

[ÉCOLE DOCTORALE]

[Unité de recherche ou Laboratoire]

THÈSE présentée par / DISSERTATION presented by :

Assaad ZEGHINA

soutenue le /defended on : [06 Décembre 2024]

pour obtenir le grade de / to obtain the grade of :

Docteur de l'Université de Strasbourg / Strasbourg University Doctor

Discipline/Spécialité / Discipline/Specialty : Informatique

**Recherche de motifs spatio-temporels
fréquents dans un grand graphe via des
méthodes d'apprentissage profond**

THÈSE dirigée par / DISSERTATION supervisor :

Mme Florence LE BER
M. Antoine VACAVANT

Directeur de recherche, Université de Strasbourg
Professeur, Université Clermont Auvergne

RAPPORTEURS :

M. Christophe CLARAMUNT
M. Dino IENCO

Professeur, Ecole Navale
Directeur de recherches, INRAE

AUTRES MEMBRES DU JURY / OTHER MEMBERS OF THE JURY :

Mme Christine SOLNON
Mme Aurélie Leborgne

Professeur, INSA Lyon
Maitresse de Conférence, Université de Strasbourg

Thèse de Doctorat



Université de Strasbourg

ED 269 Mathématiques, Sciences de l'Information et de l'Ingénieur (MSII)
UMR 7357 ICube

Thèse pour obtenir le grade de
Docteur de l'Université de Strasbourg
Informatique

Présentée et soutenue par

Assaad ZEGHINA

Le 06 Décembre 2024

**RECHERCHE DE MOTIFS SPATIO-TEMPORELS FREQUENTS DANS UN GRAND GRAPHE VIA
DES METHODES D'APPRENTISSAGE PROFOND**

Thèse dirigée par Florence LE BER et Antoine VACAVANT

JURY :

Rapporteurs

M. Christophe CLARAMUNT, Professeur – Ecole Navale

M. Dino IENCO, Directeur de recherches – INRAE

Examineurs

Mme Christine SOLNON, Professeur – INSA Lyon

Mme Aurélie LEBORGNE, Maîtresse de Conférences – Université de Strasbourg

If I have seen further, it is by standing on the shoulders of Giants.

Sir Isaac Newton

Remerciements

Je tiens à exprimer ma sincère gratitude à tous ceux qui m'ont soutenu tout au long de mon parcours de recherche.

Tout d'abord, je souhaite remercier mes directeurs de thèse et mon encadrante, Aurélie Leborgne, Florence Leber et Antoine Vacavant, pour leurs conseils inestimables, leurs encouragements et leurs idées. Vos expertises et votre engagement envers l'excellence ont profondément influencé mon travail et mon développement personnel.

Je suis également reconnaissant envers mes membres de jury, M. Christophe Claramunt, M. Dino Ienco et Mme Christine Solnon, pour avoir accepté d'être membres de mon jury, ainsi que pour leurs retours constructifs et leur soutien, qui ont contribué à façonner cette thèse.

Un remerciement particulier à mes collègues et amis d'Icube, dont la camaraderie a été précieuse tout au long de ce projet.

Je souhaite également remercier ma famille, dont la foi inébranlable en mes capacités a été une source de force et de motivation. Merci pour votre soutien émotionnel et financier. Votre patience et votre compréhension pendant les moments difficiles de ma recherche ont été essentielles.

Enfin, j'exprime ma gratitude à tous les participants et contributeurs qui ont fait partie de cette recherche. Vos idées et votre volonté de vous engager dans ce travail ont été inestimables.

Merci à tous d'avoir été partie prenante de ce parcours.

Résumé

Les graphes sont des outils puissants pour représenter des relations complexes dans des domaines variés comme les réseaux sociaux, la biologie, et les systèmes de transport. L'extraction de motifs fréquents consiste à identifier des sous-graphes récurrents au sein d'un ou de plusieurs graphes. Ces motifs, révélateurs de structures communes, apportent des informations précieuses sur les relations, propriétés et interactions sous-jacentes aux données. Cependant, la complexité algorithmique des méthodes traditionnelles d'extraction de motifs pose un défi majeur. L'énumération exhaustive de tous les motifs possibles a une croissance exponentielle, faisant du dénombrement des occurrences d'un motif donné un problème NP-complet. Ce défi devient encore plus important lorsqu'il s'agit de multigraphes, qui permettent de représenter des relations variées et potentiellement superposées à l'aide de différents types d'arcs et de directions.

Les graphes spatio-temporels, une catégorie spécifique de multigraphes, constituent un cadre dynamique et interconnecté où les nœuds représentent des entités, les arêtes codent les relations, et les attributs spatiaux et temporels apportent une information contextuelle riche. Cette thèse s'attaque au défi de l'extraction de motifs fréquents à partir de grands graphes spatio-temporels, une tâche cruciale à l'ère du big data. Nous proposons pour cela des approches innovantes basées sur l'apprentissage profond et l'apprentissage par renforcement.

Nos contributions principales sont les suivantes : premièrement, nous introduisons Multi-SPMiner, une adaptation de la méthode SPMiner (R. Ying et al., 2020) pour les multigraphes spatio-temporels. Cette approche repose sur un modèle d'apprentissage profond spécialement conçu (Multigraph convolution Network) pour projeter efficacement les nœuds et leur voisinages dans un espace ordonné, tout en préservant la propriété d'inclusion de graphes. Nous améliorons cette méthode en intégrant diverses architectures de réseaux de neurones pour graphes, y compris des modèles basés sur l'attention, et en optimisant la fonction de perte et le processus d'apprentissage. Deuxièmement, nous présentons Deep-QMiner, une méthode novatrice fondée sur l'apprentissage par renforcement profond multi-agents. Cette approche reformule l'exploration des graphes en un processus de prise de décision séquentiel, offrant une nouvelle perspective sur la découverte de motifs fréquents.

Les deux méthodes ont été évaluées sur des ensembles de données synthétiques et des données réelles, notamment les jeux de données Corine Land Cover (CLC) et du Registre Parcellaire Graphique (RPG). Nos expériences montrent l'efficacité

de nos approches pour traiter des graphes allant jusqu'à 2000 nœuds et extraire des motifs de tailles variées (de 4 à 10 nœuds). Les résultats clés concernent la performance supérieure de Multi-SPMiner basé sur l'attention, la robustesse de Multi-SPMiner face aux données manquantes, et les résultats prometteurs de Deep-QMiner pour les graphes de plus petite taille. Nous observons également des performances constantes sur différents ensembles de données, validant la caractère général de notre approche. Enfin, nous démontrons la capacité des deux méthodes à généraliser, en les entraînant sur des données synthétiques et en les testant sur des données réelles CLC et RPG, avec une perte minimale de performance.

Abstract

Graphs are powerful tools for representing complex relationships across various domains such as social networks, biology, and transportation systems. Frequent motif extraction involves identifying recurring subgraphs within one or more graphs. These motifs, indicative of common structures, provide valuable insights into the underlying relationships, properties, and interactions present in the data. However, the algorithmic complexity of traditional motif extraction methods poses a significant challenge. The exhaustive enumeration of all possible motifs exhibits exponential growth, making the counting of occurrences of a given motif an NP-complete problem. This challenge becomes even more pronounced when dealing with multigraphs, which enable the representation of varied and potentially overlapping relationships through different types of edges and directions. Spatiotemporal graphs, a specific category of multigraphs, form a dynamic and interconnected framework where nodes represent entities, edges encode relationships, and spatial and temporal attributes provide rich contextual information. This thesis addresses the challenge of extracting frequent motifs from large spatiotemporal graphs, a crucial task in the era of big data. To this end, we propose innovative approaches based on deep learning and reinforcement learning.

Our main contributions are as follows : First, we introduce Multi-SPMiner, an adaptation of the SPMIner method (R. Ying et al., 2020) for spatiotemporal multigraphs. This approach relies on a specially designed deep learning model (Multigraph Convolution Network) to efficiently project nodes and their neighborhoods into an ordered space while preserving the inclusion property of graphs. We enhance this method by integrating various graph neural network architectures, including attention-based models, and optimizing the loss function and learning process. Second, we present Deep-QMiner, an innovative method based on multi-agent deep reinforcement learning. This approach reformulates graph exploration as a sequential decision-making process, offering a new perspective on frequent motif discovery.

Both methods have been evaluated on synthetic datasets and real-world data, including the Corine Land Cover (CLC) and Parcellaire Graphique (RPG) datasets. Our experiments demonstrate the effectiveness of our approaches in handling graphs of up to 2000 nodes and extracting motifs of varying sizes (from 4 to 10 nodes). Key results include the superior performance of attention-based Multi-SPMiner, the robustness of Multi-SPMiner in the presence of missing data, and the promising outcomes of Deep-QMiner for smaller graphs. We also observe consistent performance across different

datasets, validating the generality of our approach. Finally, we demonstrate the ability of both methods to generalize by training them on synthetic data and testing them on real CLC and RPG data, with minimal loss in performance.

Table des Matières

I	Problématique et Etat de l'art	16
1	Introduction	17
1.1	Contexte et motivation	18
1.2	Objectifs et approche	19
1.3	Importance et implications	21
1.4	Structure de la thèse	22
1.5	Contributions de la thèse	23
1.6	Résumé	24
2	Notions de base et concepts fondamentaux	25
2.1	Définitions	26
2.1.1	Analyse des graphes	27
2.1.2	Recherche de motifs fréquents dans les graphes	28
2.1.3	Définitions et Notations	29
2.2	Graphes spatio-temporels	33
2.2.1	Différentes structures de représentation des graphes spatio-temporels	34
2.2.2	Calcul des connexions entre régions (RCC8)	38
2.3	Fondements des Réseaux de Neurones pour les graphes	40
2.3.1	Réseaux de neurones pour le traitement des graphes	40
2.4	Apprentissage par renforcement	46
2.4.1	Notions de Base	46
2.4.2	Apprentissage par renforcement sur les graphes	49
2.4.3	Apprentissage par renforcement profond sur les graphes	50
2.5	Conclusion	51
3	Etat de l'art	52
3.1	Introduction	53
3.2	Applications de l'apprentissage profond sur les graphes spatio-temporels	53
3.2.1	Imagerie médicale	54

3.2.2	Environnement	56
3.2.3	Reconnaissance d'activités et de mouvements	58
3.2.4	Tâches liées au trafic routier	61
3.2.5	Applications diverses	67
3.3	Exploration des Graphes avec l'Apprentissage par Renforcement	73
3.3.1	Applications Réelles avec l'Apprentissage par Renforcement sur les Graphes	75
3.4	Discussion	77
3.5	Recherche des motifs fréquents dans les graphes	80
II	Développements et résultats	85
4	Méthode Multi-SPMiner	86
4.1	Introduction	87
4.2	La méthode de référence SPMIner	88
4.2.1	Première phase : Encodage des Sous-graphes Candidats	89
4.2.2	Deuxième phase : Procédure de Recherche de Motifs	90
4.2.3	Analyse et Évaluation	92
4.3	Développement de l'approche Multi-SPMiner	93
4.4	Tests du modèle Multi-SPMiner sur des données synthétiques	97
4.4.1	Génération de données synthétiques	97
4.4.2	Modalités des tests et résultats	97
4.5	Conclusion	101
5	Méthode Multi-SPMiner améliorée	103
5.1	Extension vers des Modèles Basés sur l'Attention	104
5.2	Amélioration techniques	108
5.3	Tests des modèles améliorés Multi-SPMiner sur des données synthétiques	110
5.3.1	Réseau de neurones basé sur l'attention :	110
5.3.2	Amélioration sur la fonction du perte et l'entraînement des GNN :	112
5.4	Conclusion	114
6	Méthode Deep QMiner	116
6.1	Introduction	117
6.2	Formulation du problème	118
6.3	Configuration de l'Apprentissage par Renforcement	118
6.3.1	Système de récompense intermédiaire pour l'extraction de motifs	120

6.3.2	Système de récompense progressive pour l'ajout de nœuds . . .	120
6.3.3	Pénalités pour Étapes Invalides	121
6.4	Phase d'Entraînement	122
6.4.1	Exploration Epsilon-Greedy	122
6.4.2	Découverte de motifs multiples au cours des épisodes	124
6.5	Phase de test	126
6.6	Tests de l'approche Multi Agent Deep-QMiner sur des données synthétiques	126
6.7	Conclusion	130
7	Évaluation des méthodes sur les données environnementales et discussion	133
7.1	Données environnementales	134
7.1.1	Registre Parcellaire Graphique	134
7.1.2	Corine Land Cover	135
7.1.3	Génération des graphes spatio-temporels	136
7.1.4	Vérité terrain	137
7.2	Tests sur les données environnementales	137
7.3	Comparaison de la performance computationnelle des approches proposées :	140
8	Conclusion et perspectives	144
8.1	Résumé des contributions	145
8.2	Discussion des résultats	146
8.2.1	Performance de Multi-SPMiner	146
8.2.2	Efficacité de Deep-QMiner	147
8.2.3	Performance sur les données du monde réel	148
8.3	Objectifs et leur réalisation	148
8.4	Limitations et défis	149
8.5	Directions futures	150
A	Annexes	153
A.1	Ajustement des hyperparamètres pour Deep-QMiner	154
A.1.1	Hyperparamètres testés	154
A.1.2	Résultats et observations	154
A.2	Complexité temporelle et spatiale Deep Qminer	155
A.2.1	1. Complexité temporelle	155
A.2.2	2. Complexité spatiale	157

B Bibliographie **159**
Références 160
Liste des travaux 174

Table des Figures

28figure.caption.1		
2.2	La connectivité pondérée entre les sommets proposé par Yu <i>et al.</i> [14].	36
2.3	Grappe-ST basé sur la corrélation proposé par Yu <i>et al.</i> [14].	36
2.4	Représentation d'un graphe-ST, les lignes rouges représentent les relations spatio-temporelles, les lignes vertes représentent les relations spatiales et les lignes bleues représentent les relations de filiation [18]	38
2.5	Relations topologiques du RCC8 [16]	39
2.6	Diagramme de base de l'apprentissage par renforcement	49
3.1	Pipeline du réseau de convolution de graphes spatio-temporels utilisé par [10]	55
3.2	Le framework présenté par [8] pour la prédiction de $PM_{2.5}$	57
3.3	Illustration de Yan <i>et al.</i> [36] expliquant la méthode présentée. À partir de vidéos, des graphe-STs sont construits sur des séquences de squelettes. Plusieurs couches de convolution spatio-temporelle (ST-GCN) sont appliquées et génèrent progressivement des cartes de caractéristiques de plus haut niveau sur le graphe. Un classificateur softmax standard est utilisé pour obtenir la catégorie d'action correspondante.	59
3.4	Illustration de l'approche proposée dans [37]. La séquence de squelettes 3D en entrée est d'abord transformée en squelettes et en trajectoires de nœuds respectivement. Ensuite, le SGR et le TGR produisent respectivement de nouveaux graphes de connectivité de squelettes et d'articulations. ST-GCN reçoit ces graphes et génère une classe d'action.	60
3.5	Vue d'ensemble du travail de Pallabi <i>et al.</i> [38], démontrant les propriétés de leur graphe, permettant divers types de nœuds et supportant des connexions temporelles flexibles pouvant s'étendre sur plusieurs étapes temporelles. Le graphe spatio-temporel alimente une pile de blocs STGCN en sablier pour obtenir une séquence d'actions prédites observées dans la vidéo.	60

3.6	Les trois graphes générés dans le travail de Zhang <i>et al.</i> [41].	63
3.7	L'architecture de LC-RNN telle que démontrée dans Zhong <i>et al.</i> [47]. LC : Convolution par table de correspondance ; BN : Normalisation par lot ; FC : Couche entièrement connectée ; LSTM : Mémoire à long court terme.	64
3.8	Illustration des différents composants du modèle DSTGCNN proposé par [49].	64
3.9	Architecture du système du réseau de convolution multi-graphe spatio- temporel (ST-MGCN) proposé par [50]. Ils encodent différents aspects des relations entre les régions, y compris le voisinage, la similarité fon- ctionnelle et la connectivité de transport, en utilisant plusieurs graphes. Ensuite, ils utilisent un réseau de neurones récurrents contextuels à portes (CGRNN) pour agréger les observations à différents moments en considérant l'information contextuelle globale. Après cela, la convolution multi-graphe est utilisée pour modéliser les corrélations non euclidiennes entre les régions.	65
3.10	Diagramme du framework proposé dans [11]	66
3.11	Diagramme du framework proposé dans [60]	68
3.12	Comparaison entre la méthode conventionnelle pour entraîner le modèle ST et l'approche AutoST [67]	71
4.1	SPMiner se compose de deux étapes principales : la phase d'encodage, où les données sont transformées en une représentation appropriée, suivie de la phase de recherche, au cours de laquelle des motifs fréquents sont identifiés à partir des données encodées.	88
4.2	Région Super-graphe : Tous les points de la région ombragée en rouge correspondent à des voisinages dans G_T qui contiennent G_Q (le motif).	91
4.3	Le framework Multi-SPMiner, contrairement au modèle SPMiner de base, utilise un MGCN pour exploiter les relations multigraphes afin de calculer les voisinages, tout en s'appuyant sur le même principe de l'espace ordonné de plongement.	93
4.4	Création des ensembles de paires positives et négatives pour l'entraîne- ment du MGCN. P (les paires (A, B) où A est un sous-graphe de B), N (les paires (A, B) où A n'est pas un sous-graphe de B)	94

4.5	À chaque étape, Multi SPMiner cherche à ajouter le nœud qui donne la « plus petite violation totale », en se basant sur la fonction apprise lors de la phase d’encodage.	95
4.6	Effet de la suppression des nœuds sur la découverte de motifs dans le graphe : Les pourcentages indiqués sous les barres représentent la proportion de nœuds supprimés, tandis que les valeurs à gauche représentent le ratio correspondant de motifs découverts.	100
6.1	Mécanisme Epsilon-Greedy pour la sélection de nœuds	122
6.2	Approche d’apprentissage profond par Q multi-agents pour la découverte de motifs	127
6.3	Interaction des agents avec l’environnement dans chaque épisode	128
7.1	Extrait du millésime 2017 du RPG sur la plaine d’Alsace	135
7.2	Un extrait d’exemple du jeu de données Corine Land Cover (CLC) illustrant l’évolution de la région parisienne, où différents types de terres sont codés par couleur. Chaque segment se voit attribuer un identifiant unique (ID) essentiel pour le suivi des identités, tandis que les types de terres servent de caractéristiques des nœuds dans le processus de création du graphe spatiotemporel. A droite, les différents types de zones utilisant le 2ème niveau de hiérarchie	136
7.3	Exemple de construction d’un graphe spatio-temporel. Les données représentent une région spatiale évoluant dans le temps, où P1, P2, P3, P4 et P5 correspondent, par exemple, à des parcelles agricoles.	137
7.4	Comparaison des temps d’exécution (en secondes) en utilisant différents modèles.	141
7.5	L’évolution du taux de motifs trouvés des modèles en fonction du temps d’exécution pour différentes tailles de motifs (6, 8, et 10). Chaque courbe représente un modèle spécifique : Baseline, GAT, RGCN, GGCN, HAN, MGAT, et MMAGAT.	142

Première partie

Problématique et Etat de l'art

1

Introduction

Sommaire

1.1	Contexte et motivation	18
1.2	Objectifs et approche	19
1.3	Importance et implications	21
1.4	Structure de la thèse	22
1.5	Contributions de la thèse	23
1.6	Résumé	24

1.1 Contexte et motivation

Dans l'ère des données massives, nous sommes confrontés à une croissance exponentielle des informations spatio-temporelles dans divers domaines, allant de l'environnement à l'urbanisme, en passant par les réseaux sociaux et les systèmes de transport. Ces données, riches en informations sur l'évolution des phénomènes dans le temps et l'espace, offrent un potentiel immense pour la compréhension de dynamiques complexes et la prise de décisions éclairées. Cependant, la valorisation de ces grandes masses de données spatio-temporelles nécessite le développement de nouvelles approches, capables de traiter conjointement les aspects spatiaux et temporels, une tâche que les méthodes actuelles peinent à accomplir de manière efficace.

Les graphes spatio-temporels offrent un modèle puissant pour modéliser des phénomènes évolutifs complexes et peuvent être utilisés pour représenter ces données spatio-temporelles. Ils permettent de capturer non seulement les relations spatiales entre les entités à un moment donné, mais aussi leur évolution dans le temps et les liens de succession entre différents états d'une même entité. Cette représentation s'avère particulièrement intuitive et facilement compréhensible pour le cerveau humain, offrant ainsi une perspective naturelle sur les dynamiques complexes des données spatio-temporelles.

Les graphes spatio-temporels se distinguent par leur capacité à intégrer simultanément les dimensions spatiales et temporelles des données. La structure de ces graphes peut varier dans le temps, avec des nœuds et des arêtes qui apparaissent, disparaissent ou se modifient, reflétant ainsi la nature dynamique des systèmes qu'ils représentent. Cette richesse de représentation ouvre de nouvelles perspectives pour l'analyse de données complexes, mais pose également des défis significatifs en termes de traitement et d'interprétation.

L'extraction de motifs fréquents dans les graphes est un domaine fondamental de la fouille de données, visant à identifier des structures récurrentes significatives au sein de réseaux complexes. Cette tâche a démontré son utilité dans de nombreux domaines, de l'analyse des réseaux sociaux à la bioinformatique. Cependant, lorsqu'il s'agit de graphes spatio-temporels, qui incorporent à la fois des relations spatiales et des évolutions temporelles, les défis deviennent considérablement plus importants.

Cependant, les méthodes classiques d'extraction de motifs fréquents dans les graphes se heurtent à des obstacles significatifs lorsqu'elles sont appliquées aux graphes spatio-temporels de grande taille. Ces approches, souvent basées sur des algorithmes d'énumération exhaustive ou de croissance de motifs, deviennent rapidement impraticables en termes de temps de calcul et de ressources mémoire lorsque la taille et la complexité du graphe augmentent. De plus, elles peinent généralement à capturer efficacement l'information portée par des relations spatio-temporelles, traitant souvent les dimensions spatiales et temporelles de manière séparée, ce qui peut conduire à une perte de précision et de capacité d'interprétation des résultats.

Face à ces défis, l'utilisation de méthodes d'apprentissage profond apparaît comme une piste prometteuse. Ces techniques ont démontré leur capacité à apprendre des représentations complexes et à généraliser à partir de grands ensembles de données dans divers domaines. Appliquées à l'extraction de motifs fréquents dans les graphes spatio-temporels, elles pourraient permettre de surmonter les limitations des approches traditionnelles en termes de temps de calcul et de capacité à capturer des motifs complexes. La recherche de motifs fréquents dans ces graphes spatio-temporels présente un intérêt majeur pour de nombreuses applications. Dans le domaine de l'environnement, par exemple, l'identification de motifs récurrents dans l'évolution de l'occupation de sols peut révéler des tendances importantes dans l'utilisation des terres, la progression de l'urbanisation, ou les changements écologiques. Ces informations sont cruciales pour la planification urbaine, la gestion des ressources naturelles et l'élaboration de politiques environnementales.

1.2 Objectifs et approche

L'objectif principal de cette thèse est d'étudier et de développer des méthodes d'apprentissage profond innovantes pour l'extraction de motifs fréquents (sous-graphes spatio-temporels fréquents) à partir de grands graphes spatio-temporels. Notre ambition est de concevoir des approches qui peuvent être entraînées une seule fois et ensuite appliquées efficacement à un nombre illimité de graphes, offrant ainsi une solution générique et évolutive à ce problème complexe.

Plus spécifiquement, nous visons à développer des méthodes capables de traiter des graphes de grande taille, allant jusqu'à 2000 nœuds, avec l'objectif à long terme de gérer des graphes de 10 000 nœuds. Nous nous concentrons sur l'extraction de motifs de tailles variées, allant de 4 à 10 nœuds, couvrant ainsi un large éventail de structures potentiellement significatives.

Notre approche se distingue des méthodes d'apprentissage profond en ce qu'elle vise à apprendre le processus même de découverte de motifs fréquents, plutôt que d'apprendre simplement la forme des motifs eux-mêmes dans les graphes, ou les tendances dans les données. Cette orientation vers l'apprentissage du processus offre la promesse d'une meilleure généralisation et d'une plus grande efficacité computationnelle.

Pour atteindre ces objectifs, notre méthodologie s'est articulée autour de plusieurs axes clés :

1. **État de l'art approfondi** : Nous avons mené une recherche approfondie sur l'état de l'art pour comprendre les avancées récentes et les défis persistants dans le domaine de l'extraction de motifs fréquents dans les graphes spatio-temporels. Cette revue nous a permis d'identifier les lacunes dans les approches existantes et d'orienter notre recherche vers des solutions innovantes. Ce travail a été valorisé par la publication d'un article dans le journal *Neurocomputing* [1].
2. **Adaptation et extension de SPMiner** : Au cours de notre recherche, nous avons identifié SPMiner comme une méthode prometteuse pour l'extraction de motifs fréquents dans les graphes non orientés et non étiquetés. Nous avons donc choisi d'adapter cette approche aux multigraphes spatio-temporels, qui présentent des défis supplémentaires en termes de complexité et de dimensionnalité. Cette adaptation implique de gérer les multiples types d'arêtes caractéristiques des graphes spatio-temporels. Cette approche a été présentée et publiée lors de la conférence internationale KES [2].
3. **Développement d'une approche de Deep-Q Learning** : Nous avons développé une méthode novatrice qui combine les forces de l'apprentissage profond et de l'apprentissage par renforcement. Cette approche vise à tirer parti de la capacité de l'apprentissage profond à apprendre des représentations complexes, tout en exploitant la nature séquentielle et décisionnelle de l'apprentissage par renforcement pour guider efficacement la recherche de motifs dans l'espace des graphes (Publication à venir).
4. **Validation sur données synthétiques et réelles** : Nous avons validé les méthodes sur des graphes synthétiques générés de manière contrôlée. Cette étape nous permet d'évaluer rigoureusement les performances et la robustesse de nos approches dans des conditions variées et bien définies. Par la suite, nous appliquons nos méthodes à des données environnementales réelles, concernant l'occupation du sol à différentes échelles, notamment issues du Registre Parcellaire Graphique, pour démontrer leur efficacité et leur pertinence dans des contextes pratiques.

5. **Optimisation pour le passage à l'échelle** : Un aspect crucial de notre recherche est l'optimisation de nos méthodes pour qu'elles puissent traiter efficacement des graphes de grande taille. Nous explorons diverses techniques d'optimisation, tant au niveau algorithmique qu'au niveau de l'implémentation, pour atteindre notre objectif de traiter des graphes de 2000 nœuds ou plus.

1.3 Importance et implications

L'importance de cette recherche réside dans son potentiel à transformer notre capacité à analyser et comprendre les dynamiques complexes capturées dans les grands graphes spatio-temporels. Les implications de ce travail sont vastes et multidisciplinaires :

- **Avancées méthodologiques** : En développant des méthodes d'apprentissage profond pour l'extraction de motifs fréquents dans les graphes spatio-temporels, nous contribuons à l'avancement des techniques d'analyse de données complexes. Ces avancées peuvent avoir des répercussions bien au-delà du domaine spécifique des graphes spatio-temporels, influençant potentiellement d'autres domaines de l'apprentissage automatique et de la fouille de données.
- **Efficacité computationnelle** : En visant à développer des méthodes qui peuvent être entraînées une fois et appliquées à de multiples graphes, nous abordons le défi crucial de l'efficacité computationnelle dans l'analyse de grands ensembles de données. Cette approche peut ouvrir la voie à des applications en temps réel ou quasi-réel de l'analyse de graphes spatio-temporels.
- **Généralisation et adaptabilité** : Notre objectif de créer des méthodes génériques capables de traiter divers types de graphes spatio-temporels promet une grande adaptabilité. Cela pourrait faciliter l'application de ces techniques à une variété de domaines au-delà de l'environnement, comme les réseaux de transport, les systèmes sociaux, ou même les réseaux biologiques cérébraux.
- **Applications environnementales** : Dans le contexte de l'étude de l'évolution du territoire, notre approche peut révéler des motifs significatifs dans les changements d'utilisation des terres, la progression de l'urbanisation, ou les dynamiques écologiques. Ces informations sont cruciales pour la planification urbaine, la gestion des ressources naturelles et l'élaboration de politiques environnementales informées.

1.4 Structure de la thèse

Cette thèse est organisée en deux parties : la première partie est dédiée au cadre théorique et à l'état de l'art, tandis que la seconde partie se regroupe sur le développement, les contributions et la discussion. Chaque partie de cette thèse est divisée en plusieurs chapitres, chacun abordant un aspect spécifique de notre recherche, et suit la chronologie de nos réflexions et le développement des contributions.

