



# Utilisation d'Ensemble Indépendant Maximal (MIS) dans le Pooling sur Graphes – Contributions et futurs projets

Stevan Stanovic



## Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES
- 4 MIES
- 5 Résultats préliminaires
- 6 Line Graph
- 7 Architecture Adaptative
- 8 Futures Perspectives

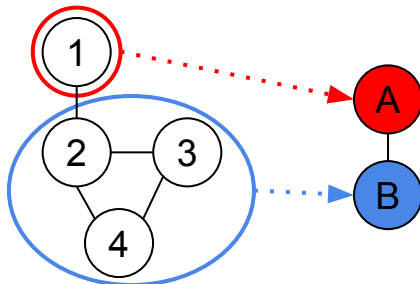


# Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES
- 4 MIES
- 5 Résultats préliminaires
- 6 Line Graph
- 7 Architecture Adaptative
- 8 Futures Perspectives



# Graph Pooling



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$\mathbf{A}^{(l)}$

$$\begin{aligned} \mathbf{X}^{(l+1)} &= \mathbf{S}^T \mathbf{X}^{(l)} \\ \mathbf{A}^{(l+1)} &= \mathbf{S}^T \mathbf{A}^{(l)} \mathbf{S} \end{aligned}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 6 \end{pmatrix}$$

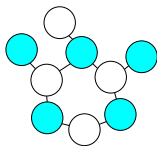
$\mathbf{A}^{(l+1)}$

$$\begin{matrix} \mathbf{A} & \mathbf{B} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

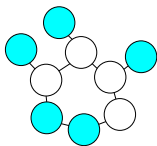
**Matrix S**



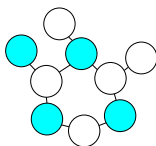
## Maximal Independent Vertex Set (MIVS) [1]



**Maximal  
Independent Vertex  
Set**



**First property not  
respected**



**Second property not  
respected**

### Properties

1) Two surviving vertices can't be neighbors:

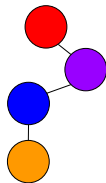
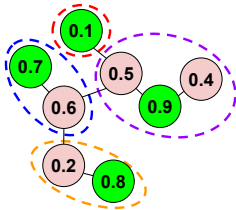
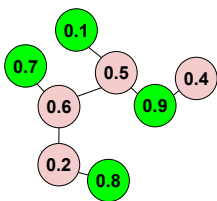
$$\forall (v, v') \in (\mathcal{V}^{(l+1)})^2 : (v, v') \notin \mathcal{E}^{(l)}$$

2) A non-surviving vertex has at least one surviving neighbor:

$$\forall v \in \mathcal{V}^{(l)} - \mathcal{V}^{(l+1)}, \exists v' \in \mathcal{V}^{(l+1)} : (v, v') \in \mathcal{E}^{(l)}$$



## MIVS : affectation



Ratio de décimation :  $< 50\%$



# Sommaire

- 1 Introduction
- 2 EdgePool**
- 3 MIDES
- 4 MIES
- 5 Résultats préliminaires
- 6 Line Graph
- 7 Architecture Adaptative
- 8 Futures Perspectives



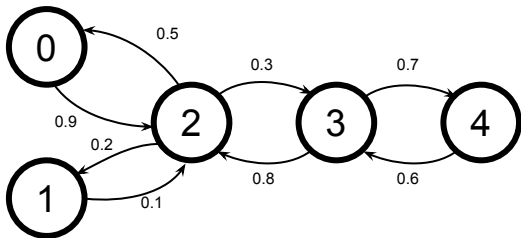
## EdgePool : Généralités

- Méthode sélectionnant les arêtes comme un MIVS  $\Rightarrow$  MIES
- ICML 2019 [2]
- Voisinage d'une arête : toutes arêtes entrantes ou sortantes des deux sommets associés



