

Matrice d’adjacence courbée : intégration de la courbure des arêtes dans la transmission des messages

Hugo Attali*, Davide Buscaldi*,
Nathalie Pernelle *

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

Résumé. Les réseaux de neurones opérant sur les graphes suivent un processus itératif de mise à jour des représentations de nœuds qui se base sur l’agrégation de l’information issue des nœuds voisins, i.e. passage de message. Cela permet à ces architectures d’apprendre des représentations dépendant de la structure du graphe. Néanmoins, pour certaines topologies de graphe, cela conduit à une propagation inefficace de l’information connue sous le nom d’over-squashing. Récemment, il a été démontré que ce phénomène vient d’une partie du graphe qui peut être identifiée par une mesure de la courbure des arêtes. Dans cet article, nous proposons une nouvelle manière de réaliser le passage de message en diffusant l’information en fonction de la courbure des arêtes. Les expériences menées sur une tâche de classification de nœuds montrent que notre méthode s’avère plus efficace que les méthodes de pré-traitement de graphes existantes.

1 Introduction

L’apprentissage de représentations est un domaine de recherche en pleine expansion pour la modélisation de données structurées comme les graphes. La majorité des réseaux neuronaux opérant sur des graphes (*GNNs*) se base sur une technique appelée passage de messages (en anglais *message passing*) (Gilmer et al., 2017), où l’information contenue dans les plongements de nœuds est transmise aux nœuds voisins via les arêtes du graphe. Les *GNNs* se sont imposés dans de nombreux domaines (Zhou et al., 2020) (Guille et Attali, 2022), notamment la chimie, la recherche d’informations, les réseaux sociaux et les graphes de connaissances. Ces architectures ont démontré des résultats très intéressants pour résoudre des tâches au niveau du nœud, du graphe ou de l’arête. Bien qu’ils soient de plus en plus utilisés, les *GNNs* sont confrontés à un certain nombre de problèmes, notamment des performances limitées pour capturer de l’information à une longue distance (Alon et Yahav, 2021). La plupart des *GNNs* comme le GCN (Kipf et Welling, 2017) ou le GAT (Veličković et al., 2018) diffusent de l’information uniquement localement i.e aux voisins. Cela provoque une détérioration des résultats lorsqu’on travaille dans un environnement hétérophile, autrement dit lorsque les nœuds voisins ont des labels différents (Zhu et al., 2020) (Platonov et al., 2023).

Soient deux nœuds i et j situés à une distance k . Pour qu’ils puissent échanger de l’information au moins k couches doivent être empilées. Ainsi pour échanger de l’information avec

des nœuds situés à longue distance une approche naïve consiste à augmenter le nombre de couches. Toutefois, cette méthodologie conduit à deux problèmes bien connus pour les GNNs. D'une part le phénomène de surlissage qui survient lorsque le passage des messages est itéré de manière excessive. Dans ce cas, toutes les représentations des nœuds deviennent similaires et cela entraîne une détérioration des résultats Oono et Suzuki (2020) Cai et Wang (2020). D'autre part, à mesure que le nombre de couches d'un GNN augmente, un trop grand nombre d'informations peut avoir été agrégé dans un vecteur de taille fixe. Ainsi la représentation de ce nœud ne sera que très peu impactée par les représentations des nœuds à longue distance. Ce phénomène est appelé over-squashing.

Pour surmonter ce problème, les GNNs doivent être en mesure d'incorporer des représentations pas seulement locales mais aussi globales dans le processus de mise à jour des représentations. Certaines approches effectuent un pré-traitement de la matrice d'adjacence du graphe d'afin d'en améliorer la connectivité et d'éviter les goulots d'étranglement qui provoquent l'over-squashing (Alon et Yahav, 2021)(Topping et al., 2022) (Kedar Karhadkar et Montúfar, 2023).

Récemment, il a été montré que le phénomène d'over-squashing provient de certaines zones des graphes, zones que l'on peut identifier par une mesure de courbure des arêtes. Tandis qu'une solution consiste à modifier le graphe dans ces zones en ajoutant et en supprimant des arêtes, nous proposons de différencier la propagation des messages en fonction de la courbure des arêtes.

Contributions Nous proposons (1) une nouvelle mesure de l'homophilie qui se base sur la courbure des arêtes du graphe et qui permet de mieux caractériser le comportement communautaire du voisinage.

Motivés par cette nouvelle mesure, nous proposons (2) un nouveau modèle de transmission de passage de message, appelé CCMP (Courvature Constrained Message Passing) qui prend en compte la courbure des arêtes pour guider l'apprentissage en différenciant l'information venant des arêtes à courbure positive de celles à courbure négative. L'information peut alors être uniquement diffusée à travers les arêtes ayant une courbure négative, une courbure positive ou une combinaison des deux. Nous élargissons aussi cette diffusion à un voisinage de taille 2.