Cette thèse est structurée en deux parties principales : la Problématique et l'État de l'art, suivis des Développements et résultats. La première partie, Problématique et État de l'art, se compose de trois chapitres. Le chapitre d'introduction présente le contexte, les motivations, les objectifs et l'importance de notre recherche. Il est suivi par un chapitre sur les fondements théoriques, qui expose les concepts fondamentaux des graphes spatio-temporels, de l'apprentissage profond et de l'apprentissage par renforcement nécessaires à la compréhension de notre approche. Ce chapitre introduit également les notions utilisées dans le reste du manuscrit. Le troisième chapitre de cette partie propose une revue approfondie des méthodes d'apprentissage profond appliquées aux graphes spatio-temporels, ainsi que de l'extraction de motifs fréquents dans les graphes. La deuxième partie, Développements et résultats, comprend cinq chapitres. Le premier chapitre de cette partie présente Multi-SPMiner, notre adaptation de SPMiner pour les multigraphes spatio-temporels, incluant les détails de l'architecture, les modifications apportées et les premières évaluations. Le chapitre suivant détaille les améliorations apportées à Multi-SPMiner, notamment l'introduction de mécanismes d'attention et l'optimisation de la fonction de perte et du processus d'entraînement. Le troisième chapitre de cette partie décrit Deep-QMiner, notre approche basée sur l'apprentissage profond par renforcement pour l'extraction de motifs fréquents, incluant l'architecture, le processus d'apprentissage et les défis spécifiques abordés. L'avant-dernier chapitre présente en détail nos expériences sur des données réelles environnementales, propose une analyse critique de nos résultats, discute des implications de notre travail et identifie des pistes prometteuses pour les recherches futures. Enfin, la thèse se conclut par un résumé des contributions principales et des réflexions sur l'impact potentiel de notre travail dans le domaine de la recherche des motifs fréquents dans des graphes spatio-temporels.

À travers cette structure, nous visons à présenter une exploration complète et cohérente de notre approche pour l'extraction de motifs fréquents dans les grands graphes spatio-temporels, depuis les fondements théoriques jusqu'aux applications pratiques et aux perspectives futures.

1.5 Contributions de la thèse

Les contributions attendues de cette thèse sont multiples et s'étendent sur plusieurs aspects de la recherche en fouille de données et en apprentissage automatique :

1. **Avancées théoriques** : Nos travaux contribuent à l'avancement des méthodes de représentation et d'analyse de données spatio-temporelles pour l'extraction de motifs fréquents. Cela inclut le développement de nouvelles approches pour le plongement de graphes spatio-temporels et l'adaptation des techniques d'apprentissage profond à ce contexte spécifique.
2. **Nouvelle méthode d'extraction de motifs** : L'adaptation et l'amélioration de SPMiner pour les multigraphes spatio-temporels, ainsi que le développement de Deep-QMiner, représentent des contributions méthodologiques significatives. Ces approches promettent d'offrir des solutions plus efficaces et plus adaptables que les méthodes existantes pour l'extraction de motifs fréquents dans les grands graphes spatio-temporels.
3. **Prototype générique** : Le développement d'un prototype capable de traiter divers types de données spatio-temporelles représente une contribution pratique importante. Ce prototype pourrait servir de base pour de futures applications dans divers domaines nécessitant l'analyse de données spatio-temporelles complexes.
4. **Optimisations pour le passage à l'échelle** : Les techniques que nous avons développées permettent le traitement efficace de grands graphes (jusqu'à 2000 nœuds) et ont des applications au-delà de notre domaine spécifique, contribuant à l'amélioration générale de l'analyse de grands graphes. Ces optimisations peuvent être appliquées à divers types de données représentées sous forme de graphes, y compris des données cérébrales.
5. **Intégration de l'apprentissage profond et de l'apprentissage par renforcement** : Notre approche combinant ces deux paradigmes d'apprentissage automatique pour l'extraction de motifs ouvre de nouvelles perspectives dans le domaine de l'apprentissage sur graphes. Cette intégration pourrait notamment permettre le développement de systèmes plus adaptatifs et autonomes, capables d'ajuster dynamiquement leurs stratégies d'extraction de motifs en fonction de la structure et de l'évolution des graphes spatio-temporels.

1.6 Résumé

L'extraction de motifs fréquents dans les grands graphes spatio-temporels représente un défi majeur à l'intersection de plusieurs domaines : la fouille de données, l'apprentissage automatique et l'analyse spatio-temporelle. Notre recherche vise à repousser les frontières de ce domaine en développant des méthodes innovantes basées sur l'apprentissage profond et l'apprentissage par renforcement.

Notre travail s'articule autour de contributions théoriques significatives dans le domaine de l'analyse des graphes spatio-temporels. Nous proposons de nouvelles approches algorithmiques qui visent à améliorer l'efficacité et la précision de la détection de motifs fréquents dans ces structures complexes. Ces méthodes ont le potentiel de généraliser les techniques existantes et d'ouvrir de nouvelles perspectives pour l'analyse de graphes dynamiques dans divers contextes. En combinant l'apprentissage profond et l'apprentissage par renforcement, nous explorons des paradigmes novateurs pour aborder la complexité inhérente aux graphes spatio-temporels, offrant ainsi un cadre théorique robuste pour futures recherches dans ce domaine.

Cette thèse s'inscrit dans un contexte plus large de valorisation des grandes masses de données spatio-temporelles, un enjeu crucial à l'ère du *big data*. En proposant des approches capables de traiter efficacement ces données complexes, nous espérons ouvrir la voie à de nouvelles perspectives d'analyse et de compréhension des phénomènes évolutifs dans divers domaines. Nos contributions théoriques visent à établir un fondement solide pour le développement d'applications pratiques dans de nombreux secteurs, allant de l'analyse des réseaux sociaux à la modélisation de systèmes complexes en passant par l'étude des dynamiques urbaines et environnementales.

Les chapitres suivants détailleront notre démarche, depuis l'état de l'art jusqu'aux expérimentations et résultats, en passant par le développement de nos méthodes innovantes. À travers ce travail, nous espérons non seulement apporter des contributions significatives au domaine de l'analyse des graphes spatio-temporels, mais aussi inspirer de futures recherches et applications dans ce domaine en constante évolution. Notre objectif est de fournir un cadre théorique solide qui pourra être adapté et appliqué à une variété de problèmes concrets, démontrant ainsi la puissance et la flexibilité de nos approches dans différents contextes d'analyse de données spatio-temporelles.

2

Notions de base et concepts fondamentaux

Sommaire

2.1	Définitions	26
2.1.1	Analyse des graphes	27
2.1.2	Recherche de motifs fréquents dans les graphes	28
2.1.3	Définitions et Notations	29
2.2	Graphes spatio-temporels	33
2.2.1	Différentes structures de représentation des graphes spatio-temporels	34
2.2.2	Calcul des connexions entre régions (RCC8)	38
2.3	Fondements des Réseaux de Neurones pour les graphes	40
2.3.1	Réseaux de neurones pour le traitement des graphes	40
2.4	Apprentissage par renforcement	46
2.4.1	Notions de Base	46
2.4.2	Apprentissage par renforcement sur les graphes	49
2.4.3	Apprentissage par renforcement profond sur les graphes	50
2.5	Conclusion	51

2.1 Définitions

L'analyse des données spatiales et temporelles est devenue un domaine de recherche crucial en raison de la quantité massive de données générées par diverses applications telles que la surveillance environnementale, la gestion urbaine, les réseaux sociaux [3], *etc...* Dans ce contexte, les graphes spatio-temporels jouent un rôle central en modélisant les relations dynamiques entre entités dans le temps et l'espace, fournissant ainsi une représentation riche des phénomènes dynamiques. L'extraction de motifs fréquents dans les graphes est une technique puissante pour découvrir des structures récurrentes et significatives au sein de ces réseaux complexes. En effet, ces motifs permettent de révéler des structures récurrentes et significatives au sein des réseaux, qu'ils soient statiques ou évolutifs. Dans le contexte des graphes spatio-temporels, la découverte de motifs fréquents peut ainsi offrir des éclairages précieux sur les interactions typiques qui émergent au fil du temps.

Dans ce chapitre, nous explorons les concepts fondamentaux liés aux graphes spatio-temporels, leur définition, leurs propriétés et leur importance dans l'analyse des données. Nous abordons également l'analyse des données environnementales, en soulignant les différents types de données environnementales couramment utilisés et les défis associés à leur traitement.

Ensuite, nous introduisons les réseaux de neurones pour graphes (GNNs), une classe de modèles d'apprentissage profond spécialement conçus pour l'analyse des données de graphes. Ces modèles ont montré des performances prometteuses dans diverses tâches, telles que la classification des nœuds, la prédiction des liens et la détection de motifs.

Enfin, nous discutons de l'apprentissage par renforcement appliqué aux données de graphes. Cette approche permet de formuler des stratégies optimales pour explorer et exploiter les structures des graphes, en apprenant par interaction avec l'environnement de données.

Ce chapitre fournit une base théorique et pratique pour comprendre les techniques et les méthodes avancées utilisées dans la suite du manuscrit pour l'exploration des motifs fréquents dans les grands graphes spatio-temporels à l'aide de l'apprentissage profond.

2.1.1 Analyse des graphes

Dans le traitement des graphes, les tâches peuvent être catégorisées en deux types principaux : les tâches focalisées sur les nœuds et les tâches focalisées sur les graphes [4], comme illustré dans la Figure 2.1. Les tâches focalisées sur les nœuds impliquent la prédiction des propriétés des nœuds individuels dans un graphe, comme l'apprentissage d'une représentation pour chaque nœud en vue de prédire une variable cible associée au nœud, telle que son étiquette, sa centralité de degré, ou encore sa fonction dans un réseau. Par exemple, la classification des nœuds vise à attribuer une étiquette à chaque nœud, elle est souvent utilisée dans des contextes tels que la classification des utilisateurs dans des réseaux sociaux ou des molécules dans des réseaux chimiques. Le regroupement des nœuds, quant à lui, cherche à identifier des communautés ou des clusters dans lesquels les nœuds sont fortement interconnectés, ce qui est utile dans l'analyse de réseaux sociaux pour détecter des groupes d'intérêts communs ou des sous-structures denses dans les réseaux biologiques. Enfin, la régression des nœuds se concentre sur la prédiction de valeurs numériques associées à chaque nœud, telles que la popularité d'un utilisateur ou la réactivité d'une molécule. Les tâches focalisées sur les graphes, quant à elles, visent à apprendre des représentations de graphes entiers et à prédire des propriétés globales du réseau. Ces tâches jouent un rôle crucial dans l'analyse de structures complexes où la totalité du graphe, plutôt que les nœuds individuels, est d'intérêt. Par exemple, la classification des graphes consiste à attribuer une étiquette à un graphe complet, comme dans le cas de la classification de molécules en fonction de leurs propriétés chimiques ou biologiques [5]. Cela est particulièrement utilisé dans des domaines comme la bio-informatique, où il s'agit de prédire si une molécule aura un effet thérapeutique ou toxique en fonction de sa structure. Le regroupement de graphes cherche à identifier des groupes de graphes ayant des similitudes structurelles, un processus essentiel dans la comparaison de réseaux de communication ou dans la recherche de structures similaires dans des bases de données moléculaires. Enfin, la génération de graphes vise à produire de nouveaux graphes ayant certaines propriétés, ce qui est utile dans des contextes comme la conception de molécules avec des caractéristiques spécifiques ou la modélisation de réseaux de transport. Ces tâches focalisées sur les graphes nécessitent des approches différentes de celles focalisées sur les nœuds, car elles visent à capturer les relations complexes entre les nœuds au niveau global. Cela inclut non seulement les connexions entre les nœuds mais aussi des propriétés plus abstraites telles que la connectivité générale, la densité du graphe, et des métriques comme le coefficient de clustering ou le spectre laplacien, qui offrent des indications sur la structure globale du réseau. L'une des tâches clés dans l'analyse des graphes est la découverte de motifs fréquents, un processus qui consiste à identifier des sous-structures récurrentes au sein de

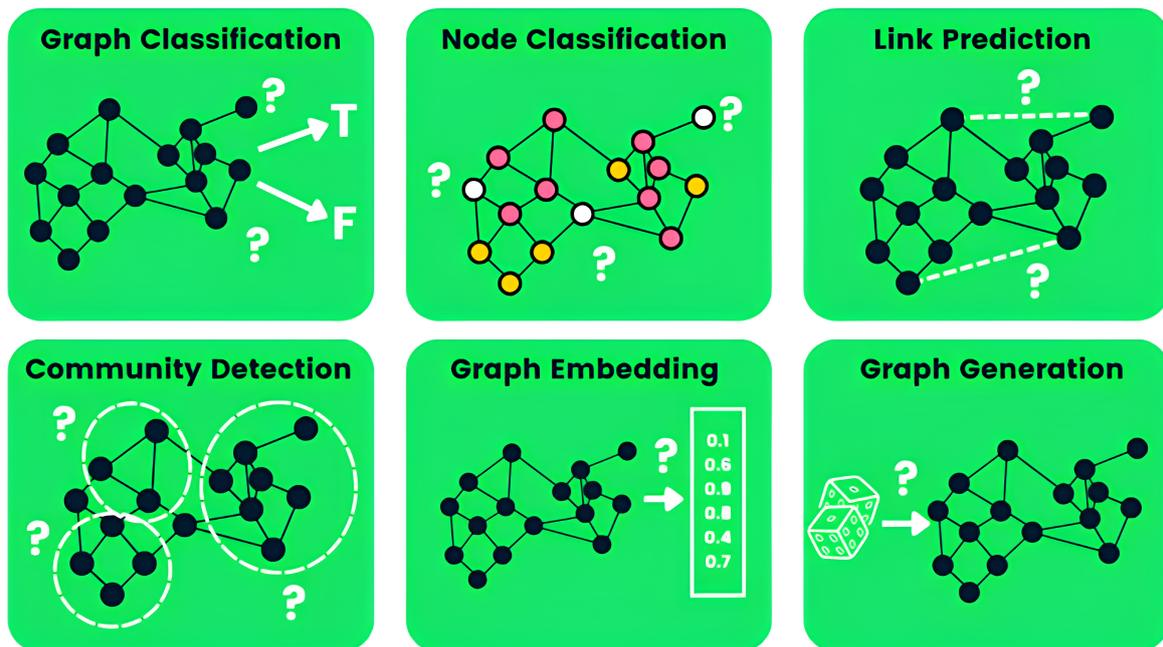


FIGURE 2.1 – Différents tâches focalisées sur les nœuds et les graphes¹.

grands réseaux. Ces motifs fréquents, souvent appelés motifs récurrents ou sous-graphes fréquents, peuvent révéler des informations essentielles sur la manière dont les entités interagissent dans un réseau. En capturant des motifs communs, ça permet non seulement de mieux comprendre la structure globale du graphe, mais aussi de déduire des comportements ou des tendances cachées. Dans la section suivante, nous allons explorer en détail la notion de recherche de motifs fréquents et son importance dans le contexte des graphes.

2.1.2 Recherche de motifs fréquents dans les graphes

La recherche de motifs fréquents dans les graphes est le processus d'identification de motifs de sous-graphes (motifs) qui apparaissent fréquemment dans un graphe plus grand ou dans un ensemble de graphes. Ces motifs sont de relativement petites structures récurrentes et significatives qui peuvent fournir des informations sur les processus ou les fonctionnalités sous-jacents des données de graphe. Cette technique est largement utilisée dans divers domaines tels que la bioinformatique, l'analyse des réseaux sociaux et les réseaux de communication pour découvrir des motifs et des relations cachés [6]. Dans le domaine de la bioinformatique, la recherche de motifs fréquents dans les graphes joue un rôle crucial dans l'analyse des réseaux de protéines. Par exemple, dans un réseau d'interactions protéine-protéine, chaque

nœud représente une protéine, et chaque arête représente une interaction entre deux protéines. Identifier des motifs fréquents dans ces réseaux permet de détecter des sous-structures qui sont associées à des fonctions biologiques spécifiques [7]. Un motif récurrent pourrait, par exemple, correspondre à un ensemble de protéines qui coopèrent pour catalyser une réaction enzymatique. En découvrant ces motifs, les chercheurs peuvent mieux comprendre les mécanismes biologiques sous-jacents et potentiellement identifier des cibles thérapeutiques pour des maladies liées à des dysfonctionnements dans ces interactions.

Le processus de recherche de motifs fréquents se déroule généralement comme suit :

- **Énumération des sous-graphes** : Identifier tous les sous-graphes possibles au sein du graphe principal. Cette étape consiste à générer des sous-graphes candidats de différentes tailles et types.
- **Test d'isomorphisme** : Vérifier si deux sous-graphes sont isomorphes, c'est-à-dire s'ils peuvent être transformés l'un en l'autre en re-étiquetant leurs nœuds. Cette étape permet de s'assurer que les motifs identiques sont comptés correctement et ne sont pas comptés plusieurs fois.
- **Comptage des fréquences** : Compter les occurrences de chaque motif de sous-graphe dans le graphe. Cela peut être fait en utilisant des méthodes de comptage exactes ou des algorithmes approximatifs pour plus d'efficacité.
- **Filtrage des motifs** : Appliquer des seuils de fréquence pour filtrer les sous-graphes rares, ne conservant que ceux qui apparaissent suffisamment fréquemment pour être considérés comme des motifs.

2.1.3 Définitions et Notations

Dans le cadre de l'analyse de données complexes, les graphes jouent un rôle fondamental en permettant de représenter des objets et leurs relations sous forme de réseaux. Que ce soit dans des contextes sociaux, biologiques, ou informatiques, les graphes offrent un cadre flexible et puissant pour modéliser des systèmes composés de nombreuses entités interconnectées. Avant d'aller plus loin dans des applications spécifiques, il est essentiel de commencer par définir formellement ce qu'est un graphe, car cette structure constitue la base des concepts que nous développerons par la suite.

Definition 1 *Un graphe G est une paire d'ensembles finis $G = (V; E)$ où V est l'ensemble des sommets ou nœuds et $E \subseteq V \times V$ est l'ensemble des arêtes. Lorsque deux sommets $x, y \in V$ sont adjacents dans le graphe, ils sont reliés par une arête, relation ou lien, ce qui se traduit par une entrée correspondante dans l'ensemble des arêtes $x; y \in E$.*

Definition 2 Deux arêtes sont également **adjacentes** si elles partagent un sommet. Les ensembles des sommets et respectivement des arêtes d'un graphe G sont aussi notés $V(G)$ et respectivement $E(G)$.

Definition 3 L'ordre n d'un graphe est défini par le nombre de ses sommets, c'est-à-dire, la **cardinalité** de l'ensemble des sommets V . Le nombre d'arêtes dans un graphe est sa taille m , notée comme la cardinalité de l'ensemble des arêtes $|E|$. Un graphe avec n sommets peut également être désigné comme un **n -graphe**, noté G_n .

Les graphes peuvent avoir des caractéristiques différentes. En effet, certains graphes présentent des propriétés spéciales qui les rendent particulièrement utiles dans divers contextes d'analyse. Parmi ces variantes, on trouve les graphes étiquetés, où des informations supplémentaires sont assignées à chaque sommet et arête, permettant de capturer plus de détails sur la nature des entités et des relations qu'ils modélisent.

Definition 4 (Graphe étiqueté) Un graphe étiqueté est un triplet $G = (V, E, \ell)$, où V est un ensemble de sommets, $E \subseteq V \times V$ est un ensemble d'arêtes, et $\ell : V \rightarrow L$ est une fonction d'étiquetage qui associe chaque sommet de V à une étiquette de L .

Un graphe ne contient généralement pas de boucles, c'est-à-dire, d'arêtes reliant un sommet à lui-même, ni plus d'une arête entre une paire de sommets. Parfois, les entités d'un graphe peuvent entretenir plusieurs types d'interactions ou de relations simultanément. Dans ces situations, les graphes simples ne suffisent plus à modéliser cette complexité. C'est là qu'interviennent les multigraphes, qui permettent de représenter plusieurs arêtes entre une même paire de sommets, chacune pouvant correspondre à une relation différente.

Definition 5 (Multigraphe) Un multigraphe $G = (V, E, l_V, l_E, L_V, L_E)$ comprend un ensemble V de sommets, un ensemble d'arêtes $E \subseteq V \times V$, et deux ensembles d'étiquettes L_V, L_E . Il est caractérisé par des fonctions d'étiquetage $l_V : V \rightarrow 2^{L_V}$ pour les sommets et $l_E : E \rightarrow 2^{L_E}$ pour les arêtes.

Cette généralisation permet de modéliser des relations multiples et variées entre les entités du graphe, offrant ainsi une représentation encore plus riche des interactions.

Les graphes, en particulier lorsqu'ils deviennent de grande taille, peuvent être difficiles à représenter de manière formelle et à manipuler efficacement. Pour surmonter cette complexité, les matrices offrent une solution pratique en permettant de traduire les structures de graphe en une forme plus compacte et manipulable. Parmi les outils les plus couramment utilisés, la matrice d'adjacence et la matrice de degré jouent un rôle clé dans la description des connexions entre les sommets et des propriétés locales du graphe.

Definition 6 (Matrice d'adjacence et Matrice de degré) Une matrice d'adjacence A d'un n -graphe est une matrice binaire $n \times n$, où chaque entrée non nulle correspond à une arête allant du sommet v_i au sommet v_j . Une matrice de degré D pour un graphe G est une matrice diagonale $n \times n$ définie comme suit :

$$D_{i,j} := \begin{cases} \deg(v_i) & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

où le degré $\deg(v_i)$ d'un sommet compte le nombre d'arête ayant v_i comme extrémité.

En traitement des graphes, il ne suffit souvent pas de connaître uniquement les connexions entre les sommets (capturées par la matrice d'adjacence) ou le degré des sommets (décrit par la matrice de degré). Il est également essentiel de comprendre comment ces deux types d'informations interagissent pour révéler des propriétés plus globales du graphe, telles que la connectivité ou les chemins possibles entre les sommets. C'est ici qu'intervient la matrice laplacienne. En combinant les données fournies par la matrice d'adjacence et la matrice de degré, la matrice laplacienne permet de capturer à la fois la structure locale et globale du graphe, et elle joue un rôle central dans de nombreuses applications, allant de l'analyse spectrale à l'optimisation de flux.

Definition 7 (Matrice de Laplacien) La matrice de Laplacien d'un graphe, souvent abrégée en Laplacien, est définie par $\mathbb{L} = D - A$. La matrice de Laplacien normalisée est $L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$. La décomposition spectrale du Laplacien est définie par $\mathbb{L} = Q\Lambda Q^T$, où $\Lambda \in \mathbb{R}^{N \times N}$ est une matrice diagonale des valeurs propres triées par ordre croissant et $Q \in \mathbb{R}^{N \times N}$ sont les vecteurs propres correspondants.

Il est parfois nécessaire de se concentrer sur une partie spécifique du graphe. Cela nous amène au concept de sous-graphe, qui permet d'examiner une portion du graphe tout en conservant certaines propriétés de la structure originale. Les sous-graphes sont particulièrement utiles lorsqu'on cherche à étudier des sous-ensembles d'entités ou de relations au sein de grands réseaux complexes.

Definition 8 (Sous-graphe) Un sous-graphe d'un multigraphe $G = (V, E, l_V, l_E, L_V, L_E)$ est un multigraphe $G' = (V', E', l_{V'}, l_{E'}, L_{V'}, L_{E'})$ tel que $V' \subseteq V$, $E' \subseteq E$, et $\forall v \in V', l_{V'}(v) \subseteq l_V(v)$ et $\forall e \in E', l_{E'}(e) \subseteq l_E(e)$. Deux types de sous-graphes peuvent être définis : G' est un sous-graphe induit par les nœuds si et seulement si $V' \subseteq V$, $E' = \{(u, v) \in E \mid u \in V', v \in V'\}$. G' est un sous-graphe induit par les arêtes si et seulement si $E' \subseteq E$, et $V' = \{v \in V \mid \exists u \in V \text{ et } (u, v) \in E'\}$.

Dans de nombreuses applications, comme la recherche de motifs fréquents ou l'analyse des similarités dans des réseaux complexes, il est crucial de pouvoir identifier des structures équivalentes. Bien que deux graphes puissent avoir des étiquettes ou des

présentations différentes, ils peuvent néanmoins partager la même organisation. C'est là qu'intervient l'isomorphisme, une propriété qui permet de déterminer deux graphes ou sous-graphes structurellement équivalents.

Definition 9 (Isomorphisme) $G_1 = (V_1, E_1, l_{V_1}, l_{E_1}, L_V, L_E)$ et $G_2 = (V_2, E_2, l_{V_2}, l_{E_2}, L_V, L_E)$ sont isomorphes s'il existe une bijection $f : V_1 \rightarrow V_2$ telle que $(u, v) \in E_1$ si et seulement si $(f(u), f(v)) \in E_2$, $l_{V_1}(u) = l_{V_2}(f(u))$, $l_{V_1}(v) = l_{V_2}(f(v))$ et $l_{E_1}(u, v) = l_{E_2}(f(u), f(v))$. La bijection est une correspondance un-à-un entre les nœuds d'un graphe et d'un autre, assurant que toutes les arêtes et étiquettes sont préservées.

Definition 10 (Isomorphisme de sous-graphe) L'isomorphisme de sous-graphe est la propriété de déterminer si un graphe donné G contient un autre graphe H comme sous-graphe. Étant donné $G = (V_G, E_G, l_{V_G}, l_{E_G}, L_V, L_E)$ et $H = (V_H, E_H, l_{V_H}, l_{E_H}, L_V, L_E)$, nous disons que H est isomorphe à un sous-graphe de G , noté $H \subseteq G$, s'il existe une correspondance bijective $\phi : V_H \rightarrow V_G$ telle que $(u, v) \in E_H$ si et seulement si $(\phi(u), \phi(v)) \in E_G$, $l_{V_H}(u) \subseteq l_{V_G}(\phi(u))$, $l_{V_H}(v) \subseteq l_{V_G}(\phi(v))$ et $l_{E_H}(u, v) \subseteq l_{E_G}(\phi(u), \phi(v))$.

Definition 11 Une relation d'ordre partiel \leq est définie sur l'ensemble des graphes \mathcal{G} par : $A \leq B$ si et seulement si le graphe A est isomorphe à un sous-graphe de B .

Après avoir établi les concepts fondamentaux tels que les graphes, les sous-graphes et l'isomorphisme, nous avons désormais les outils nécessaires pour analyser et comparer les structures de réseau de manière formelle. Ces notions permettent d'explorer tant la structure globale du graphe que les relations locales entre ses entités. Cette compréhension approfondie des similarités structurelles entre graphes et sous-graphes ouvre la voie à l'identification de motifs récurrents au sein des réseaux complexes.

Definition 12 (La fréquence d'un sous-graphe) La fréquence d'un sous-graphe G' peut être définie en termes du nombre de fois que G' ou ses sous-graphes isomorphes apparaissent dans G . Soit $\mathcal{M}(G')$ l'ensemble de tous les sous-graphes dans G qui sont isomorphes à G' . La fréquence de G' dans G est donnée par :

$$freq(G') = |\mathcal{M}(G')|$$

Un sous-graphe G' est considéré comme un motif fréquent si :

$$freq(G') \geq \theta$$

où $\theta \in \mathbb{N}$ est un seuil de fréquence prédéfini.

Definition 13 (Motif) Étant donné un graphe $G = (V, E)$ où V est l'ensemble des sommets et E est l'ensemble des arêtes, un sous-graphe $G' = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$ est appelé **motif** s'il apparaît fréquemment dans G ou dans plusieurs graphes.

Les multigraphes qui permettent plusieurs arêtes entre une même paire de nœuds, introduisent des nuances supplémentaires dans l'analyse des motifs. En effet, la présence de multiples types de relations ou d'interactions entre les entités d'un réseau rend la découverte de motifs fréquents plus riche et plus complexe. Nous allons maintenant explorer comment la notion de motifs fréquents s'applique aux multigraphes et en quoi elle permet de capturer des structures récurrentes encore plus diversifiées.

Definition 14 *Un motif fréquent est un sous-graphe G' qui apparaît fréquemment dans G ou dans plusieurs multigraphes. La fréquence d'un motif G' peut être définie comme le nombre de fois où G' ou ses sous-graphes isomorphes apparaissent dans G . Soit $\mathcal{M}(G')$ l'ensemble de tous les sous-graphes dans G qui sont isomorphes à G' . La fréquence de G' dans G est donnée par :*

$$freq(G') = |\mathcal{M}(G')|$$

Un sous-graphe G' est considéré comme un motif fréquent si :

$$freq(G') \geq \theta$$

Motifs Ancrés sur les Nœuds et les Arêtes

Les motifs fréquents offrent une vue précieuse sur les structures récurrentes dans divers types de graphes. Cependant, lorsque les relations évoluent non seulement dans l'espace mais aussi dans le temps, la modélisation devient plus complexe. C'est ici que les graphes spatio-temporels entrent en jeu. Ces graphes permettent de capturer à la fois les interactions dynamiques entre les entités et leur évolution temporelle, offrant ainsi une représentation plus riche et fidèle des systèmes complexes.

