Harvard John A. Paulson School of Engineering and Applied Sciences

Higher-Order Learning with Graph Neural Networks via Hypergraph Encodings

Raphael Pellegrin* Lukas Fesser* Melanie Weber

NEURAL INFORMATION PROCESSING SYSTEMS

Punchline

- Many real systems have multi-way relations that are better modeled as hypergraphs rather than graphs.
- Hypergraph NNs (HNNs) under-perform relative to GNNs on expanded hypergraphs.
- We bridge the gap by computing *hypergraph* versions of popular Positional and Structural Encodings.
- Accuracies increase by as much as 10% on social networks for graph classification tasks.
- Hypergraph Encodings are sometimes strong enough that a simple MLP performs close to SOTA GNNs

Graph vs. hypergraph at a glance

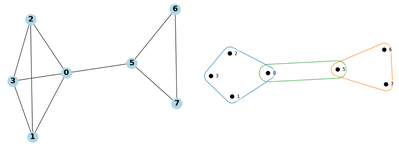


Figure: A graph (left) represents pairwise relations, while a hypergraph (right) can represent multi-way relations. One can re-parametrize a graph into a hypergraph via lifting.

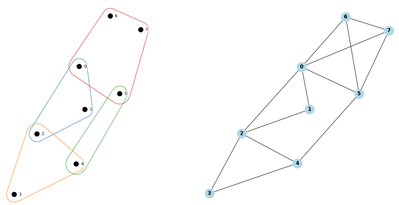


Figure: One can re-parametrize a hypergraph into a graph via clique expansions

Background and motivation

Graphs connect pairs of entities. **Hypergraphs** connect sets of entities. Many tasks - co-authorship teams, cellular pathways, recommendation sets - are naturally multi-way.

Recent progress in graph learning shows that **Positional and Structural Encodings** (eigenvectors of graph Laplacians, random-walk statistics, curvature, degree profiles) are simple, effective signals that improve representation power and accuracy of GNNs.

However, directly designing new HNN architectures has lagged behind: empirically, **clique expansion plus strong GNNs often outperform dedicated HNNs**.

Our idea: keep the powerful GNNs, **compute the Encodings on the *hypergraph*** (where higher-order structure lives), and then feed them to the graph model.

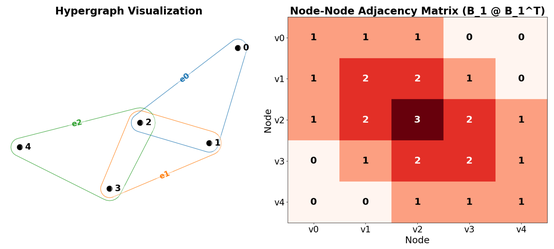
Hypergraph Encodings that plug into GNNs

Design principle: compute Encodings on the hypergraph where higher-order structure is explicit, then attach them as node features for a standard GNN trained on a graph representation.

Hypergraph Laplacian Positional Encodings: top-k eigenvectors of a symmetric hypergraph Laplacian. These global Positional Encodings capture the global position of nodes.

Hypergraph Random-Walk Positional Encodings: summary vectors from hypergraph random walks. These capture the neighborhood topology.

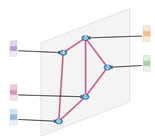
Hypergraph curvature profiles: local summaries based on discrete curvature (for example, Forman or Ollivier curvature) aggregated over incident hyperedges, providing shape and bottleneck information.



Theory supports empirical results

- Message Passing GNNs are at most as powerful as the 1-WL test, with GIN designed to be as powerful as the 1-WL test.
- Hypergraph Encodings can distinguish pairs of non-isomorphic graphs that the 1-WL test and graph-based Encodings cannot, increasing the expressive power of GNNs.