(3) Nous démontrons empiriquement un gain de performance sur des jeux de données hétérophiles et nous montrons que l'utilisation d'un passage de message restreint selon la courbure atténue l'over-squashing. *Le code pour reproduire les résultats est disponible.*¹

2 Travaux connexes

Réseaux de neurone à passage de messages Le succès de l'apprentissage profond dans le domaine euclidien a suscité un grand intérêt pour la généralisation des réseaux de neurones à des domaines non euclidiens comme les graphes. Soit $G = (V, E)$ un graphe simple non dirigé et connexe avec des représentations pour chaque nœud $h_i \in R^d, i \in V$. $\mathcal{N}(i)$ représente l'ensemble des voisins du nœuds i . L'objectif principal de l'approche par passage de messages (MPNNs) est de calculer une représentation de nœud qui capture les informations du voisinage (Gilmer et al., 2017). Ainsi, la mise à jour des représentations dépend grandement du graphe d'entrée, autrement dit de la matrice d'adjacence. Le passage de messages s'effectue en itérant

1. <https://github.com/Hugo-Attali/Curvature-Constrained-Message-Passing>

deux phases qui appliquent successivement les fonctions AGGREGATE et UPDATE afin de calculer les représentations h_i^ℓ pour la couche ℓ sur la base du message $m_i^{(\ell)}$ contenant des informations sur les voisins :

$$\begin{aligned} m_i^{(\ell)} &= \text{AGGREGATE}^{(\ell)} \left(h_i^{(\ell-1)}, \{h_j^{(\ell-1)} \mid j \in \mathcal{N}(i)\} \right), \\ h_i^\ell &= \text{UPDATE}^{(\ell)} \left(h_i^{(\ell-1)}, m_i^{(\ell)} \right) \end{aligned} \quad (1)$$

Dans le cas du Graph Convolutional Networks (GCN), la mise à jour des représentations s'effectue en normalisant par le degré de chaque nœud (Kipf et Welling, 2017) :

$$m_i^\ell = \sum_{j \in \mathcal{N}(i)} \frac{\mathbf{h}_j^\ell}{\sqrt{|\mathcal{N}(i)| |\mathcal{N}(j)|}}$$

où \parallel représente la concaténation et \mathcal{N}_i est l'ensemble des voisins du sommet i .

Pour les Graph Attention Network (GAT) les représentations sont mises à jour en utilisant un mécanisme d'attention qui pondère l'importance des représentations des nœuds voisins (Veličković et al., 2018) : $m_i^\ell = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^\ell \mathbf{h}_j$.

avec

$$\alpha_{ij}^{(\ell)} = \frac{\exp(\text{LeakyReLU}(z^{(\ell)} \cdot [h_i^{(\ell-1)} \mathbf{W}^{(\ell)} \parallel h_j^{(\ell-1)} \mathbf{W}^{(\ell)}]))}{\sum_{j \in \mathcal{N}_i} \exp(\text{LeakyReLU}(z^{(\ell)} \cdot [h_i^{(\ell-1)} \mathbf{W}^{(\ell)} \parallel h_j^{(\ell-1)} \mathbf{W}^{(\ell)}]))}$$

Ce score est paramétré par $z^{(l)}$ et $\mathbf{W}^{(l)}$, qui représentent respectivement un vecteur de poids et une transformation linéaire. L'activation LeakyReLU est calculée avec une pente négative de 0,2.

Le fait que les MPNNs classiques diffusent des messages par l'intermédiaire des arêtes du graphe s'avère particulièrement intéressant lorsque les nœuds adjacents partagent le même label (cas homophile). En revanche, travailler dans un environnement hétérophile peut alors conduire à de faibles performances. En effet, l'un des principaux inconvénients des MPNNs classiques est qu'ils ne propagent des messages qu'à un voisinage à distance 1. Des couches supplémentaires doivent être empilées pour propager à distance plus grande. Toutefois, cela conduit dans ce cas au phénomène d'over-squashing présenté en section 2.2.

2.1 Courbure dans les graphes

La notion de courbure est un bon moyen de caractériser le comportement local d'un graphe. La courbure des arêtes d'un graphe décrit la manière dont les voisins de deux nœuds sont structurellement connectés (fig 1). Forman (2003) et Ollivier (2007) ont été les premiers à proposer une mesure de la courbure pour les graphes. De nombreux travaux ont démontré l'utilité de la courbure des arêtes pour différentes tâches liées aux graphes. Ye et al. (2019) Attali et al. (2023) mettent en place un mécanisme d'attention incorporant la courbure des arêtes montrant les avantages d'une telle architecture pour une tâche de classification de nœuds.

Ni et al. (2019) Sia et al. (2019) utilise la courbure d'Ollivier pour la détection de communautés, en proposant notamment d'effectuer une segmentation des communautés en éliminant les arêtes à courbure négatives jusqu'à obtention de nouvelles composantes connexes. Plus récemment, Samal et al. (2018), Topping et al. (2022) ont proposé des extensions à la courbure de Forman pour améliorer son expressivité. Les travaux de Topping et al. (2022) ont montré la corrélation entre la courbure des arêtes et le phénomène d'over-squashing. Dans ce travail, nous utilisons la courbure d'Ollivier et la courbure proposé par Samal et al. (2018).