2.2 Graphes spatio-temporels

Les graphes spatio-temporels constituent un outil essentiel pour modéliser et analyser des systèmes complexes où les interactions entre les entités évoluent simultanément dans l'espace et dans le temps. Contrairement aux graphes traditionnels, qui capturent des relations statiques à un moment donné, les graphes spatio-temporels intègrent la dimension temporelle, permettant de suivre l'évolution dynamique des connexions au fil du temps. Ce type de graphe est particulièrement pertinent pour l'étude de différents types de données. Prenons par exemple l'analyse de la qualité de l'air dans une région donnée [8]. Dans ce contexte, les nœuds d'un graphe spatio-temporel pourraient représenter des capteurs dispersés sur un territoire, tandis que les arêtes symboliseraient les corrélations entre les niveaux de pollution enregistrés dans différents

endroits à différents moments. La dimension temporelle permettrait ici de modéliser comment les niveaux de pollution évoluent d'une période à l'autre, en tenant compte des facteurs environnementaux tels que les conditions météorologiques ou les événements locaux (par exemple, les incendies de forêt ou les activités industrielles). En intégrant ces données dans un graphe spatio-temporel, il devient possible de détecter des motifs récurrents, tels que les zones régulièrement touchées par des pics de pollution ou les périodes critiques de l'année.

Cette approche permet non seulement de mieux comprendre les dynamiques locales, mais aussi d'anticiper les interactions futures entre les différents facteurs environnementaux. Dans cette thèse, les graphes spatio-temporels sont représentés sous forme de multigraphes, une structure permettant de modéliser plusieurs types de relations entre les entités à différents moments. Cela nous permet d'obtenir une vue d'ensemble plus fine et plus complète des systèmes complexes étudiés, en capturant non seulement l'évolution des interactions dans le temps, mais aussi la diversité des relations entre les entités.

Definition 15 *Un graphe spatio-temporel est un 4-tuple $G = (V, E, T, X)$, où : $V = \{v_1, \dots, v_n\}$ est un ensemble de sommets représentant des entités spatiales, $E = \{e_1, \dots, e_m\}$ est un ensemble d'arêtes représentant les relations ou interactions entre les sommets, chaque arête e_i reliant deux sommets v_j et v_k , $T = \{t_1, \dots, t_l\}$ est un ensemble d'intervalles de temps ou de timestamps, indiquant quand les interactions ou relations se produisent, $X = \{x_1, \dots, x_n\} \cup \{y_1, y_2, \dots, y_m\}$ est un ensemble d'attributs ou de propriétés associés aux sommets et aux arêtes, où x_i représente les attributs du sommet v_i , et y_i représente les attributs de l'arête e_i .*

2.2.1 Différentes structures de représentation des graphes spatio-temporels

Dans cette section, nous présentons différentes représentations de graphe spatio-temporel (graphe-ST) utilisées dans divers domaines et applications. Les graphes-ST sont des outils puissants pour aborder les applications nécessitant un raisonnement spatial et temporel. Leur dynamisme permet l'attribution de caractéristiques tant aux arêtes qu'aux nœuds.

Un composant essentiel d'une chaîne de traitement en apprentissage profond est constitué par les données d'entrée elles-mêmes. La modélisation d'un graphe spatio-temporel représente une tâche complexe, consistant à modéliser les connexions entre les nœuds en fonction du temps et de l'espace. Une représentation adéquate d'un graphe-ST doit exprimer avec précision les changements dans la structure topologique au fil du temps, ce qui permet un apprentissage plus rapide et plus efficace. De nombreux

facteurs peuvent influencer la structure de la représentation du graphe-ST ; par exemple, un graphe-ST représentant un réseau social [9] peut différer fondamentalement d'un graphe-ST modélisant les changements de connectivité fonctionnelle du cerveau à l'aide de signaux de résonance magnétique cardiaque (IRM) au repos [10].

Un autre facteur crucial est le dynamisme du graphe : dans un réseau social, par exemple, les connexions entre utilisateurs apparaissent et disparaissent [9], contrairement aux graphes de flux de trafic, où les connexions entre les routes sont, dans la plupart des cas, permanentes [11]. Par exemple, Lu *et al.* [12] construisent un graphe-ST cardiaque structurel basé sur la séquence de la structure cardiaque au cours de l'examen d'IRM, représentée par les coordonnées 2D des nœuds sur l'endocarde et l'épicarde. Dans chaque image d'IRM, les arêtes correspondent aux connexions entre les coordonnées.

Des informations supplémentaires peuvent être associées au graphe, comme dans [13], où une méthode pour une estimation précise de la durée de vie restante (RUL) a été développée. Les auteurs définissent le graphe-ST comme un ensemble $\mathcal{G}^t = (V, E, A^t)$; le graphe est construit à partir des données des capteurs d'un système industriel (le type de données des capteurs variant d'un système à l'autre), où A^t est une matrice d'adjacence dynamique, calculée à partir des données des capteurs et de la durée de vie restante estimée.

Dans un autre travail, Qi *et al.* [8] développent une méthode pour prédire les concentrations de différents polluants atmosphériques. Ils adoptent une définition similaire, où chaque nœud représente une station de surveillance à chaque instant t . Les relations spatiales existant entre différents emplacements peuvent être représentées sous la forme d'un graphe non orienté $G = (V, E, A)$ où, pour chaque temporalité t , $A \in \mathbb{R}^{N \times N}$ est la matrice de poids spatiaux, et chaque élément $A_{i,j}$ représente la corrélation spatiale quantitative entre v_i et v_j .

Yu *et al.* [14] définissent un graphe-ST comme $G = (V, E, W, X)$, où $W \in \mathbb{R}^{N \times N}$ est une matrice d'adjacence enregistrant la connectivité pondérée entre deux sommets, tandis que X est l'ensemble des caractéristiques des nœuds qui évoluent dans le temps, illustré dans la figure 2.2.

Dans le contexte des problèmes liés au trafic (prévision du trafic, disponibilité des places de stationnement, flux piétonnier, etc.), un graphe-ST est défini comme un graphe non orienté $G = (V, E, \mathbf{A})$. $\mathbf{A} \in \mathbb{R}^{N \times N}$ est la matrice d'adjacence du réseau de trafic G . \mathbf{A} peut représenter différents types de caractéristiques structurelles ; par exemple, dans [11], $A_{i,j}$ est défini à 1 si les segments de route v_i et v_j sont connectés, et à 0 sinon. Dans [15], la matrice \mathbf{A} représente la corrélation entre les nœuds pour chaque temporalité. Le poids de l'arête e_{ij} représente la force de corrélation entre v_i et v_j . Un poids plus élevé indique une corrélation plus forte entre les deux routes (figure 2.3). Les

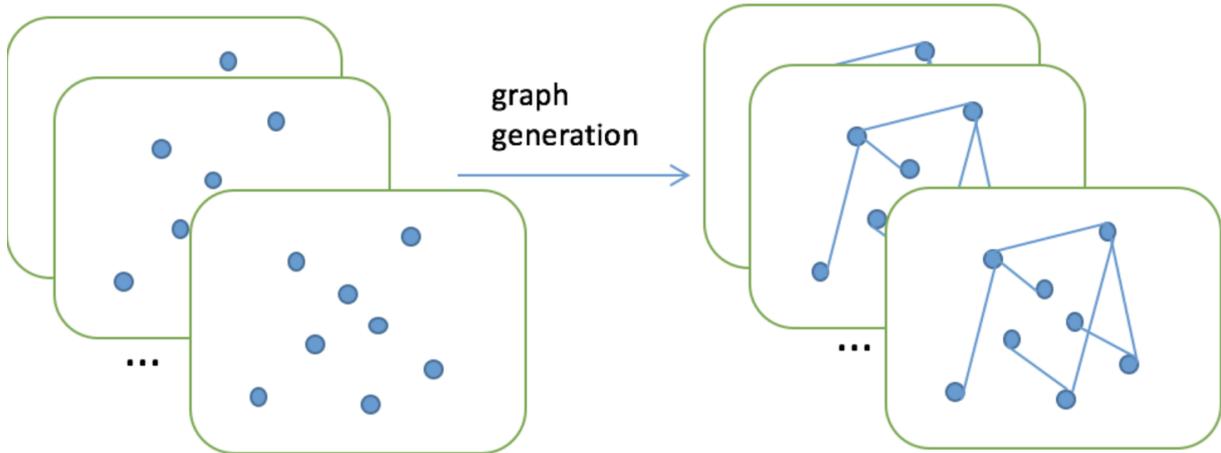


FIGURE 2.2 – La connectivité pondérée entre les sommets proposé par Yu *et al.* [14].

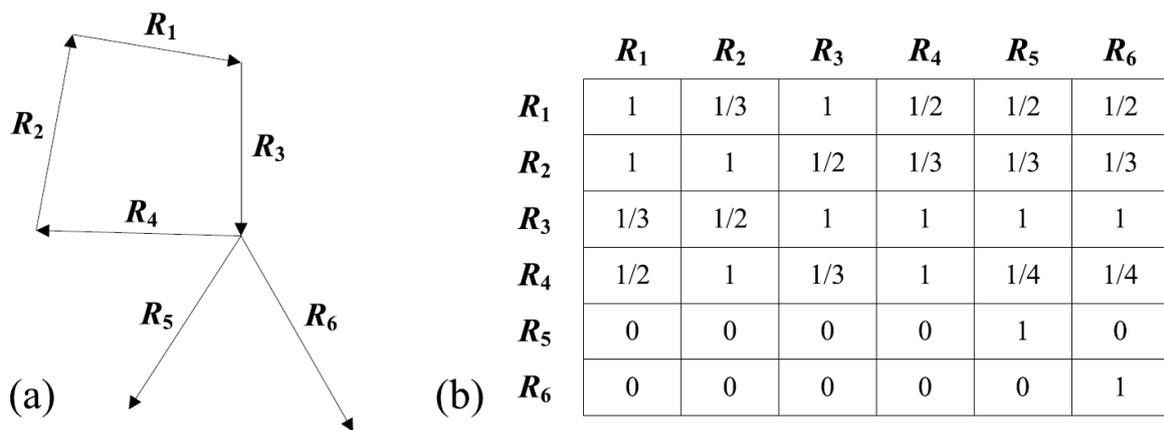


FIGURE 2.3 – Graphe-ST basé sur la corrélation proposé par Yu *et al.* [14].

valeurs de corrélation sont calculées après la construction de quatre graphes routiers, chacun représentant un aspect différent de la relation entre les nœuds (les valeurs de corrélation sont basées sur la structure topologique des routes, le modèle de trafic et les similitudes fonctionnelles de la zone locale).

Del Mundo *et al.* [16] introduisent une représentation de graphe spatio-temporel basée sur les multigraphes. Pour chaque pas de temps, les entités sont présentées comme N nœuds. Les auteues définissent trois graphes : un *graphe de relations spatiales*, qui représente les interactions spatiales entre les entités dans la même temporalité. Ces relations sont basées sur la théorie RCC8 [17] (que nous aborderons dans la section 2.2.2). Le second graphe est le *graphe de relations spatio-temporelles*, qui représente les interactions spatiales entre les entités à des temporalités successives, et utilise

également les relations RCC8. Enfin, un troisième graphe qui représente la transmission de l'identité entre les entités à différents moments, appelé *graphe de filiation*. Cette présentation tend à être plus expressive et permet d'ajouter des propriétés utiles.

Suivant la dernière définition, les graphes spatio-temporels prennent en compte un ensemble d'entités temporelles et leurs relations, incluant des relations sémantiques générales (spatiales, de corrélation, etc., principalement des relations spatiales dans notre contexte) et des relations de filiation. Les relations de filiation sont importantes pour suivre les entités dans le temps. Le modèle est limité à la contrainte qu'une relation lie deux entités au même instant ou à deux instants consécutifs, avec au plus une relation sémantique et une relation de filiation entre deux entités. Cependant, le modèle peut être généralisé pour accepter plusieurs, mais un nombre limité de relations sémantiques. Dans cette thèse, nous suivons la définition de Leborgne *et al.*[18] (qui s'inspire elle-même de la définition de Del Mundo *et al.*[16]) où un graphe-ST est défini sur un domaine temporel, $\mathcal{T} = t_1, t_2, \dots, t_n$, où t_i représente un instant donné avec une granularité spécifique et $t_i < t_{i+1}$ pour tous les $i \in [1, n]$. Δ est un ensemble d'entités, e_1, e_2, \dots, e_m . Nous introduisons également Σ , un ensemble de relations spatiales, et Φ , un ensemble de relations de filiation.

Definition 16 *Un graphe spatio-temporel \mathcal{G} est un triplet (V, E_Σ, E_Φ) , où V est un ensemble de sommets $(e_i, t_i) \in \Delta \times \mathcal{T}$, E_Σ est un ensemble de triplets $(T, (e_i, t_i), (e_j, t_j))$ où $(e_i, t_i), (e_j, t_j) \in V$, $t_i \leq t_j \leq t_{i+1}$, et $T \in \Sigma$, et E_Φ est un ensemble de triplets $(\rho, (e_i, t_i), (e_j, t_{i+1}))$ où $(e_i, t_i), (e_j, t_{i+1}) \in V$, et $\rho \in \Phi$.*

Ce modèle peut être vu comme l'union de trois sous-graphes (comme illustré dans la figure 2.4) dans lesquels les entités sont regroupées selon les instants :

- **Le sous-graphe des relations spatiales** représente les interactions spatiales entre les entités à un instant donné t_i . Dans ce travail, les relations spatiales proviennent de la théorie du Calcul des Connexions Régionales (RCC8)[19].
- **Le sous-graphe des relations spatio-temporelles** représente les interactions spatio-temporelles entre les entités à deux instants successifs. Dans ce travail, les relations spatio-temporelles proviennent également de la théorie RCC8.
- **Le sous-graphe des relations de filiation** représente la transmission de l'identité entre les entités à différents moments, deux types de relations de filiation sont considérés, la continuation γ et la dérivation δ . Une relation de continuation entre (v_i, t_i) et (e_j, t_{i+1}) signifie que v_i et v_j ont la même identité ; une relation de dérivation signifie que v_j contient une partie de l'identité de v_i .

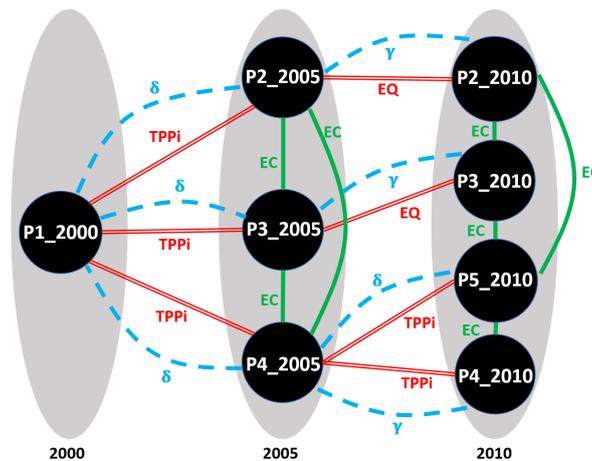


FIGURE 2.4 – Représentation d'un graphe-ST, les lignes rouges représentent les relations spatio-temporelles, les lignes vertes représentent les relations spatiales et les lignes bleues représentent les relations de filiation [18]

2.2.2 Calcul des connexions entre régions (RCC8)

Le Calcul des Connexions entre Régions (RCC) est un formalisme utilisé dans la représentation et le raisonnement spatial, développé à l'origine par Randell, Cui et Cohn en 1992 [19]. Le RCC8, un sous-ensemble du RCC, est particulièrement adapté au raisonnement spatial qualitatif, offrant un ensemble de huit relations de base qui décrivent les différentes manières dont deux régions peuvent être connectées. Cet ensemble de relations est exhaustif et mutuellement exclusif, ce qui en fait un outil puissant pour la représentation de la connaissance spatiale.

Relations RCC8

Le modèle RCC8 est un formalisme permettant de décrire les interactions spatiales qualitatives entre des régions. Il définit huit types de relations, comme la disjonction, l'inclusion ou le chevauchement, sans nécessiter de données précises sur les positions. Utilisé dans des domaines comme la géographie, RCC8 permet de raisonner sur les relations spatiales de manière qualitative. Les huit relations de base du RCC8 (Figure 2.5) sont les suivantes :

Déconnecté (DC) Les deux régions ne partagent aucun point ; elles sont complètement séparées.

Connecté Externe (EC) Les régions partagent un ou plusieurs points de frontière mais aucun point intérieur.

Égal (EQ) Les deux régions sont identiques, partageant tous les points de frontière et intérieurs.

Chevauchement Partiel (PO) Les régions se chevauchent, partageant certains points mais pas tous, et aucune n'est entièrement contenue dans l'autre.

Partie Propre Tangentielle (TPP) Une région est une partie propre tangentielle de l'autre, partageant certains points de frontière, mais leurs intérieurs ne se chevauchent pas complètement.

Partie Propre Tangentielle Inverse (TPPi) L'inverse de TPP ; une région est une partie propre tangentielle de l'autre.

Partie Propre Non Tangentielle (NTPP) Une région est une partie propre non tangentielle de l'autre, où une région est entièrement contenue dans l'autre sans partager de points de frontière.

Partie Propre Non Tangentielle Inverse (NTPPi) L'inverse de NTPP ; une région contient l'autre, et elles ne partagent pas de points de frontière.

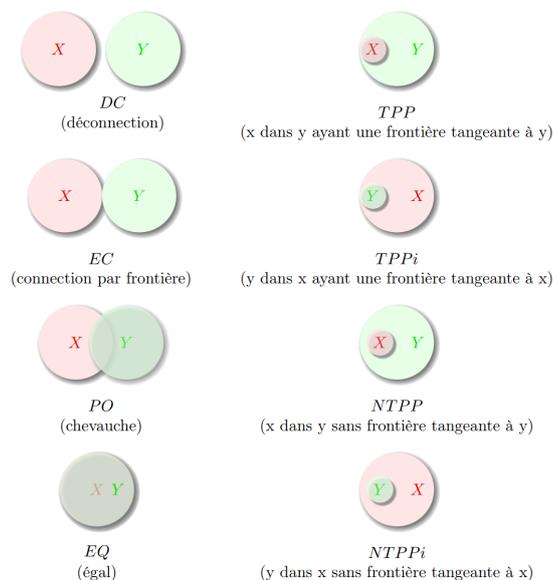


FIGURE 2.5 – Relations topologiques du RCC8 [16]

Application du RCC8 dans les graphes spatio-temporels

Dans le contexte des graphes spatio-temporels, les relations RCC8 fournissent un cadre formel pour décrire comment les configurations spatiales des entités évoluent dans le temps. En utilisant le RCC8, on peut modéliser des interactions spatiales complexes et des transitions, telles que :

- Modéliser et suivre le mouvement des régions géographiques et leurs interactions les unes par rapport aux autres au cours d'une série temporelle.

- Analyser les changements spatiaux et prédire les états futurs à partir des données historiques.
- Intégrer le raisonnement spatial qualitatif dans les processus décisionnels, particulièrement dans des domaines tels que l'urbanisme, la surveillance environnementale et la gestion des ressources.

Avec la montée en complexité des graphes, des outils plus sophistiqués sont requis pour traiter et extraire les informations pertinentes. Les réseaux de neurones conçus spécifiquement pour les graphes, appelés réseaux neuronaux de graphes, se sont imposés comme des méthodes puissantes pour analyser ces structures. En combinant la capacité des réseaux neuronaux de graphes à apprendre des représentations des nœuds et des motifs tout en tenant compte de la structure du graphe, nous pouvons traiter des réseaux de manière plus efficace et identifier des relations complexes qui seraient autrement difficiles à détecter. Nous allons maintenant nous pencher sur le rôle des réseaux neuronaux de graphes dans le traitement des graphes, et plus spécifiquement dans le cadre des graphes spatio-temporels.

2.3 Fondements des Réseaux de Neurones pour les graphes

2.3.1 Réseaux de neurones pour le traitement des graphes

Dans cette thèse, l'application des réseaux neuronaux de graphes (GNNs) joue un rôle clé dans le traitement des données spatio-temporelles. En permettant à la fois d'exploiter les relations locales à l'échelle des nœuds et les propriétés globales du graphe, les GNNs offrent une approche flexible et puissante pour analyser les graphes dans des contextes variés. Nous explorerons comment ces techniques peuvent être appliquées aux graphes spatio-temporels, en utilisant à la fois les caractéristiques des nœuds individuels et la structure globale du réseau pour effectuer des prédictions complexes.

Réseaux de neurones de graphes

Les réseaux de neurones de graphes (Graph Neural Networks, GNN) sont la forme de base de l'apprentissage profond des graphes. Ils sont également appelés réseaux graphiques récurrents. Un GNN est basé sur l'encodage de l'information structurelle du

graphe, où, pour chaque nœud v_i , on obtient récursivement un vecteur de représentation basse s_i . Un GNN applique la définition suivante [20] :

$$\mathbf{s}_i = \sum_{j \in \mathcal{N}(i)} \mathcal{F}(\mathbf{s}_i, \mathbf{s}_j, \mathbf{F}_i^V, \mathbf{F}_j^V, \mathbf{F}_{i,j}^E) \quad (2.1)$$

où $\mathcal{N}(i)$ est le voisinage du nœud v_i , \mathbf{F} sont les vecteurs de caractéristiques, et \mathcal{F} est la fonction à apprendre. \mathcal{F} est une fonction de propagation. La représentation finale h_i du nœud v_i peut être obtenue en appliquant une fonction de sortie O à la représentation basse s_i , définie comme suit :

$$\mathbf{h}_i = O(\mathbf{s}_i, \mathbf{F}_i^V) \quad (2.2)$$

où \mathbf{h}_i est la représentation finale du nœud v_i .

Réseau convolutionnel des graphes

Les réseaux de neurones convolutifs sur graphes (Graph Convolutional Networks, GCN) sont l'adaptation des Réseaux de Neurones Convolutifs (CNN) aux graphes. Similaire aux CNN, l'opération principale des GCN sont les convolutions. En raison de l'absence de structure en grille, l'opération de convolution ne peut pas être appliquée directement aux graphes. Nous pouvons définir deux types de convolution de graphe : spectrale et spatiale. La convolution spectrale effectue la convolution en transformant la représentation des nœuds dans l'espace spectral, en utilisant la transformée de Fourier du graphe, tandis que la convolution spatiale utilise le voisinage spatial. L'opération de convolution pour un graphe peut être définie par l'équation suivante :

$$\mathbf{u}_1 *_G \mathbf{u}_2 = \mathbf{Q}((\mathbf{Q}^T \mathbf{u}_1) \odot (\mathbf{Q}^T \mathbf{u}_2)) \quad (2.3)$$

où $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N$ sont deux vecteurs de nœuds et \mathbf{Q} sont les vecteurs propres de la matrice laplacienne du graphe \mathbf{L} .

Une couche convolutive est définie en appliquant différents filtres à différentes paires de signaux d'entrée-sortie comme suit Zhang22 :

$$\mathbf{u}_j^{l+1} = \rho \left(\sum_{i=1}^{f_l} \mathbf{Q} \Theta_{i,j}^l \mathbf{Q}^T \mathbf{u}_i^l \right) \quad j = 1, \dots, f_{l+1} \quad (2.4)$$

où l est la couche, $\mathbf{u}_j^l \in \mathbb{R}^N$ est la j^{th} représentation cachée pour les nœuds dans la $l^{ème}$ couche, $\Theta_{i,j}^l$ sont des filtres apprenables, et f_l est la dimensionnalité de la représentation cachée dans la $l^{ème}$ couche.

En utilisant les caractéristiques des nœuds \mathbb{F}^V comme couche d'entrée et en empilant plusieurs couches convolutives, l'architecture globale est similaire à celle d'un CNN. Suite à l'équation précédente, le nombre de paramètres apprenables est énorme. Comme solution, Chebnet a été proposé, en utilisant un filtre polynomial :

$$\Theta(\Lambda) = \sum_{k=0}^K \theta_k \Lambda^k \quad (2.5)$$

où $\theta_0, \dots, \theta_K$ sont les paramètres apprenables et K est l'ordre du polynôme. Pour résoudre des tâches centrées sur les graphes, une opération de lecture est nécessaire dans le GCN, afin d'agréger les informations des nœuds. De nombreuses méthodes statistiques ont été utilisées telles que la moyenne et la somme, tandis que d'autres travaux utilisent le regroupement hiérarchique ou des approches plus sophistiquées comme les réseaux Long-Short Term Memory (LSTM).

Réseaux attentionnels de graphes

Les réseaux attentionnels de graphes (Graph Attention Networks, GAT) utilisent le mécanisme d'attention pour pondérer différemment les contributions des nœuds voisins lors de l'agrégation des informations. Un GAT utilise des poids d'attention appris pour chaque paire de nœuds connectés, permettant ainsi une agrégation adaptative des informations. La définition formelle de GAT est donnée par :

$$\mathbf{h}'_i = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W} \mathbf{h}_j \right) \quad (2.6)$$

où \mathbf{h}'_i est la nouvelle représentation du nœud v_i , α_{ij} est le poids d'attention pour l'arête (i, j) , \mathbf{W} est la matrice de poids, et σ est une fonction d'activation. Les poids d'attention α_{ij} sont calculés en utilisant un mécanisme d'attention, tel que :

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W} \mathbf{h}_i | \mathbf{W} \mathbf{h}_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W} \mathbf{h}_i | \mathbf{W} \mathbf{h}_k]))} \quad (2.7)$$

où \mathbf{a} est un vecteur de poids d'attention, et $|$ représente la concaténation.

GraphSAGE

GraphSAGE (SAmple and aggreGatE) est un modèle d'apprentissage des plongements de nœuds dans des données structurées en graphe. Étant donné un graphe $G = (V, E)$

avec N nœuds et M arêtes, GraphSAGE apprend une fonction $f : V \rightarrow \mathbb{R}^d$ qui associe à chaque nœud $v \in V$ un vecteur de plongement de dimension d . L'algorithme GraphSAGE se compose de deux éléments principaux : une fonction d'agrégation de voisinage et une fonction entraînable. La fonction d'agrégation de voisinage prend en entrée un nœud v et son voisinage local $N(v)$, et agrège les informations des voisins pour produire une représentation vectorielle de taille fixe du voisinage. La fonction entraînable combine ensuite la représentation du voisinage avec le vecteur de caractéristiques du nœud courant pour produire le plongement final. Soit $h_v^{(0)}$ le vecteur de caractéristiques initial du nœud v . La fonction d'agrégation de voisinage prend la forme :

$$h_{N(v)}^{(k)} = \text{AGGREGATE}_k(\{h_u^{(k-1)} \mid u \in N(v)\})$$

où AGGREGATE_k est une fonction d'agrégation différentiable qui combine les vecteurs de caractéristiques des voisins dans le voisinage à k sauts de v . La fonction entraînable prend la forme :

$$h_v^{(k)} = \sigma(W^{(k)}[h_v^{(k-1)} \parallel h_{N(v)}^{(k)}])$$

où $W^{(k)}$ est une matrice de poids entraînable, \parallel est l'opérateur de concaténation, et σ est une fonction d'activation. GraphSAGE peut être entraîné de bout en bout en utilisant une fonction de perte qui compare les plongements de nœuds prédits aux vraies étiquettes pour une tâche donnée. Le modèle a démontré des performances à l'état de l'art sur diverses tâches basées sur les graphes, notamment la classification de nœuds et la prédiction de liens.

AutoEncodeurs de Graphes (AE)

Les autoencodeurs sont des modèles d'apprentissage profond non supervisés qui ont une large gamme d'utilité dans le domaine du traitement d'images, de la vision par ordinateur et de la réduction de dimensionnalité, etc. [21]. Les autoencodeurs et leurs variantes peuvent être utilisés pour l'apprentissage des encodages de graphes et de nœuds. Les AE peuvent apprendre des encodages de nœuds de faible dimension en réduisant la dimensionnalité de la matrice d'adjacence. Un simple AE applique une perte de reconstruction L2 sur une matrice d'entrée, la perte de reconstruction L2 définie comme suit :

$$\begin{aligned} \min_{\Theta} \mathcal{L}_2 &= \sum_{i=1}^N \|\mathbf{P}(i, :) - \hat{\mathbf{P}}(i, :)\|_2 \\ \hat{\mathbf{P}}(i, :) &= \mathcal{G}(\mathbf{h}_i), \mathbf{h}_i = \mathcal{F}(\mathbf{P}(i, :)) \end{aligned} \quad (2.8)$$

où \mathbf{P} est la matrice de transition (latente), $\hat{\mathbf{P}}$ est la matrice reconstruite, $\mathbf{h}_i \in \mathbb{R}^d$ est la représentation de faible dimension du nœud v_i , $\mathcal{F}(\cdot)$ est l'encodeur, et $\mathcal{G}(\cdot)$ est le décodeur.

