

Graphe de Delaunay : Résoudre le problème d'écrasement excessif et de lissage excessif à l'aide de la triangulation de Delaunay

Hugo Attali*, Davide Buscaldi*
Nathalie Pernelle *

* Université Paris Sorbonne Nord, LIPN UMR 7030
attali@lipn.univ-paris13.fr

Résumé. Les GNNs (Graph Neural Networks) reposent sur l'échange de messages pour diffuser l'information le long des arêtes du graphe. Cette approche rend l'efficacité des architectures fortement dépendante de la structure spécifique du graphe en entrée. Certaines topologies de graphes entraînent une propagation inefficace de l'information, ce qui conduit à un phénomène connu sous le nom de sur-compression (over-squashing). Alors que la majorité des méthodes existantes traitent la sur-compression en réorganisant les arêtes du graphe (rewiring method) en entrée, notre approche novatrice consiste à construire un graphe directement à partir des caractéristiques en utilisant la triangulation de Delaunay. Nous montrons que les propriétés topologiques du graphe ainsi obtenu sont avantageuses pour atténuer à la fois la sur-lissage (oversmoothing) et la sur-compression. Nos expérimentations approfondies démontrent que notre méthode surpasse les techniques de rewiring de graphes déjà existantes.

1 Introduction

La prolifération de données structurées sous forme de graphes a fait émerger de nombreux algorithmes d'apprentissage automatique spécialisés dans le traitement de ce format de données. Les GNNs jouent un rôle crucial dans la résolution de diverses tâches associées aux graphes et ces architectures trouvent des applications dans des domaines variés tels que la chimie, la recherche d'information, l'analyse des réseaux sociaux ou les systèmes de recommandation. Les GNNs utilisent un processus dit de passage de message, mettant à jour les représentations des nœuds par l'agrégation locale des informations provenant des nœuds voisins. À chaque étape, les nœuds captent des informations de plus en plus éloignées dans le graphe, permettant de capturer à la fois des interactions locales (voisins immédiats) et, avec suffisamment de couches, des interactions plus globales (Gilmer et al., 2017). Lorsque les nœuds adjacents présentent des étiquettes dissemblables (cas des graphes hétérophiles), les MPNNs classiques peuvent voir leur performance nettement diminuer. Cette limitation majeure nécessite l'empilement de couches supplémentaires pour capturer des interactions non locales. Malheureusement, ce passage excessif de messages tend à rendre les caractéristiques

des nœuds de plus en plus similaires, conduisant à un lissage excessif des représentations des nœuds.

Un autre problème bien étudié est le phénomène dit de surcompression ou d'écrasement excessif (Alon et Yahav, 2021), qui se produit lorsque une quantité exponentiellement croissante d'informations est compressée dans un vecteur de taille fixe, ce qui peut considérablement diminuer l'efficacité du passage de messages. De nombreuses méthodes existantes atténuent ce phénomène en réorganisant le graphe d'entrée pour minimiser les goulets d'étranglement structurels (i.e. méthodes dites de rewiring). Pour ce faire, la plupart des méthodes s'appuient sur des caractéristiques structurelles du graphe, qui peuvent être locales, telles que la courbure des arêtes (Topping et al., 2022; Nguyen et al., 2023), ou plus globales, comme la résistance (Black et al., 2023). Leur principale limitation réside dans la nécessité d'analyser la structure du graphe d'entrée à l'aide de méthodes qui passent difficilement à l'échelle, et le choix des hyperparamètres dépend souvent fortement des caractéristiques du jeu de données. De plus, les méthodes de rewiring modifient un graphe original qui n'est pas toujours disponible (Attali et Tomeh, 2024).

Dans cet article, nous présentons Delaunay Rewiring (DR), une méthode de réorganisation structurelle qui intègre les caractéristiques des nœuds avec une complexité raisonnable pour atténuer à la fois les problèmes de surcompression et de lissage excessif (Attali et al., 2024b)¹. Au lieu de modifier la structure existante du graphe, notre approche consiste à reconstruire le graphe en utilisant des caractéristiques extraites par triangulation de Delaunay.

L'efficacité de cette méthode sur des graphes hétérophiles, mais aussi sur des graphes homophiles, est démontrée par une évaluation approfondie sur 10 jeux de données.

2 MPPNs

Les réseaux de neurones de graphes (Graph Neural Networks) fonctionnent selon le paradigme des réseaux de neurones à passage de messages (MPNNs). Ce paradigme consiste à trouver une bonne manière de mettre à jour la représentation d'un nœud en agrégeant les représentations des nœuds voisins.