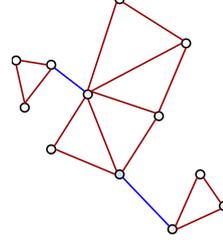


FIG. 1 – En rouge, les arêtes à courbure positive relient les nœuds d'une même communauté, en bleu, les arêtes à courbure négative relient les nœuds de communautés différentes.

2.2 Over-squashing

De nombreuses tâches sur les graphes exigent d'échanger de l'information entre nœuds à longue distance et nécessitent l'empilage de plusieurs couches de GNNs. Les représentations des nœuds sont ainsi agrégées à chaque étape avant d'être transmises au nœud suivant. La taille des vecteurs de représentations des nœuds restant constante, ils épuisent rapidement leur capacité à capturer les représentations de chaque nœud afin de conserver toutes les informations précédemment intégrées pouvant provoquer un goulot d'étranglement. Le phénomène d'over-squashing se produit lorsqu'une quantité d'informations trop importante est compressée dans un vecteur de taille fixe (Alon et Yahav, 2021).

Pour quantifier ce phénomène, certaines approches exploitent le trou spectral (Kedar Kahrhadkar et Montúfar, 2023), qui est étroitement lié à la constante de Cheeger (Chung et Graham, 1997) :

$$Ch(G) = \min_{1 \leq |S| \leq \frac{|V|}{2}} \frac{|\partial S|}{|S|}, \quad (2)$$

Avec $S \subset V$ et où $\partial S = \{(i, j) : i \in S, j \in \bar{S}, (i, j) \in E\}$

Si la valeur de la constante de Cheeger est petite, il existe un goulot d'étranglement qui provoque le phénomène d'over-squashing. Le calcul de la valeur précise de $Ch(G)$ est très coûteux. L'inégalité de Cheeger Alon et Milman (1984) Cheeger (1970) montre le lien entre le trou spectral et la constante de Cheeger. Nous notons A la matrice d'adjacence, D la matrice diagonale des degrés et $L = I - D^{-1/2}AD^{-1/2}$ le laplacien normalisé de G . Le trou spectral de G est la différence entre les deux premières valeurs propres $\lambda_2 - \lambda_1$ de L avec $\lambda_1 = 0$.

$$\frac{\lambda_2}{2} \leq Ch(G) \leq \sqrt{2\lambda_2} \quad (3)$$

Méthodes de pré-traitement De nombreuses méthodes effectuent un pré-traitement sur le graphe d'entrée *ie* modifier la matrice d'adjacence pour atténuer un goulot d'étranglement et ainsi limiter le phénomène d'over-squashing. Alon et Yahav (2021) sont les premiers à mettre

en évidence le problème d'oversquashing pour les GNNs. Ils proposent de modifier la dernière couche du GNNs en utilisant une matrice d'adjacence représentant le graphe complet. Topping et al. (2022) démontre que ce sont les arêtes très négativement courbées qui caractérisent le phénomène de goulôt d'étranglement et qui perturbent la qualité du passage de message. Aussi, les auteurs proposent une méthode de prétraitement, appelée Stochastic Discrete Ricci Flow (SDRF), qui va augmenter la courbure des arêtes à courbure fortement négative en rajoutant ou supprimant des arêtes. Kedar Karhadkar et Montúfar (2023) propose un algorithme (FOSR) permettant de rajouter des arêtes afin de maximiser le trou spectral. Comme calculer le trou spectral pour chaque ajout d'arêtes est coûteux, FOSR utilise une approximation de la première valeur non nulle i.e. λ_2 du Laplacien. Sans avoir pour objectif de diminuer le phénomène d'oversquashing, d'autres méthodes comme Klicpera et al. (2019) modifie la matrice d'adjacence pour améliorer la connectivité du graphe. Cette méthode rajoute des arêtes basée sur l'algorithme PageRank. Comme PageRank est basé sur des marches aléatoires, DIGL va améliorer la connectivité entre nœuds au sein de la même communauté du graphe.

3 Curvature Message Passing

Nous proposons de s'appuyer sur deux notions de courbure sur les arêtes pour la diffusion de l'information, la courbure d'Ollivier (Ollivier, 2007) et la "Augmented Forman Curvature" (Samal et al., 2018).