Réseaux d'Isomorphisme de Graphes

Les Réseaux d'Isomorphisme de Graphes (Graph Isomorphism Networks, GIN) Les Réseaux d'Isomorphisme de Graphes (GINs) sont un type de réseau neuronal conçu pour apprendre des plongements de nœuds pour des données structurées en graphe. Étant donné un graphe $G = (V, E)$ avec N nœuds et M arêtes, les GINs apprennent une fonction $f : V \rightarrow \mathbb{R}^d$ qui associe à chaque nœud $v \in V$ un vecteur de plongement de dimension d . L'algorithme GIN consiste en une série d'étapes de passage de messages, où l'information est agrégée à partir des nœuds voisins pour produire une représentation vectorielle de taille fixe pour chaque nœud. L'idée clé derrière les GINs est d'utiliser une fonction récursive pour calculer les plongements des nœuds, où la fonction est appliquée aux vecteurs de caractéristiques agrégés à chaque itération. Pour $h_v^{(0)}$ le vecteur de caractéristiques initial du nœud v , l'étape de passage de message peut s'écrire :

$$h_v^{(k)} = MLP^{(k)} \left((1 + \epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)} \right)$$

où $MLP^{(k)}$ est un perceptron multicouche (MLP) avec L couches, $\epsilon^{(k)}$ est un scalaire apprenable. GIN est conçu pour des tâches de prédiction au niveau du graphe et apprend l'information structurelle d'un graphe de manière invariante par permutation. Les plongements des nœuds peuvent être agrégés pour obtenir une représentation au niveau du graphe comme suit :

$$h_G = \text{CONCAT}(\text{READOUT}(h_v \mid v \in G)) \quad (2.9)$$

où h_G est la représentation finale au niveau du graphe et K est le nombre de couches dans le modèle GIN et l'opération READOUT peut être une somme ou une opération de Pooling. Une approche courante est d'utiliser une simple fonction d'agrégation invariante par permutation telle que la somme ou la moyenne comme suit :

$$h_G = \sum_{v \in G} h_v^{(K)}$$

Mécanismes d'Auto-Attention et Transformer de Graphes

Les mécanismes d'auto-attention sont utilisés pour pondérer l'importance de différentes parties d'une séquence d'entrée en reliant différentes positions de la séquence entre

elles. Étant donné une séquence de longueur N , les mécanismes d'auto-attention calculent un ensemble de vecteurs de requête, clé et valeur de dimension N . Une tête d'auto-attention peut alors être définie comme une fonction qui prend ces vecteurs en entrée et produit un ensemble de vecteurs de contexte de dimension N qui sont des moyennes pondérées des vecteurs de valeur, les poids étant déterminés par le produit scalaire entre les vecteurs de requête et les vecteurs de clé. Cela peut être exprimé mathématiquement comme suit :

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

où Q , K et V sont les vecteurs de requête, clé et valeur, respectivement, et d_k est la dimension des vecteurs de clé. La fonction softmax est appliquée pour normaliser les poids en une distribution de probabilité. Le produit scalaire est mis à l'échelle par $\sqrt{d_k}$ pour réduire la variance dans les poids. Les mécanismes d'auto-attention dans l'apprentissage profond sur graphes et les transformers de graphes permettent de modéliser les interactions entre les nœuds d'un graphe. Étant donné un ensemble de nœuds V et leurs vecteurs de caractéristiques correspondants \mathbf{x}_i , le mécanisme d'auto-attention calcule un vecteur de poids \mathbf{a}_i pour chaque nœud i comme suit :

$$\mathbf{a}_i = \text{softmax} \left(\frac{\mathbf{x}_i \mathbf{W}_{\text{query}} (\mathbf{X} \mathbf{W}_{\text{key}})^T}{\sqrt{d_k}} \right)$$

où \mathbf{X} est la matrice de tous les vecteurs de caractéristiques des nœuds, $\mathbf{W}_{\text{query}}$ et \mathbf{W}_{key} sont des matrices de projection apprises, et d_k est la dimension des vecteurs de clé. Le vecteur de poids \mathbf{a}_i indique l'importance de chaque nœud j dans le calcul de la représentation du nœud i , et est utilisé pour calculer une somme pondérée des vecteurs de caractéristiques :

$$\mathbf{z}_i = \sum_{j \in V} \mathbf{a}_{i,j} \mathbf{x}_j$$

où $\mathbf{a}_{i,j}$ est le poids d'attention entre les nœuds i et j . Le résultat \mathbf{z}_i est utilisé comme entrée pour les couches suivantes dans le réseau neuronal de graphes.

Réseaux de Neurones Récurrents pour l'Apprentissage Profond sur les Graphes

Les réseaux de neurones récurrents (RNN) sont un modèle largement utilisé qui incorpore une mémoire pour prendre en compte toute information des entrées antérieures afin d'influencer l'entrée et la sortie actuelles. Le RNN est un outil puissant pour attribuer un contexte à la sortie du modèle. Les RNN sont particulièrement adaptés aux données

dépendantes du temps, telles que les données séquentielles ou les séries temporelles. Parmi les exemples de RNN, on trouve à la fois l'Unité Récurrente à Portes (GRU) et la Mémoire à Long Court Terme (LSTM). La principale différence entre GRU et LSTM est que GRU est moins complexe que LSTM ; GRU a deux portes (réinitialisation et mise à jour), tandis que LSTM en a trois (entrée, sortie et oubli). Étant donné la nature des graphes spatio-temporels représentant des relations temporelles, les RNN ont été largement incorporés dans les cadres d'apprentissage profond sur les graphes spatio-temporels pour résoudre des problèmes de prédiction et de prévision.

2.4 Apprentissage par renforcement

Alors que les réseaux de neurones et les méthodes d'apprentissage profond se sont révélés extrêmement efficaces pour capturer des représentations complexes à partir des données, certaines problématiques nécessitent des approches plus dynamiques, capables d'apprendre à partir d'interactions séquentielles avec un environnement. C'est là que l'apprentissage par renforcement prend tout son sens. Contrairement aux méthodes traditionnelles d'apprentissage supervisé qui reposent sur des ensembles de données fixes et des labels, l'apprentissage par renforcement permet à un agent d'explorer un environnement, de prendre des actions et d'apprendre de manière itérative en fonction des récompenses reçues. Nous allons maintenant explorer comment cette approche s'applique aux graphes, notamment dans des scénarios où les agents doivent découvrir des structures ou motifs optimaux à travers une exploration active du réseau.

2.4.1 Notions de Base

L'apprentissage par renforcement (AR) est une branche de l'apprentissage automatique qui se concentre sur la manière dont un agent peut apprendre à prendre des décisions optimales en interagissant avec un environnement dynamique et potentiellement incertain. Contrairement à l'apprentissage supervisé, où un modèle est entraîné sur des données étiquetées, l'AR repose sur un mécanisme d'apprentissage par essais et erreurs, guidé par des récompenses reçues suite aux actions entreprises en réponse aux observations de l'environnement.

Agent et Environnement

- **Agent** : L'agent est une entité autonome qui prend des décisions en effectuant des actions. Il peut s'agir d'un robot, d'un programme informatique ou de tout système capable d'apprendre à partir de l'interaction avec l'environnement. L'agent cherche

à maximiser une fonction de récompense en adaptant sa politique de décision en fonction des expériences passées.

- **Environnement** : L'environnement représente tout ce qui est extérieur à l'agent et avec lequel il interagit. Il réagit aux actions de l'agent en modifiant son état et en fournissant des observations et des récompenses à l'agent. L'environnement peut être déterministe ou stochastique, stationnaire ou non stationnaire, pleinement observable ou partiellement observable.

Observations ($o \in \mathcal{O}$) et États ($s \in \mathcal{S}$)

- **État** (s) : L'état est une description complète de la situation courante de l'environnement. Il englobe toutes les informations nécessaires pour prédire le comportement futur de l'environnement (sous l'hypothèse de la propriété de Markov 10.5555/3279266). L'ensemble des états possibles est noté \mathcal{S} .
- **Observation** (o) : Dans de nombreux cas, l'agent n'a pas accès à l'état complet de l'environnement mais perçoit une observation o , qui peut être une version partielle ou bruitée de l'état réel. L'ensemble des observations possibles est noté \mathcal{O} .

Actions ($a \in \mathcal{A}$)

- **Action** (a) : L'action est le choix effectué par l'agent à un instant donné pour influencer l'état de l'environnement. L'ensemble des actions possibles est noté \mathcal{A} . Les actions peuvent être :
 - **Discrètes** : L'agent choisit parmi un ensemble fini d'actions (par exemple, déplacer un personnage à gauche, à droite, sauter).
 - **Continues** : L'agent choisit des valeurs dans un espace continu (par exemple, la force à appliquer sur un robot, la vitesse d'un véhicule).

Récompenses ($r \in \mathbb{R}$)

- **Récompense** (r) : La récompense est un signal numérique reçu par l'agent après l'exécution d'une action dans un état donné. Elle représente le feedback de l'environnement sur la performance de l'agent par rapport à l'objectif à atteindre. La fonction de récompense est essentielle car elle définit ce que l'agent doit apprendre à optimiser.

Caractéristiques des récompenses :

- **Immédiates vs Différées** : Les récompenses peuvent être reçues immédiatement après une action ou différées dans le temps.

- **Positives, Négatives ou Nulles** : Les récompenses peuvent encourager (positives), décourager (négatives) ou être neutres (nulles).

Le cycle d'interaction se déroule généralement comme suit :

1. **Observation** : L'agent perçoit une observation o_t de l'état courant s_t de l'environnement.
2. **Action** : En se basant sur cette observation, l'agent choisit et exécute une action a_t .
3. **Transition** : L'environnement passe à un nouvel état s_{t+1} en réponse à l'action de l'agent.
4. **Récompense** : L'agent reçoit une récompense r_{t+1} associée à la transition.

Ce processus se répète à chaque pas de temps t , et l'objectif de l'agent est d'apprendre une politique qui maximise la récompense cumulée sur le long terme.

Interaction Agent-Environnement L'interaction entre l'agent et l'environnement se déroule selon un cycle itératif à chaque pas de temps t :

1. **Perception** : L'agent reçoit une observation o_t de l'état courant s_t de l'environnement. Cette observation peut être influencée par des bruits ou être incomplète.
2. **Décision** : En se basant sur sa politique π et l'observation reçue, l'agent sélectionne une action a_t . La politique peut être :
 - **Déterministe** : $\pi(o_t) = a_t$, l'action est une fonction déterministe de l'observation.
 - **Stochastique** : $\pi(a_t|o_t) = P[a_t|o_t]$, l'action est choisie selon une distribution de probabilité conditionnée sur l'observation.
3. **Action** : L'agent exécute l'action a_t , influençant l'état de l'environnement.
4. **Transition** : L'environnement évolue vers un nouvel état s_{t+1} selon la dynamique $P(s_{t+1}|s_t, a_t)$, qui peut être connue ou inconnue de l'agent.
5. **Récompense** : L'agent reçoit une récompense r_{t+1} en fonction de l'état précédent, de l'action entreprise et du nouvel état.
6. **Mise à jour** : L'agent utilise l'expérience $(o_t, a_t, r_{t+1}, o_{t+1})$ pour mettre à jour sa politique ou ses estimations de valeur afin d'améliorer ses décisions futures.

Ce processus (démonstré dans la figure 2.6) se répète à chaque pas de temps, formant une séquence ou un épisode d'interactions. L'objectif ultime de l'agent est d'apprendre une politique optimale π^* qui maximise la valeur espérée des récompenses cumulées futures.

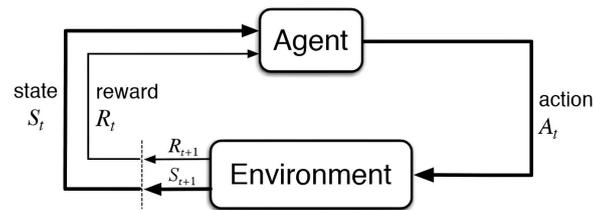


FIGURE 2.6 – Diagramme de base de l'apprentissage par renforcement

Processus de Décision Markovien (PDM) Pour modéliser mathématiquement cette interaction, on utilise souvent un Processus de Décision Markovien, défini par le tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, où :

- \mathcal{S} : Ensemble des états possibles.
- \mathcal{A} : Ensemble des actions possibles.
- $P(s' | s, a)$: Probabilité de transition vers l'état s' après avoir pris l'action a dans l'état s .
- $R(s, a)$: Récompense espérée en prenant l'action a dans l'état s .
- $\gamma \in [0, 1]$: Facteur d'escompte déterminant l'importance des récompenses futures.

L'agent cherche à apprendre une politique optimale π^* qui maximise la valeur espérée des récompenses cumulées :

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \right]$$

2.4.2 Apprentissage par renforcement sur les graphes

L'apprentissage par renforcement (Reinforcement Learning, RL) est un type d'apprentissage automatique où un agent apprend à prendre des décisions en effectuant des actions dans un environnement et en recevant des retours par le biais de récompenses. Cette approche est particulièrement efficace dans des scénarios où une supervision explicite est indisponible, et les décisions doivent être prises de manière séquentielle. Les avancées récentes étendent le RL aux données structurées en graphes, permettant des applications novatrices dans des réseaux complexes tels que les réseaux sociaux, les systèmes de trafic et les structures moléculaires.

Dans l'apprentissage par renforcement basé sur les graphes, l'environnement est représenté sous forme de graphe où les nœuds et les arêtes contiennent des informations pertinentes pour la tâche d'apprentissage. L'agent interagit avec le graphe, prenant des décisions à chaque étape qui modifient son état. Par exemple, dans un réseau social,

un agent peut décider quelles connexions recommander pour améliorer l'engagement des utilisateurs, ou dans un graphe chimique, il pourrait explorer des combinaisons d'éléments pour découvrir de nouveaux matériaux.

Le principal défi de l'application du RL aux graphes est de définir l'espace d'état, l'espace d'action et la fonction de récompense de manière à capturer la structure relationnelle et la dynamique du graphe. L'état peut être défini comme la configuration actuelle du graphe, incluant les caractéristiques des nœuds et les attributs des arêtes. Les actions pourraient impliquer l'ajout ou la suppression de nœuds ou d'arêtes, et la fonction de récompense doit quantifier l'impact de ces actions pour atteindre l'objectif global.

2.4.3 Apprentissage par renforcement profond sur les graphes

L'apprentissage par renforcement profond (Deep Q Learning, DQN) intègre l'apprentissage profond avec l'apprentissage par renforcement, une méthode de RL basée sur la valeur, où un agent apprend à estimer la qualité (valeur Q) des actions prises dans des états particuliers. Appliqué aux graphes, l'apprentissage par renforcement profond implique l'entraînement d'un réseau de neurones profond, typiquement un réseau neuronal de graphes (GNN), pour approximer les valeurs Q . Le GNN prend l'état du graphe comme entrée et produit des valeurs Q pour toutes les actions possibles dans cet état. Le processus d'entraînement implique :

- D'observer l'état actuel du graphe.
- De sélectionner et d'effectuer une action basée sur la politique dérivée des valeurs Q (souvent une politique ϵ -gloutonne).
- D'observer la récompense et le nouvel état résultant de l'action.
- De mettre à jour les estimations des valeurs Q en fonction de la récompense observée et de la valeur Q maximale de l'état suivant, en utilisant l'équation de Bellman :

$$Q_{\text{new}}(s, a) = Q(s, a) + \alpha \left[r + \gamma \max_{a'} (Q(s', a') - Q(s, a)) \right] \quad (2.10)$$

où α est le taux d'apprentissage, γ est le facteur d'actualisation, r est la récompense, s, a sont l'état et l'action courants, et s', a' sont le nouvel état et les actions potentielles.

Cette approche permet à l'agent d'apprendre des stratégies optimales pour manipuler des données structurées en graphes au fil du temps. Par exemple, dans un problème de routage sur un graphe de trafic, l'apprentissage par renforcement profond peut aider à apprendre des politiques qui minimisent le temps de voyage et évitent la congestion.

2.5 Conclusion

Dans ce chapitre, nous avons parcouru les concepts fondamentaux nécessaires à l'analyse et au traitement des graphes complexes. Nous avons commencé par introduire les structures de base des graphes, en mettant un accent particulier sur les graphes spatio-temporels et leur importance pour modéliser des systèmes évoluant à la fois dans l'espace et dans le temps. Ensuite, nous avons approfondi l'exploration de la recherche des motifs fréquents dans ces graphes, une tâche clé pour révéler des structures récurrentes et significatives dans des réseaux complexes. En particulier, nous avons donné une définition précise de la recherche des motifs fréquents dans les graphes et les multigraphes, en expliquant la problématique spécifique abordée dans cette thèse et son importance.

Nous avons ensuite abordé l'importance des réseaux neuronaux pour les graphes, qui permettent d'apprendre des représentations efficaces des nœuds et des graphes entiers, tout en capturant à la fois les propriétés locales et globales des réseaux. Enfin, nous avons exploré l'apprentissage par renforcement, et plus spécifiquement son application aux graphes, ainsi que l'apprentissage par renforcement profond, une approche qui permet aux agents d'apprendre des stratégies optimales à travers des interactions itératives avec le graphe.

Ainsi, ce chapitre a fourni une vue d'ensemble complète des outils théoriques nécessaires pour aborder les défis posés par la recherche des motifs fréquents dans des graphes spatio-temporels, tout en posant les bases pour les développements plus avancés que nous explorerons dans les chapitres suivants.

3

Etat de l'art

Sommaire

3.1	Introduction	53
3.2	Applications de l'apprentissage profond sur les graphes spatio-temporels	53
3.2.1	Imagerie médicale	54
3.2.2	Environnement	56
3.2.3	Reconnaissance d'activités et de mouvements	58
3.2.4	Tâches liées au trafic routier	61
3.2.5	Applications diverses	67
3.3	Exploration des Graphes avec l'Apprentissage par Renforcement . .	73
3.3.1	Applications Réelles avec l'Apprentissage par Renforcement sur les Graphes	75
3.4	Discussion	77
3.5	Recherche des motifs fréquents dans les graphes	80

3.1 Introduction

Dans ce chapitre, nous explorerons comment l’apprentissage profond (apprentissage profond, DL) et l’apprentissage par renforcement (Reinforcement Learning, RL) révolutionnent l’analyse de ces graphes. Ces méthodes offrent des outils puissants pour extraire des structures complexes et dynamiques à partir des données qui évoluent. L’apprentissage profond, avec sa capacité à apprendre des représentations hiérarchiques, et l’apprentissage par renforcement, axé sur la prise de décision et le développement de politiques (ou stratégie) dans des environnements dynamiques, sont particulièrement prometteurs pour ces applications. Finalement, nous aborderons le problème spécifique de l’extraction de motifs fréquents dans les graphes spatio-temporels. Les motifs, qui sont des sous-graphes récurrents et significatifs, révèlent des schémas communs pouvant permettre de prédire des comportements ou d’identifier des anomalies. Nous discuterons des méthodes actuelles disponibles pour l’extraction de motifs fréquents, tout en soulignant les défis associés à cette tâche, tels que la complexité computationnelle et la nature dynamique des graphes spatio-temporels. Ce chapitre vise à fournir une compréhension approfondie de l’intersection entre les technologies d’apprentissage avancées et leur application aux graphes spatio-temporels, ouvrant la voie à des solutions innovantes dans des systèmes complexes et dynamiques.

3.2 Applications de l’apprentissage profond sur les graphes spatio-temporels

Les graphes ont longtemps été utilisés pour représenter des données dans divers domaines tels que les sciences sociales, l’imagerie médicale et la biologie. Les graphes spatio-temporels (graphes-ST) offrent une expressivité accrue en permettant de modéliser les changements des entités et des événements au fil du temps en intégrant des informations spatiales et temporelles. Bien que la plupart des applications de l’apprentissage profond sur les graphes-STs soient principalement axées sur les tâches de prévision [22], l’apprentissage profond est également utilisé pour résoudre des

problèmes de classification et de catégorisation basés sur les motifs spatio-temporels appris à partir des graphe-STs [23]. Dans ce chapitre, nous explorons les différents domaines d’application de l’apprentissage profond sur les graphe-STs.

3.2.1 Imagerie médicale

Les modèles d’apprentissage profond ont été largement explorés dans le domaine de l’imagerie médicale pour étudier la structure et la fonction du corps, ainsi que pour la détection précoce des maladies [24]. Les applications incluent l’analyse des données d’images cliniques, radiologiques et pathologiques [25]. Récemment, certains travaux ont examiné l’utilisation des graphe-STs pour représenter des données non structurées en grille, comme les enregistrements physiologiques souvent irréguliers et désordonnés. Par exemple, Azevedo *et al.* [23] ont exploré la possibilité de prédire les résultats et de classer les patients en utilisant des données cérébrales, principalement des données d’IRMf représentées sous forme de graphe-ST. Les auteurs construisent le graphe-ST où les relations représentent les valeurs de corrélation à travers les séries temporelles entre les régions du cerveau (nœuds). Pour chaque temporalité, les auteurs définissent un ensemble de matrices $N \times T$, où T est la taille de la série temporelle et N le nombre de régions cérébrales enregistrées. Au lieu de s’appuyer uniquement sur des modèles basés sur des graphes, ils calculent d’abord les plongements pour chaque nœud en utilisant des couches de convolution 1D, suivies d’une couche GCN pour transformer les caractéristiques de chaque nœud en fonction des informations transmises par ses voisins. Une représentation finale du graphe est obtenue en utilisant un pooling hiérarchique, et finalement, ils emploient un ensemble de transformations linéaires pour générer la prédiction finale.

De même, Gadgil *et al.* [10] utilisent les graphes-ST pour représenter le signal BOLD (Blood Oxygen Level Dependent, qui représente les changements dans les niveaux d’oxygénation du sang, utilisés comme mesure indirecte de l’activité neuronale dans le cerveau) de l’IRMf en état de repos. Un graphe-ST a été créé en supposant que l’activation spontanée dans les N régions d’intérêt (ROI) peut être quantifiée par la moyenne des séries temporelles BOLD au sein de la région pour T temporalités. Le signal $\mathbf{f} \in \mathbb{R}^{N \times T}$ est alors défini sur un graphe ST non orienté $\mathcal{G} = (V, E)$, où V est l’ensemble des nœuds et E contient les arêtes caractérisant les connexions temporelles et spatiales. Chaque arête dans le graphe temporel relie une ROI à la même ROI au point temporel suivant. Chaque arête dans le graphe spatial (associé à un point temporel) relie deux ROI conformément à l’organisation fonctionnelle du cerveau. Ils définissent ensuite leur opération de convolution graphe-ST en considérant à la fois les voisinages spatiaux et temporels, et utilisent un noyau de convolution pour normaliser les caractéristiques des nœuds. Le processus est présenté dans la figure 3.1.

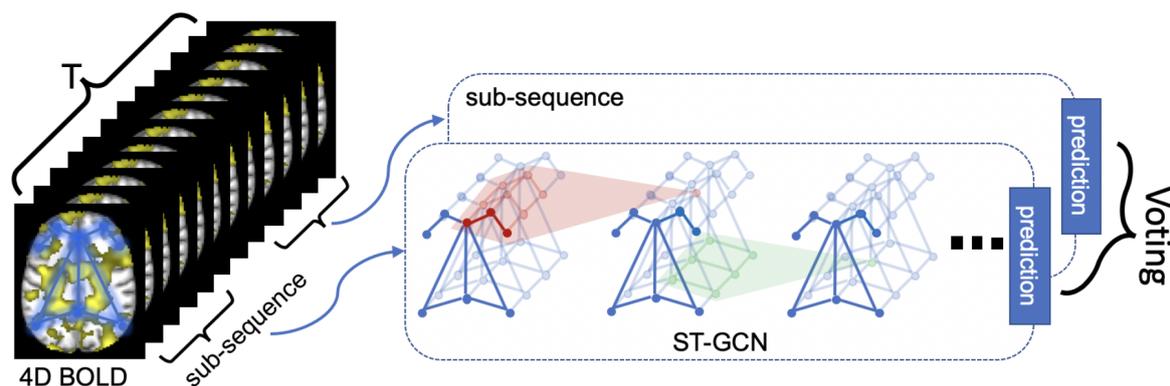


FIGURE 3.1 – Pipeline du réseau de convolution de graphes spatio-temporels utilisé par [10]

Par ailleurs, Tiago *et al.* [26] proposent un modèle utilisant un bloc de GNN pour prendre en compte les interrelations spatiales entre les régions du cerveau et des réseaux de convolution temporelle pour capturer les dynamiques intra-temporelles des séries temporelles BOLD. Ils soutiennent que l’utilisation des GNN élimine la limitation de supposer la linéarité dans les interactions entre les séries temporelles spécifiques à chaque région du cerveau en capturant les interactions de ordre plus élevé entre les régions d’intérêt. Le modèle utilise une architecture TCN pour extraire une représentation de plongement de dimension inférieure des séries temporelles IRMf (dans l’état de repos) pour chaque nœud.

Enfin, Lu *et al.* [12] proposent une approche dynamique de réseau de convolution sur graphes spatio-temporels (DST-GCN) pour apprendre les motifs de mouvement du ventricule gauche à partir d’images d’IRM cardiaque. Ils formulent l’estimation du mouvement cardiaque comme un problème de prévision qui estime les trajectoires futures de mouvement en se basant sur la structure cardiaque historique. La séquence de structures cardiaques est représentée par les coordonnées 2D des nœuds sur l’endocarde et l’épicarde dans chaque trame d’IRM cardiaque. En utilisant des convolutions de graphes spatiaux suivies de convolutions temporelles 1D, ils construisent un graphe-ST non orienté pour modéliser les trajectoires de mouvement cardiaque.

Une étude a proposé un nouveau cadre de réseau de convolution spatio-temporel pour identifier les biomarqueurs de connectivité fonctionnelle pour la maladie d’Alzheimer [27]. Des données IRMf sont utilisées pour construire un réseau dynamique de connectivité fonctionnelle du cerveau. Le réseau de convolution spatio-temporel (STGC-GCAM) capture à la fois les informations temporelles et spatiales des données IRMf, permettant l’extraction de caractéristiques détaillées qui aident au diagnostic précoce de la maladie d’Alzheimer. L’architecture du modèle comprend une convolution spatiale dans le domaine spectral et des couches de convolution temporelle, ce

qui aide à identifier les régions cérébrales pertinentes associées à Alzheimer et ses stades. Cette approche a été validée sur des données provenant de plusieurs cohortes, démontrant son efficacité dans les tâches de diagnostic précoce.

Une autre étude a introduit le Réseau de Neurones Spatio-Temporel Fonctionnel et Efficace (FE-STGNN) pour diagnostiquer les troubles cognitifs légers (MCI) [28]. Ce modèle intègre des mesures de connectivité fonctionnelle et efficace à partir de données IRMf pour construire des graphes spatio-temporels. Le modèle FE-STGNN utilise des couches de convolution de graphes pour capturer les dépendances spatiales et des couches de convolution temporelle pour comprendre les dynamiques temporelles dans le cerveau. Le modèle a montré des résultats prometteurs dans le diagnostic précis de MCI en exploitant l’information combinée de connectivité, ce qui améliore la compréhension des interactions complexes du cerveau.

Ces exemples montrent comment les graphe-STs et l’apprentissage profond peuvent être combinés pour extraire des motifs complexes et pertinents des données médicales, offrant des perspectives nouvelles pour le diagnostic et la recherche biomédicale.

3.2.2 Environnement

Les graphes spatio-temporels sont un outil puissant pour modéliser les données environnementales et géographiques. En plus de leur expressivité des relations topologiques, les graphe-STs peuvent représenter les changements au fil du temps dans les zones étudiées et permettent de surveiller les changements sur les cartes [29]. Dans cette section, nous examinons les cas d’utilisation de l’apprentissage profond sur les graphe-STs dans le contexte des données environnementales et géographiques.

Khodayar *et al.* [30] ont présenté un cadre pour modéliser et prévoir la vitesse du vent à court terme en utilisant l’apprentissage profond. Ils ont d’abord obtenu des séries temporelles de la vitesse et de la direction du vent provenant de fermes éoliennes environnantes, chaque ferme étant représentée comme un nœud. Les arêtes sont établies en définissant des relations de voisinage à l’aide de l’information mutuelle [31] entre les données historiques de tous les nœuds, similaire à une méthode de calcul de corrélation. L’étape suivante consiste en l’extraction des caractéristiques temporelles à l’aide d’un modèle supervisé de réseau Long Short Term Memory (LSTM) récurrent. Les caractéristiques sont ensuite passées à huit couches de convolution spatiale, exploitant la théorie des ensembles approximatifs [32] pour extraire des caractéristiques spatio-temporelles significatives.

