Machine Learning with graphs - Project Defense

Delaunay Graph: Addressing Over-Squashing and Over-Smoothing Using Delaunay Triangulation
by Attali H., Duscaldi D. and Pernelle N. [2]

Edwin Roussin and Tristan Waddington

Supervised by Jhony H. Giraldo
IP-Paris, CEMST

26/03/2025

Delauney triangulation

Reconstruct a graph completely from projected features using the Delaunay triangulation.
$\Rightarrow$ Avoid **over-smoothing** and **over-squashing**.

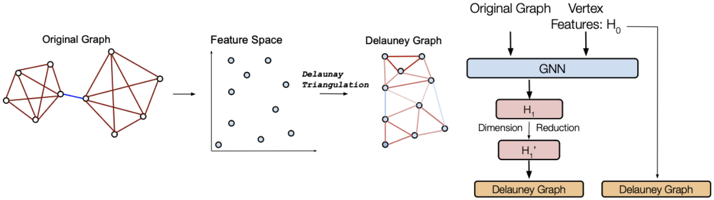


Figure: Illustration of the Delaunay rewiring [2, Attali al., 2024]

Outline

Need of graph rewiring

Over-Squashing: inefficient information propagation

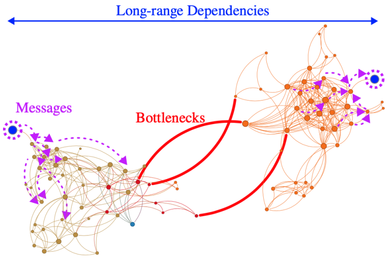


Figure: Illustration of Bottlenecks [Giraldo, Lecture GNNs, 2025]

GNNs struggle to propagate info to distant nodes: **bottleneck** when aggregating messages across a long path [1, Alon et al., 2021].
Causes **over-squashing** of exponentially growing info into fixed-size vectors. $\Rightarrow$ *Perform poorly when prediction task depends on long-range interaction.*

Vulnerable GNNs

GNC and GIN *absorb incoming edges equally*, more susceptible to over-squashing than GAT and GGNN.

Curvature metric

Discrete Ricci curvature [7, Topping et al. 2021] to identify bottlenecks.

Over-Smoothing: consequence of message passing paradigm

Message-passing neural networks (MPNN):

Iterative approach, updating node representations through the local aggregation of information from neighboring nodes.

Causes **over-smoothing** by the need to stack additional layers to capture non-local interactions. Will smooth-out heterophilic graphs. $\Rightarrow$ *Nodes' representations are similar.*

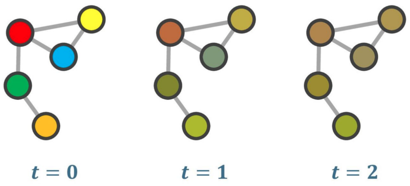


Figure: Illustration of Over-smoothing by Alex Ganose

Existing solutions

Identify the quality of the message passing:

- **Graph structure analysis** using curvature, but does not scale.
  Highly positive curved edges $\rightarrow$ over-smoothing [5, Nguyen et al., 2023].
  Highly negative curved edges $\rightarrow$ over-squashing [7, Topping et al., 2021].
- **Need original graph** but sometimes only features available (NER, documents, ...).

Avoid over-smoothing in preventing the embedding to become the same:

- **Normalization** with PairNorm [9, Zaho, 2020].
- **Rewiring** Drop edges, at random [6, Rong, 2019] or in finding the potential good ones [3, Giraldo, 2023]

Over-smoothing and over-squashing are intrinsically related

Inevitable trade-off between these two issues, as they cannot be alleviated simultaneously.
Quadratic complexity in the number of nodes (or edges).

Key technical novelty of the paper

Theoretical analysis

Delaunay rewiring

Is an extreme **4 steps rewiring** method.

1. First GNN[a] construct **node embeddings** .
2. Reduce the embedding with **UMAP** in dim 2.
3. **Rebuilt edges with Delaunay triangulation**.
4. Second GNN **mix with the original features** of the graph.

[a] GCN from [4, Kipf and Welling, 2017]

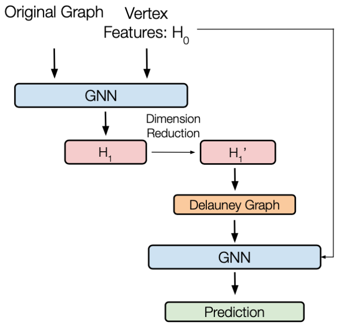


Figure 2: Illustration of the rewiring method using the features obtained by a GNN.

Initial Thoughts

Simplicity of the Method

No hyper-parameters = no grid-search.
Complexity of $\mathcal{O}(N \log N)$

Graph creation method

Create a graph from the embedding $\Rightarrow$ no need for the original graph.

Umap in 2 dimensions only

Triangulation in higher dimensions $\Rightarrow$ longer time + denser resulting graphs [a] + worse accuracy.

First GNN

Embed the initial smoothing and squashing? But needed for quality of embedding. Long range dependencies?

[a] Generalized triangles in dim=3: have 6 edges, 10 in dim=4

Delaunay Graph properties

Sparse graphs
Raise the homophily value of heterophilic graphs.

Reduce over-squashing

$\iff$ Reduce the negative curved edges
$\iff$ maximize triangles + minimize squares.

Reduce over-smoothing

Largest cliques limited to 3 nodes $\Rightarrow$ no over-smoothing [5, Nguyen et al, 2023].