Malgré leur utilisation répandue, les GNNs font face à divers défis, en particulier dans des environnements hétérophiles où les nœuds voisins tendent à avoir des étiquettes différentes. De plus, compte tenu de leur fonctionnement local, les GNNs peinent à modéliser efficacement les interactions à longue distance. Pour échanger des informations entre deux nœuds à une distance k , il est nécessaire d'empiler au moins k couches. Cette augmentation du nombre de couches entraîne deux défauts majeurs : le lissage excessif (oversmoothing) et la sur-compression (oversquashing).

Oversmoothing : Quand le nombre de couches augmente, le passage de messages s'intensifie et à chaque couche, les caractéristiques des nœuds deviennent progressivement plus similaires ce qui entraîne une baisse de performance (Cai et Wang, 2020a). Une stratégie efficace pour atténuer le lissage excessif consiste à incorporer des connexions résiduelles dans les réseaux profonds. D'autres approches reposent sur l'idée de réduire le nombre d'arêtes (Cai et Wang, 2020b; Chen et al., 2020). Certaines structures de sous-graphes, telles que les cliques

1. Code disponible à cette adresse : <https://github.com/Hugo-Attali/Delaunay-Rewiring>

de grande taille, peuvent accentuer l'effet de lissage excessif et certaines approches cherchent à réduire la densité de ces structures (Nguyen et al., 2023).

Over-squashing : Certaines tâches nécessitent une propagation de l'information à longue portée, à travers plusieurs couches. Cependant, la taille fixe des vecteurs de caractéristiques des nœuds entraîne un épuisement rapide de leur capacité de représentation. Cela conduit à la surcompression lorsque une quantité d'informations croissant de manière exponentielle est compressée dans un vecteur de taille fixe. Dans de tels scénarios, les GNNs peuvent intégrer des caractéristiques globales du graphe lors du processus d'apprentissage de la représentation (Gilmer et al., 2017). Une autre stratégie consiste à réorganiser le graphe d'entrée pour améliorer la connectivité et atténuer les goulets d'étranglement structurels (Alon et Yahav, 2021).

Homophilie : L'homophilie d'un graphe joue un rôle crucial dans l'efficacité des architectures pour les tâches de classification de nœuds. Diverses mesures de l'homophilie ont été proposées dont deux sont couramment utilisées : l'homophilie des nœuds qui calcule pour chaque nœud la proportion moyenne de voisins partageant la même classe (Pei et al., 2020), et l'homophilie des arêtes, qui représente la fraction des arêtes connectant des nœuds de même classe (Zhu et al., 2020). Pour les graphes présentant une faible homophilie, il est crucial de rechercher des informations sur de plus longues distances. Mais dans ce cas, les structures de graphes présentant des goulets d'étranglement posent de réels défis. Exploiter la courbure des arêtes du graphe est une façon de capturer l'information locale du graphe : les arêtes de courbure positive établissent des connexions entre des nœuds appartenant à la même communauté, tandis que les arêtes de courbure négative connectent des nœuds issus de communautés différentes. La courbure discrète caractérise en effet la connectivité entre les voisins de deux nœuds (Forman, 2003; Ollivier, 2007). De nombreuses études ont montré l'intérêt de la courbure des arêtes dans diverses tâches liées aux graphes, comme la détection de communautés (Ni et al., 2019) ou la classification de nœuds en utilisant un mécanisme d'attention (Ye et al., 2019). De nombreuses mesures de la courbure des arêtes ont été étudiées :

Augmented Forman Curvature (AF) : La mesure de courbure proposée par (Samal et al., 2018) étend la courbure de Forman en prenant en compte les triangles dans le graphe afin de la rendre plus expressive. Pour un graphe non orienté, la courbure c_{ij} d'une arête e_{ij} est calculée par : $c_{ij} = 4 - d_i - d_j + 3m$ où m est le nombre de triangles qui contient e_{ij} .

Balanced Forman Curvature (BF) : La mesure de courbure proposée par (Topping et al., 2022) permet de considérer également des cycles de taille 4 :

$$c_{ij} = \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2 \frac{m}{\max\{d_i, d_j\}} + \frac{m}{\min\{d_i, d_j\}} + \frac{(\Gamma_{\max})^{-1}}{\max\{d_i, d_j\}} (\gamma_i + \gamma_j)$$

où $\Gamma_{\max}(i, j)$ est le nombre max de 4-cycles auquel appartient e_{ij} , et γ_i est le nombre de 4-cycles basé sur e_{ij} ne présentant pas de diagonales.

