L2G-Net: Local to Global Spectral GNNs via Cauchy Factorizations

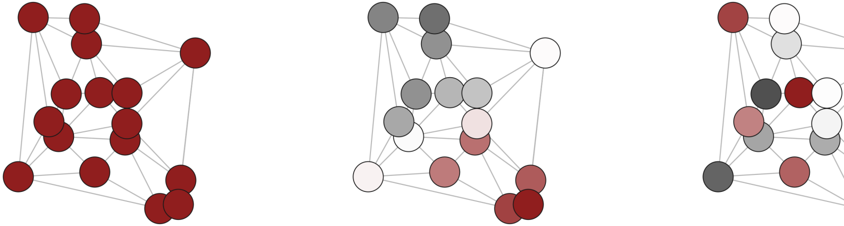
S. Fernández Menduiña, E. Pavez, A. Ortega

University of Southern California

Overview

- GNNs and GFT
- Cauchy matrices and Cauchy factorization
- L2G-Net
- Empirical results
- Conclusions and future work

Spectral GNNs and the GFT



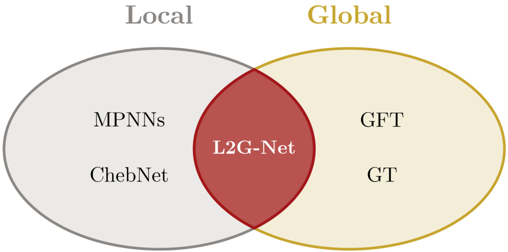
$\lambda = 0.00$ $\lambda = 0.33$ $\lambda = 1.57$

Graph Fourier transform (GFT) [Ortega et al., 2018]: meaningful frequencies.

Cubic complexity $O(n^3) \implies$ impractical for large graphs.

Fully global $\implies$ no local inductive bias for short-range tasks.

Existing approaches



- **MPNNs** [Kipf and Welling, 2017] and **polynomial filters** [Defferrard et al., 2016]: local, but oversquashing, instabilities.
- **Graph transformers** [Dwivedi and Bresson, 2021]: global, but many parameters, lose topological inductive bias.
- Our goal: **GFT** without cubic complexity and **local-to-global** processing.

Our contribution

We introduce an **exact factorization** of the GFT via Cauchy matrices.

1) **Quadratic complexity** $O(kn^2)$ scaled by the subgraph cut size.

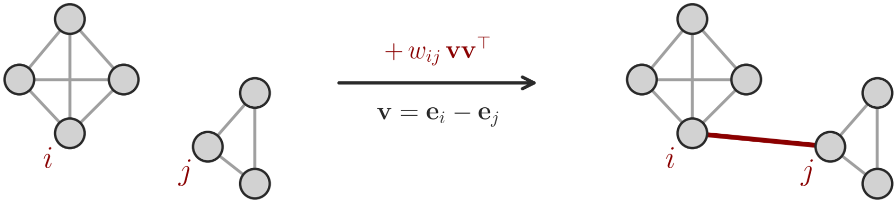
2) **Graph partitioning** algorithm to minimize factorization cost.

3) **L2G-Net**: a new class of spectral GNNs with local-to-global bias.

Background: rank-one updates and Cauchy matrices

Adding an edge (i, j) to a graph yields a **rank-one update** of the Laplacian $\mathbf{L}$:

$$\tilde{\mathbf{L}} = \mathbf{L} + w_{ij}\,\mathbf{v}\mathbf{v}^\top, \quad \mathbf{v} = \mathbf{e}_i - \mathbf{e}_j. \tag{1}$$

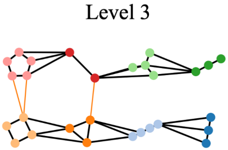


Cauchy update of the eigenbasis [Fernández et al., 2025]

Rank-one update $\implies$ rotate by an **orthogonal Cauchy-like matrix**:

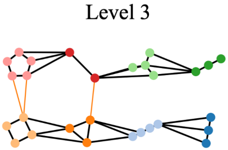
$$\tilde{\mathbf{U}}^\top = \mathbf{C}(\tilde{\boldsymbol{\lambda}}, \boldsymbol{\lambda})\,\mathbf{U}^\top. \tag{2}$$

GFT via divide and conquer



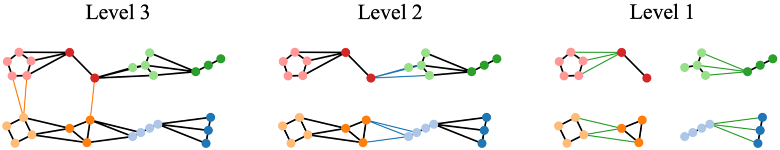
1. **Partition** the graph into subgraphs (e.g., balanced cuts).