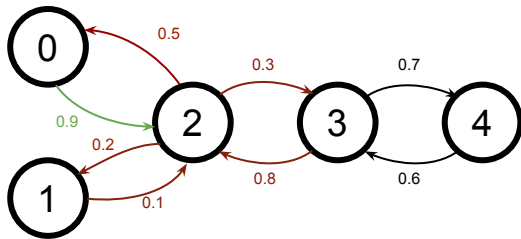


## EdgePool : Schéma



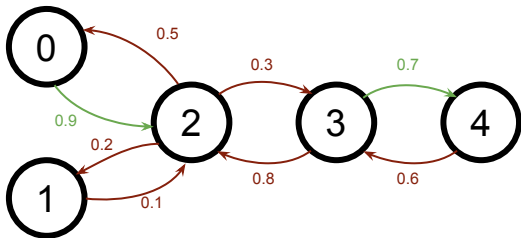


## EdgePool : Schéma



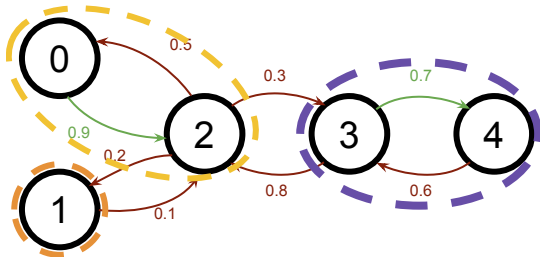


## EdgePool : Schéma





## EdgePool : Schéma





## Bilan de la méthode EdgePool

- Graphe doit être orienté
- Sélection itérative des arêtes et non parallélisé
- Méthode parallélisable présenté à ESANN 2022 par une autre équipe [3]
- Ratio de décimation  $> 50\%$



# Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES**
- 4 MIES
- 5 Résultats préliminaires
- 6 Line Graph
- 7 Architecture Adaptative
- 8 Futures Perspectives

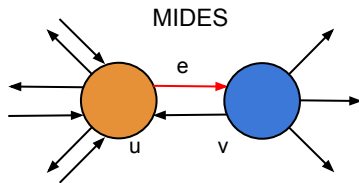
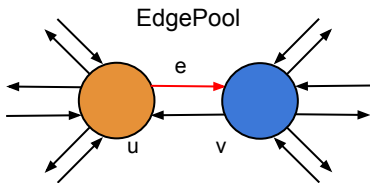


## MIDES : Généralités

- Maximal Independent Directed Edge Set (MIDES)
- Méthode parallélisable
- Si une arête  $u \rightarrow v$ , les arêtes voisines entrantes dans  $v$  sont toujours candidates (Autorise la création d'étoile)
- Ratio de décimation  $\approx 50\%$



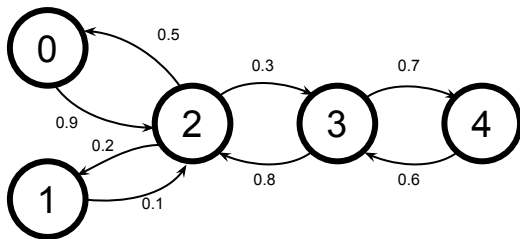
## MIDES : Modification voisinage





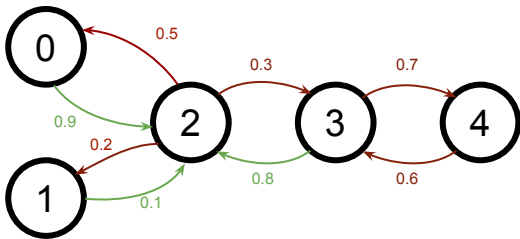


## MIDES : Schéma



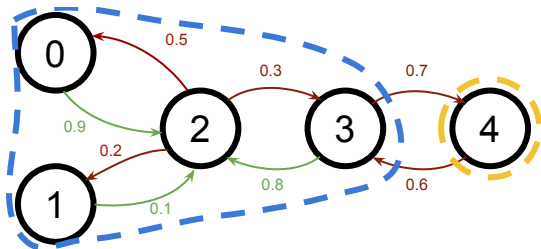


## MIDES : Schéma





## MIDES : Schéma





# Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES
- 4 MIES**
- 5 Résultats préliminaires
- 6 Line Graph
- 7 Architecture Adaptative
- 8 Futures Perspectives