Courbure d'Ollivier Définissons une distribution de probabilité μ_i sur les nœuds du graphe de manière à appliquer à chaque nœud i une mesure de probabilité basée sur une marche aléatoire π :

$$\mu_i : j \mapsto \begin{cases} \pi & \text{if } j = i \\ (1 - \pi)/D_{ii} & \text{if } j \in \mathcal{N}(i) \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

Suivant les précédents travaux (Ni et al., 2018), nous choisissons $\pi = 0,5$. Nous considérons ensuite la distance de Wasserstein d'ordre 1, $W_1(i, j)$, correspondant au transport optimal des masses de probabilité des voisins de i vers les voisins de j .

$$W_1(\mu_i, \mu_j) = \inf_M \sum_{i,j \in V} dist(i, j) M(i, j), \quad (5)$$

où $M(i, j)$ est la "quantité de masse" déplacée de i à j en suivant le plus court chemin entre i et j . Enfin, la courbure d'Ollivier c_{ij} d'une arête e_{ij} peut être définie comme :

$$c_{ij} = 1 - \frac{W_1(\mu_i, \mu_j)}{dist(i, j)}, \quad (6)$$

où $dist(i, j)$ est le plus court chemin entre le nœud i et le nœud j . Le résultat de la courbure d'Ollivier est borné entre -1 et 1. Intuitivement, cette courbure (définie pour deux sommets du graphe) compare la distance entre les deux sommets et la moyenne des distances entre les voisins de ces deux sommets.

Matrice d’adjacence courbée : intégration de la courbure dans la transmission des messages

Augmented Forman Curvature La mesure de la courbure proposée par Samal et al. (2018) propose d’extendre la courbure de Forman pour la rendre plus expressive.

$$c_{ij} = 4 - D_{ii} - D_{jj} + 3m, \quad (7)$$

où m est le nombre de triangles contenu dans e_{ij} . Cette mesure n’est pas bornée et est moins coûteuse que la courbure d’Ollivier.

3.1 Homophilie basée sur la courbure

L’homophilie d’un graphe a un rôle très important dans la performance des GNNs pour une tâche de classification de nœuds. En effet, comme la plupart des GNNs propagent de l’information localement, si les nœuds voisins ont des labels différents (cas hétérophile) ces architectures obtiennent de mauvaises performances. Dans ce cas échanger de l’information à longue distance est nécessaire. Plusieurs mesures de l’homophilie existent dans la littérature (Pei et al.) (Zhu et al., 2020) (Platonov et al., 2022). Les plus communes sont (1) la mesure d’homophilie sur les nœuds Pei et al. qui calcule la moyenne des proportions des voisins qui ont le même label noté y et (2) la mesure d’homophilie sur les arêtes qui calcule la proportion d’arêtes (Zhu et al., 2020) qui connectent deux nœuds ayant le même label. Une limitation de ces mesures d’homophilie est qu’elles capturent peu les caractéristiques structurelles du voisinage du graphe. Pour dépasser cette limitation, nous proposons une mesure de l’homophilie qui prend en compte la courbure des arêtes du graphe :

$$\beta^+ = \frac{|\{(i, j) : (i, j) \in E^+ \wedge y_i = y_j\}|}{|E^+|} \quad (8)$$

où E^+ représente l’ensemble des arêtes du graphe (i, j) tel que $c_{ij} \geq \theta$. β^- est défini de la même manière en utilisant E^- , i.e. l’ensemble des arêtes (i, j) tel que $c_{ij} < \theta$ (où θ représente un seuil variant selon le type de courbure). Notons que si $E^+ = E$ on retrouve la définition de l’homophilie sur les arêtes β (Zhu et al., 2020). Une valeur élevée de β^+ signifie que la proportion d’arêtes qui connectent des nœuds au sein de la même communauté ont tendance à avoir le même label. Les valeurs de β^+ et β^- des jeux de données sont affichées à la table 1. Nous considérons l’homophilie basée sur la courbure d’Ollivier avec $\theta = 0$ pour un voisinage de 1 et de 2. Nous présentons également le Gain Max d’homophilie par rapport à la mesure originale (Zhu et al., 2020).

	Jeu de données	β	β^+	β^-	2-hop β^+	2-hop β^-	Max Homophilic Gain -
Heterophilic	Squirrel	0.23	0.28	0.29	0.25	0.29	25%
	Chameleon	0.26	0.29	0.28	0.32	0.23	24%
	Texas	0.31	0.44	0.43	0.59	0.47	92%
	Wisconsin	0.36	0.44	0.50	0.49	0.38	37%
	Cornell	0.34	0.48	0.46	0.37	0.40	40%
	R-empire/	0.29	0.40	0.48	0.05	0.07	65%
	Actor	0.32	0.73	0.32	0.32	0.18	131%

TAB. 1 – Comparaison des mesures d’homophilie sur les arêtes.

Pour les jeux de données homophiles les nœuds au sein de la même communauté (*arêtes à courbure positive*) ont tendance à avoir des labels plus similaires que les nœuds inter-communauté

(*Arêtes à courbure négative*). Pour les jeux de données hétérophiles, prendre en compte uniquement les arêtes à courbure positive n’améliore pas l’homophilie. Zhu et al. (2020) indique qu’élargir le voisinage à une distance de deux (i.e. voisins des voisins) améliore l’homophilie pour les jeux de données hétérophiles. Néanmoins, selon notre mesure, ce résultat est valide pour les petits jeux de données. De manière globale nous notons que l’homophilie augmente en prenant en compte uniquement les arêtes à courbure négative.