Des recherches récentes ont démontré qu'il existait un lien entre la courbure des arêtes et le lissage excessif ou la surcompression. (Nguyen et al., 2023) montre que le lissage excessif peut être attribué à des régions du graphe ayant des arêtes fortement courbées positivement. D'autre part, (Topping et al., 2022) démontrent que les arêtes fortement courbées négativement jouent un rôle central dans la présence de goulets d'étranglement. .

Méthodes de Rewiring : Pour atténuer le phénomène de surcompression, la modification du graphe d'entrée i.e la matrice d'adjacence est une méthode couramment utilisée pour diminuer les structures de graphes qui nuisent à la diffusion de l'information. (Alon et Yahav, 2021) proposent de modifier la dernière couche du GNN pour établir des connexions entre tous les nœuds. (Topping et al., 2022) introduisent une approche de rewiring basé sur SDRF (Stochastic Discrete Ricci Flow), visant à améliorer la courbure de Forman des arêtes courbées négativement en ajoutant et en supprimant des arêtes. Des lors de nombreuses méthodes se basent sur la courbure pour effectuer le rewiring (Topping et al., 2022; Nguyen et al., 2023; Attali et al., 2024a). (Karhadkar et al., 2023) ont proposé d'ajouter des arêtes à chaque étape pour maximiser l'écart spectral. (Nguyen et al., 2023) propose une approche permettant de supprimer les arêtes dont la courbure est trop positive pour réduire le lissage excessif et d'éliminer également les arêtes de courbure trop négative afin d'atténuer la surcompression. Par ailleurs, (Black et al., 2023) montrent que les arêtes ayant une mesure de résistance élevée contribuent à la surcompression et proposent une méthode de rewiring qui ajoute des arêtes au graphe afin d'en minimiser la résistance totale. Plus de détails sur les méthodes de rewiring sont présentes dans ce survey (Attali et al., 2024c).

3 Graphe de Delaunay

Alors que la majorité des méthodes de rewiring modifient la structure originale du graphe, nous proposons une reconstruction complète du graphe, basée uniquement sur les caractéristiques des nœuds, et ignorant les arêtes du graphe original. Les nouvelles arêtes du graphe sont générées en appliquant une triangulation de Delaunay sur les caractéristiques des nœuds, comme le montre la figure 1. Nous démontrons à travers l'analyse de la courbure du nouveau graphe, que cette réorganisation présente de bonnes propriétés topologiques et atténue efficacement les phénomènes de surcompression, surlissage, de goulet d'étranglement.

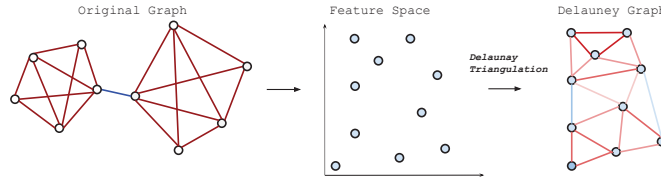


FIG. 1 – Construction du graphe de Delaunay, en rouge les arêtes à courbure positive, en bleu celles à courbure négative

Une **triangulation de Delaunay**, notée $DT(P)$, pour un ensemble P de points dans l'espace euclidien de dimension d , est une triangulation où aucun point de P ne se trouve à l'intérieur de l'hypersphère circonscrite de tout d -simplex de $DT(P)$. En deux dimensions, les triangulations de Delaunay maximisent les angles des triangles formés par un ensemble de points, s'efforçant de créer des triangles qui sont aussi proches que possible d'être équilatéraux. Elles garantissent également que le cercle circonscrit de chaque triangle ne contient aucun autre point de l'ensemble. Dans ce contexte, la proximité de deux points augmente la probabilité qu'ils soient inclus dans le même triangle.

Pour atténuer le problème de la surcompression, nous souhaitons réduire le nombre d’arêtes à courbure négative. Or, le nombre de triangles incidents à une arête joue un rôle dans l’augmentation de sa courbure. Appliquer une triangulation permet de maximiser la valeur de m dans les calculs des courbures AF et BF, tout en garantissant une taille maximale de clique.