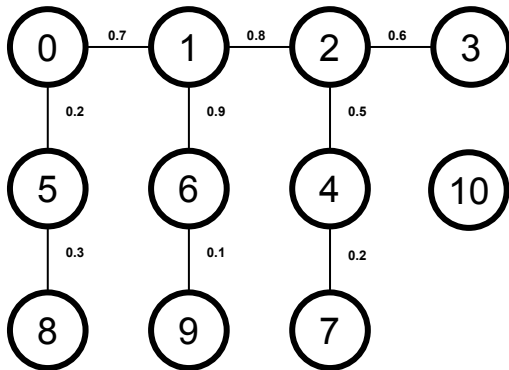


## MIES : Généralités

- Maximal Independent Edge Set (MIES)
- Méthode parallélisable
- Graphe non orienté
- Voisinage des arêtes : même que celui de EdgePool
- Proposition 1 : faire la même chose que EdgePool mais avec un graphe non orienté
- Proposition 2 : après avoir fait l'étape EdgePool, raccorder les sommets isolés connectés et couper les regroupements dont le chemin est supérieur à 2
- Ratio de décimation pour la proposition 1 :  $> 50\%$
- Ratio de décimation pour la proposition 2 :  $\approx 50\%$

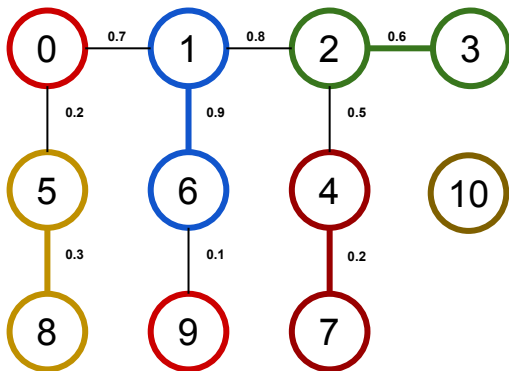


## MIES : Schéma



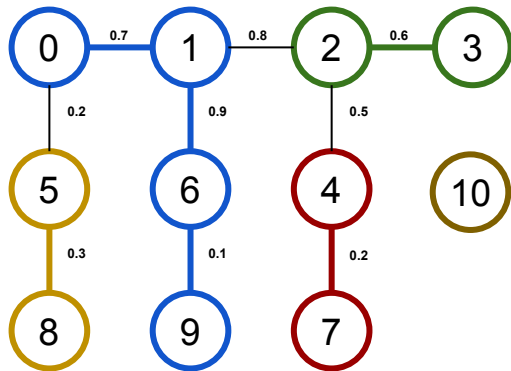


## MIES : Schéma





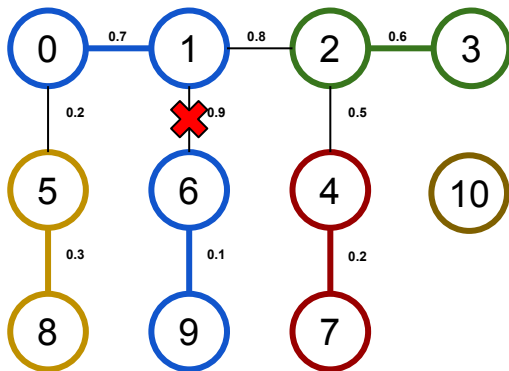
## MIES : Schéma







## MIES : Schéma





# Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES
- 4 MIES
- 5 Résultats préliminaires**
- 6 Line Graph
- 7 Architecture Adaptative
- 8 Futures Perspectives