GFT via divide and conquer



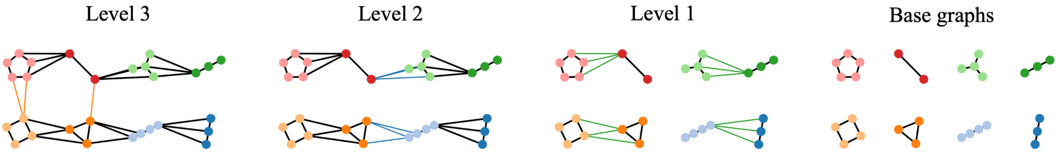
1. **Partition** the graph into subgraphs (e.g., balanced cuts).

GFT via divide and conquer



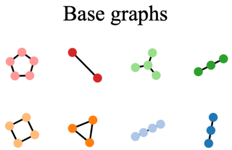
1. **Partition** the graph into subgraphs (e.g., balanced cuts).

GFT via divide and conquer



1. **Partition** the graph into subgraphs (e.g., balanced cuts).
2. **Compute** GFT of base graphs.

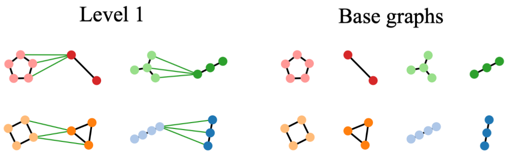
GFT via divide and conquer



1. **Partition** the graph into subgraphs (e.g., balanced cuts).
2. **Compute** GFT of base graphs.
3. Progressively **merge** base GFTs using edge information.

Key: adding one edge adds one Cauchy factor [Fernández et al., 2025].

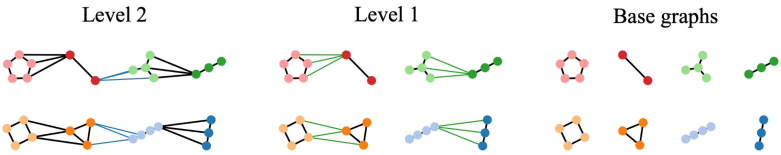
GFT via divide and conquer



1. **Partition** the graph into subgraphs (e.g., balanced cuts).
2. **Compute** GFT of base graphs.
3. Progressively **merge** base GFTs using edge information.

Key: adding one edge adds one Cauchy factor [Fernández et al., 2025].

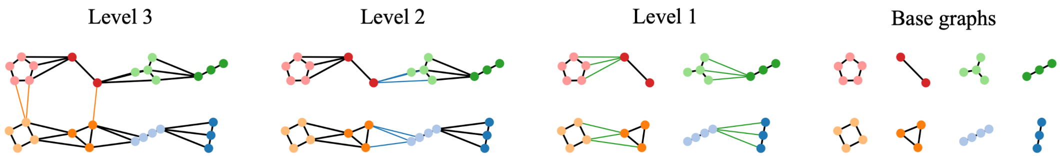
GFT via divide and conquer



1. **Partition** the graph into subgraphs (e.g., balanced cuts).
2. **Compute** GFT of base graphs.
3. Progressively **merge** base GFTs using edge information.

Key: adding one edge adds one Cauchy factor [Fernández et al., 2025].

GFT via divide and conquer



1. **Partition** the graph into subgraphs (e.g., balanced cuts).
2. **Compute** GFT of base graphs.
3. Progressively **merge** base GFTs using edge information.

Key: adding one edge adds one Cauchy factor [Fernández et al., 2025].

Cauchy factorization of the GFT

GFT factorization

For any graph $\mathcal{G}$ with GFT basis $\mathbf{U}^\top$:

$$\mathbf{U}^\top = \mathbf{D}(\boldsymbol{\lambda}, \tilde{\boldsymbol{\lambda}}_{K-1}) \cdots \mathbf{D}(\tilde{\boldsymbol{\lambda}}_1, \tilde{\boldsymbol{\lambda}}_0)\, \mathbf{U}_0^\top . \tag{3}$$

Start from **independent subgraph GFTs**, merge one cut edge at a time.