Nous effectuons la triangulation en considérant l’ensemble des nœuds, dont la position dans un espace de dimension d est représentée par d caractéristiques. Comme nous voulons maximiser les triangles nous devons effectuer une triangulation en 2D (2-simplex). Nous réduisons les dimensions à 2 en appliquant UMAP (McInnes et al., 2018). Nous avons tout d’abord triangulé les graphes en considérant les caractéristiques originales des nœuds. Cependant, ces caractéristiques manquent souvent d’expressivité ou souffrent d’une mauvaise qualité et nous avons observé que, si les résultats étaient meilleurs pour le graphe de Delaunay que pour le graphe original (voir (Attali et al., 2024b)), une autre stratégie pouvait être envisagée. Nous proposons d’effectuer la triangulation sur les embeddings des nœuds obtenus à l’aide d’un GNN. Plus précisément, les caractéristiques originales sont passées dans un premier GNN (i.e. un réseau de convolution de graphes suivant le modèle proposé par (Kipf et Welling, 2017) pour obtenir les embeddings utilisés pour construire le graphe de Delaunay (après la réduction de dimension avec UMAP). Ensuite, nous effectuons la tâche de prédiction en prenant en compte la structure du graphe issue de la triangulation de Delaunay et les caractéristiques originales (H_0) des nœuds.

3.1 Propriétés des graphes de Delaunay

La table 1 montre que le nombre d’arêtes du graphe de Delaunay est 6 fois plus faible que dans le graphe original (3 fois si le graphe est dirigé). Cette réduction importante améliore non seulement l’efficacité globale (i.e. réduction des nœuds à trop faible ou fort degré), mais apporte également des avantages notables en termes de performances computationnelles.

La table 1 montre également que la valeur d’homophilie est beaucoup plus élevée pour les graphes obtenus après la triangulation de Delaunay pour les graphes hétérophiles (de 33 % à 1060 %).

Type	Dataset	# Nodes	# Classes	# Original Edges	O-Homophily	# Delaunay Edges	D-Homophily	Homophily Gain
Heterophile	Squirrel	5 021	5	217 073	0.22	31 170	0.59	+168%
	Chameleon	2 277	5	36 101	0.25	13 630	0.69	+176%
	Texas	181	5	309	0.06	1 072	0.63	+950%
	Wisconsin	251	5	499	0.06	1 470	0.55	+817%
	Cornell	181	5	295	0.11	1 064	0.67	+509%
	Roman-empire	22 662	18	32 927	0.06	135 922	0.58	+1060%
	Actor	7 600	5	33 544	0.24	45 520	0.40	+33%
Homophile	Citeseer	3 312	6	4 715	0.71	19 923	0.78	+10%
	Cora	2 708	7	5 429	0.83	16 214	0.88	+6%
	Pubmed	19 717	3	44 348	0.77	118 192	0.86	+9%

TAB. 1 – Statistiques des jeux de données originaux (O) et après triangulation de Delaunay (D).

Courbure des arêtes dans un graphe de Delaunay : Dans l’article original (Attali et al., 2024b), nous montrons que le graphe de Delaunay, ne comporte pas d’arêtes fortement négativement courbées, qui sont responsables des goulets d’étranglement (Topping et al., 2022). De plus, il ne possède pas d’arêtes fortement positivement courbées, ce qui atténue le sur-lissage (Nguyen et al., 2023).

Resistance dans les graphes de Delaunay : La résistance est une métrique efficace pour quantifier l’over-squashing (Black et al., 2023). Nous avons montré que la résistance moyenne entre chaque paire de nœuds était fortement réduite sur des petits jeux de données tels que Wisconsin, Cornell ou Texas (plus de 50%).

Complexité : La complexité de notre méthode de rewiring est en $\mathcal{O}(N \log(N))$, ce qui la rend plus efficace que la majorité des méthodes. De plus elle est exempte d’hyperparamètres, car elle se base exclusivement sur les caractéristiques des nœuds. Cela assure une plus grande robustesse et une plus grande facilité d’utilisation.

4 Experimentations

Datasets : Nous avons évalué cette approche sur dix ensembles de données pour la tâche de classification de nœuds (sept jeux de données hétérophiles et trois jeux de données homophiles). Les statistiques décrivant ces ensembles de données sont présentées dans la Table 1.

Baseline : Nous avons effectué une analyse comparative de notre méthode avec six autres techniques basées sur des stratégies de rewiring. Les méthodes incluent FA (Alon et Yahav, 2021), DIGL (Klicpera et al., 2019), SDRF (Topping et al., 2022), FOSR (Karhadkar et al., 2023), BORF (Nguyen et al., 2023) et GTR (Black et al., 2023).