Dataset	PROTEINS [4]	NCI1 [5]	NCI109 [5]
Baseline	<b>76.43 ± 4.05</b>	<b>72.51 ± 3.19</b>	71.65 ± 3.35
Top-k [6]	73.16 ± 4.13	66.09 ± 4.67	63.94 ± 4.02
SagPool [7]	72.32 ± 2.74	66.76 ± 5.18	66.15 ± 4.58
EdgePool [2]	73.55 ± 4.61		
MIVSPool [1]	74.93 ± 3.90	71.04 ± 3.84	69.32 ± 3.30
MIESPool <sub>rand</sub>	74.45 ± 3.67	69.38 ± 4.31	70.00 ± 3.64
MIESPool	75.65 ± 2.84	71.85 ± 4.43	<b>71.66 ± 4.29</b>
MIESPoolCut <sub>rand</sub>	74.57 ± 4.08	71.64 ± 4.04	69.74 ± 5.25
MIESPoolCut	70.98 ± 1.92	71.70 ± 3.57	70.86 ± 3.84
MIDESPool <sub>rand</sub>	74.50 ± 4.03	68.52 ± 3.48	67.79 ± 3.99
MIDESPool		69.98 ± 3.32	69.42 ± 3.61



# Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES
- 4 MIES
- 5 Résultats préliminaires
- 6 Line Graph**
- 7 Architecture Adaptative
- 8 Futures Perspectives



Étant donné un graphe  $\mathcal{G}$ , son line graph  $L(\mathcal{G})$  est le graphe défini de la façon suivante :

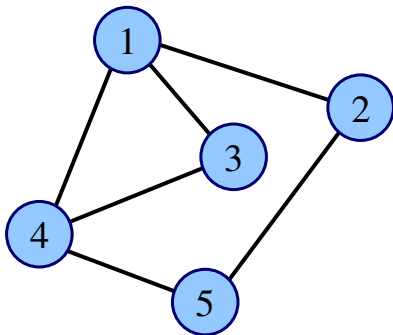
- chaque sommet de  $L(\mathcal{G})$  représente une arête de  $\mathcal{G}$
- deux sommets de  $L(\mathcal{G})$  sont adjacents si et seulement si les arêtes correspondantes partagent une extrémité commune dans  $\mathcal{G}$

Intérêt général :

- utiliser le MIVSPool sur le Line Graph pour obtenir un MIES ou un MIDES
- tester une forme de MIVSPool différentiable

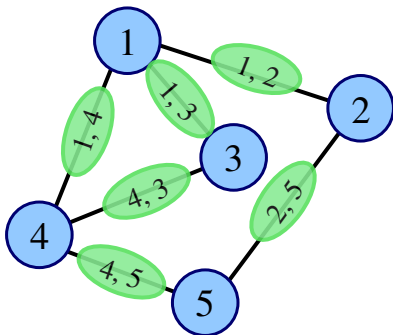


## Line Graph : Schéma



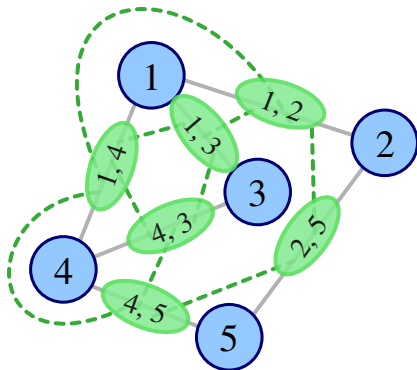


## Line Graph : Schéma





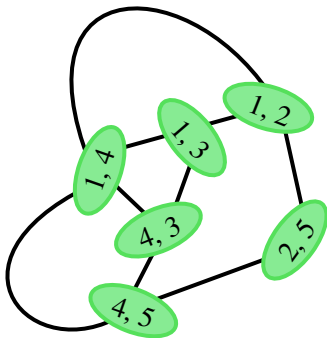
## Line Graph : Schéma







## Line Graph : Schéma





# Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES
- 4 MIES
- 5 Résultats préliminaires
- 6 Line Graph
- 7 Architecture Adaptative**
- 8 Futures Perspectives