Each Cauchy factor is **localized** to the two subgraphs it connects.

Computable in **quadratic time**: $O(kn^2)$.

Similar eigenvalues

Orthogonal Cauchy-like matrices

Let $\mathbf{L} = \mathbf{U}\mathrm{diag}(\boldsymbol{\lambda})\mathbf{U}^\top$ and $\tilde{\mathbf{L}} = \mathbf{L} + \rho\,\mathbf{v}\mathbf{v}^\top = \tilde{\mathbf{U}}\mathrm{diag}(\tilde{\boldsymbol{\lambda}})\tilde{\mathbf{U}}^\top$. Then,

$$\tilde{\mathbf{U}}^\top = -\,\mathrm{diag}(\mathbf{a})\mathbf{C}(\tilde{\boldsymbol{\lambda}}, \boldsymbol{\lambda})\,\mathrm{diag}(\mathbf{z})\mathbf{U}^\top, \tag{4}$$

where $\mathbf{z} \doteq \mathbf{U}^\top\mathbf{v}$ and $\mathbf{a} \in \mathbb{R}^n$ normalizes each column.

Same eigenvalues after update $\implies$ eigenvectors remain the same (deflation).

Eigenvalues are close $\implies$ numerator and denominator decrease equally fast.

Operation is numerically well-conditioned.

Finding the new eigenvalues

Eigenvalue interleaving

Let $\mathbf{L} = \mathbf{U}\mathrm{diag}(\boldsymbol{\lambda})\mathbf{U}^\top$ and $\tilde{\mathbf{L}} = \mathbf{L} + \rho\,\mathbf{v}\mathbf{v}^\top = \tilde{\mathbf{U}}\mathrm{diag}(\tilde{\boldsymbol{\lambda}})\tilde{\mathbf{U}}^\top$. Then, if $\rho > 0$,

$$\lambda_1 \leqslant \tilde{\lambda}_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n \leqslant \tilde{\lambda}_n, \tag{5}$$

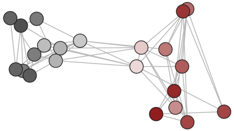
with the inequalities reversed when $\rho < 0$.

Given $\mathbf{z} = \mathbf{U}^\top \mathbf{v}$, it can be shown that the new eigenvalues are the roots of

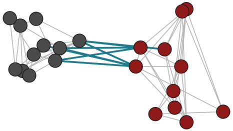
$$1 + \rho \sum_{i=1}^{n} z_i^2 \,/\, (\tilde{\lambda}_j - \lambda_i) = 0. \tag{6}$$

Solved efficiently using Netwon's method [Gu and Eisenstat, 1996].

Finding a suitable hierarchy



Fiedler vector



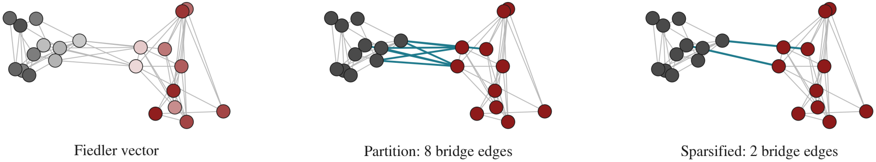
Partition: 8 bridge edges

We want small k. **Balanced cuts** via spectral bisection (Fiedler vector).

Accept partition only if theoretical complexity reduces.

Cut sparsification: reduce k via sparsification [Spielman and Srivastava, 2008].

Finding a suitable hierarchy

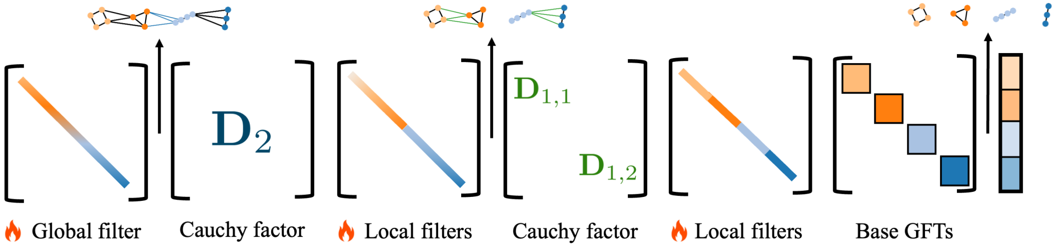


We want small k. **Balanced cuts** via spectral bisection (Fiedler vector).