Dans une autre étude, Qi *et al.* [8] ont étudié la qualité de l’air à partir de stations de surveillance au sol. Les auteurs proposent un modèle d’apprentissage profond intégrant des GCNx et des réseaux LSTM pour prévoir la variation spatio-temporelle des particules fines ($PM_{2.5}$, particules de diamètre 2.5 microns ou moins). Ils construisent

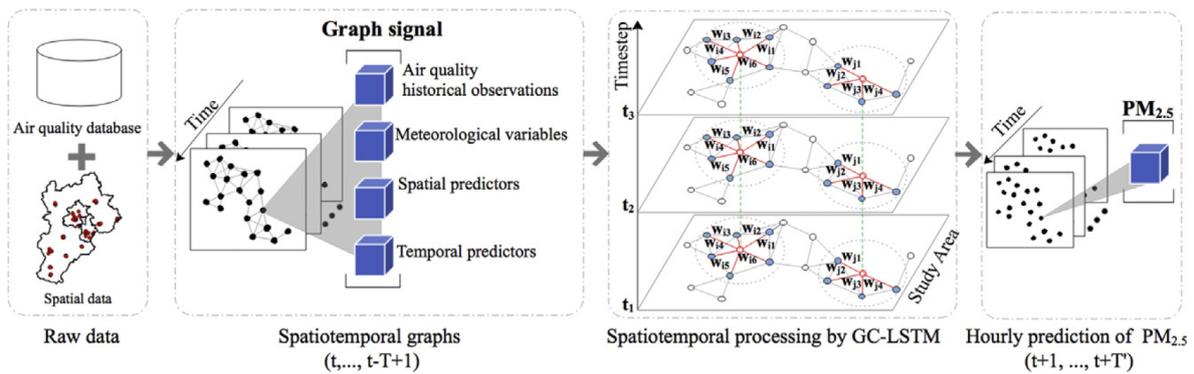


FIGURE 3.2 – Le framework présenté par [8] pour la prédiction de $PM_{2.5}$

d’abord une structure de graphe pour représenter les relations géospatiales entre les stations. Les relations spatiales entre différents lieux sont représentées comme un graphe non orienté $G = (V, E, \mathbf{A})$ avec N nœuds $v_i \in V$ et chaque arête $(v_i, v_j) \in E$. $\mathbf{A} \in \mathbb{R}^{N \times N}$ est la matrice de poids spatial, où chaque élément $A_{i,j}$ représente la corrélation spatiale quantitative entre v_i et v_j .

Le modèle proposé est composé de blocs contenant chacun une opération de convolution de graphe prenant en entrée les signaux du graphe X_t à chaque instant t , ainsi qu’une matrice de poids spatial \mathbf{A} , pour extraire les caractéristiques spatiales H_t . Ces caractéristiques sont concaténées avec les signaux originaux X_t puis passées à un LSTM pour l’extraction des caractéristiques temporelles. Enfin, la sortie du LSTM est traitée par une couche entièrement connectée (FC) pour prédire la concentration de $PM_{2.5}$ à un moment donné, comme illustré à la figure 3.2.

Dans [33], les auteurs développent une approche de prévision pour la prédiction des cas de COVID-19 en utilisant des GNN et des données de mobilité. Les données de mobilité consistent en des flux d’activité humaine inter- et intra-régionaux, à partir desquels des graphe-STs sont créés. Chaque nœud représente un lieu connecté à un nombre arbitraire d’autres nœuds, et les poids des arêtes correspondent aux mesures de mobilité humaine entre les nœuds. Les auteurs créent un multigraphe avec des arêtes spatiales représentant les mouvements directs entre lieux, et des arêtes temporelles représentant des connexions binaires aux jours précédents. Le graphe résultant est composé de 100 couches empilées, chaque couche modélisant le graphe de connectivité de ce jour. Pour la phase d’apprentissage, les auteurs utilisent un modèle GNN simple avec connexion (de saut) à 2 sauts. Les plongements spatio-temporels sont obtenus en empilant plusieurs couches d’agrégation spatiale. Le plongement final est passé à un perceptron multicouche (MLP) pour obtenir la prédiction finale. Les résultats obtenus sont comparés avec les modèles de référence "Autoregressive integrated moving average" ARIMA et LSTM.

L’utilisation des graphes-ST permet de représenter les changements au fil du temps dans les zones et de surveiller les changements sur les cartes. Les études présentées dans cette section démontrent le potentiel des graphe-STs pour modéliser et prévoir les variables environnementales telles que la vitesse du vent et la qualité de l’air. L’utilisation de techniques d’apprentissage profond telles que les réseaux LSTM et les modèles basés sur la convolution de graphes permet d’extraire des caractéristiques spatio-temporelles significatives des données. Les modèles résultants sont capables de prédire les variables environnementales avec une précision améliorée par rapport aux modèles traditionnels. De plus, l’utilisation des graphe-STs a montré des résultats prometteurs pour prédire la propagation des maladies infectieuses, comme la COVID-19, en incorporant les données de mobilité dans la structure du graphe. Ces exemples démontrent le potentiel de l’apprentissage profond sur les graphe-STs pour relever des défis environnementaux et géographiques importants.

3.2.3 Reconnaissance d’activités et de mouvements

Un domaine majeur des applications de l’IA et de la vision par ordinateur est la compréhension de scènes [34]. Une interaction homme-machine plus efficace peut être réalisée en améliorant la capacité des systèmes sensoriels et oculaires à analyser leur environnement et les activités humaines présentes dans les scènes. De nombreuses méthodes basées sur les réseaux de neurones convolutionnels, les GANs et les autoencodeurs ont été proposées pour répondre à ce besoin [35]. Récemment, une tendance croissante à utiliser l’apprentissage profond basé sur les graphes spatio-temporels pour reconnaître l’activité et le mouvement humain a été observée. Dans cette section, nous discutons des méthodes récemment publiées dans ce domaine.

Un des premiers travaux démontrant l’efficacité de l’apprentissage profond sur graphes pour la reconnaissance d’activités est celui de Yan et *al.* [36]. Leur objectif est d’identifier l’activité humaine basée sur les squelettes. Contrairement aux approches classiques utilisant des caractéristiques définies manuellement, les auteurs proposent un modèle de squelettes dynamiques reposant sur des réseaux spatio-temporels. Les données squelettiques, obtenues à partir de dispositifs de capture de mouvement ou d’algorithmes d’estimation de pose, sont utilisées comme entrée, représentées par les coordonnées des articulations. Un graphe spatio-temporel non orienté $G = (V, E)$ est construit à partir des séquences de squelettes comportant N articulations et T images. Les auteurs définissent la convolution spatio-temporelle comme une extension de la convolution de graphes spatiaux en incluant les articulations connectées temporellement. Le modèle ST-GCN est composé de 9 couches de convolution spatio-temporelle suivies d’un pooling global, et le vecteur résultant alimente à un réseau de neurones avec une couche finale softmax, comme illustré à la figure 3.3.

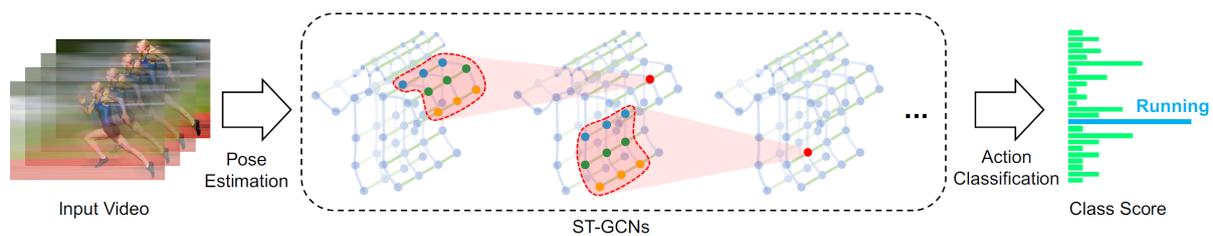


FIGURE 3.3 – Illustration de Yan et *al.*[36] expliquant la méthode présentée. À partir de vidéos, des graphe-STs sont construits sur des séquences de squelettes. Plusieurs couches de convolution spatio-temporelle (ST-GCN) sont appliquées et génèrent progressivement des cartes de caractéristiques de plus haut niveau sur le graphe. Un classificateur softmax standard est utilisé pour obtenir la catégorie d’action correspondante.

Dans une autre étude, Zhang *et al.* [37] proposent des étapes de prétraitement avant de transmettre la représentation graphique des squelettes au réseau de convolution de graphes spatio-temporels pour reconnaître l’action (figure 3.4). En considérant le squelette humain en 3D pour construire le graphe-ST, chaque individu est représenté comme un vecteur de caractéristiques en coordonnées xyz pour chaque articulation au n -ième instant. Une matrice d’adjacence $A \in \mathbb{R}^{N \times N}$ est définie, où chaque entrée A_{ij} indique si l’articulation i et l’articulation j sont connectées. La séquence de squelettes est d’abord transformée en squelettes image-par-image et en trajectoires des nœuds, qui sont ensuite passées à un routeur de graphes spatiaux (SGR) et à un routeur de graphes temporels (TGR) respectivement pour produire de nouveaux graphes de connectivité des articulations squelettiques. Finalement, le ST-GNN reçoit les graphes et les sorties du routage des graphes pour produire la classification.

Pallabi *et al.* [38] proposent un modèle de réseau de convolution spatio-temporelle empilé pour prédire et segmenter les vidéos en sections d’actions. Ils utilisent une méthode de reconnaissance de situation pour obtenir le graphe-ST, où les nœuds du graphe correspondent aux objets dans la scène, tandis que les arêtes représentent leurs relations spatiales et/ou fonctionnelles. Les auteurs mettent à jour la convolution classique de graphes spatio-temporels en permettant des connexions temporelles arbitraires pour supporter des configurations flexibles et des déformations fréquentes du graphe dans les activités complexes. L’architecture du modèle suit un design en sablier, consistant en une série d’opérations de sous-échantillonnage et de sur-échantillonnage avec des connexions de saut (comme illustré à la figure 3.5).

Pratyusha *et al.* [39] ont utilisé un réseau de neurones convolutionnel spatio-temporel pour analyser les activités complexes basées sur les squelettes des mains. Initialement, un graphe-ST est construit à partir des séquences temporelles des positions des articulations de la main dans l’espace, qui sont transmises à plusieurs couches de convolution de graphes spatio-temporels pour extraire des caractéristiques de haut

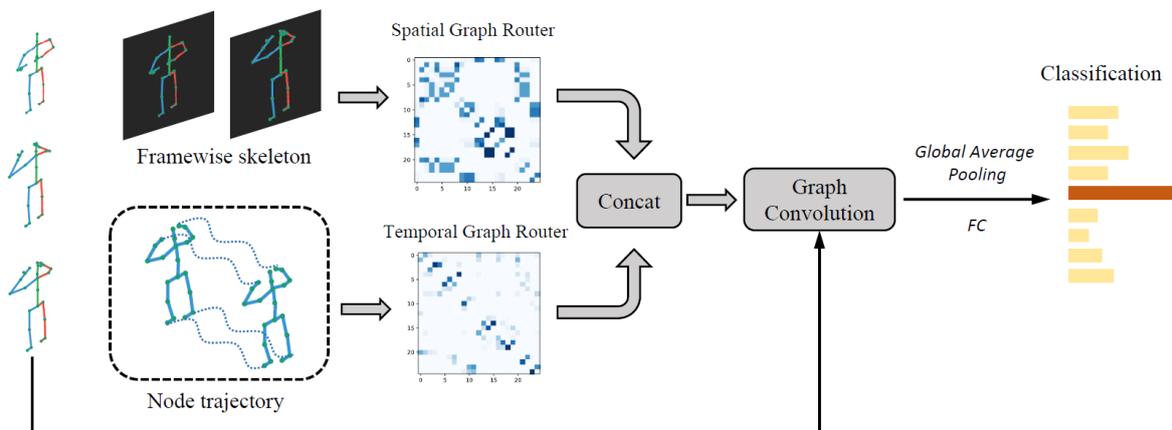


FIGURE 3.4 – Illustration de l’approche proposée dans [37]. La séquence de squelettes 3D en entrée est d’abord transformée en squelettes et en trajectoires de nœuds respectivement. Ensuite, le SGR et le TGR produisent respectivement de nouveaux graphes de connectivité de squelettes et d’articulations. ST-GCN reçoit ces graphes et génère une classe d’action.

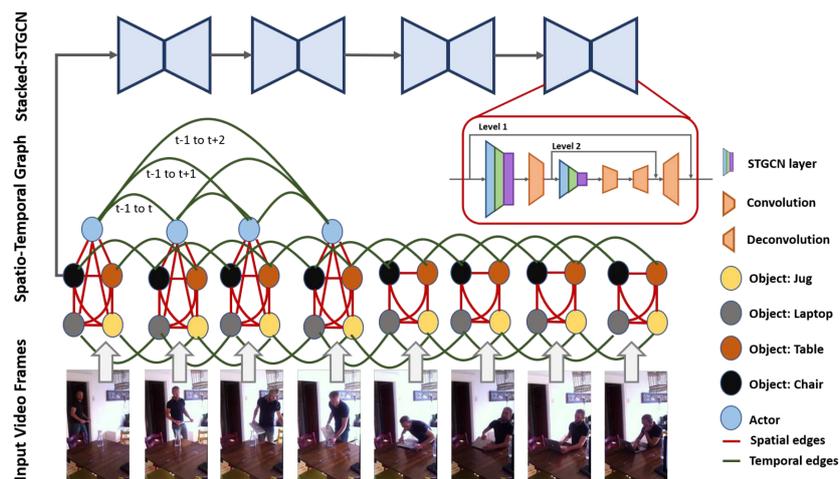


FIGURE 3.5 – Vue d’ensemble du travail de Pallabi *et al.*[38], démontrant les propriétés de leur graphe, permettant divers types de nœuds et supportant des connexions temporelles flexibles pouvant s’étendre sur plusieurs étapes temporelles. Le graphe spatio-temporel alimente une pile de blocs STGCN en sauto pour obtenir une séquence d’actions prédites observées dans la vidéo.

niveau. Les caractéristiques obtenues sont concaténées et suivies d’un pooling global, et enfin, un prédicteur softmax est utilisé pour prédire l’action.

Ces travaux représentent une contribution significative au domaine de la reconnaissance d’activités humaines et démontrent le potentiel de l’apprentissage profond sur graphes spatio-temporels dans ce domaine. Contrairement aux méthodes traditionnelles reposant sur des caractéristiques définies manuellement, ces modèles exploitent les réseaux

spatio-temporels pour étendre le concept de voisinage aux articulations connectées temporellement, permettant ainsi une reconnaissance plus précise des activités.

3.2.4 Tâches liées au trafic routier

Les réseaux de graphes spatio-temporels (graphe-STs) sont largement utilisés pour les tâches liées au trafic, notamment la prévision du flux de trafic et des vitesses, ainsi que la prédiction de la demande de covoiturage et des accidents. Cette section résume les travaux récents dans ce domaine, divisés en sous-catégories pour une meilleure lisibilité.

Prévision du flux de trafic

L’un des premiers travaux utilisant l’apprentissage profond sur graphe-STs pour résoudre des problèmes liés au trafic est celui de Yu *et al.* [40], qui propose un modèle pour la prévision du trafic capturant les dépendances spatiales et temporelles. Le modèle utilise un réseau de neurones convolutionnels spatio-temporels (STGCN) pour modéliser les dépendances spatio-temporelles dans les données de trafic, combinant un modèle GNN pour l’aspect spatial et un CNN 1D pour traiter le domaine temporel. Zhang *et al.* [41] ont proposé un modèle basé sur des réseaux convolutifs pour prédire le flux de trafic en divisant la carte spatiale initiale en un format de grille présenté dans la figure 3.6. Chai *et al.* [22] ont développé une approche de réseau de convolution multi-graphe pour la prévision du flux de vélos en libre-service, en utilisant des graphes représentant les stations de vélos et leurs relations spatiales. Le multigraphe généré est composé de trois sous-graphes : un graphe de distance, un graphe d’interaction et un graphe de corrélation.

Chen *et al.* [42] ont proposé un modèle combinant des unités récurrentes et des réseaux de graphes pour capturer les dépendances spatio-temporelles dans les données de trafic. Leur modèle combine des unités récurrentes avec des connexions résiduelles pour capturer à la fois les dépendances spatiales et temporelles, utilisant un réseau de convolution de graphes (GCN) pour modéliser les dépendances spatiales entre les segments de route et un réseau récurrent résiduel (GRRU) pour capturer les dépendances temporelles.

Une approche de convolution multi-graphe temporelle a été proposée par Ming *et al.* [15] pour la prévision du flux de trafic. Les auteurs construisent quatre graphes pour encoder correctement la corrélation entre les nœuds : un graphe représentant la structure topologique du réseau routier, un graphe topologique pondéré basé sur la longueur des chemins entre les nœuds, un graphe de motifs de trafic et un graphe de fonctionnalité. Les graphes sont passés à un réseau de neurones multi-graphe

pour capturer les interactions entre les nœuds, suivi d’une couche récurrente pour capturer la corrélation temporelle.

Chen *et al.* [43] ont proposé une approche similaire pour modéliser la dépendance spatio-temporelle complexe entre les flux de trafic et les conditions de trafic en évolution rapide. Leur modèle prend en entrée les volumes de trafic précédents, représentés par des graphes empilés, et les conditions de trafic précédentes, représentées par des matrices d’affinité, intégrées dans un réseau à deux flux pour effectuer la prédiction. Pour la prédiction de la demande origine-destination (OD), Jintao *et al.* [44] ont proposé un ensemble encodeur-décodeur composé de deux encodeurs, spatial et temporel. L’encodeur spatial utilise plusieurs convolutions multi-graphe résiduelles (RMGC) pour modéliser les corrélations spatiales entre les paires OD. L’encodeur temporel utilise un modèle LSTM spatial pour apprendre les dépendances temporelles de chaque paire OD. Les sorties des encodeurs sont concaténées et passées à travers des réseaux RMGC pour prédire la demande OD.

Une étude s’est concentrée sur l’amélioration de la prédiction du flux de trafic en introduisant un réseau de neurones de graphes spatio-temporels dynamiques (DSTGCNN) [45]. Ce modèle incorpore la décomposition tensorielle pour capturer les changements dynamiques dans la matrice de Laplacien du réseau de trafic, pour s’adapter aux fluctuations et incidents de trafic à court terme. Le DSTGCNN intègre des couches de convolution temporelle à portes et des couches de convolution de graphes spatiaux pour modéliser efficacement les dépendances spatiales et temporelles. Cette approche a démontré une précision améliorée dans la prédiction du flux de trafic par rapport aux méthodes traditionnelles.

Une autre étude a présenté un modèle appelé Réseau de Séquences de Graphes avec Mécanisme d’Attention (GSeqAtt) pour la prévision de la vitesse du trafic [46]. Le modèle GSeqAtt combine des mécanismes d’attention horizontale et verticale pour capturer les dépendances spatiales changeantes au fil du temps. Cette approche à double attention augmente la capacité du modèle à traiter des séquences de graphes et à améliorer la précision des prédictions de vitesse. Le cadre exploite les mécanismes d’attention pour ajuster dynamiquement l’importance des différentes caractéristiques spatiales et temporelles, ce qui se traduit par une prévision plus précise de la vitesse du trafic.

Prévision de la vitesse de circulation

Zhong *et al.* [47] ont proposé un modèle nommé LC-RNN pour des prédictions précises de la vitesse de circulation en intégrant des modèles RNN et CNN (figure 3.7). Le modèle utilise une couche de convolution de look-up, une couche récurrente, une

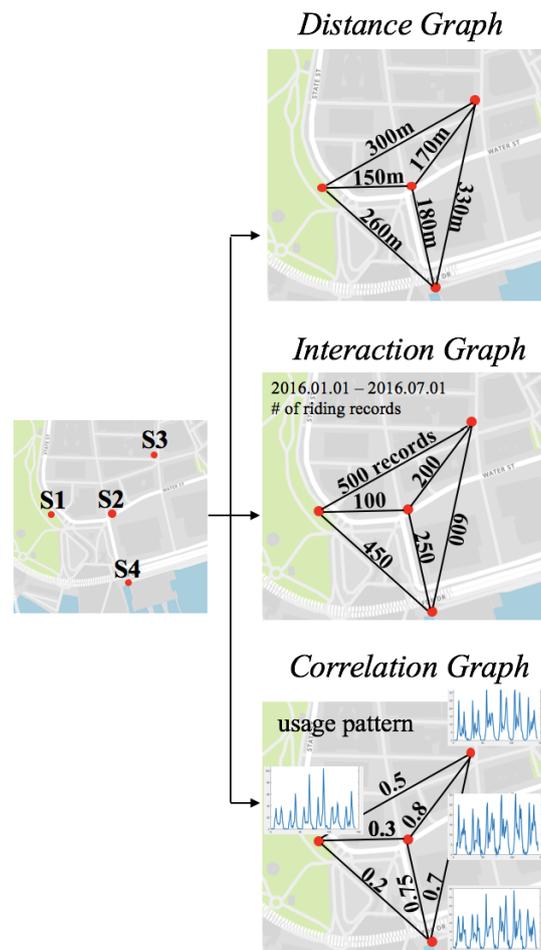


FIGURE 3.6 – Les trois graphes générés dans le travail de Zhang *et al.* [41].

couche d’extraction de périodicité et une couche d’extraction de contexte pour modéliser l’évolution spatiale locale, les dépendances temporelles longues, la périodicité et les facteurs de contexte.

Han *et al.* [48] utilisent des réseaux de graphes dynamiques pour capturer les caractéristiques spatio-temporelles multifacettes des données de trafic. Les graphes sont générés à l’aide d’un constructeur de graphes dynamiques et passés à deux blocs : un bloc de caractéristiques primaires et un bloc de caractéristiques auxiliaires, chacun composé d’une couche de convolution temporelle et d’une convolution de graphes dynamiques. Le module de fusion multifacettes combine les sorties des blocs pour la prédiction finale. Diao *et al.* [49] proposent un modèle hybride graphe-ST intégrant la décomposition tensorielle pour la prédiction du temps de trajet. Ils utilisent une matrice de Laplacien dynamique calculée pendant la phase d’apprentissage pour capturer les fluctuations de trafic à court terme. Le modèle (illustré dans la figure 3.8) est composé de trois

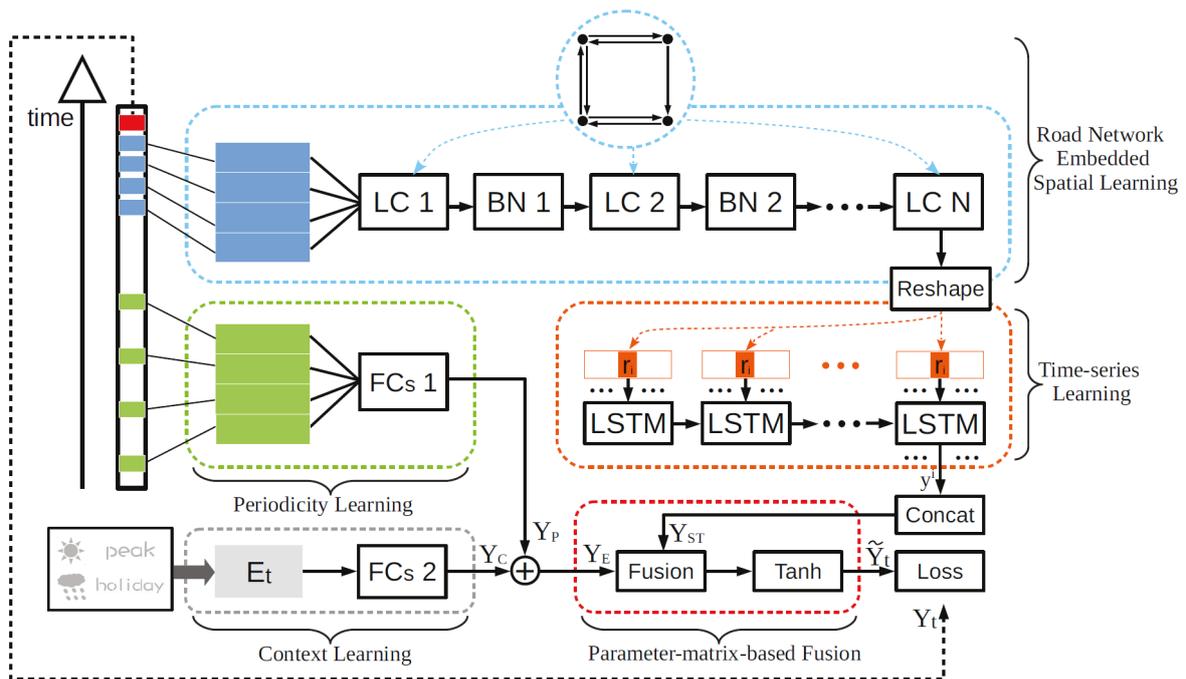


FIGURE 3.7 – L’architecture de LC-RNN telle que démontrée dans Zhong *et al.* [47]. LC : Convolution par table de correspondance ; BN : Normalisation par lot ; FC : Couche entièrement connectée ; LSTM : Mémoire à long court terme.

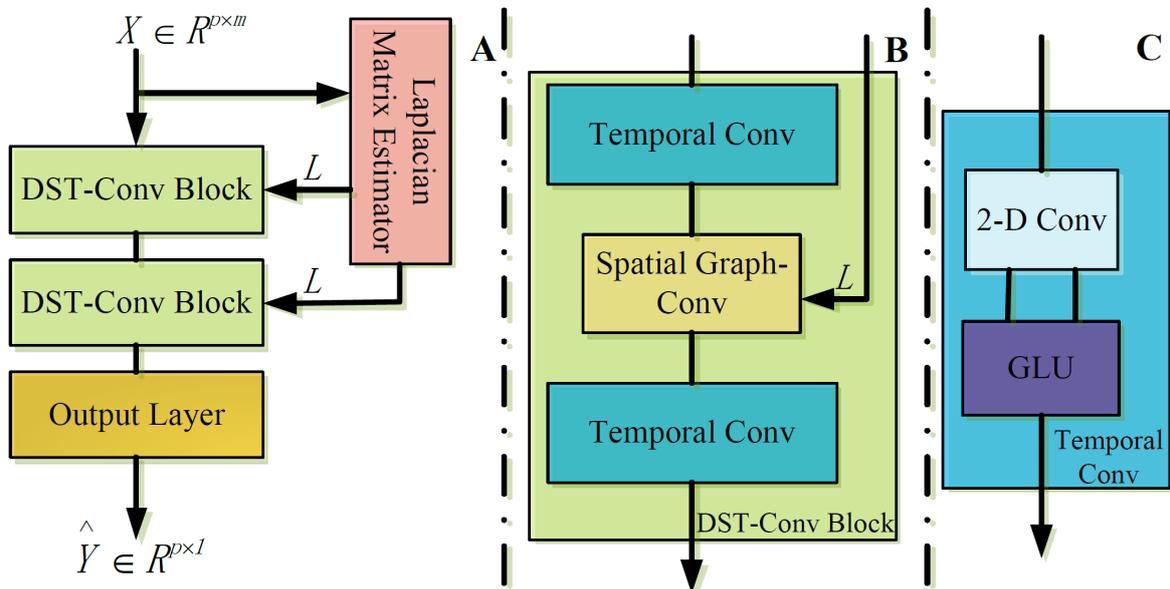


FIGURE 3.8 – Illustration des différents composants du modèle DSTGCNN proposé par [49].

modules : un estimateur de matrice de Laplacien, deux blocs de convolution spatio-temporelle et une couche de sortie.

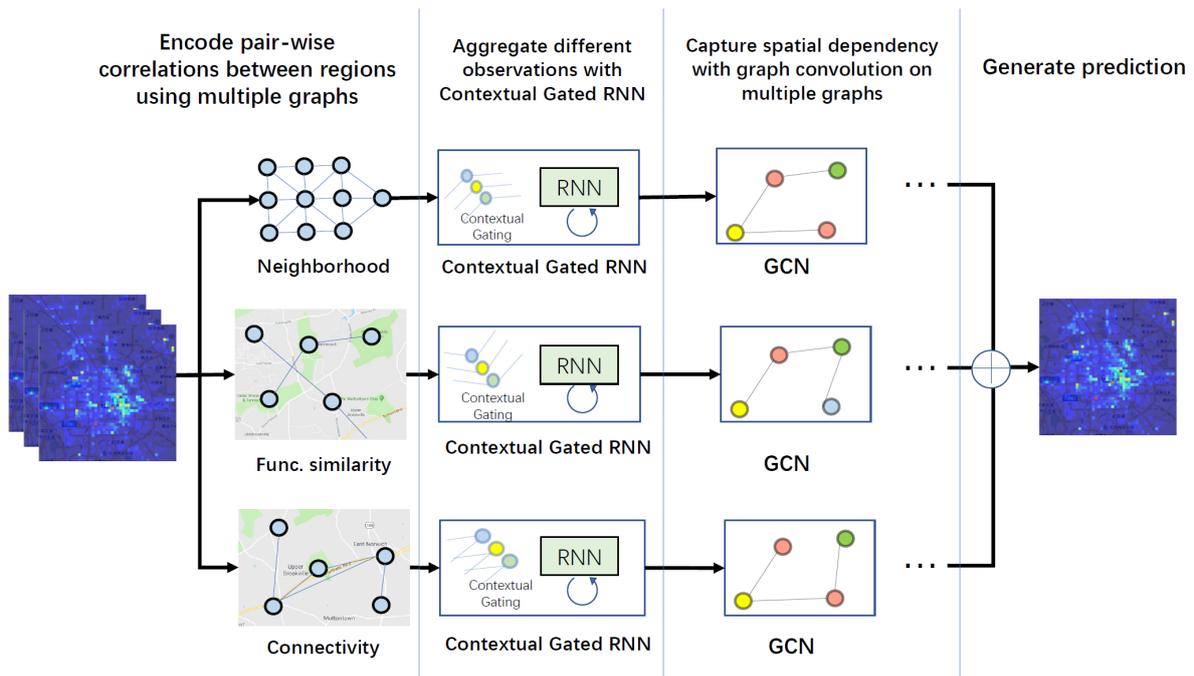


FIGURE 3.9 – Architecture du système du réseau de convolution multi-graphe spatio-temporel (ST-MGCN) proposé par [50]. Ils encodent différents aspects des relations entre les régions, y compris le voisinage, la similarité fonctionnelle et la connectivité de transport, en utilisant plusieurs graphes. Ensuite, ils utilisent un réseau de neurones récurrents contextuels à portes (CGRNN) pour agréger les observations à différents moments en considérant l’information contextuelle globale. Après cela, la convolution multi-graphe est utilisée pour modéliser les corrélations non euclidiennes entre les régions.