3.2 Curvature-Constrained Message Passing

En se basant sur nos précédentes mesures de l’homophilie sur la courbure, nous proposons de différencier la diffusion de l’information en fonction des arêtes à courbure négative ou positive. Nous considérons une diffusion à distance 1 ou 2).

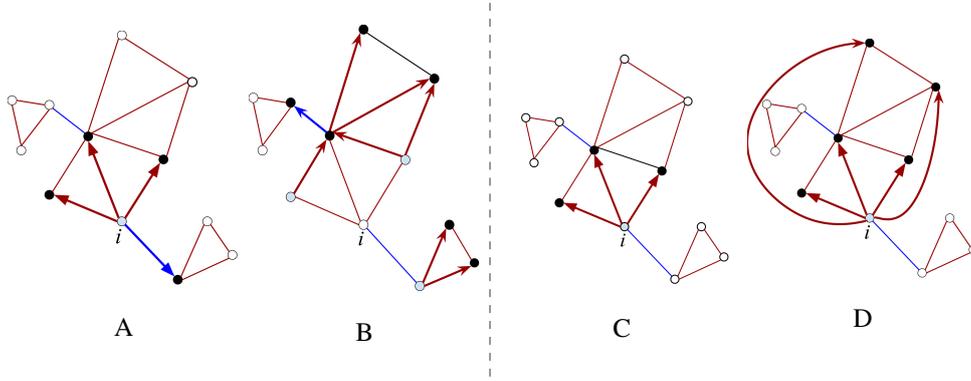


FIG. 2 – À gauche, un passage de messages classique pour la première couche (A) et la deuxième couche (B) à partir d’un nœud donné i . À droite, un exemple de transmission de messages à courbure positive à un voisinage de taille 1 (C) et à courbure positive un voisinage de taille 2 (D). Le message est propagé non seulement aux nœuds adjacents, mais aussi à ceux qui se trouvent à une distance de deux le long des arêtes à courbure positive.

Nous proposons de modifier la partie d’agrégation du MPPNs classique 1 :

$$\begin{aligned} Curv_{m_i}^+(\ell) &= \text{AGGREGATE}^{(\ell)} \left(\left\{ h_j^{(\ell)} : j \in \mathcal{N}^+(i) \right\} \right) \\ h_i^\ell &= \text{UPDATE}^{(\ell)} \left(h_i^{(\ell-1)}, Curv_{m_i}^+(\ell) \right) \end{aligned} \quad (9)$$

Où \mathcal{N}^+ représente le voisinage des nœuds qui sont connecté par des arêtes à courbure positive de i . De la même manière $Curv_{m_i}^-$ est définie en considérant \mathcal{N}^- au lieu de \mathcal{N}^+ .

La propagation de l’information par le biais de la courbure des arêtes offre une plus grande flexibilité dans le processus d’apprentissage via GNNs. En effet, nous pouvons soit échanger uniquement de l’information sur les arêtes à courbure négative ou positive soit en utiliser des matrices d’adjacence courbé différentes pour chaque couche.

Matrice d’adjacence courbée : intégration de la courbure dans la transmission des messages

Matrice d’adjacence courbée à un voisinage de taille 1 L’utilisation d’un voisinage de courbure à une distance de 1 nous permet de distribuer des informations non pas à toutes les arêtes du graphe, mais uniquement aux arêtes ayant une certaine courbure. Nous éliminons soit les arêtes à courbure négative, soit les arêtes positives, ce qui simplifie la connectivité du graphe. Dans ce cas, $W_1(\mu_i, \mu_j)$ de l’équation 5 diminue ; par conséquent, le nombre d’arêtes à courbure fortement négative (Topping et al., 2022), responsables des goulets d’étranglement, est réduit. En outre, la sparsification du graphe présente plusieurs avantages : (1) elle contribue à réduire le surlissage (Rong et al., 2019), (2) elle permet de réduire considérablement le diamètre du graphe, ce qui réduit la distance nécessaire pour relier deux nœuds du graphe et ainsi contribue également à diminuer l’over-squashing (Deac et al., 2022).

Matrice d’adjacence courbée à un voisinage de taille 2 Diffuser de l’information sur une distance de voisinage multiple permet d’atténuer les limites des MPNNs classiques, où les nœuds ne peuvent communiquer que par l’intermédiaire de leurs voisins directs. En effet, en densifiant le graphe avec des sauts multiples, nous pouvons désormais transmettre des informations directement aux nœuds éloignés (Brüel-Gabrielsson et al., 2022)(Abboud et al., 2022). Toutefois, en fonction de la taille du graphe, cela peut considérablement augmenter le coût de calcul de la méthode . En travaillant uniquement sur un voisinage de taille deux tout en restreignant la propagation en fonction d’une certaine courbure des arêtes, il est possible de limiter la densification du graphe et donc de réduire le coût de calcul tout en facilitant l’échange d’informations entre les nœuds distants.

