

Topological Graph Neural Networks [1]

Afandi Mohammad, Ait Ouazzou Zakari, Reboud Jehanne

Introduction

La **classification de graphes** est utilisée dans divers domaines comme la bio-informatique pour la classification de molécules, dans le traitement d'images ou la classification de graphes de réseaux sociaux.

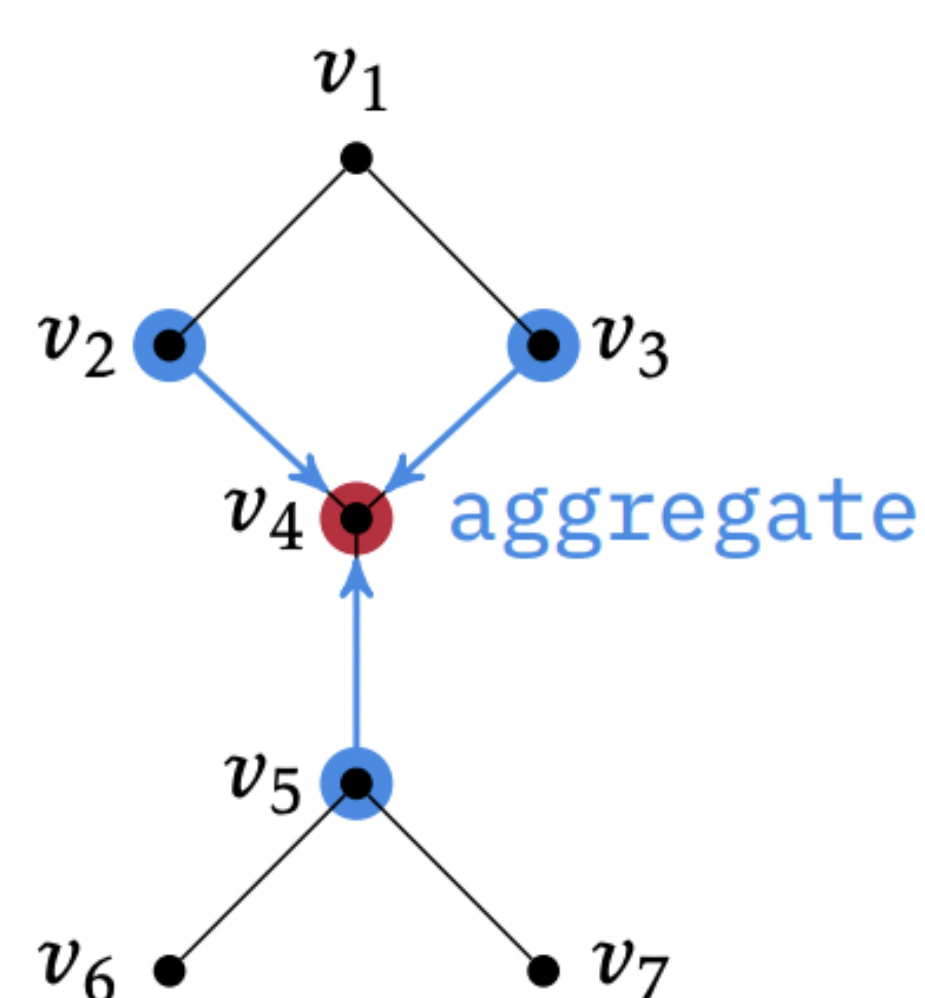
Les **GNNs (Graph Neural Network)** sont particulièrement adaptés pour traiter les données sous forme de graphes mais ne parviennent pas à reconnaître certaines structures topologiques.

Pour améliorer l'expressivité de ces modèles, l'article propose d'intégrer à différents GNNs **une couche de graphe topologique (TOGL)**, fondée sur les méthodes classiques d'analyse de données topologiques (TDA).

Cette méthode est aussi applicable dans la **classification de noeuds** si on s'intéresse au rôle de chaque noeud dans le graphe.

GNN

Un GNN travaille avec des inputs de taille fixe et nécessite une représentation vectorisée d'un graphe G . $f : G \rightarrow \mathbb{R}^d$. f invariante aux permutations.



$$a_v^{(k)} := \text{aggregate}^{(k)}\left(\left\{h_u^{(k-1)} \mid u \in \mathcal{N}_G(v)\right\}\right)$$

$$h_v^{(k)} := \text{combine}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$

$$h_G := \text{readout}\left(\left\{h_v^{(K)} \mid v \in \mathcal{V}_G\right\}\right)$$

- Pour chaque $v_i \in \mathbb{R}^d$ on combine l'information des voisins et on réitère.
- Un GNN est incapable de reconnaître certaines structures topologiques.