Prévision de la demande de covoiturage

Geng *et al.* [50] ont proposé une méthode de prévision de la demande de covoiturage à l’échelle régionale en utilisant des convolutions sur graphe-ST. Pour cela, ils divisent une ville en grilles de taille égale, chaque grille étant définie comme une région, et prédisent la demande sur la base des données historiques. Le modèle intègre des relations de voisinage, de similarité fonctionnelle et de connectivité des transports à l’aide de multiples graphes, et utilise un réseau récurrent contextuellement verrouillé (CGRNN) pour agréger les observations à différents moments (figure 3.9).

Guang *et al.* [51] ont proposé un modèle de graphes virtuels pour se concentrer sur les régions à forte demande en utilisant des réseaux de graphes spatio-temporels multi-vue. Leur modèle comprend un générateur de graphes pour obtenir le graphe virtuel spatio-temporel à partir des vues de distance, de corrélation et de mobilité. Chaque nœud est ensuite passé à un CNN 1D pour capturer les dépendances temporelles à court terme, et des couches GAT empilées sont utilisées pour capturer les

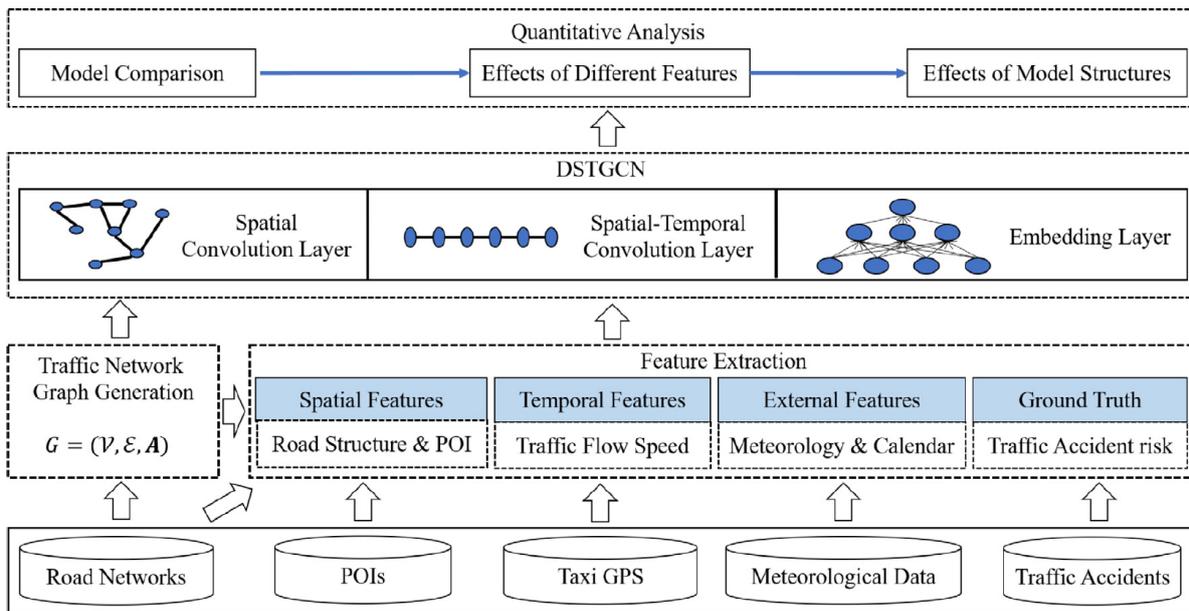


FIGURE 3.10 – Diagramme du framework proposé dans [11]

dépendances spatiales. Enfin, une couche de transformateur est ajoutée pour capturer les dépendances temporelles à long terme.

Un problème similaire a été abordé par Hong *et al.* [52] pour prédire la demande de taxis. Ils créent un graphe où chaque nœud représente une localisation et chaque arête représente la corrélation spatio-temporelle entre les localisations. Le modèle d'apprentissage profond comprend une couche d'entrée, une couche de convolution multi-graphe, une couche encodeur-décodeur composée de LSTM, et une couche de sortie. Une méthode similaire a été explorée par [53] pour prédire les trajectoires des véhicules.

Prévision des accidents et événements

Yu *et al.* [11] ont développé un système pour la prévision et la prévention des accidents de la circulation en utilisant des données telles que les accidents de la route, les données GPS des taxis, les points d'intérêt et les données météorologiques. Ils créent un graphe-ST en divisant la carte en grille et en associant les caractéristiques précédemment mentionnées à chaque segment de route. Les caractéristiques sont divisées en trois catégories : spatiales, temporelles et externes, chacune étant passée à une couche adéquate pour l'extraction des caractéristiques. L'approche est résumée dans la figure 3.10. Yang *et al.* [54] ont proposé un modèle de diffusion locale pour la prédiction des événements spatio-temporels rares, en utilisant une approche de réseau basé sur un graphe pour modéliser la propagation des événements. Les segments de route sont représentés comme des nœuds de graphe, et les intersections entre les segments

adjacents sont transformées en arêtes, avec des poids calculés via une fonction de pondération de noyau gaussien.

Prévision de la demande et de l’occupation des parkings

Xiao *et al.* [55] ont proposé un réseau convolutionnel spatio-temporel pour prédire la demande de vélos en libre-service, en explorant les informations multi-vue. Leurs modèles intègrent des relations de voisinage, de similarité fonctionnelle et de connectivité de transport à l’aide de multiples graphes. Ils utilisent des couches de convolution spatio-temporelle pour capturer les dépendances spatiales et temporelles dans les données de trafic. Yang *et al.* [56] ont proposé un modèle combinant des GCNN et des LSTM pour prédire l’occupation des parkings à l’échelle des blocs urbains.

Autres applications liées au trafic

Pour étudier la corrélation entre les routes distantes partageant des schémas similaires, Bing Yu *et al.* [14] ont proposé des réseaux de graphes temporels 3D (3D-TGCN) permettant d’apprendre conjointement la structure des graphes et les dynamiques temporelles. Le cadre proposé est composé de quatre blocs de convolution de graphes 3D et d’une couche de sortie pour obtenir les prédictions finales. Dans une autre application de prédiction des points chauds urbains, Guangyin *et al.* [57] ont proposé un réseau de neurones géographique-sémantique intégré. Leur approche intègre des réseaux de neurones prédictifs récurrents et des réseaux de neurones de graphes convolutifs prédictifs pour capturer les dynamiques spatio-temporelles sous différents angles. Les travaux présentés démontrent l’efficacité des graphe-STs et des réseaux de neurones profonds pour capturer les dépendances spatio-temporelles dans les données de trafic, améliorant ainsi la précision des prédictions et la gestion des systèmes de transport.

3.2.5 Applications diverses

Les graphes spatio-temporels sont une structure de modélisation expressive utilisée dans divers cas d’utilisation. Dans cette section, nous explorons les applications de l’apprentissage profond sur les graphe-STs, dans des domaines qui n’ont pas été au centre des recherches mais qui offrent néanmoins des perspectives intéressantes.

Xuan *et al.* [58] proposent un modèle basé sur un réseau de neurones multi-graphe pour résoudre le problème de la prévision des flux de foule. Ils construisent deux graphes K-NN en utilisant la distance euclidienne et le coefficient de corrélation de Pearson, avec différentes matrices adjacentes. Ils encodent les flux de foule historiques

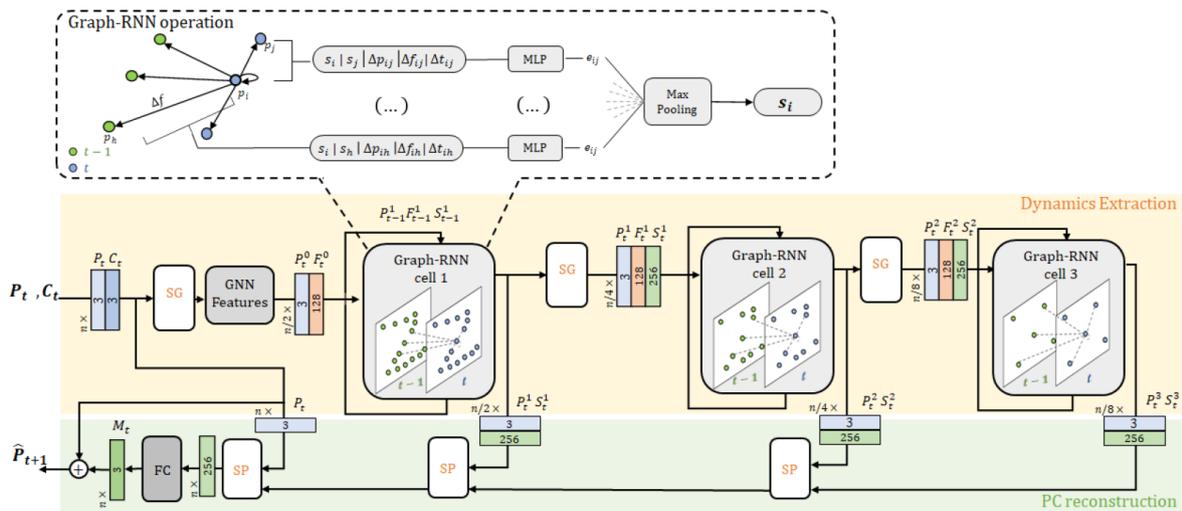


FIGURE 3.11 – Diagramme du framework proposé dans [60]

en plusieurs graphes, chacun étant traité par un bloc spatial incluant GAT et ChebNet. Les sorties sont fusionnées et passées à un modèle temporel composé d’un LSTM et d’un mécanisme d’auto-attention. La prédiction finale est obtenue via un réseau de neurones entièrement connecté.

Abduallah *et al.* [59] ont utilisé un modèle de réseau neuronal convolucional basé sur le graphe-ST pour prédire les trajectoires des piétons. Le modèle est capable de prédire les trajectoires des piétons à partir d’un ensemble de positions observées prises dans des scènes. La sortie du modèle consiste en une distribution de probabilité aléatoire des coordonnées des piétons, incorporant deux composants principaux : un réseau de neurones convolucional graphe-ST et un réseau de convolution pour l’extrapolation temporelle. Le stGCNN effectue des opérations de convolution spatio-temporelle sur la représentation graphique des trajectoires des piétons pour extraire des caractéristiques, qui sont ensuite passées au second module pour prédire les trajectoires futures.

Gomes *et al.* [60] proposent un cadre pour prédire les futures séquences de nuages de points. Le cadre est conçu pour prédire les coordonnées euclidiennes des futurs nuages de points dans un espace 3D en se basant sur des séquences précédentes de nuages de points étiquetés (figure 3.11). Les composants principaux du cadre incluent un modèle GNN transformant l’espace 3D initial en un espace de caractéristiques de dimension supérieure et envoie la sortie à un Graph-RNN pour calculer l’encodage lié au voisinage. L’étape finale consiste à reconstruire le nuage de points à partir des caractéristiques apprises en appliquant une opération de mise à l’échelle. Le modèle est conçu avec une architecture hiérarchique pour capturer des informations à différentes échelles. La méthode a été testée sur diverses applications, comme la prédiction d’activités de corps humains synthétiques.

Pour résoudre un problème similaire, Feng *et al.* [61] proposent un modèle de réseau de convolution graphe-ST pour prédire les trajectoires d’objets et leurs relations dynamiques. Ils abstraient les vidéos en graphes-ST entièrement connectés, puis les passent à plusieurs blocs de convolution graphe-ST et couches entièrement connectées pour obtenir des prédictions de catégorie d’objets.

Bing Yu *et al.* [62] présentent une architecture inspirée du U-net multi-échelle pour extraire des caractéristiques dynamiques et complexes à partir de structures spatio-temporelles. Le ST-Unet repose principalement sur des opérations de pooling et de dépooling spatio-temporelles. Pour le pooling des graphes spatiaux, les auteurs utilisent l’algorithme de croissance de chemin global, réduisant la taille d’un graphe par un facteur de deux pour produire un graphe plus grossier correspondant à une observation des données à une résolution différente. Pour le pooling temporel, les auteurs utilisent le modèle GRU et les couches GCN comme unités récurrentes de graphes convolutifs (GCGRU) pour découvrir des motifs temporels à partir de séries temporelles structurées en graphes. L’opération de dépooling est appliquée en utilisant trois stratégies différentes : copie directe, déconvolution ordonnée et déconvolution pondérée. L’architecture générale du réseau consiste en une première couche de convolution de graphes pour agréger les informations de voisinage, suivie d’une couche ST-Pool. L’encodage obtenu est ensuite passé à une couche de suréchantillonnage, dont la sortie est passée à une dernière couche de convolution de graphes pour obtenir la prédiction finale.

Pour estimer la durée de vie restante dans les applications industrielles, Yuxuan *et al.* [13] proposent une approche adaptative basée sur les réseaux de neurones convolutionnels graphe-ST. L’approche vise d’abord à apprendre la structure des graphes à partir des données de capteurs, réalisée en utilisant la convolution de graphes. La matrice adjacente adaptative est calculée en utilisant la métrique de Mahalanobis et un noyau gaussien avec la formule suivante :

$$A_{ij}^t = f(x_i^t, x_j^t) = e^{-D(x_i^t, x_j^t)^2 / 2\sigma^2}$$

où D est la distance de Mahalanobis, et x_i^t est la donnée du i^{me} capteur au temps t . Après avoir obtenu les matrices adjacentes, les graphes sont passés à des couches de convolution adaptatives graphe-ST empilées, consistant en une couche de convolution temporelle pour capturer la dépendance dans le domaine temporel, et une couche de convolution de graphes adaptative pour apprendre l’information sous-jacente des graphes. Une régression finale est appliquée sur la sortie obtenue pour estimer la durée de vie restante.

Dans un autre article, Wang *et al.* [63] proposent une approche similaire pour l’estimation de la durée de vie restante des moteurs d’avions. Comme le travail précédent,

les auteurs proposent une architecture composée de blocs spatio-temporels empilés, chaque bloc consistant en deux GCN et deux couches de convolution temporelle, suivi d’un mécanisme de verrouillage inspiré des réseaux récurrents pour fusionner les caractéristiques spatiales et temporelles. La sortie des blocs ST empilés est finalement passée à une couche entièrement connectée pour obtenir le résultat final.

L’apprentissage profond automatisé (AutoDL) est une approche prometteuse pour démocratiser l’utilisation de l’apprentissage profond dans divers domaines, y compris l’apprentissage profond sur les graphes. Les techniques AutoDL ont le potentiel d’automatiser le processus de conception et d’optimisation des réseaux de neurones de graphes (GNN). Les avancées récentes en recherche d’architecture neuronale (NAS) exigent des ressources informatiques importantes, ce qui entrave la reproductibilité et constitue une barrière pour les utilisateurs n’ayant pas accès à des calculs à grande échelle. Pour résoudre ce problème, Ying *et al.* [64] ont introduit NAS-Bench-101, le premier ensemble de données d’architecture public pour la recherche NAS. Zhou *et al.* [65] ont proposé le cadre Automated Graph Neural Networks (AGNN), visant à trouver une architecture GNN optimale dans un espace de recherche prédéfini. Néanmoins, l’architecture ayant la plus haute précision de validation pendant la phase de recherche peut ne pas bien performer pendant l’évaluation. Pour atténuer ce problème, Guohao *et al.* [66] ont introduit la recherche d’architecture séquentielle gloutonne (SGAS), une méthode efficace pour la recherche d’architecture neuronale.

Dans le contexte des graphe-ST, Li *et al.* [67] proposent une méthode efficace de recherche d’architecture neuronale spatio-temporelle appelée AutoST pour automatiser la conception de modèles de graphes spatio-temporels profonds pour des tâches comme la prédiction d’événements spatio-temporels tels que les flux de foule. Contrairement aux modèles actuels de recherche d’architecture neuronale (NAS) qui optimisent les stratégies de recherche dans un espace de recherche fixe, l’espace de recherche innovant d’AutoST est spécifiquement conçu pour le domaine spatio-temporel. Il comprend des opérations de convolution optionnelles à chaque couche pour extraire automatiquement des dépendances spatio-temporelles multi-portées et des connexions de saut apprenables pour fusionner dynamiquement les caractéristiques de bas et de haut niveau.

De même, Zheyi Pan *et al.* [68] abordent les défis de la recherche automatisée d’architecture neuronale pour les graphes spatio-temporels avec l’application à la prédiction du trafic urbain. Les défis incluent la définition d’un espace de recherche pour capturer les corrélations spatio-temporelles complexes et l’apprentissage des paramètres de poids du réseau liés au graphe attribué correspondant à un graphe spatio-temporel. Pour surmonter ces défis, les auteurs proposent un cadre appelé AutoSTG pour la prédiction automatisée de graphes spatio-temporels. Le cadre proposé utilise la convolution de graphes spatiaux et des opérations de convolution temporelle dans l’espace

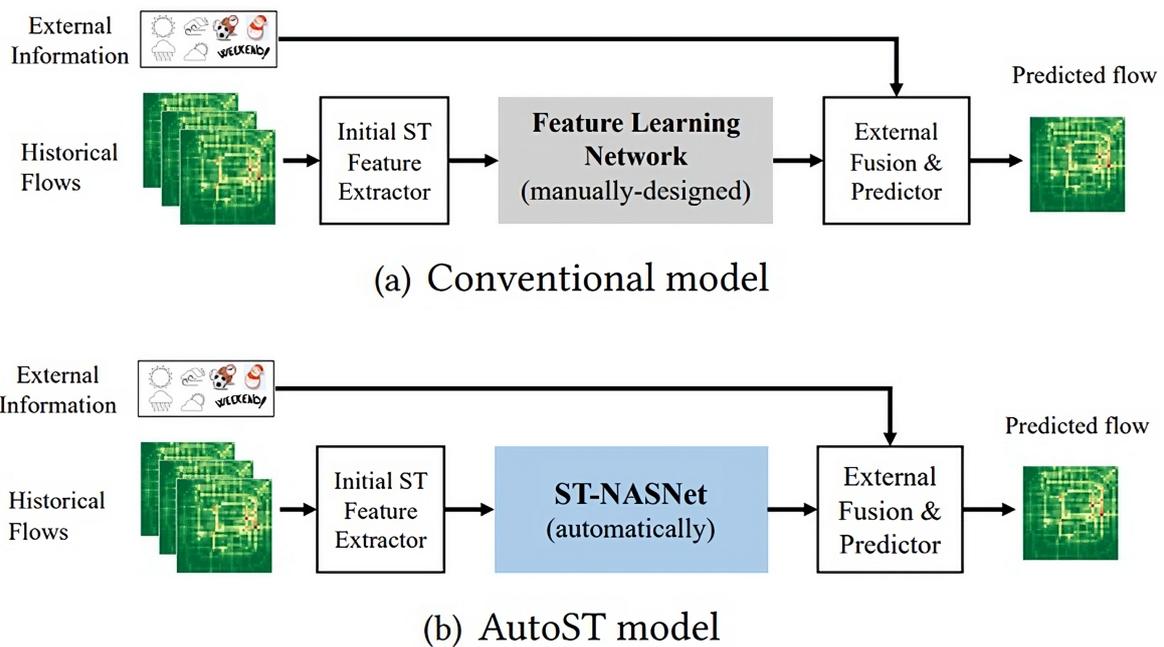


FIGURE 3.12 – Comparaison entre la méthode conventionnelle pour entraîner le modèle ST et l’approche AutoST [67]

de recherche pour capturer les corrélations spatio-temporelles complexes. De plus, les auteurs utilisent des techniques de méta-apprentissage pour apprendre les matrices adjacentes des couches de convolution de graphes spatiaux et les noyaux des couches de convolution temporelle à partir de la méta-connaissance du graphe attribué. Une version étendue introduisant des améliorations supplémentaires a été introduit [69]. Les auteurs introduisent le processus de construction de méta-graphes, y compris l’extraction de similarité spatiale, de similarité temporelle et de similarité de caractéristiques, et améliorent la structure du méta-apprentissage des graphes pour apprendre la méta-connaissance à partir de multiples méta-graphes. En outre, ils proposent un module de graphe dynamique partiel appelé Graphes Dynamiques Quantifiés par Vecteur (VQDG) pour apprendre la représentation à partir de l’information temporelle dynamique et ensuite générer le méta-graphe dynamique correspondant.

Les réseaux de neurones de graphes ont également été explorés dans le domaine de la prévision de séries temporelles multivariées [70]. STGNN utilise la théorie spectrale des graphes et les techniques d’apprentissage profond pour capturer les dépendances entre différentes variables et les motifs temporels dans les données de séries temporelles. Le modèle se compose de deux composants principaux : le réseau de convolution de graphes spectraux (SGCN) et le réseau de convolution temporelle (TCN). Le SGCN est utilisé pour capturer les dépendances spatiales entre les variables, tandis que le TCN est utilisé pour capturer les motifs temporels dans les données de séries temporelles.

Le SGCN utilise l’opération de convolution de graphes spectraux, qui fonctionne sur le domaine de la transformée de Fourier du graphe, pour capturer l’information topologique du graphe et la représentation des caractéristiques des nœuds. Le TCN utilise une série de couches de convolution temporelle 1D avec des connexions résiduelles et des convolutions dilatées pour capturer les dépendances temporelles entre les étapes de temps. Le modèle STGNN introduit également un mécanisme d’attention novateur appelé module d’Attention Spectrale Temporelle (STA), qui ajuste dynamiquement l’importance des caractéristiques spatiales et temporelles à chaque pas de temps. Le module STA utilise le mécanisme d’attention pour apprendre les relations entre différentes variables et les motifs temporels dans les données de séries temporelles.

Xu *et al.* [71] ont proposé un modèle basé sur des réseaux de neurones multigraphes pour la prévision du flux de foule. Le modèle construit des graphes K-NN basés sur la distance euclidienne et le coefficient de corrélation pour représenter les données historiques de flux de foule. Ces graphes sont traités par un bloc spatial composé de Réseaux de Neurones à Attention de Graphes (GAT) et de ChebNet, suivis d’un bloc temporel avec LSTM et mécanismes d’attention. La prévision finale du flux de foule est obtenue grâce à un réseau de neurones entièrement connecté, démontrant des améliorations significatives de la précision des prévisions. Dans une autre instance, Abdullah *et al* [72] ont introduit un réseau de neurones de convolution spatio-temporel pour prédire les trajectoires des piétons. Ce modèle prédit les positions futures des piétons en fonction des trajectoires observées en effectuant des opérations de convolution spatio-temporelle sur la représentation graphique des mouvements des piétons. Le cadre comprend un réseau de convolution pour l’extrapolation temporelle afin d’améliorer la précision de la prédiction, offrant une solution robuste pour la prévision dynamique des trajectoires dans divers environnements.

Différentes applications ont été discutées dans cette section, y compris la prévision des flux de foule, la prédiction des trajectoires des piétons, la prédiction des futures séquences de nuages de points, la prédiction des trajectoires d’objets, la prédiction des relations dynamiques, l’extraction de caractéristiques et l’estimation de la durée de vie restante dans les applications industrielles. Les techniques couramment utilisées dans ces travaux sont les réseaux de neurones de graphes, les réseaux de neurones convolutionnels, les réseaux de neurones récurrents et les mécanismes d’attention. Les graphes spatio-temporels sont construits différemment en fonction des exigences de chaque application. Les travaux emploient également diverses opérations de pooling et de dépooling, y compris l’algorithme de croissance de chemin global, les couches GCN et la métrique de Mahalanobis, pour réduire la taille des graphes et obtenir des détails plus fins dans les prédictions. Les architectures générales des modèles se composent de couches de convolution de graphes suivies de couches de convolution temporelle,

avec des mécanismes de verrouillage pour fusionner les caractéristiques spatiales et temporelles. Les approches présentées dans ces travaux démontrent la polyvalence des graphes spatio-temporels et des modèles d’apprentissage profond pour aborder divers problèmes dans différents domaines.

En résumé, les travaux discutés dans cette section illustrent la polyvalence des graphes spatio-temporels et des modèles d’apprentissage profond pour résoudre une variété de problèmes complexes, allant de la prévision de flux de foule à la prédiction des trajectoires d’objets. Les techniques telles que les réseaux de neurones de graphes, les mécanismes d’attention et les architectures convolutionnelles spatio-temporelles se révèlent particulièrement efficaces pour capturer les dynamiques spatiales et temporelles. Ces approches ouvrent la voie à des applications encore plus sophistiquées dans des domaines variés, avec un potentiel d’amélioration continue en matière de précision et d’efficacité des prédictions.

3.3 Exploration des Graphes avec l’Apprentissage par Renforcement

L’apprentissage par renforcement (RL/DRL) appliqué aux graphes est un domaine ré-émergent [73] qui combine les principes du RL avec les structures complexes et riches des données modélisées par des graphes. L’intégration du RL avec les GNNs a ouvert de nouvelles voies pour résoudre une large gamme de problèmes, de l’optimisation du flux de trafic dans les villes intelligentes à la découverte de médicaments dans le secteur de la santé. Cette section explore les techniques, méthodologies et applications réelles de pointe de l’apprentissage par renforcement sur les graphes, en mettant en lumière les défis et les solutions innovantes développées dans ce domaine en évolution rapide. La recherche sur l’apprentissage par renforcement sur les graphes (GRL) connaît un essor considérable ces dernières années, et les méthodes GRL jouent un rôle important dans la résolution de diverses applications réelles et attirent une attention substantielle. Une utilisation majeure du RL sur les graphes consiste en l’apprentissage de la représentation des réseaux, qui vise à apprendre une cartographie qui intègre les nœuds d’un graphe en tant que vecteurs de faible dimension tout en encodant une variété d’informations structurelles et sémantiques. Ces méthodes optimisent les représentations de sorte que les relations géométriques dans l’espace de projection préservent la structure du graphe original. Les représentations des nœuds obtiennent efficacement des tâches telles que la classification des nœuds, le clustering des nœuds, la prédiction des liens et la classification des graphes. SUGAR [74] conserve les informations structurelles des niveaux "Nœuds/Sous-graphes/Graphes" en sélectionnant

de manière adaptative des sous-graphes significatifs avec le DQL pour représenter les informations discriminantes du graphe. Cette approche d’apprentissage hiérarchique fournit des représentations puissantes, de la généralisation et de l’explicabilité.

Lai *et al.* [75] ont défini un module de méta-politique et un module GNN pour apprendre les relations et les représentations de réseau entre les attributs des nœuds et les itérations d’agrégation. Le processus décisionnel de Markov multi-étapes (MDP) utilisé échantillonne le nombre de sauts allant du nœud courant et aux nœuds voisins de manière itérative avec la méta-politique et entraîne le GNN en agrégeant les informations des nœuds atteint par les sauts échantillonnés. L’algorithme *Policy-GNN* résout le défi de la détermination de la portée d’agrégation des nœuds dans les réseaux de grande échelle avec l’algorithme DQN.

Hong *et al.* [76] se sont concentrés sur une méthode RL modulaire et ont conçu un modèle de transformeur sans information morphologique pour encoder efficacement les relations dans les graphes, résolvant ainsi le défi du transfert d’informations multi-sauts. En plus du partage de paramètres et du mécanisme de tampon pour l’apprentissage de la représentation des réseaux dans Policy-GNN, qui permettent d’améliorer l’efficacité du modèle, Yan *et al.* [77] ont proposé un algorithme novateur et efficace pour l’intégration automatique de réseaux virtuels. Cette méthode utilise le GCN pour extraire automatiquement les caractéristiques spatiales dans une topologie de graphe irrégulière chez les agents d’apprentissage.

GDPNet [78] emploie deux phases de sélection de voisinage des nœuds et d’apprentissage d’une représentation pour éliminer les voisinages bruités afin d’optimiser le voisinage de chaque nœud cible, en apprenant une politique avec des récompenses spécifiques à la tâche, reçues de la phase d’apprentissage de représentation.

NetRL [79] améliore le réseau en détectant les liens bruités et en prédisant les liens manquants pour améliorer les capacités d’analyse et de modélisation du réseau.

Cependant, les algorithmes ci-dessus ne considèrent pas les voisinages hétérogènes dans l’agrégation des nœuds, ce qui conduit à négliger ou simplifier les diverses relations des nœuds et des arêtes dans les réseaux réels.