Accept partition only if theoretical complexity reduces.

Cut sparsification: reduce k via sparsification [Spielman and Srivastava, 2008].

L2G-Net architecture



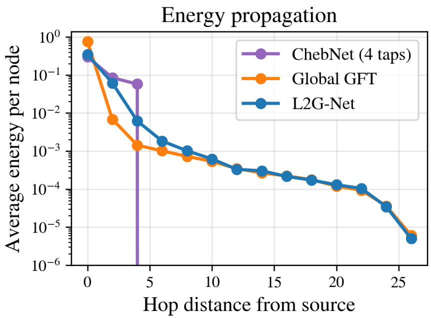
Local filters capture patterns within subgraphs.

Global filter models long-range subgraph interactions.

Intuition: architecture **adapts to graph structure** before training begins.

L2G-Net: inductive bias

L2G-Net provides a trade-off between MPNN locality and GFT globality.

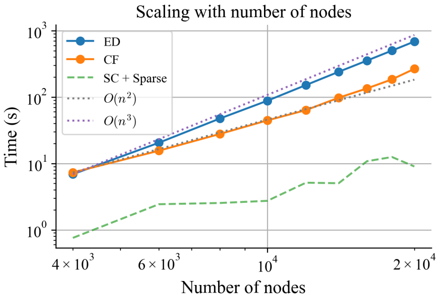


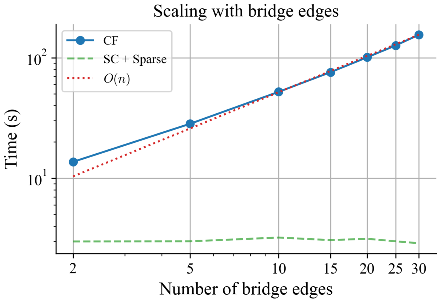
- Insert an impulse in a node.
- Apply the operator. Use randomly initialized filters.
- Measure energy k-hops away.
- Average over random filters.

Empirical complexity validation

Compute eigendecomposition (ED) and Cauchy Factorization (CF).

Barabási–Albert random graphs.





Runtime scales as $O(n^2)$ and $O(k)$, matching theoretical predictions.

Experimental setup

We test L2G-Net in:

- A transductive benchmark for heterophilous graphs [Platonov et al., 2023].

  Quantifies performance against graph transformers.

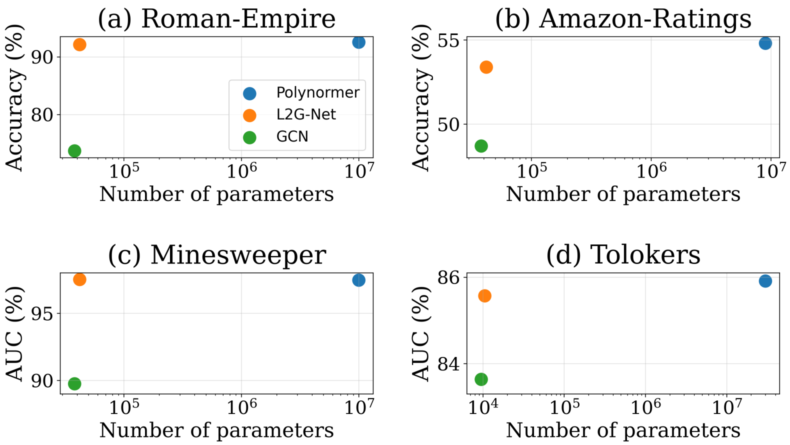
- A transductive benchmark of large scale graphs.

  Demonstrates L2G-Net scales to graphs beyond reach for the GFT.

- Inductive tasks on small graphs.

  Tests L2G-Net in graph-level benchmarks.

GNNs in [Platonov et al., 2023]

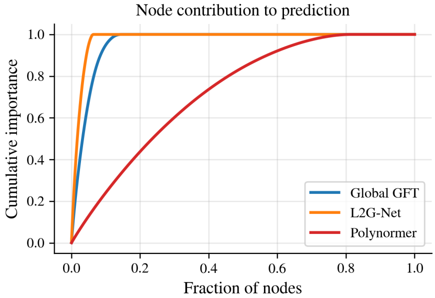


Similar or better performance than Polynormer, same parameters as GCN.