TOGL: Topological Graph Layer

- Persistent Homology

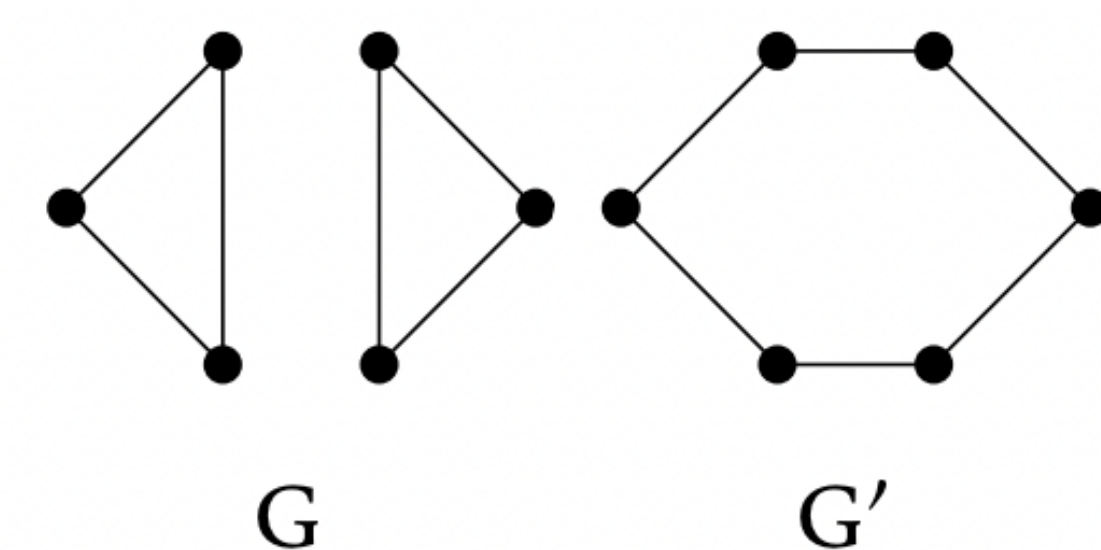


Figure: Exemple de graphes non isomorphes que les GNNs ne distinguent pas. $\beta_0(G)=\beta_1(G)=2$ et $\beta_0(G')=\beta_1(G')=1$

- Persistent Homology (PH) est une méthode algébrique qui analyse la topologie d'un graphe G à plusieurs échelles.
- PH se base sur les caractéristiques topologiques β_i de chaque sous graphe G_i de G . Des diagrammes de persistance sont construits à partir de : β_0 (nombre de composants connectés) et β_1 (nombre de cycles) invariants aux isomorphismes.
- PH permet de distinguer des graphes que les GNNs ne distinguent pas.

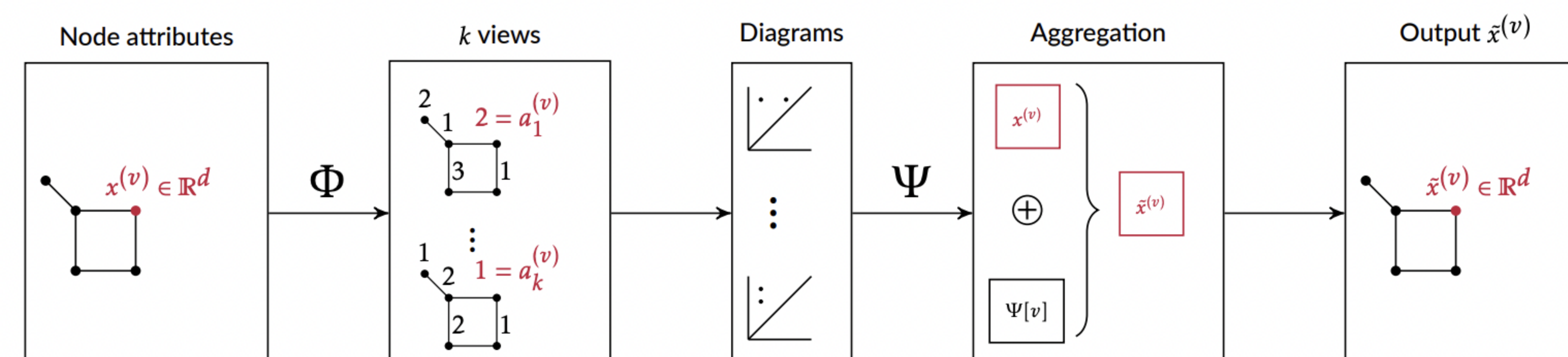


Figure: Architecture de TOGL

Datasets et Initialisation

On évalue le score de prédiction sur :

- **2 datasets synthétiques** (Cycles et Necklaces) : contiennent des graphes topologiques que le GNN ne parvient pas à distinguer facilement ;
- **datasets de référence** (DD, ENZYMES, MNIST, PROTEINS, MOLHIV, PATTERN).