Comparing GNNs and HNNs



GNNs often use Message Passing:

$$x_v^{l+1} = \phi_l\Big(\bigoplus_{p\in\mathcal{N}_v} \psi_l\left(x_p^l\right)\Big)$$

HNNs often employ a two-phase scheme:

$$h_e^{l+1} = \phi_1\left(\left\{x_j^l\right\}_{j\in e}\right), \qquad \tilde{x}_i^{l+1} = \phi_2\left(x_i^l, \left\{h_e^{l+1}\right\}_{e\in E_i}\right).$$

Model (Encodings)	citeseer-CC (↑)	cora-CC (↑)	pubmed-CC (↑)	DBLP (↑)
GCN (No Encoding)	$\mathbf{69.28 \pm 0.28}$	$\mathbf{75.43 \pm 0.26}$	$\mathbf{84.66 \pm 0.49}$	$\mathbf{75.66 \pm 0.81}$
UniGCN (No Encoding)	63.36 ± 1.76	71.10 ± 1.37	75.32 ± 1.09	71.05 ± 1.40

Figure: GCN and UniGCN performance on hypergraph datasets. We report mean accuracy and standard deviation over 50 runs.

Ablations and practical guidance

We compare a wide selection of GNNs (Message-Passing, transformer-based, hybrid) and HNN architectures, with and without multiple graph and hypergraph Encodings.

Random-Walk and Laplacian hypergraph Encodings often perform best.

Ablations on BREC

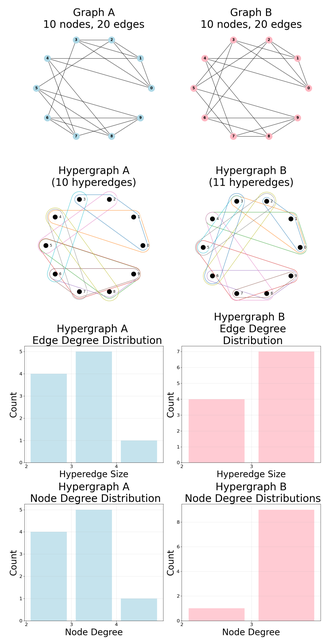


Figure: A pair of non-isomorphic graphs that cannot be distinguished using the 1-WL test. The graphs are easily seen to be non-isomorphic by studying their liftings.

Encodings are beneficial for GNNs

Compute Encodings at the hypergraph level (either on native hypergraph data, or on lifted data); convert hypergraphs to graphs by clique expansion; train strong GNNs.

Adding our hypergraph Encodings as node features consistently improves accuracy, often beyond graph Encodings.

Largest gains appear on social and co-authorship datasets with strong higher-order effects.

Hypergraph Encodings do not appear to be beneficial for HNNs, and we don't understand why!

Model (Encodings)	Imdb (↑)	Reddit (↑)	Mutag (↑)
GCN (No Encoding)	48.10 ± 1.02	67.87 ± 1.38	65.96 ± 1.76
GCN (LCP-FRC)	63.42 ± 1.47	79.53 ± 1.62	67.04 ± 1.49
GCN (LCP-ORC)	67.93 ± 1.55	80.75 ± 1.54	83.09 ± 1.71
GCN (RWPE)	50.41 ± 1.26	78.93 ± 1.60	71.75 ± 2.08
GCN (LAPE)	48.82 ± 1.31	77.26 ± 1.58	73.30 ± 1.95
GCN (HCP-FRC)	64.64 ± 0.88	82.09 ± 0.58	80.85 ± 1.77
GCN (EE H-RWPE)	$\mathbf{73.96 \pm 0.65}$	$\mathbf{82.79 \pm 0.62}$	$\mathbf{85.32 \pm 1.63}$
GCN (EN H-RWPE)	$\mathbf{73.84 \pm 0.48}$	$\mathbf{83.30 \pm 0.79}$	82.34 ± 2.68
GCN (Hodge H-LAPE)	71.38 ± 0.75	79.46 ± 0.82	83.66 ± 1.90
GCN (Norm. H-LAPE)	71.05 ± 0.82	80.08 ± 0.67	81.68 ± 1.79

Figure: GCN performance with no Encoding (top), graph level Encodings (middle) and hypergraph level Encodings (bottom). We report mean and standard deviation across 50 runs.

Takeaways

- Keep the strong GNNs baseline; compute Encodings at the hypergraph level.
- Hypergraph Encodings are plug-and-play features that increase expressive power and accuracy with minimal engineering.
- This path is a simple, effective alternative to designing new HNN architectures.

Compute your Positional and Structural Encodings at the hypergraph level!

Code and paper

github.com/Weber-GeoML/Hypergraph_Encodings
https://arxiv.org/abs/2502.09570