Utilisation de la Matrice d’adjacence courbée à une distance de 1 et 2 entre les couches

Rappelons que si entre les nœuds i et j sont à une distance de k , leur interaction ne se produira qu’à la k^{eme} couches. Pour les GNNs à deux couches, l’utilisation d’une courbure à une distance de 1 puis de deux entre les couches permet une interaction plus rapide entre les nœuds distants. Cette procédure est une restriction du travail (DRew) de Gutteridge et al. (2023) en restreignant les arêtes à courbure positive ou négative. L’utilisation de ce cadre, comme le démontre Gutteridge et al. (2023), permet d’atténuer l’ over-squashing. La restriction de ce processus permet également de remédier à l’une des limites de l’article, à savoir qu’il ne peut être utilisé que pour des modèles GNN très profonds.

4 Expériences

Datasets Nous avons évalué notre approche sur 7 jeux de données hétérophiles utilisés dans la littérature, sur une tâche de classification de nœuds. Ces articles Pei et al. Platonov et al. (2023) décrivent plus en détails les différents jeux de données . Pour ces jeux de données le temps de calcul de la courbure d’Ollivier et AFC est respectivement entre 1 et 830 secondes et entre 1 et 85 secondes.

Baseline Nous comparons notre méthode à quatre méthodes basées sur des techniques de pré-traitement. Nous présentons les résultats de FA Alon et Yahav (2021), qui ne modifie que la matrice d’adjacence de la dernière couche, DIGL (Klicpera et al., 2019)², SDRF (Topping

2. <https://github.com/gasteigerjo/gdc>

et al., 2022)³ et FOSR(Kedar Karhadkar et Montúfar, 2023)⁴. Nous utilisons les hyperparamètres définis dans la publication originale et ajustons le nombre d'itérations. Pour DIGL, nous affinons le top k pour $\{8, 16, 32, 64, 128\}$ et $\{0.05, 0.1, 0.15\}$ pour le PageRank personnalisé.

Setup : Pour les expériences, nous utilisons la même stratégie d'apprentissage que Pei et al.. Ainsi nous fixons un nombre de couche de 2, dropout = 0.5, pas d'apprentissage de 0.005, patience de 100 "epochs", et un "weight decay" de $5E^{-6}$ (Texas/Wisconsin/Cornell) ou $5E^{-5}$ (autres datasets). Pour Amazon Photo et Computers nous utilisons la stratégie habituelle présentée dans Shchur et al. (2018). Pour tous les jeux de données nous prenons un échantillon aléatoire de 60% pour l'entraînement et 20% pour la validation et le test. Nous notons la moyenne d'accuracy pour chaque méthode sur 100 échantillons aléatoires.

Nous utilisons comme base les deux GNNs les plus populaires, le GCN et le GAT, et nous comparons les différentes méthodes après le pré-traitement de la matrice d'adjacence.

Curvature-Constrained Message Passing (CCMP) configuration : En fonction de la méthode de la courbure nous utilisons différentes configurations pour les jeux de données. Comme la mesure de la courbure d'Olliver est bornée nous considérons une matrice d'adjacence courbée négativement/positivement respectivement de telle manière à ce que $e_{ij} \leq 0$ et $e_{ij} \geq 0$. Pour la mesure "Augmented Forman curvature (AFC)" nous considérons une matrice d'adjacence courbée négativement/positivement respectivement de telle manière à ce que la courbure soit moins/plus importante que la moyenne de la courbure des arêtes sur le graphe. Nous avons testé les différentes configurations de CCMP en fonction du graphe et de l'homophilie basée sur la courbure. Les configurations optimales sont les suivantes :

Pour l'utilisation de la **courbure d'Olliver** (1) pour Cora et Citeseer, pour la première couche, une matrice d'adjacence à courbure négative sur un voisinage de taille 1 et pour la deuxième couche, une matrice d'adjacence à courbure positive à une distance de taille 1, (2) pour les petits jeux de données comme Texas, Wisconsin et Cornell, une matrice d'adjacence à courbure négative sur un voisinage de taille 2 pour les deux couches, (3) pour les jeux de données Squirrel et Roman-Empire, une matrice d'adjacence à courbure négative sur un voisinage de taille 1 sur les deux couches, (4) pour les jeux de données Amazon Computers et Actor, une matrice d'adjacence positive à sur un voisinage de taille 1 sur deux couches, (5) pour Chameleon et Amazon photo, pour la première couche, une matrice d'adjacence positive sur un voisinage de taille 1 et pour la deuxième couche, une matrice d'adjacence positive à un voisinage de taille 2.

Pour l'utilisation de l'**Augmented Forman Curvature** nous utilisons pour (1) Chameleon, Squirrel, Citeseer, Amazon Computers une matrice d'adjacence positive à un voisinage de taille 1 sur deux couches, (2) pour Actor et Romain empire une matrice d'adjacence négative à sur un voisinage de taille 1 sur deux couches, (3) Pour les autres jeux de données la configuration est la même que pour la **courbure d'Olliver**. Nous désignons l'utilisation de la courbure d'Olliver avec $CCMP_O$ et la courbure AFC avec $CCMP_A$.