Experimental Evaluation

Methodology

Aim to reproduce as closely as possible the experiments of the authors.

- Get same datasets, and preprocess them.
- Train the models with the same hyperparameters.
- Finally, we evaluate the models with the same metrics.

Results

Text Dataset preparation

- Text8 (English Wikipedia) processed.
- OpenWebText (GPT2 training material) processed in 18 hours.
- train/validation/test split according to experimental methodology.

We were unable to run the models

- Training impossible on our machines (size of model).
- Does not work on Colab (tensorflow numpy2 incompatibilities).
- No trained model shared by the authors.

Discussion

We faced huge challenges

- Graphs

Future paper that will be explored in the report:

- *Cayley Graph Propagation* by JJ Wilson, Maya Bechler-Speicher, Petar Veličković [8]

Conclusion

Conclusion

- Cannot confirm the results o
- Not able to reproduce.
- Hard time digging in code and documentation.

Do you have any question?

References I

Uri Alon and Eran Yahav.
On the bottleneck of graph neural networks and its practical implications, 2021.

Hugo Attali, Davide Buscaldi, and Nathalie Pernelle.
Delaunay graph: Addressing over-squashing and over-smoothing using delaunay triangulation.
In *Forty-first International Conference on Machine Learning*, 2024.

Jhony H. Giraldo, Konstantinos Skianis, Thierry Bouwmans, and Fragkiskos D. Malliaros.
On the trade-off between over-smoothing and over-squashing in deep graph neural networks.
In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 566–576. ACM, October 2023.

T. N. Kipf and M. Welling.
Semi-supervised classification with graph convolutional networks.
In *Proceedings of the International Conference on Learning Representations*, 2017.

K. Nguyen, N. M. Hieu, V. D. Nguyen, N. Ho, S. Osher, and T. M. Nguyen.
Revisiting over-smoothing and over-squashing using ollivier-ricci curvature.
In *International Conference on Machine Learning*, pages 25956–25979. PMLR, 2023.

Y. Rong, W. Huang, T. Xu, and J. Huang.
Dropedge: Towards deep graph convolutional networks on node classification.
In *International Conference on Learning Representations*, 2019.

References II

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein.
Understanding over-squashing and bottlenecks on graphs via curvature, 2022.

JJ Wilson, Maya Bechler-Speicher, and Petar Veličković.
Cayley graph propagation, 2024.

Lingxiao Zhao and Leman Akoglu.
Pairnorm: Tackling oversmoothing in gnns.
In *International Conference on Learning Representations*, 2020.

Curvature

Balance Forman Curvature [7, Topping, 2022] is computed over cycles of size 4.

$$c_{ij} = \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{\sharp_\Delta}{\max(d_i, d_j)} + \frac{\sharp_\Delta}{\min(d_i, d_j)} + \frac{\max(\sharp^i_\Box, \sharp^j_\Box)^{-1}}{\max(d_i, d_j)}(\sharp^i_\Box + \sharp^j_\Box)$$

where $\sharp_\Delta$ is the number of triangles based at e_{ij}, $\sharp^i_\Box$ is the number of 4-cycles based at e_{ij} starting from i without diagonals inside.

Curvature of graph edges

- Positive curvature edges establish connections between nodes belonging to the same community. Highly positive curved edges $\rightarrow$ over-smoothing [5, Nguyen et al., 2023].
- Negative curvature edges connect nodes from different communities. Highly negative curved edges $\rightarrow$ over-squashing [7, Topping et al., 2021].

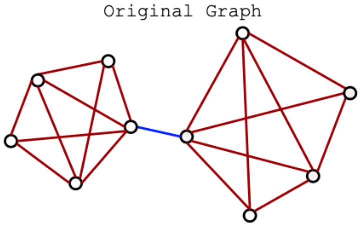


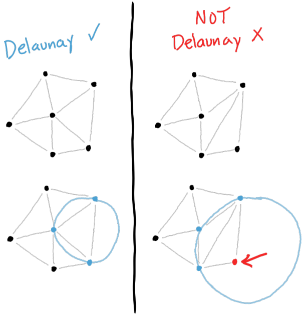
Figure: Example graph: in red the edges with positive curvature ($\sim$ 3), in blue with negative curvature (-1.2) [2, Attali al., 2024]

Delaunay Triangulation

Definition

A Delaunay triangulation, denoted as $DT(P)$, for a set P of points in the d-dimensional Euclidean space, is a triangulation where no point in P resides within the circum-hypersphere of any d-simplex in $DT(P)$.

In two dimensions, Delaunay triangulations maximize the angles of triangles formed by a set of points $\rightarrow$ triangle $\sim$ equilateral.
Figure: Sam Westrick



UMAP

Uniform Manifold Approximation and Projection (UMAP) is a dimensionality reduction technique that can be used for visualisation similarly to t-SNE, but also for general non-linear dimension reduction. UMAP constructs a high dimensional graph representation of the data then optimizes a low-dimensional graph to be as structurally similar as possible.

Advantages

- **Speed**: UMAP is faster than t-SNE.
- **Global structure**: UMAP preserves more of the global structure.
- **Separation**: clearly separate groups of similar categories.

Dimensionality reduction technique is perfect - by necessity, we're distorting the data to fit it into lower dimensions - and UMAP is no exception. But it is a powerful tool to visualize and understand large, high-dimensional datasets.

Hyperparameters choice

Most common: `n_neighbors` and `min_dist`, control the balance between local and global structure.

Figure: Illustration of UMAP hyperparameters from Google PAIR