Wang *et al.* [80] proposent la méthode CORL, intégrant les méthodes RL dans un paradigme basé sur l’ordre. La méthode décrit le problème de recherche d’ordre comme un processus MDP et met en œuvre le processus de génération d’ordre avec des structures encodeur-décodeur, optimisant le modèle proposé avec le RL basé sur un mécanisme de récompense conçu pour chaque ordre. L’ordre généré est ensuite traité en utilisant une sélection de variables pour obtenir le graphe final.

Chen *et al.* [81] proposent un modèle Graph-to-Sequence (Graph2Seq) basé sur RL, utilisant l’algorithme de formation de séquence auto-critique (SCST) [82] pour optimiser directement la métrique d’évaluation pour la génération de questions en langues naturelles.

Découvrir et comprendre les mécanismes causaux peut être défini comme la recherche de graphes acycliques dirigés (Directed Acyclic Graph, DAG) qui minimisent des fonctions de score définies. La recherche de l’espace DAG ou la découverte de conditions implicites a généralement une complexité élevée. Zhu *et al.* [83] exploitent le RL pour trouver le DAG sous-jacent à partir de l’espace des graphes sans avoir besoin de fonctions de score lisses. Cet algorithme utilise l’Actor-Critic comme algorithme de recherche et produit le graphe avec la meilleure récompense parmi tous les graphes générés pendant l’entraînement. Cependant, cette méthode a un coût de calcul élevé, et l’espace composé de graphes dirigés est généralement difficile à explorer complètement. L’apprentissage par renforcement appliqué aux graphes représente une avancée prometteuse dans l’exploitation des structures de données complexes pour résoudre des problèmes variés allant de la classification des nœuds à la découverte de motifs causaux. Les méthodes explorées dans cette section démontrent l’efficacité du RL et des GNNs pour améliorer les représentations de graphes et optimiser les processus de décision à l’échelle des graphes. Cependant, malgré les avancées notables, des défis subsistent, notamment la gestion de la complexité des graphes dirigés et l’intégration des relations hétérogènes des nœuds et arêtes. Ces limitations offrent des pistes de recherche futures pour affiner ces approches et renforcer leur applicabilité dans des contextes plus larges et diversifiés.

3.3.1 Applications Réelles avec l’Apprentissage par Renforcement sur les Graphes

Pour des applications plus pratiques, par exemple la prédiction du flux de trafic, les travaux se concentrent exploitant les informations statistiques de flux de trafic à l’intérieur ou entre des régions et les données de trafic fournies par les institutions ou organisations gouvernementales concernées. IG-RL [84] dédié aux réseaux de transport dynamiques se concentre sur le problème de contrôle des signaux de trafic, en utilisant le GCN pour apprendre les entités en tant d’agrégations de nœuds et l’algorithme Q-learning est employé pour entraîner le modèle. Cette méthode permet la représentation et l’utilisation de la demande de trafic et de la structure du réseau routier de manière appropriée, indépendamment du nombre d’entités et de leur emplacement dans le réseau routier. Qiu *et al.* [85] utilisent un GCN pour représenter la fonction de valeur et la fonction de politique dans le problème de péage électronique dynamique (DETC) et résolvent ce problème avec un algorithme RL multi-agents collaboratif. De plus, l’optimisation des services de covoiturage à la demande [86], la prise de décision de changement de voie pour les véhicules autonomes connectés [87], et le contrôle adaptatif des signaux de trafic [88] sont également étudiés.

Yang *et al.* [89] proposent un modèle de prédiction de diffusion d’information à grande échelle permettant d’intégrer des informations macro et micro. Meiom *et al.* [90] proposent une méthode RL pour contrôler un processus dynamique partiellement observé sur un graphe avec un nombre limité d’interventions, qui est appliqué avec succès pour freiner la propagation d’une épidémie. Dans une autre application de contrôle des épidémies, l’algorithme RAI [91] exploite les relations sociales entre les dispositifs mobiles dans l’Internet social des objets (SIoT) pour aider à contrôler la propagation du COVID-19 en allouant des ressources de protection limitées aux individus influents par une identification précoce des cas suspects. De plus, certains algorithmes pour la recherche de nœuds clés [92], et le démantèlement des réseaux [93] peuvent être appliqué au problème de contrôle des épidémies. À ce jour, il y a eu peu de recherches sur l’utilisation des méthodes GRL pour la dynamique des réseaux, et les méthodes existantes se sont concentrées uniquement sur des problèmes tels que le contrôle des maladies épidémiques. Nous croyons que le modèle de dynamique des réseaux basé sur GRL mérite une recherche et une investigation plus approfondies.

La version dynamique de nombreuses données de fouille de graphes est un objet d’étude critique pour traiter les réseaux évolutifs. L’algorithme GTA-RL [94] est un algorithme d’apprentissage heuristique basé conçu pour les graphes, afin de résoudre des problèmes d’optimisation combinatoire dynamique NP-difficile, en proposant un encodeur de caractéristiques temporelles capable d’intégrer des instances et un décodeur capable de se concentrer sur les caractéristiques intégrées pour trouver des instances d’un problème combinatoire donné. DeepOpt [95] utilise une méthode DRL pour résoudre le problème des politiques de placement pour plusieurs types de ressources dans une fonction de réseau virtuel (VNF). L’algorithme proposé par Silver *et al.* [96] construit la méthode de recherche de chemin le plus court dans les graphes spatio-temporels basés sur des sous-graphes statiques et dynamiques, et utilise un algorithme de Proximal Policy Optimization (PPO) [97] pour entraîner un agent à optimiser la politique état-action pour résoudre le problème de planification de chemin dans les tâches réelles.

Les techniques GRL sont couramment utilisées dans les applications de soutien à la décision clinique. Wang *et al.* [98] proposent un modèle de convolution de graphes RL pour résoudre les problèmes de corrélation et de séquence des médicaments. La méthode conçoit un mécanisme DQN pour apprendre les interactions entre les médicaments. BN-GNN [99] est un cadre de représentation du réseau cérébral, qui permet de déterminer le nombre optimal d’agrégations de caractéristiques avec DQN. D’autres auteurs ont proposé de combiner un graphe de connaissances médicales et des méthodes DRL pour aider les patients à décrire les caractéristiques des maladies et fournir une précision dans le diagnostic des maladies [100].

Les applications réelles de l’apprentissage par renforcement sur les graphes démontrent son potentiel à résoudre des problèmes complexes dans divers domaines, tels que la gestion du trafic, le contrôle des épidémies et la prise de décision clinique. Les méthodes combinant les GNNs avec le RL permettent d’exploiter efficacement la structure des graphes dynamiques pour des tâches telles que l’optimisation des signaux de trafic ou le diagnostic médical. Malgré les avancées significatives, de nombreux défis demeurent, notamment dans la gestion de la dynamique des réseaux en temps réel.

3.4 Discussion

Le modèle de convolutions des graphe-ST (ST-graphe convolution) sont couramment employés dans diverses tâches, notamment l’imagerie médicale [10], la reconnaissance d’activités humaines [38] et les tâches liées à la gestion du trafic [55]. Ce modèle considère à la fois les voisinages temporels et spatiaux. Un autre modèle important est le réseau de convolution multi-graphe, utilisé dans des tâches où des graphes de caractéristiques supplémentaires sont intégrés [15], [22], [44]. Les RNN et LSTM ont montré leur efficacité pour extraire les caractéristiques temporelles [8], [22], tandis que les GNN ou GraphRNN sont utilisés pour améliorer l’encodage des nœuds.

Il existe deux approches principales pour traiter les informations spatiales et temporelles (résumés dans les tableaux 3.1 et 3.2, le traitement combiné et le traitement séparé : **Les modèles de traitement combiné**, tels que 3D GCN [14], Dynamic STGCN [49], H-STGCN [102] et D-STG-CNN [43], traitent simultanément les informations spatiales et temporelles. Certains modèles intègrent des modules LSTM ou GRU pour traiter conjointement les informations temporelles et spatiales [62], capturant ainsi les deux dimensions de manière holistique pour améliorer les performances.

L’approche séparée L’approche de traitement séparé utilise différents modèles pour les informations spatiales (GCN) et temporelles (RNN ou 1D-CNN) [13]. Cette approche est plus répandue, avec de nombreux modèles basés sur des architectures GCN. Le domaine a évolué vers des méthodes sophistiquées pour gérer les données spatiales et temporelles, incluant des mécanismes d’attention [101], [103], des modèles dynamiques [107], [108] et des matrices d’entrée personnalisées pour s’adapter aux changements de structure du graphe [49].

Il existe des variations dans la mise en œuvre de l’architecture STGCN. Certains modèles utilisent un seul GCN pour les voisinages spatiaux et temporels [10], [39], [59], tandis que d’autres utilisent des couches séparées ou combinent le GCN avec des convolutions 1D [11], [12]. Certains modèles utilisent des mécanismes d’attention, tandis que d’autres non, ce qui impacte les performances.

TABLE 3.1 – Résumé des des méthodes (partie I).

Modèle / Article	Traitement spatiotemporel	Traitement spatial	Traitement temporel	Commentaire	Tâche
3D GCN [14]	Combiné	3DGNN			Prévision du trafic
GNet [56]	Séparé	GCN	LSTM		Prévision du trafic
MGCN [22]	Séparé	MGCN	LSTM		Prévision du trafic
DMV-STVGNN [51]	Séparé	GAT	1D-CNN		Prévision du trafic
DSTGCN [11]	Séparé	Spatial-GCN	Temporal-GCN		Prévision du trafic
DMf-ST Framework [48]	Séparé	GCN spatial dynamique	GCN temporel		Prévision du trafic
Dynamic STGCN [49]	Combiné	GCN spatial	GCN temporel à portes	Matrice laplacienne dynamique définie	Prévision du trafic
STN [41]	Combiné	GCN/FCN		GCN/CNN/FCN multimodèles	Prévision du trafic
LDNGRU [54]	Séparé	LDN(GCN)	GRU		Prévision du trafic
Attention ST-graphs [101]	Séparé	GCN	GRU	Le cadre utilise l'attention pour évaluer l'impact des nœuds	Prévision du trafic
Ensemble DL [57]	Combiné	GCN/LSTM*		*LSTM avec mémoire spatiotemporelle	Prévision du trafic
ST-MGCN [50]	Séparé	MGCN	RNN		Prévision du trafic
LCRNN [47]	Séparé	Look-up CNN	LSTM	Modèle basé sur des modèles d'apprentissage profond non liés aux graphes	Prévision du trafic
S-STGCN [59]	Combiné	STGCN		GCN avec des voisins spatiaux et temporels	Prévision du trafic
H-STGCN [102]	Combiné	GCN/GRU			Prévision du trafic
STGCN [55]	Séparé	Spatial-GCN	Temporal-GCN		Prévision du trafic
D-STG-CNN [43]	Combiné	STGCN	1D convolution ajoutée pour des entrées supplémentaires	GCN avec voisins spatiaux et temporels	Prévision du trafic
STE-DRMGCN [44]	Séparé	Res MGCN	LSTM		Prévision du trafic
STGNN [103]	Combiné	STGCN*		GCN avec voisins spatiaux et temporels, *GCN utilise des mécanismes d'attention	Prévision du trafic
ASTG [12]	Séparé	GCN	1D-CNN/GRU		Prévision du trafic
STG-LSTM [53]	Séparé	GCN	LSTM	Encodeur-décodeur LSTM-GCN empilé	Prévision du trafic
T-MGCN [15]	Séparé	MGCN	RNN		Prévision du trafic
ST-MGCN [52]	Séparé	MGCN	LSTM	Couches encodeur-décodeur LSTM	Prévision du trafic
DDP-GCN [104]	Séparé	1D CNN	1D CNN		Prévision du trafic
MGCN [58]	Séparé	GAT/ChebNet	LSTM	Modèle temporel incluant LSTM et mécanisme d'auto-attention	Prévision du trafic
STGNN [40]	Séparé	GNN	1D-CNN		Prévision du trafic

TABLE 3.2 – Résumé des méthodes (partie II).

Modèle / Article	Traitement spatio-temporel	Traitement spatial	Traitement temporel	Commentaire	Tâche
STCGN [105]	Séparé	GCN	3D GCN/LSTM	3DGCN combiné avec LSTM	Prévision du trafic
GCN-GRRU [42]	Séparé	GCN	GRU		Prévision du trafic
GCN-3DCNN [45]	Combiné	GCN/3D CNN		GCN utilisé pour capturer l’information spatio-temporelle, 3DCNN utilisé pour intégrer les motifs	Prévision du trafic
TGCN [106]	Séparé	GCN	GRU		Prévision du trafic
GSNN [46]	Combiné	Transformers			Prévision du trafic
STGNN [70]		GCN spectrale	1D CNN	GCN opérant sur la transformation de Fourier des graphes	Général
STG-RNN [60]	Combiné	GNN			Général
AutoST [67]	Combiné	GCN		Travaux focalisés sur AutoDL	Général
St-unet [62]	Combiné	GCN	GRU	Modules GCN/GRU empilés dans une structure U-Net (pooling-unpooling)	Général
GRR-GNN [42]	Séparé	GNN	LSTM/GRU		Prévision du trafic
DMGFN [107]	Combiné	STGCN		Modèle agrège des caractéristiques spatio-temporelles de différents voisins et les transmet à GCN	Prévision du trafic
STGSAM [108]	Combiné	STGCN		GCN avec une méthode d’agrégation sur mesure pour extraire les caractéristiques spatio-temporelles en une seule passe	Prévision du trafic

Lorsqu’il s’agit de l’apprentissage par renforcement dans le contexte des graphes-ST ou des multi-graphes, il est évident qu’il existe un chevauchement significatif entre les objectifs et les tâches abordés par l’apprentissage profond et les techniques de RL. Les deux paradigmes visent souvent à résoudre des problèmes de prévision et utilisent fréquemment des réseaux neuronaux similaires pour modéliser des dépendances complexes dans les données.

Cependant, l’application du RL aux graphes-STs reste relativement limitée par rapport à son utilisation plus large dans d’autres domaines. Cette rareté révèle un domaine de recherche en pleine expansion et prometteur avec un potentiel énorme pour l’exploration et l’innovation. Alors que les méthodes de apprentissage profond ont connu une application et un succès généralisés dans divers domaines, les défis et opportunités uniques présentés par les données spatio-temporelles nécessitent des approches spécialisées qui peuvent tirer parti des forces du RL.

L’utilisation des multi-graphes dans le RL peut enrichir le processus d’apprentissage en incorporant divers types de relations et d’interactions dans les données. Par exemple, dans le contexte des villes intelligentes, les multi-graphes peuvent représenter divers

types d’interactions entre les entités, telles que les réseaux de transport, les réseaux de communication et les systèmes de distribution d’énergie [109]. Les algorithmes de RL peuvent alors être conçus pour optimiser simultanément plusieurs objectifs, comme la réduction de la congestion, l’amélioration de l’efficacité des communications et l’équilibrage des charges énergétiques.

Les tâches de DL et RL sur les graphe-STs peuvent être résumées comme des problèmes de prévision. Quelle que soit l’application, qu’il s’agisse d’une tâche liée au trafic, à l’imagerie médicale ou orientée vers l’ingénierie, l’objectif peut être résumé à la prédiction de valeurs futures, soit pour les nœuds, soit pour les arêtes, en se basant sur des observations antérieures. Un manque de diversité de tâches peut facilement être observé. En outre, on remarque une rareté des travaux investiguant l’utilisation du apprentissage profond pour résoudre des problèmes théoriques classiques et fondamentaux des graphes-ST comme la découverte de séparateurs temporels, le clustering de nœuds spatio-temporels, la découverte de motifs spatio-temporels fréquents (ce qui est le sujet de cette thèse). De plus, bien que le pré-entraînement et l’apprentissage par transfert aient été largement explorés dans le domaine du apprentissage profond, leurs applications aux graphe-STs restent relativement inexplorées. Il y a un potentiel pour le pré-entraînement des réseaux de neurones sur les graphes-ST sur de grands ensembles de données avec des structures similaires et l’utilisation de l’apprentissage par transfert pour adapter le modèle à des ensembles de données plus petits et spécifiques. Cela pourrait mener à une amélioration des performances et à une réduction des temps d’entraînement et d’exécution pour les modèles d’apprentissage profond sur les graphes-ST.

Les applications des deux approches, l’apprentissage profond et l’apprentissage par renforcement, sont pour l’instant limitées à des problèmes pratiques. Cependant, mettre davantage l’accent sur leur utilisation pour résoudre des problèmes fondamentaux des graphes pourrait offrir des améliorations majeures en termes de temps d’exécution. Dans cette thèse, nous visons à employer ces deux approches pour la tâche de la découverte de motifs fréquents dans les multi-graphes. Ainsi, dans la section suivante, nous présentons l’état de l’art de la découverte de motifs fréquents dans les graphes en général, dans les graphes spatio-temporels et les multi-graphes, et nous discutons du potentiel de l’apprentissage profond et de l’apprentissage par renforcement pour cette tâche.

3.5 Recherche des motifs fréquents dans les graphes

Les algorithmes de fouille de motifs procèdent généralement par une exploration itérative des représentations possibles de sous-graphes, en commençant par des motifs simples tels que des sommets ou des arêtes. Au cours de ce processus, le système analyse

les motifs candidats et élimine ceux qui ne peuvent pas contribuer à la solution finale, tout en développant les motifs restants en ajoutant progressivement des sommets ou des arêtes. L’exploration continue jusqu’à ce qu’il ne soit plus possible de développer davantage les motifs. Il existe un large corpus de travaux a étudié des approches exactes et approximatives pour la fouille de sous-graphes fréquents, en proposant diverses stratégies visant à optimiser l’espace de recherche. Parmi ces techniques, on retrouve l’élagage, la compression ainsi que l’emploi d’étiquetages canoniques, qui permettent de réduire de manière significative la complexité de la recherche.

Des algorithmes représentatifs pour la découverte de sous-graphes fréquents dans des graphes, tels que SIGRAM [110], GRAMI [111], et MuGram [112], ont été développés pour relever ce défi. SIGRAM identifie les sommets apparaissant au moins un certain nombre de fois dans le graphe d’entrée, stocke leurs représentations et les utilise pour étendre les motifs et trouver des sous-graphes plus grands. Cependant, cette approche peut être gourmande en mémoire. GRAMI contourne ce problème en trouvant uniquement un nombre suffisant d’instances de tout sous-graphe pour établir sa fréquence en traitant les calculs de support comme un problème de satisfaction de contraintes. GRAMI est capable de gérer à la fois des graphes dirigés et non dirigés, des contraintes simples et une recherche approximative. Bien qu’efficaces, ni SIGRAM ni GRAMI ne sont adaptés à la fouille de graphes multi-relationnels, dans lesquels plusieurs arêtes de types distincts peuvent connecter une même paire de nœuds. MuGram a été proposé pour résoudre ce cas, en effectuant une recherche en profondeur à partir des arêtes fréquentes pour trouver tous les sous-graphes fréquents et en appliquant des heuristiques pour confirmer la fréquence d’un sous-graphe [112].

Apriori-based Subgraph Mining (AGM) est un algorithme pour la fouille de sous-graphes fréquents dans des graphes non dirigés. Il génère des sous-graphes candidats en étendant de manière itérative des sous-graphes fréquents plus petits et élimine ceux qui ne répondent pas au seuil de support minimal [113]. gSpan (graph-based Substructure pattern mining) [114] utilise une approche de croissance de motif et introduit un ordre lexicographique novateur pour les sous-graphes. Cette méthode explore efficacement l’espace de recherche en étendant les sous-graphes fréquents. CloseGraph [115] est une extension de gSpan conçue pour découvrir des sous-graphes fréquents fermés, qui sont des sous-graphes maximaux apparaissant fréquemment. Cette approche réduit la redondance en se concentrant sur les motifs fermés, qui sont plus informatifs et concis. MoFa (Molecular Fragment miner) [116] est adapté à la fouille de sous-graphes fréquents dans les graphes moléculaires. Il emploie une approche de croissance de motif similaire à gSpan mais est optimisé pour les besoins spécifiques de l’informatique chimique, en se concentrant sur les fragments moléculaires. SPIN (SPecific Interaction Network) [117] cible les réseaux biologiques, en utilisant des heuristiques spécifiques au domaine pour fouiller les motifs d’interaction. Cet algorithme est conçu pour gérer la complexité et la

spécificité des données biologiques, en se concentrant sur les interactions significatives au sein du réseau. Gaston [118] combine des approches basées sur les chemins et les fragments pour fouiller les sous-graphes fréquents dans les graphes étiquetés. Il traite efficacement les données d’informatique chimique en générant des sous-graphes candidats par extensions de chemins et compositions de fragments. SUBDUE [119] emploie une approche basée sur des heuristiques pour découvrir des sous-graphes qui compressent le graphe d’entrée. Cette méthode équilibre la fréquence et la complexité structurelle, la rendant adaptée aux graphes étiquetés et non étiquetés. Elle utilise un principe de longueur minimale de description (MDL) pour trouver des motifs offrant la meilleure compression. MUSE (MULTiple Subgraph Enumeration) [120] est conçu pour les graphes dynamiques, il effectue la mise à jour des motifs fréquents à mesure que le graphe évolue. Il gère efficacement les changements dans la structure du graphe en mettant à jour de manière incrémentale les comptes de fréquence des sous-graphes. Dans [121], Damiand et al. proposent un algorithme polynomial pour résoudre le problème d’isomorphisme de sous-graphes dans le cas de graphes plongés dans des espaces n -dimensionnels, représentés par des cartes combinatoires. Cet algorithme permet de déterminer efficacement si une carte combinatoire est isomorphe à une sous-partie d’une autre, avec des applications potentielles en vision par ordinateur et en modélisation géométrique. Par ailleurs, dans [122], Solnon et al. introduisent PLAGRAM, un algorithme de fouille conçu pour identifier des motifs fréquents dans des graphes spatio-temporels. En exploitant la planéité des graphes et en adoptant une nouvelle stratégie d’extension des motifs, PLAGRAM propose une solution efficace pour l’analyse de données complexes où les relations spatiales et temporelles jouent un rôle central. Enfin, dans [123], Deville et al. présentent une version optimisée de PLAGRAM adaptée aux graphes structurés en grilles spatio-temporelles (2D+t). Cette approche est particulièrement utile pour analyser les automates cellulaires et identifier des motifs récurrents dans des données temporelles structurées, ouvrant ainsi des perspectives pour l’étude de systèmes dynamiques. Ces contributions mettent en avant des solutions innovantes pour l’isomorphisme de sous-graphes et la fouille de motifs dans des graphes, tout en offrant des applications variées en modélisation géométrique, vision par ordinateur et analyse de données spatio-temporelles complexes.

Ces algorithmes représentent une large gamme de techniques et d’optimisations pour la fouille de divers types de graphes, démontrant la polyvalence et l’adaptabilité des méthodes de fouille de sous-graphes à différents domaines et structures de données. Les avancées récentes en DL ont encore élargi les capacités de ces algorithmes. Des travaux récents d’apprentissage profond tels que GraphSAGE [124] et GCN [125] ont été intégrés avec des algorithmes de fouille de motifs fréquents. GraphSAGE génère des plongements de nœuds en échantillonnant et en agrégeant des caractéristiques à partir du voisinage local d’un nœud, ce qui est utilisé pour identifier des motifs de

sous-graphes fréquents, en intégrant les informations sur le voisinage du nœud pendant le processus de recherche. GCN utilise des convolutions spectrales de graphes pour capturer la structure du graphe et les interactions des nœuds, améliorant l’identification des motifs de sous-graphes dans les grands graphes. Ces développements mettent en lumière l’innovation et le raffinement continu dans le domaine de la fouille de sous-graphes, relevant de nouveaux défis et élargissant l’applicabilité de ces techniques à des données plus complexes et dynamiques.

Dans [121], Damiand et al. proposent un algorithme polynomial pour résoudre le problème d’isomorphisme de sous-graphes dans le cas de graphes plongés dans des espaces n -dimensionnels, représentés par des cartes combinatoires. Cet algorithme permet de déterminer efficacement si une carte combinatoire est isomorphe à une sous-partie d’une autre, avec des applications potentielles en vision par ordinateur et en modélisation géométrique. Par ailleurs, dans [122], Solnon et al. introduisent PLAGRAM, un algorithme de fouille conçu pour identifier des motifs fréquents dans des graphes spatio-temporels. En exploitant la planéité des graphes et en adoptant une nouvelle stratégie d’extension des motifs, PLAGRAM propose une solution efficace pour l’analyse de données complexes où les relations spatiales et temporelles jouent un rôle central. Enfin, dans [123], Deville et al. présentent une version optimisée de PLAGRAM adaptée aux graphes structurés en grilles spatio-temporelles ($2D+t$). Cette approche est particulièrement utile pour analyser les automates cellulaires et identifier des motifs récurrents dans des données temporelles structurées, ouvrant ainsi des perspectives pour l’étude de systèmes dynamiques. Ces contributions mettent en avant des solutions innovantes pour l’isomorphisme de sous-graphes et la fouille de motifs dans des graphes, tout en offrant des applications variées en modélisation géométrique, vision par ordinateur et analyse de données spatio-temporelles complexes.

En faisant un pas de plus dans l’utilisation du DL pour la découverte de motifs, Rex Ying *et al.* [126] ont proposé SPMiner, la première approche neuronale dédiée à l’extraction de motifs fréquents dans de grands graphes. Le fonctionnement de SPMiner repose sur deux étapes fondamentales. Dans un premier temps, l’algorithme utilise un GNN pour projeter les voisinages ancrés sur les nœuds des graphes d’entrée dans un espace d’intégration ordonné. Ensuite, il identifie les motifs fréquents en effectuant des parcours dans cet espace de plongement. Le GNN de SPMiner est capable d’extraire des motifs grâce à sa capacité à discerner l’isomorphisme de sous-graphes, facilitant ainsi l’extraction de motifs indépendamment du type de réseau. Cette capacité est obtenue en apprenant une approche méthodologique plutôt que des instances de données spécifiques [127].

Malgré l’efficacité démontrée de SPMiner dans l’extraction de motifs à partir de graphes simples, son application à d’autres formats de graphes, en particulier les multi-graphes, reste inexplorée. Cela souligne l’enjeu de cette thèse, qui vise à adapter et à étendre les

capacités de SPMiner pour permettre la découverte de motifs dans les multi-graphes. SPMiner sera discuté plus en détail dans le prochain chapitre, car il est considéré comme référence pour notre première contribution dans cette thèse.

Deuxième partie

Développements et résultats

4

Méthode Multi-SPMiner

Sommaire

4.1	Introduction	87
4.2	La méthode de référence SPMiner	88
4.2.1	Première phase : Encodage des Sous-graphes Candidats	89
4.2.2	Deuxième phase : Procédure de Recherche de Motifs	90
4.2.3	Analyse et Évaluation	92
4.3	Développement de l'approche Multi-SPMiner	93
4.4	Tests du modèle Multi-SPMiner sur des données synthétiques	97
4.4.1	Génération de données synthétiques	97
4.4.2	Modalités des tests et résultats	97
4.5	Conclusion	101

4.1 Introduction

Ce chapitre présente la première étape de notre recherche : l'adaptation de la méthode SPMiner aux graphes spatio-temporels, une avancée que nous avons nommée Multi-SPMiner. Face au défi complexe de l'extraction de motifs fréquents dans les grands graphes spatio-temporels, nous avons choisi SPMiner comme point de départ, un algorithme initialement conçu pour les graphes simples non orientés et non étiquetés. Les graphes spatio-temporels, représentés dans notre travail sous forme de multigraphes, présentent des défis spécifiques en raison de leur structure intégrant des relations spatiales, temporelles et de filiation. Notre objectif avec Multi-SPMiner est d'étendre les capacités de SPMiner pour gérer efficacement ces structures de données multidimensionnelles, tout en préservant ses atouts en termes d'efficacité computationnelle. Dans ce chapitre, nous commencerons par une analyse approfondie du fonctionnement de SPMiner, essentielle pour identifier les points d'adaptation nécessaires. Nous détaillerons ensuite les modifications et améliorations proposées pour transformer SPMiner en Multi-SPMiner, en mettant l'accent sur la prise en compte de la nature des graphes spatio-temporels. Une part importante de notre discussion sera consacrée à l'évaluation de Multi-SPMiner sur des ensembles de données synthétiques, nous permettant de valider notre approche dans un environnement contrôlé. Ensuite, nous effectuerons une comparaison détaillée des performances entre Multi-SPMiner et SPMiner, quantifiant ainsi les améliorations apportées et identifiant les forces et les limites de notre approche adaptée. Pour évaluer la fiabilité de Multi-SPMiner, nous présenterons des tests de robustesse face à diverses altérations des graphes d'entrée, reflétant des scénarios réels où les données peuvent être bruitées ou incomplètes. Enfin, nous examinerons les capacités de généralisation de Multi-SPMiner, un aspect crucial pour déterminer son adaptabilité à différents types et tailles de graphes spatio-temporels. À travers cette analyse complète, ce chapitre pose les fondements de notre approche Multi-SPMiner, ouvrant la voie à des développements ultérieurs proposés dans cette thèse.