4.1 Résultats

Les tableaux 2 et 3 montrent que pour 6 des 7 jeux de données hétérophiles, notre méthode obtient les meilleurs résultats avec une différence significative. En moyenne, pour tous ces jeux

3. <https://github.com/jctops/understanding-oversquashing/tree/main>

4. <https://github.com/kedar2/FoSR/tree/main>

Matrice d’adjacence courbée : intégration de la courbure dans la transmission des messages

	Base (GCN)	DIGL	FA	SRDF	FOSR	CCMP _O	CCMP _A
Cham.	65.35±0.54	54.82±0.48	26.34±0.61	63.08±0.37	67.98 ±0.40	63.22±0.45	65.66±0.44
Squir.	51.30±0.38	40.53±0.29	22.88±0.42	49.11±0.28	<u>52.63</u> ±0.30	<u>53.36</u> ±0.22	54.79 ±0.31
Actor	30.02±0.22	26.75±0.23	26.03±0.30	31.85±0.22	29.26±0.23	33.57±0.22	34.59 ±0.24
Texas	56.19±1.61	45.95±1.58	55.93±1.76	59.79±1.71	61.35±1.25	<u>64.67</u> ±1.81	69.67 ±1.64
Wisc.	55.12±1.51	46.90±1.28	46.77±1.48	58.49±1.23	55.60±1.25	<u>66.40</u> ±1.24	67.80 ±1.49
Corn.	44.78±1.45	44.46±1.37	45.33±1.55	47.73±1.51	45.11±1.47	58.91 ±1.82	58.54±1.57
R-emp.	51.66±0.17	53.93±0.14	OOM	52.53±0.13	52.38±0.21	58.58±0.14	58.91 ±0.19

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

	Base (GAT)	DIGL	FA	SRDF	FOSR	CCMP _O	CCMP _A
Cham.	65.07±0.41	56.34±0.43	27.11±0.56	63.15±0.44	66.61 ±0.45	63.09±0.52	65.59±0.43
Squi.	50.87±0.56	41.65±0.68	21.49±0.71	50.36±0.38	<u>52.02</u> ±0.43	51.82±0.32	54.74 ±0.52
Actor	29.92±0.23	31.22±0.47	28.20±0.51	31.47±0.25	29.73±0.24	<u>33.23</u> ±0.22	34.23 ±0.23
Texas	56.84±1.61	46.49±1.63	56.17±1.71	57.45±1.62	61.85±1.41	71.78 ±1.39	70.65±1.36
Wisc.	53.58±1.39	46.29±1.47	46.95±1.52	56.80±1.29	54.06±1.27	<u>65.77</u> ±1.32	68.59 ±1.41
Cornell	46.05±1.49	44.05±1.44	44.60±1.74	48.03±1.66	48.30±1.61	60.43 ±1.47	59.81±1.49
R-Emp.	49.23±0.33	53.89±0.16	OOM	50.75±0.17	49.54±0.31	57.36 ±0.19	<u>56.78</u> ±0.39

TAB. 3 – Résultats (accuracy) sur les jeux de données **hétérophiles** avec **GAT** comme base. Meilleur score en gras et second souligné.

de données, CCMP_O et CCMP_A surpassent les résultats obtenus avec la matrice d’adjacence originale pour une base de GCN et GAT respectivement de 14,24 % et de 16,55 %. Notons que SRDF, FOSR et CCMP surpassent DIGL sur les ensembles de données hétérophiles car DIGL tend à améliorer la connectivité entre les nœuds avec une diffusion courte et ici les nœuds voisins ont tendance à ne pas avoir les mêmes labels.

Nous présentons ici certains résultats basés sur CCMP_O. Conformément à (Topping et al., 2022), nous nous intéressons aux arêtes fortement négativement courbées sur les graphes originaux. Utiliser une diffusion à une distance de taille 2 supprime complètement les arêtes à courbure fortement négative en récupérant les premiers déciles de la courbure (triés par ordre croissant). Pour la matrice d’adjacence originale du Texas, du Wisconsin et de Cornell, les premiers déciles de la courbure passent de -0.33 , -0.33 , -0.25 à $+0.37$, $+0.33$, $+0.06$ pour les deux matrices d’adjacence à courbure négative.

L’utilisation d’une matrice d’adjacence courbée à distance 1 permet de réduire la taille du graphe. Par conséquent, sur Squirrel, Actor et Roman-Empire, le coût de calcul est réduit de 10 à 40%. En outre, nous réduisons considérablement le diamètre, ce qui contribue à atténuer l’écrasement excessif. Ce résultat est confirmé empiriquement par une augmentation de l’écart spectral normalisé, allant de 5 à 87% sur ces jeux de données.

5 Conclusion

Dans cet article, nous présentons une méthode qui peut être appliquée à n’importe quelle architecture MPNN. Notre méthode diffuse des messages en fonction de la courbure des arêtes du graphe, ce qui permet d’atténuer l’over-squashing, l’un des principaux inconvénients des architectures MPNN classiques. En tenant compte d’une nouvelle mesure d’homophilie contrainte par la courbure, nous avons développé différentes variantes de méthode de propagation de l’in-

formation à travers les arêtes. Les expériences démontrent une amélioration significative par rapport aux méthodes de pré-traitement, prouvant empiriquement l'utilité d'une contrainte de transmission de messages basée sur la courbure pour la réduction de l'oversquashing.