Configuration de la méthode de Delaunay Rewiring : Contrairement aux autres jeux de données, pour les jeux de données de l’Empire romain, Cornell, Texas et Wisconsin, nous appliquons la triangulation sur les caractéristiques d’origine.

Setup : Nous avons utilisé le même cadre que celui décrit par (Pei et al., 2020) pour évaluer la robustesse de chaque méthode.

	Base (GCN)	DIGL	SRDF	FOSR	BORF	GTR	DR
Cham.	65.35 ± 0.54	54.82 ± 0.48	63.08 ± 0.37	67.98 ± 0.40	65.35 ± 0.51	<u>68.03</u> ± 0.61	74.28 ± 0.48
Squir.	51.30 ± 0.38	40.53 ± 0.29	49.11 ± 0.28	52.63 ± 0.30	≥ 24h	<u>53.32</u> ± 0.44	65.25 ± 0.26
Actor	30.02 ± 0.22	26.75 ± 0.23	<u>31.85</u> ± 0.22	29.26 ± 0.23	31.36 ± 0.27	31.08 ± 0.28	41.36 ± 0.20
Texas	56.19 ± 1.61	45.95 ± 1.58	59.79 ± 1.71	61.35 ± 1.25	56.30 ± 1.61	57.18 ± 1.64	70.46 ± 1.61
Wisc.	55.12 ± 1.51	46.90 ± 1.28	<u>58.49</u> ± 1.23	55.60 ± 1.25	55.37 ± 1.47	57.22 ± 1.50	70.98 ± 1.50
Corn.	44.78 ± 1.45	44.46 ± 1.37	<u>47.73</u> ± 1.51	45.11 ± 1.47	46.81 ± 1.56	47.57 ± 1.52	67.22 ± 1.48
R-emp.	51.66 ± 0.17	53.93 ± 0.14	52.53 ± 0.13	52.38 ± 0.21	<u>58.58</u> ± 0.14	53.31 ± 0.23	61.99 ± 0.14
Cora	87.73 ± 0.25	<u>88.31</u> ± 0.29	87.73 ± 0.31	87.94 ± 0.26	87.72 ± 0.27	87.86 ± 0.28	91.39 ± 0.24
Citeseer	76.01 ± 0.25	76.22 ± 0.34	76.43 ± 0.32	76.34 ± 0.27	<u>76.49</u> ± 0.28	76.12 ± 0.28	81.14 ± 0.34
Pubmed	88.20 ± 0.10	<u>88.51</u> ± 0.10	88.16 ± 0.11	88.42 ± 0.10	88.34 ± 0.10	88.44 ± 0.10	88.69 ± 0.10

TAB. 2 – Résultats (accuracy) sur des jeux de données **hétérophiles** et **homophiles** avec une base **GCN**. Meilleur score en gras et meilleur second score souligné.

Resultats : Le tableau 2 montre que la méthode de rewiring basée sur la triangulation de Delaunay (DR) a obtenu la meilleure accuracy pour les 7 ensembles de données hétérophiles et les 3 ensembles de données homophiles. En moyenne, cette méthode a démontré une augmentation substantielle de 21,8 % des performances par rapport au GCN de base, et des résultats similaires ont été obtenus avec le GAT (voir (Attali et al., 2024b)). Les méthodes de référence comme SDRF, FOSR, BORF et GTR ont montré des avancées prometteuses sur les ensembles de données hétérophiles, mais présentent des limitations pour des graphes homophiles.

5 Conclusion

Dans cet article, nous introduisons une méthode de rewiring caractérisée par une complexité raisonnable et l’absence d’hyperparamètres. Notre approche consiste à utiliser les caractéristiques des nœuds pour restructurer le graphe, en tirant parti des effets avantageux de la triangulation de Delaunay pour améliorer l’homophilie du graphe et garantir des propriétés structurelles favorables à la diffusion de l’information. En évitant les arêtes fortement négativement et positivement courbées, le nouveau graphe construit présente une sensibilité réduite à l’oversmoothing et à l’oversquashing, ce qui entraîne un gain de performance substantiel sur divers ensembles de données dans le cadre de tâches de classification de nœuds.