### Problématiques :

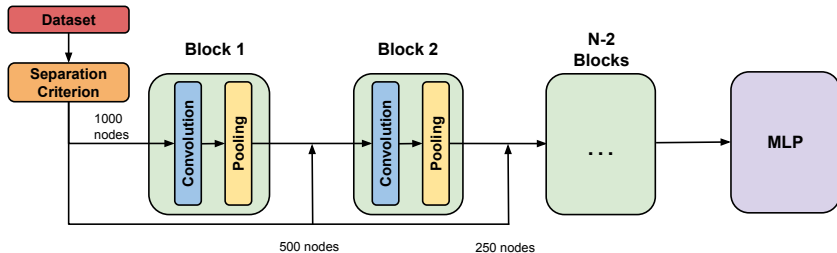
- En CNN, la taille des images dans le réseau est toujours la même  $\Rightarrow$  représentations réduites envoyés au MLP
- En GNN, les graphes ne sont pas de la même taille  $\Rightarrow$  READOUT envoyés au MLP

### Solutions :

- Pré-traiter les graphes pour qu'il est la même taille (DiffPool[8])
- Faire un nombre adaptatif de bloc de Convolution/Pooling selon la taille des graphes pour obtenir des représentations réduites de même taille que l'on envoie dans un MLP



# Architecture Adaptative : Schéma





- Méthodes denses (DiffPool [8]) : catégorie de taille à définir
- Méthodes Top-k [6, 7] : la taille d'un graphe et un ratio choisi
- Autres méthodes (MIVSPool [1], EdgePool [2]) : critère pas naïf (à trouver)



# Sommaire

- 1 Introduction
- 2 EdgePool
- 3 MIDES
- 4 MIES
- 5 Résultats préliminaires
- 6 Line Graph
- 7 Architecture Adaptative
- 8 Futures Perspectives**



## Futures Perspectives

- Utilisation la théorie spectrales [9, 10] pour préserver les propriétés spectrales d'un graphe
- Étude de méthode similaires à ce qui se fait en image pour mettre en relief les parties des graphes importantes pour une prédiction.



## References I



Stevan Stanovic, Benoit Gaüzère, and Luc Brun.

Maximal independent vertex set applied to graph pooling.

In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, S+ SSPR 2022, Montreal, QC, Canada, August 26–27, 2022, Proceedings*, pages 11–21. Springer, 2023.



Frederik Diehl, Thomas Brunner, Michael Truong Le, and Alois Knoll.

Towards graph pooling by edge contraction.

In *ICML 2019 workshop on learning and reasoning with graph-structured data*, 2019.



Francesco Landolf.

Revisiting edge pooling in graph neural networks.

In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning 2022 (ESANN 2022)*, 2022.





## References II

Paul D Dobson and Andrew J Doig.

Distinguishing enzyme structures from non-enzymes without alignments.

*Journal of molecular biology*, 330(4):771–783, 2003.

 Nikil Wale, Ian A Watson, and George Karypis.

Comparison of descriptor spaces for chemical compound retrieval and classification.

*Knowledge and Information Systems*, 14(3):347–375, 2008.

 Hongyang Gao and Shuiwang Ji.

Graph u-nets.

In *International conference on machine learning*, pages 2083–2092. PMLR, 2019.

 Junhyun Lee, Inyeop Lee, and Jaewoo Kang.

Self-attention graph pooling.

In *International conference on machine learning*, pages 3734–3743. PMLR, 2019.



## References III



Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec.

Hierarchical graph representation learning with differentiable pooling.  
*Advances in neural information processing systems*, 31, 2018.



Andreas Loukas.

Graph reduction with spectral and cut guarantees.  
*J. Mach. Learn. Res.*, 20(116):1–42, 2019.



Jie Chen, Yousef Saad, and Zechen Zhang.

Graph coarsening: from scientific computing to machine learning.  
*SeMA Journal*, pages 1–37, 2022.



Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel.

Protein function prediction via graph kernels.  
*Bioinformatics*, 21(suppl\_1):47–56, 2005.