Références

- Abboud, R., R. Dimitrov, et I. I. Ceylan (2022). Shortest path networks for graph property prediction. In *Learning on Graphs Conference*, pp. 5–1. PMLR.
- Alon, N. et V. D. Milman (1984). Eigenvalues, expanders and superconcentrators. In *25th Annual Symposium on Foundations of Computer Science, 1984.*, pp. 320–322. IEEE.
- Alon, U. et E. Yahav (2021). On the bottleneck of graph neural networks and its practical implications. In *ICLR*.
- Attali, H., A. Guille, et S. Chrétien (2023). Amélioration de l'architecture gat par la prise en compte de la courbure du graphe. In *Extraction et Gestion des Connaissances : Actes de la conférence EGC'2023*, Volume 39. BoD-Books on Demand.
- Brüel-Gabrielsson, R., M. Yurochkin, et J. Solomon (2022). Rewiring with positional encodings for graph neural networks. *arXiv preprint arXiv :2201.12674*.
- Cai, C. et Y. Wang (2020). A note on over-smoothing for graph neural networks. *Graph Representation Learning*.
- Cheeger, J. (1970). A lower bound for the smallest eigenvalue of the laplacian. In *Problems in Analysis : A Symposium in Honor of Salomon Bochner*. Princeton University Press.
- Chung, F. R. et F. C. Graham (1997). *Spectral graph theory*. American Mathematical Soc.
- Deac, A., M. Lackenby, et P. Veličković (2022). Expander graph propagation. In *Learning on Graphs Conference*, pp. 38–1. PMLR.
- 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 *ICML*, pp. 1263–1272. PMLR.
- Guille, A. et H. Attali (2022). Document classification with hierarchical graph neural networks. In *18th International Workshop on Mining and Learning with Graphs*.
- Gutteridge, B., X. Dong, M. M. Bronstein, et F. Di Giovanni (2023). Drew : Dynamically rewired message passing with delay. In *ICML*, pp. 12252–12267. PMLR.
- Kedar Karhadkar, P. K. B. et G. Montúfar (2023). Fosr : First-order spectral rewiring for addressing oversquashing in gnns. *ICLR*.
- Kipf, T. N. et M. Welling (2017). Semi-Supervised Classification with Graph Convolutional Networks. *ICLR*.
- Klicpera, J., S. Weissenberger, et S. Günnemann (2019). Diffusion improves graph learning. *NeurIPS*.
- Ni, C.-C., Y.-Y. Lin, J. Gao, et X. Gu (2018). Network alignment by discrete ollivier-ricci flow. In *International Symposium on Graph Drawing and Network Visualization*.
- 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 Mathématique* 345(11).
- Oono, K. et T. Suzuki (2020). Graph neural networks exponentially lose expressive power for node classification.
- Pei, H., B. Wei, K. C.-C. Chang, Y. Lei, et B. Yang. Geom-gcn : Geometric graph convolutional networks. ICLR.
- Platonov, O., D. Kuznedelev, A. Babenko, et L. Prokhorenkova (2022). Characterizing graph datasets for node classification : Beyond homophily-heterophily dichotomy. *arXiv preprint*.
- Platonov, O., D. Kuznedelev, M. Diskin, A. Babenko, et L. Prokhorenkova (2023). A critical look at the evaluation of gnns under heterophily : are we really making progress? In *ICLR*.
- Rong, Y., W. Huang, T. Xu, et J. Huang (2019). Dropedge : Towards deep graph convolutional networks on node classification. In *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.
- Shchur, O., M. Mumme, A. Bojchevski, et S. Günnemann (2018). Pitfalls of graph neural network evaluation. *arXiv preprint arXiv :1811.05868*.
- Sia, J., E. Jonckheere, et P. Bogdan (2019). Ollivier-ricci curvature-based method to community detection in complex networks. *Scientific reports* 9(1), 9800.
- Topping, J., F. Di Giovanni, B. P. Chamberlain, X. Dong, et M. M. Bronstein (2022). Understanding over-squashing and bottlenecks on graphs via curvature. ICLR.
- Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Liò, et Y. Bengio (2018). Graph Attention Networks. ICLR.
- Ye, Z., K. S. Liu, T. Ma, J. Gao, et C. Chen (2019). Curvature graph network. In *ICLR*.
- Zhou, J., G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, et M. Sun (2020). Graph neural networks : A review of methods and applications. *AI open* 1, 57–81.
- 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. *NEURIPS* 33, 7793–7804.

Summary

In this paper, we propose a framework appropriate for any MPNN architecture that distributes information based on the curvature of the graph’s edges. Experiments conducted on different datasets show that our method mitigates over-squashing and outperforms existing graph rewiring methods in several node classification datasets.