Les features sur les noeuds des graphes contiennent des informations issues de la structure du graphe, ce qui peut biaiser les résultats. On évalue avec 2 initialisations :

- Tous les features et labels sur les noeuds sont remplacés aléatoirement en ne gardant que les informations sur la topologie pour la classif.
- Pratique standard : évaluation en gardant l'ensemble des features.

Résultats

	Graph classification			Node classif
.	MNIST	PROTEINS	IMDB-B	PATTERN
(1)	90.0±0.3	76.1±2.4	68.6±4.9	85.5±0.4
(2)	95.5±0.2	76.0±3.9	72.0±2.3	86.6±0.1
(3)	96.1±0.3	72.3±3.3	72.8±2.5	84.8±0.0
(4)	96.1±0.1	73.6±4.8	74.2±4.2	86.7±0.1
(5)	94.8	68.6	70.0	85.7

Table: Récapitulatif des résultats

- | | |
|--------------------|----------------------------|
| (1) : GCN-4 | (4) : GIN-3-TOGL-1 |
| (2) : GCN-3-TOGL-1 | (5) : Notre implémentation |
| (3) : GIN-4 | |

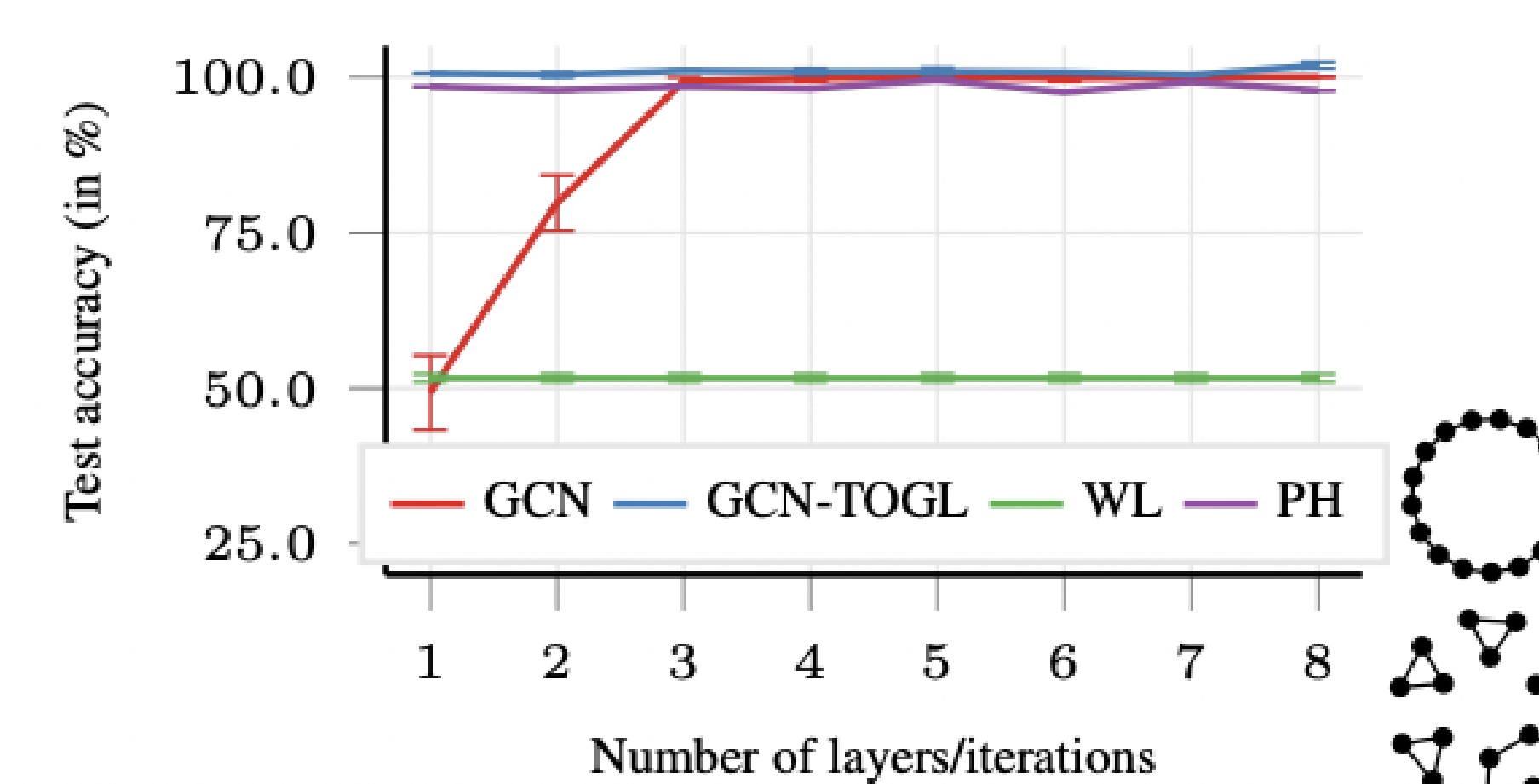
Conclusion

Les résultats que nous avons obtenus concordent avec ceux exposés dans le papier. TOGL améliore généralement l'expressivité d'un GNN et nécessite parfois de peu de couches pour avoir une bonne performance.

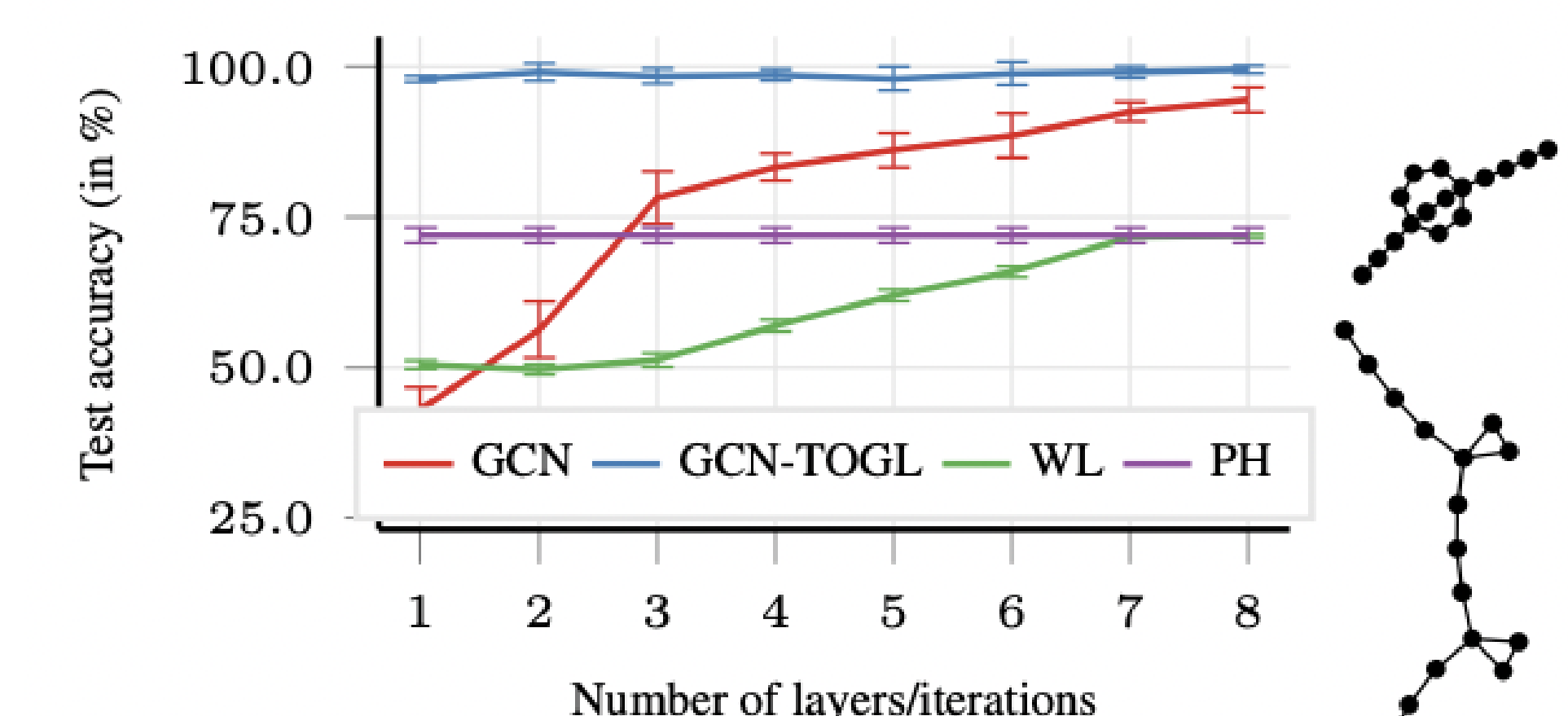
Références

- [1] Max Horne, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks.

Comparaison expressivité des méthodes



(a) CYCLES



(b) NECKLACES

Augmenter le nombre de couches dans un GNN augmente le champs receptif du réseau et donc sa performance. L'ajout de la couche TOGL, améliore grandement les performances du GNN pour les 2 datasets.