Graph topology

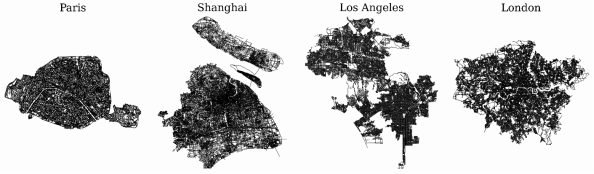
L2G-Net has more graph localization than GFT.

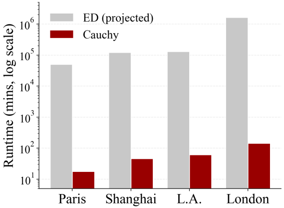
L2G-Net exploits better the graph topology than Polynormer.

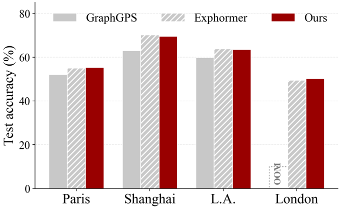


- Task: node classification.
- Use GradCAM to measure how many nodes contribute to prediction.
- Compute cumulative plot of contributions.

GNNs in CityNetworks [Liang et al., 2025]





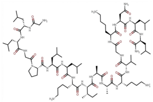


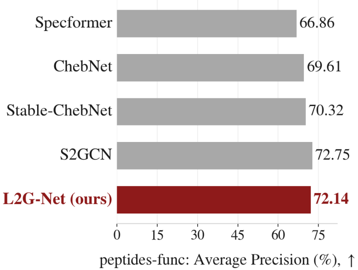
- **Global**: Estimate eccentricity of each node.
- Scales to graphs where GFT is unfeasible.
- Matches performance with less parameters.

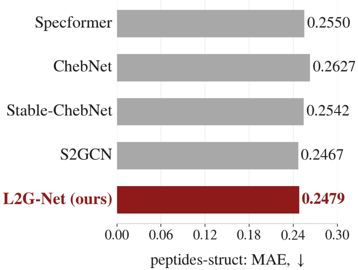
GNNs in drug discovery [Dwivedi et al., 2022]

Global: chemical properties depend on **interactions across the whole molecule.**

Local: **functional groups** govern short-range chemistry.







L2G-Net **captures both regimes** with a single architecture.

Conclusions and future work

L2G-Net: **exact GFT** with **local-to-global** inductive bias.

Cauchy factorization reduces complexity from $O(n^3)$ to $O(kn^2)$.

- Competitive accuracy with far fewer parameters.

Future work:

- Extension to **directed and dynamic graphs**.

Questions? samuelf9@usc.edu

Webpage: https://sf219.github.io/L2G_NET/

References I

[Defferrard et al., 2016] **Defferrard, M., Bresson, X., and Vandergheynst, P. (2016).**
Convolutional neural networks on graphs with fast localized spectral filtering.
Advances in neural information processing systems, 29.

[Dwivedi and Bresson, 2021] **Dwivedi, V. P. and Bresson, X. (2021).**
A generalization of transformer networks to graphs.
AAAI Workshop on Deep Learning on Graphs: Methods and Applications.

[Dwivedi et al., 2022] **Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. (2022).**
Long range graph benchmark.
Advances in Neural Information Processing Systems, 35:22326–22340.

References II

[Fernández et al., 2025] Fernández, S., Pavez, E., and Ortega, A. (2025).
Fast DCT+: A family of fast transforms based on rank-one updates of the path graph.
In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.* IEEE.

[Gu and Eisenstat, 1996] Gu, M. and Eisenstat, S. C. (1996).
Efficient algorithms for computing a strong rank-revealing qr factorization.
SIAM Journal on Scientific Computing, 17(4):848–869.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017).
Semi-supervised classification with graph convolutional networks.
In *International Conference on Learning Representations*.

References III

[Liang et al., 2025] Liang, H., Borde, H. S. d. O., Sripathmanathan, B., Bronstein, M., and Dong, X. (2025).
Towards quantifying long-range interactions in graph machine learning: a large graph dataset and a measurement.
arXiv preprint arXiv:2503.09008.

[Ortega et al., 2018] Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. (2018).
Graph signal processing: Overview, challenges, and applications.
Proceedings of the IEEE, 106(5):808–828.

References IV

[Spielman and Srivastava, 2008] Spielman, D. A. and Srivastava, N. (2008).
Graph sparsification by effective resistances.
In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 563–568.