheaf Laplacian generalises the graph Laplacian: The graph Laplacian is a trivial sheaf, by setting all the stalks to scalars (d = 1, where d is the stalk dimension) and the restriction maps to identity functions.

Sheaf-based positional encodings



We propose to construct the sheaf-based positional encodings via precomputing or learning the sheaf Laplacian.

(i) Precomputed sheaf Laplacian (ConnLap)



The connection Laplacian is a special form of the sheaf Laplacian with an orthogonal matrix. It can be thought of as a discretised representation of the vector bundle, which draws an analogy to the concept of parallel transport on a manifold. We can compute the connection Laplacian by optimally aligning the orthonormal bases [1].

Graph-level tasks:

GatedGCN	ZINC TestMAE (↓)	ZINC+LSPE TestMAE (↓)	MOLTOX21 TestAUC (↑)
No PE	0.251±0.009	N.A.	77.2±0.6
GraphLap	**0.202±0.006**	0.196±0.008	77.4±0.7
ConnLap	0.249±0.005	**0.193±0.014**	**77.9±0.2**

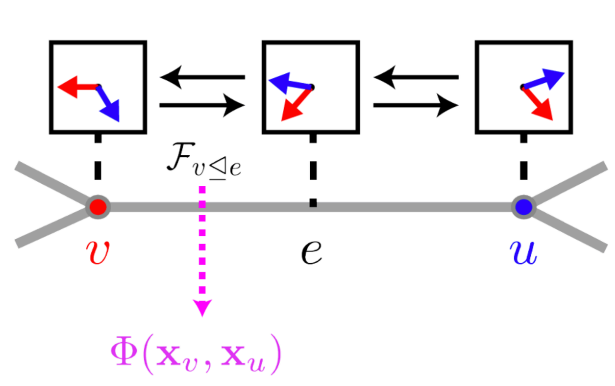
MOLTOX21

	GatedGCN	PNA	SAN
	TestAUC (↑)		
No PE	77.2±0.6	**75.5±0.8**	74.4±0.7
GraphLap	77.4±0.7	75.2±1.3	73.6±0.3
ConnLap	**77.9±0.2**	75.3±0.4	**74.5±0.4**

Mean±std MAE (↓) for ZINC and mean±std AUC (↑) for MOLTOX21.

We additionally allows ConnLap to evolve following LSPE [3].

(ii) Learnt sheaf Laplacian (SheafLap)



We approximate the restriction map using a learnable parametric function $\Phi : \mathbb{R}^{d\times 2} \to \mathbb{R}^{d\times d}$ [2]. That is, $\mathcal{F}_{v\trianglelefteq e:=(v,u)} = \Phi(\mathbf{x}_v, \mathbf{x}_u)$, where $\mathbf{x}_v$ and $\mathbf{x}_u$ are node features for nodes v and u.

Node-level tasks:

	Texas	Wisconsin	Film	Squirrel	Chameleon	Cornell	Citeseer	Pubmed	Cora
Hom level	0.11	0.21	0.22	0.22	0.23	0.30	0.74	0.80	0.81
#Nodes	183	251	7,600	5,201	2,277	183	3,327	18,717	2,708
#Edges	295	466	26,752	198,493	31,421	280	4,676	44,327	5,278
#Classes	5	5	5	5	5	5	7	3	6
No PE	57.30±5.51	49.80±6.80	**25.20±0.69**	46.62±3.62	63.97±3.10	45.95±6.84	72.34±1.41	**86.43±0.35**	84.71±1.23
GraphLap	58.22±7.03	**55.49±12.46**	25.13±0.99	47.56±3.03	64.28±3.00	**51.35±7.15**	73.83±2.07	86.43±0.36	85.05±1.47
ConnLap	**58.38±7.76**	**57.65±6.63**	**26.53±0.86**	**47.92±3.53**	**65.57±2.52**	**52.97±7.37**	**73.88±1.84**	**86.49±0.42**	**85.13±1.34**
SheafLap	**61.08±6.19**	54.51±7.22	23.80±1.10	**51.11±2.95**	**65.2±3.10**	48.38±5.05	**74.35±1.64**	85.84±0.65	**85.88±1.26**

Mean±std accuracy with decreasingly heterophilic graphs. Best and Second Best are coloured.

Conclusion & Future work

- The sheaf Laplacian outperforms the graph Laplacian in designing positional encodings by additionally taking the node data into account, especially for heterophilic graphs.
- What next? Learnt sheaf Laplacian on graph-level tasks; sign and basis invariance; theoretical proofs.

Bibliography

[1] Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael Bronstein, Petar Veličković, and Pietro Liò. *Sheaf Neural Networks with Connection Laplacians, 2022.*

[2] Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Liò, and Michael M. Bronstein. *Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs,* 2022.

[3] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. *Graph neural networks with learnable structural and positional representations,* 2021.