Références

- Alon, U. et E. Yahav (2021). On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*.
- Attali, H., D. Buscaldi, et N. Pernelle (2024a). Curvature constrained mpnns : Improving message passing with local structural properties.
- Attali, H., D. Buscaldi, et N. Pernelle (2024b). Delaunay graph : Addressing over-squashing and over-smoothing using delaunay triangulation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Attali, H., D. Buscaldi, et N. Pernelle (2024c). Rewiring techniques to mitigate oversquashing and oversmoothing in gnns : A survey. *arXiv preprint arXiv :2411.17429*.
- Attali, H. et N. Tomeh (2024). Transductive legal judgment prediction combining BERT embeddings with delaunay-based GNNs. In N. Aletras, I. Chalkidis, L. Barrett, C. Goanță, D. Preotiuc-Pietro, et G. Spanakis (Eds.), *Proceedings of the Natural Legal Language Processing Workshop 2024*, Miami, FL, USA, pp. 187–193. Association for Computational Linguistics.
- Black, M., Z. Wan, A. Nayyeri, et Y. Wang (2023). Understanding oversquashing in gnns through the lens of effective resistance. In *International Conference on Machine Learning*, pp. 2528–2547. PMLR.
- Cai, C. et Y. Wang (2020a). A note on over-smoothing for graph neural networks. In *proceedings of the Graph and representation learning (GRL@ICML)*.
- Cai, C. et Y. Wang (2020b). A note on over-smoothing for graph neural networks. *arXiv preprint arXiv :2006.13318*.
- Chen, D., Y. Lin, W. Li, P. Li, J. Zhou, et X. Sun (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 34, pp. 3438–3445.
- Forman, R. (2003). Bochner’s method for cell complexes and combinatorial ricci curvature.
- Gilmer, J., S. S. Schoenholz, P. F. Riley, O. Vinyals, et G. E. Dahl (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR.

Grphe de Delaunay

- Karhadkar, K., P. K. Banerjee, et G. Montúfar (2023). Fcsr : First-order spectral rewiring for addressing oversquashing in gnns. In *International Conference on Learning Representations*, ICLR.
- Kipf, T. N. et M. Welling (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the ICLR*.
- Klicpera, J., S. Weißenberger, et S. Günnemann (2019). Diffusion improves graph learning. In *Advances in neural information processing systems*, NeurIPS.
- McInnes, L., J. Healy, et J. Melville (2018). Umap : Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv :1802.03426*.
- Nguyen, K., N. M. Hieu, V. D. Nguyen, N. Ho, S. Osher, et T. M. Nguyen (2023). Revisiting over-smoothing and over-squashing using ollivier-ricci curvature. In *International Conference on Machine Learning*, pp. 25956–25979. PMLR.
- Ni, C.-C., Y.-Y. Lin, F. Luo, et J. Gao (2019). Community detection on networks with ricci flow. *Scientific reports* 9(1), 9984.
- Ollivier, Y. (2007). Ricci curvature of metric spaces. *Comptes Rendus Mathematique* 345(11), 643–646.
- Pei, H., B. Wei, K. C.-C. Chang, Y. Lei, et B. Yang (2020). Geom-gcn : Geometric graph convolutional networks. In *Advances in neural information processing systems*, ICLR.
- Samal, A., R. Sreejith, J. Gu, S. Liu, E. Saucan, et J. Jost (2018). Comparative analysis of two discretizations of ricci curvature for complex networks. *Scientific reports* 8(1), 8650.
- Topping, J., F. Di Giovanni, B. P. Chamberlain, X. Dong, et M. M. Bronstein (2022). Understanding over-squashing and bottlenecks on graphs via curvature. *Proceedings of the International Conference on Learning Representations*.
- Ye, Z., K. S. Liu, T. Ma, J. Gao, et C. Chen (2019). Curvature graph network. In *International conference on learning representations*.
- Zhu, J., Y. Yan, L. Zhao, M. Heimann, L. Akoglu, et D. Koutra (2020). Beyond homophily in graph neural networks : Current limitations and effective designs. *Advances in Neural Information Processing Systems* 33, 7793–7804.

Summary

GNNs rely on the exchange of messages to distribute information along the edges of the graph. This approach makes the efficiency of architectures highly dependent on the specific structure of the input graph. Certain graph topologies lead to inefficient information propagation, resulting in a phenomenon known as over-squashing. While the majority of existing methods address over-squashing by rewiring the input graph, our novel approach involves constructing a graph directly from features using Delaunay Triangulation. We posit that the topological properties of the resulting graph prove advantageous for mitigating oversmoothing and over-squashing. Our extensive experimentation demonstrates that our method consistently outperforms established graph rewiring methods.