4.2 La méthode de référence SPMIner

SPMiner (Subgraph Pattern Miner) [126] est une méthode conçue pour l'extraction de sous-graphes fréquents dans des graphes de grande taille et complexes. L'extraction de motifs de grande taille pose un défi considérable en raison de la complexité combinatoire du comptage de sous-graphes, un problème NP-difficile, et de la croissance exponentielle de l'espace de recherche à mesure que la taille des motifs augmente.

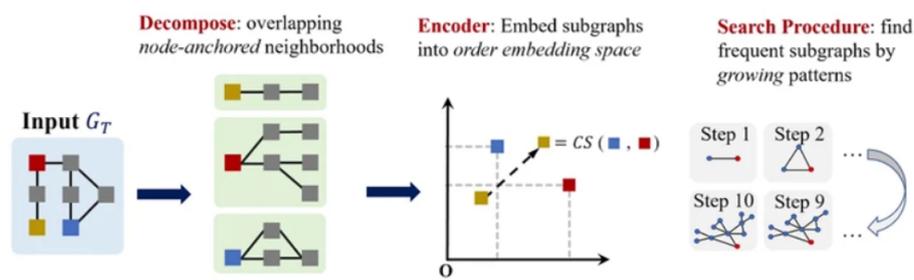


FIGURE 4.1 – SPMIner se compose de deux étapes principales : la phase d'encodage, où les données sont transformées en une représentation appropriée, suivie de la phase de recherche, au cours de laquelle des motifs fréquents sont identifiés à partir des données encodées.

Le problème de l'extraction fréquente de sous-graphes dans un graphe donné ou une collection de graphes consiste à identifier les motifs qui apparaissent le plus souvent. De manière plus formelle : étant donné un graphe cible $G_T = (V_T, E_T)$, où V_T représente l'ensemble des nœuds et E_T l'ensemble des arêtes, et un motif de requête $G_Q = (V_Q, E_Q)$, l'objectif est de déterminer si G_Q est isomorphe à un sous-graphe G'_T de G_T . Cette détermination nécessite de trouver une injection $f : V_Q \rightarrow V_T$ telle que chaque arête de G_Q soit mise en correspondance avec une arête correspondante dans G'_T , c'est-à-dire $(f(v), f(u)) \in E_T \iff (v, u) \in E_Q$. Les objectifs de SPMIner sont les suivants :

1. **Comptage de la fréquence d'un motif donné Q dans G_T** — une tâche NP-difficile nécessitant de vérifier l'inclusion de Q avec tous les sous-graphes potentiels de G_T .
2. **Recherche des motifs ayant la fréquence la plus élevée** — un problème complexe avec une croissance exponentielle de l'espace de recherche des motifs en fonction de leur taille.

SPMiner utilise une approche d'apprentissage par représentation pour surmonter ces défis, divisée en deux étapes principales : l'encodage des sous-graphes candidats et la recherche de motifs fréquents (figure 4.1). La première étape consiste à générer différentes structures de graphes échantillons et à entraîner un modèle de GNN pour distinguer si l'un des graphes est un sous-graphe de l'autre ou non. Par ce processus, le modèle apprend une fonction qui sera utilisée ultérieurement pour prendre des décisions

dans la seconde phase, qui est la phase de recherche. Dans cette deuxième phase, à partir de nœuds échantillons, nous essayons de créer des motifs. À chaque étape, le nœud ajouté au motif ou au graphe est choisi en exploitant la fonction apprise lors de la première phase. Dans la suite, nous allons introduire les différentes étapes et les modifications proposées que nous avons appliquées pour obtenir notre modèle.

4.2.1 Première phase : Encodage des Sous-graphes Candidats

La première étape de SPMiner consiste à décomposer le graphe cible $G_T = (V_T, E_T)$ en sous-graphes de voisinage à K -sauts, ancrés autour de chaque nœud $v \in V_T$. Ces sous-graphes sont ensuite encodés dans un espace ordonné de plongement à l'aide d'un réseau de neurones de graphe (GNN). Cet encodage vise à capturer les relations structurelles entre les sous-graphes tout en maintenant les relations d'ordre partiel induites par l'isomorphisme de sous-graphe.

- **Concept d'espace de plongement ordonné** : L'espace de plongement ordonné est une technique de représentation qui exploite les relations géométriques entre les plongements pour modéliser une structure d'ordre partiel. Cette approche est bien adaptée pour représenter les relations de sous-graphes car l'isomorphisme de sous-graphes induit une structure d'ordre partiel sur l'ensemble des graphes [126]. Formellement, une relation d'ordre partiel \leq est définie sur l'ensemble des graphes \mathcal{G} par : $A \leq B$ si et seulement si le graphe A est isomorphe à un sous-graphe de B .
- **Décomposition en voisinages à K -sauts** : Pour chaque nœud $v \in V_T$, le voisinage à K -sauts, noté G_v , comprend tous les nœuds situés à une distance de chemin d'au plus K de v . Cette extraction est réalisée à l'aide d'une recherche en largeur (Breadth First Search, BFS), garantissant que tous les nœuds dans le voisinage sont accessibles à partir du nœud v . Le choix de la valeur K est crucial car il détermine la taille et la complexité des sous-graphes extraits, influençant directement le taux de motifs trouvés et l'efficacité du plongement.
- **Fonction de plongement via GNN** : Une fonction de plongement $\phi : \mathcal{G} \rightarrow \mathbb{R}^n$ est apprise à l'aide d'un GNN, qui vise à préserver l'ordre d'inclusion des sous-graphes dans l'espace de plongement. La contrainte d'ordre est imposée de telle manière que, pour deux graphes $A, B \in \mathcal{G}$, $\phi(A) \leq \phi(B)$ si et seulement si A est isomorphe à un sous-graphe de B . Cela se traduit géométriquement par le fait le plongement de A doit se trouver à gauche et sous le plongement de B , dans un espace latent à 2 dimensions. Les voisinages sont encodés à l'aide d'un GNN dans un espace de plongement ordonné, suivant le principe présenté par [126].

- **Entraînement de la fonction de plongement** : L'entraînement de la fonction de plongement ϕ repose sur la création d'un ensemble de paires positives P (les paires (A, B) où A est un sous-graphe de B) et un ensemble de paires négatives N (les paires (A', B') où A' n'est pas un sous-graphe de B'). La fonction de plongement est optimisée en minimisant la fonction de perte à marge maximale suivante :

$$loss = \sum_{(A,B) \in P} E(A, B) + \sum_{(A',B') \in N} \max(0, \alpha - E(A', B')) \quad (4.1)$$

où α est un hyperparamètre de marge et E est la pénalité de plongement entre deux graphes A et B , définie comme :

$$E(A, B) = \|\max(0, \phi(A) - \phi(B))\|^2.$$

Cette pénalité de plongement $E(A, B)$ quantifie la distance entre les plongements de deux graphes, garantissant que la relation d'ordre partiel est préservée dans l'espace de plongement. Cette mesure est cruciale pour maintenir la hiérarchie des sous-graphes dans l'espace vectoriel, essentielle pour une recherche efficace de motifs.

4.2.2 Deuxième phase : Procédure de Recherche de Motifs

Après l'encodage des sous-graphes, SPMiner engage une procédure de recherche pour identifier les motifs fréquents dans l'espace ordonné de plongement. Cette recherche se déroule sous la forme d'une "marche monotone", une exploration itérative où des nœuds sont ajoutés de manière séquentielle pour générer des motifs potentiellement fréquents. L'objectif de la marche dans l'espace est d'atteindre une région où nous maximisons le nombre de voisinages contenant le motif que nous construisons (figure 4.2). Cela permet d'estimer rapidement la fréquence du motif sans avoir à le compter manuellement.

- **Marche monotone dans l'espace de plongement** : Une marche monotone est une séquence de pas où, à chaque itération, un nœud ou une arête est ajouté au motif actuel pour générer un nouveau motif G_{k-1} de taille k , en partant de $k = 1$. À chaque étape, la nouvelle configuration est choisie pour minimiser la "marge totale" $m(G)$ dans l'espace de plongement, définie comme la somme des pénalités de plongement pour tous les sous-graphes voisins $N \in \mathcal{N}$ du graphe G :

$$m(G) = \sum_{N \in \mathcal{N}} \|\max(0, \phi(G) - \phi(N))\|^2,$$

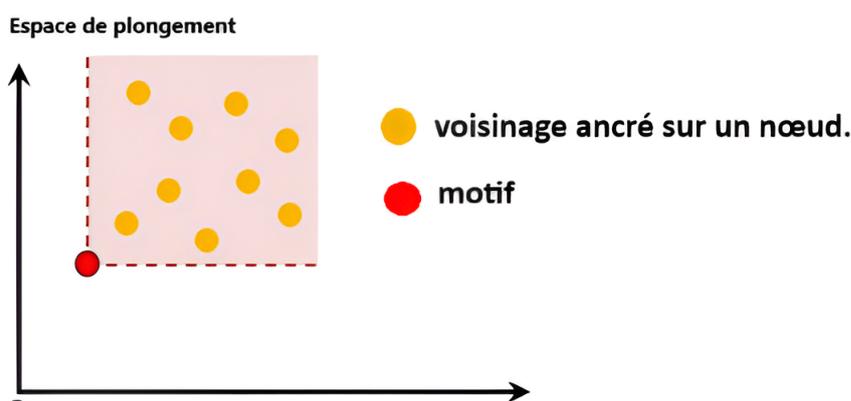


FIGURE 4.2 – Région Super-graphe : Tous les points de la région ombragée en rouge correspondent à des voisinages dans G_T qui contiennent G_Q (le motif).

où \mathcal{N} représente l'ensemble des sous-graphes de voisinage pour chaque nœud $v \in G$. Cette fonction d'objectif permet d'orienter la recherche vers des motifs qui minimisent la distance par rapport aux sous-graphes voisins dans l'espace de plongement, augmentant ainsi la probabilité de découvrir des motifs fréquents.

- **Sélection des motifs fréquents** : À chaque étape de la marche monotone, le motif est étendu en ajoutant un nœud adjacent et ses arêtes correspondantes. Pour maximiser la découverte de motifs fréquents, plusieurs nœuds de départ (ou *seed nodes*) sont utilisés. Ces nœuds sont sélectionnés aléatoirement dans le graphe, et chaque extension est choisie pour minimiser la marge totale $m(G_{k-1})$. Les motifs résultants sont évalués en fonction du nombre de fois où ils apparaissent pendant la marche, et les motifs les plus fréquents sont sélectionnés.
- **Optimisation et sélection finale** : Le motif fréquent trouvé, noté G_{freq} , est celui qui, au cours de la marche, atteint un sous-graphe de taille k minimisant la marge totale :

$$\phi(G_{\text{freq}}) = \arg \min_{G \in \mathcal{G}} m(G).$$

Cette optimisation garantit que les motifs identifiés sont non seulement fréquents mais aussi représentatifs de la structure globale du graphe cible. Le processus de minimisation de la marge totale permet de converger vers des motifs optimaux dans un espace de recherche potentiellement très vaste.

- **Stratégies de recherche avancées** : Pour améliorer la robustesse et l'efficacité de la procédure de recherche, SPMiner implémente plusieurs stratégies de recherche avancées parmi lesquelles l'utilisateur peut choisir, notamment :

- *Stratégie gloutonne* : Cette stratégie consiste à sélectionner, à chaque étape, le nœud ou l'arête qui minimise localement la fonction de perte à marge (eq. 4.1). Bien que simple, cette approche permet de trouver rapidement des solutions locales optimales.
- *Recherche en faisceau (Beam Search)* : Une approche plus sophistiquée qui explore plusieurs chemins simultanément. À chaque étape, un certain nombre de candidats sont conservés, chacun représentant une extension possible du motif courant. Cette méthode permet de maintenir un bon équilibre entre exploration et exploitation.
- *Recherche d'arbre Monte Carlo (MCTS - Monte Carlo Tree Search)* : Cette méthode utilise des simulations aléatoires pour équilibrer l'exploration de nouveaux motifs potentiels et l'exploitation des motifs connus, en optimisant globalement la recherche de motifs fréquents.

4.2.3 Analyse et Évaluation

SPMiner a été évalué sur plusieurs ensembles de données synthétiques et réels provenant de divers domaines. Ces évaluations ont montré que SPMIner offre une performance supérieure par rapport aux méthodes existantes, tant en termes de taux de motifs trouvés que de rapidité [126].

- **taux de motifs trouvés élevé** : Pour les motifs de petite taille, SPMIner a démontré une capacité exceptionnelle à identifier correctement les motifs les plus fréquents avec un taux de motifs trouvés élevé (96%).
- **Comparaison avec les méthodes exactes (gSpan, Gaston)** : Les méthodes exactes comme gSpan [114] et Gaston [118] deviennent rapidement inutilisables lorsque la taille des motifs dépasse 10 nœuds, en raison de la croissance exponentielle de l'espace de recherche. En revanche, SPMIner maintient une tendance linéaire du temps d'exécution avec l'augmentation de la taille des motifs, démontrant ainsi une meilleure scalabilité et efficacité computationnelle.
- **Capacité de généralisation et robustesse** : L'espace ordonné de plongement permet à SPMIner de généraliser efficacement à de nouveaux graphes de tailles et de structures variées, y compris ceux qui n'ont pas été vus lors de l'entraînement initial du modèle.

Ces résultats de SPMIner, notamment le taux de motifs trouvés élevé, son efficacité computationnelle et sa capacité de généralisation, nous ont fortement motivés à adapter cette méthode aux données spatio-temporelles. Nous avons donc entrepris d'adapter SPMIner pour qu'il puisse traiter efficacement les multigraphes représentant les données

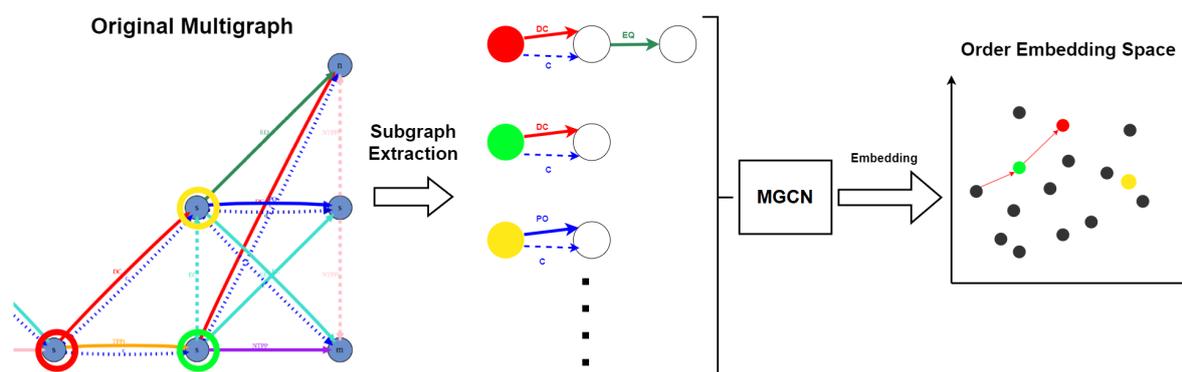


FIGURE 4.3 – Le framework Multi-SPMiner, contrairement au modèle SPMiner de base, utilise un MGCN pour exploiter les relations multigraphes afin de calculer les voisinages, tout en s'appuyant sur le même principe de l'espace ordonné de plongement.

spatio-temporelles, tout en conservant ses avantages en termes de taux de motifs trouvés et d'efficacité. Cette adaptation, que nous appelons Multi-SPMiner, vise à exploiter les forces de SPMiner tout en relevant les défis spécifiques posés par la nature complexe de nos données.

4.3 Développement de l'approche Multi-SPMiner

Multi-SPMiner est conçue pour traiter efficacement les graphes spatio-temporels. Voici comment nous avons adapté les principales étapes de SPMiner :

Première phase : Encodage des Sous-graphes Candidats

- **Modèle GNN adapté** : Multi-SPMiner utilise un réseau de convolution multigraphe (MGCN) comme modèle GNN principal. Ce choix est motivé par la capacité du MGCN à gérer efficacement les multiples types de relations présentes dans les graphes spatio-temporels, incluant les relations spatiales, temporelles et de filiation.
- **Fonction de plongement étendue** : La fonction de plongement $\phi : \mathcal{G} \rightarrow \mathbb{R}^n$ est adaptée pour prendre en compte la nature multidimensionnelle des graphes spatio-temporels. L'espace de plongement ordonné est élargi pour représenter un seul graphe à la fois, permettant ainsi d'inclure une plus grande variété de relations et d'interactions entre les nœuds.
- **Entraînement du modèle** : L'entraînement du MGCN suit un processus similaire à celui de SPMiner, utilisant des paires positives et négatives de sous-graphes

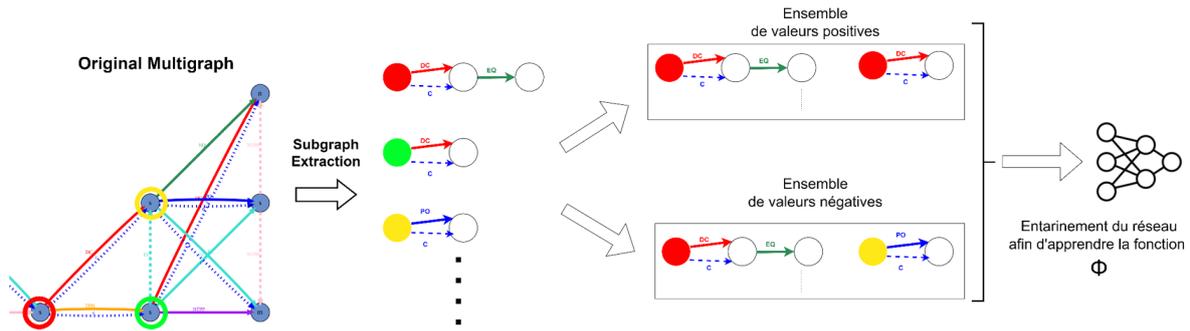


FIGURE 4.4 – Création des ensembles de paires positives et négatives pour l’entraînement du MGCN. P (les paires (A, B) où A est un sous-graphe de B), N (les paires (A, B) où A n’est pas un sous-graphe de B)

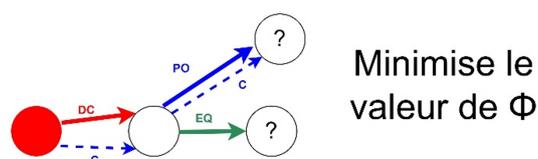
(figure 4.4). Cependant, la définition des paires est adaptée pour prendre en compte les spécificités des graphes spatio-temporels.

Deuxième phase : Procédure de Recherche de Motifs

- **Approche de recherche gloutonne** : Contrairement à SPMiner, Multi-SPMiner considère tous les nœuds pendant le processus d’ajout, adoptant une approche de recherche gloutonne. Cette modification permet une exploration plus exhaustive de l’espace de recherche.
- **Marche monotone adaptée** : Multi-SPMiner développe chaque motif de manière itérative en ajoutant un nœud adjacent, suivant une marche monotone dans l’espace de plongement ordonné.

Une procédure de recherche itérative qui fait croître le motif est utilisée, en commençant par un graphe initial trivial de taille 1 obtenu en échantillonnant aléatoirement un nœud de départ dans l’ensemble de données. Le graphe suivant est généré en ajoutant un nœud adjacent dans le graphe et ses arêtes correspondantes. Plusieurs nœuds de départ sont échantillonnés pour obtenir une estimation robuste, et les motifs résultants d’une taille donnée sont sélectionnés en fonction du nombre de fois où ils ont été rencontrés pendant la marche. Le nœud ajouté à chaque itération est sélectionné de sorte que le nouveau graphe résultant G_{k-1} de taille k minimise la marge totale $m(G_{k-1})$ où m est définie comme :

$$m(G) = \sum_{N \in \mathcal{N}} \|\max(0, \phi(G) - \phi(N))\|^2 \quad (4.2)$$



Minimise le
valeur de Φ

FIGURE 4.5 – À chaque étape, Multi SPMIner cherche à ajouter le nœud qui donne la « plus petite violation totale », en se basant sur la fonction apprise lors de la phase d’encodage.

où : \mathcal{N} est l’ensemble de tous les graphes de voisinage de tous les nœuds $v \in G$ et ϕ est la fonction de plongement d’ordre obtenue par le MGCN à l’étape précédente.

Pour chaque sous-graphe G_{i+1} formé en ajoutant un nœud adjacent à G_i (figure 4.5), l’approximation gloutonne est réalisée par une minimisation progressive à chaque étape :

$$\forall i = 0, 1, \dots, k - 1, G_{i+1} = \arg \min_{G'_i} m(\phi(G'_i))$$

où $\phi(G'_i)$ représente le plongement ordonné du sous-graphe G'_i , G'_i est l’ensemble de tous les graphes voisins possibles obtenus en ajoutant un nœud à G_i et m est la fonction de coût conçue pour minimiser la distance dans l’espace de plongement. La fonction de coût m garantissant ainsi que les motifs sélectionnés maximisent la fréquence d’apparition et la diversité.

- **Flexibilité dans la taille des motifs** : Pendant la phase de test, Multi-SPMiner examine une gamme de valeurs pour le paramètre k (la taille du motif à atteindre), permettant d’extraire des motifs de différentes tailles selon les exigences spécifiques de l’analyse.

Cette adaptation de SPMIner aux graphes spatio-temporels permet à Multi-SPMiner de naviguer efficacement dans l’espace de recherche tout en maintenant la contrainte d’ordre et en prenant en compte les complexités supplémentaires introduites par la nature des données spatio-temporelles. L’algorithme Multi-SPMiner est résumé dans l’algorithme 1.

Construction et utilisation du MGCN Le modèle MGCN de base utilisé pour apprendre la fonction ϕ du Multi-SPMiner, est construit avec plusieurs couches de convolution multigraphe avec des couches de saut multiples pour extraire les attributs structuraux des voisinages de différentes tailles. Le MGCN est spécifiquement conçu pour traiter des données structurées en graphes comportant plusieurs arêtes entre les nœuds. Chaque nœud dans le multigraphe est représenté par un vecteur de caractéristiques,

Input: Graphe G , Taille maximale du motif K , Hyperparamètre de marge α

Output: Motifs fréquents

PhasePlongement(G, K, α);

PhaseRecherche(G, K, s);

Fonction PhasePlongement(*Graphe* G , *Taille maximale de motif* K , *Hyperparamètre de marge* α) :

Décomposer G en voisinages à K sauts ancrés à chaque nœud;
Encoder les voisinages en utilisant un réseau de convolution multigraphique dans un espace de plongement ordonné en apprenant la fonction de plongement ϕ avec une perte à marge maximale;

Fonction PhaseRecherche(*Graphe* G , *Taille maximale du motif* K , *Support* s) :

Initialiser un ensemble vide Δ de motifs fréquents;

for k de 1 à K **do**

for i de 1 à N **do**

 Échantillonner aléatoirement un nœud de départ v à partir de G ;

 Initialiser un sous-graphe G_i avec v ;

while la taille du sous-graphe G_i est inférieure à k **do**

 Ajouter un nœud adjacent qui minimise la marge totale à G_i pour générer G_{i+1} ;

end

 Ajouter G_{i+1} à Δ si le nombre de fois rencontré pendant la marche est $\geq s$;

end

end

return Ensemble de motifs fréquents Δ ;

Algorithm 1: Multi-SPMiner

et plusieurs filtres (ou noyaux de convolution) sont appliqués à ces caractéristiques pour extraire des informations pertinentes.

Ces filtres sont capables de traiter différents types d'arêtes (par exemple, arêtes spatiales, temporelles, de filiation), permettant au modèle de capturer et d'exploiter les différentes relations présentes dans le graphe. Les plongements générés à chaque couche l capturent les propriétés structurelles des voisinages jusqu'à une certaine distance, déterminée par le nombre de couches dans le réseau de convolution de graphes. Cette capacité à capturer des informations locales à différentes échelles est cruciale dans le contexte des tâches d'isomorphisme de sous-graphes, car elle permet au modèle de comparer efficacement les similitudes structurelles locales entre

sous-graphes de tailles et de topologies variées.

4.4 Tests du modèle Multi-SPMiner sur des données synthétiques

4.4.1 Génération de données synthétiques

Pour évaluer la performance de Mutli-SPMiner, ainsi que les différents méthodes proposées dans les chapitre suivants, nous utilisons un générateur de graphes spatio-temporels synthétiques [18]. Ce générateur produit des graphes spatio-temporels sous forme de multigraphes, comme décrit dans le chapitre 2. Le générateur est conçu pour créer des motifs de tailles spécifiées, semblables à ceux rencontrés dans des données réelles, avec une fréquence dépassant un seuil prédéterminé fixé lors du processus de génération. Le processus de génération implique la création aléatoire d'un graphe spatio-temporel G suivant une distribution de Poisson paramétrée par le nombre des nœuds et des arêtes. Un ensemble de motifs sources sert de base, subissant des modifications et étant intégrés dans G . Il est important de noter que les nœuds de G sont attribués de manière aléatoire à des instants temporels successifs, contribuant ainsi à la nature synthétique des graphes générés.

4.4.2 Modalités des tests et résultats

Dans cette section, nous présentons le protocole de test sur des graphes synthétiques du modèle Multi-SPMiner avec MGCN. L'accent est mis sur l'identification des motifs fréquents au sein d'un seul graphes spatio-temporels à la fois. La performance des modèles est évaluée en fonction de le taux de motifs trouvés ou du ratio des motif trouvés. SPMiner, se concentrait sur l'extraction de motifs fréquents dans un ensemble de graphes, de plus, SPMiner ne peut être appliqué directement à nos données, car il a été initialement conçu pour des graphes standards. Dans cette thèse, nous nous focalisons sur l'identification de motifs fréquents au sein de multigraphes individuels (les graphes spatio-temporels). Le changement dans les données sur lesquelles repose notre approche modifie également la méthode de test. Alors que SPMiner se concentre sur l'extraction précise des M motifs les plus fréquents dans un ensemble de graphes, nous mettons l'accent sur le taux de motifs trouvés des motifs fréquents insérés dans le processus de génération d'un graphe unique. Dans cette section, les résultats des tests seront présentés sous forme de taux de motifs trouvés moyenne obtenue pour chaque test, chaque test ayant été répété 10 fois.

TABLE 4.1 – Résultats de 14 tests obtenus pour notre cadre Multi-SPMiner avec différents paramètres. Nombre de nœuds du graphe : le nombre de nœuds dans un seul graphe, Nœuds / temporalité : le nombre maximal de nœuds pour chaque période temporelle dans le graphe, Nombre de relations : une liste définissant le nombre maximal de liens (spatial, spatio-temporel, filiation) dans le motif.

Exp. N°	Nombre de nœuds du graphe	Nœuds / temporalité	Nombre de relations	Nombre de motifs	Ratio des motifs obtenus
1	200	40	[5,5,2]	16 de taille 5, 13 de taille 6	0.632 ± 0.03
2	1000	40	[5,5,2]	60 de taille 5, 50 de taille 6	0.651 ± 0.02
3	1000	40	[5,5,2]	37 de taille 8, 30 de taille 10	0.664 ± 0.03
4	1000	40	[5,5,2]	90 de taille 5, 75 de taille 6	0.612 ± 0.01
5	1000	40	[5,5,2]	56 de taille 8, 45 de taille 10	0.643 ± 0.02
6	1000	40	[5,5,2]	120 de taille 5, 100 de taille 6	0.604 ± 0.02
7	1000	40	[5,5,2]	75 de taille 8, 60 de taille 10	0.631 ± 0.03
8	1000	40	[5,5,2]	150 de taille 5, 125 de taille 6	0.592 ± 0.02
9	1000	40	[5,5,2]	93 de taille 8, 75 de taille 10	0.664 ± 0.01
10	1500	40	[5,5,2]	120 de taille 5, 100 de taille 6	0.623 ± 0.01
11	1000	40	[2,2,2]	150 de taille 4, 100 de taille 6	0.634 ± 0.01
12	1000	40	[2,2,2]	188 de taille 4, 125 de taille 6	0.661 ± 0.01
13	1000	40	[5,5,2]	40 de taille 5	0.683 ± 0.04
14	2000	40	[5,5,2]	10 motifs de 4, 5, 6, 8, 10 chacun	0.647 ± 0.03

À notre connaissance, Multi-SPMiner s'agit de la première tentative d'application de méthodes neuronales pour la recherche de motifs fréquents dans des graphes spatio-temporels et des multigraphes en général. Par conséquent, nous n'avons pas de méthodes existantes avec lesquelles comparer nos résultats.

Le tableau 4.1 présente les différents tests effectués, où, pour chacun, 200 graphes ayant les mêmes propriétés ont été générés pour l'entraînement du MGCN et 10 ont été générés graphes pour les tests. Dans le premier test, comme illustré dans la première ligne du tableau 4.1, des graphes spatio-temporels ont été générés avec 200 nœuds, comprenant 40 nœuds par temporalité. Les distributions des relations spatiales, spatio-temporelles et de filiation suivaient des valeurs moyennes de 5, 5 et 2 par nœud, respectivement. Au sein de chaque graphe, 16 motifs de taille 5 et 13 motifs de taille 6 ont été insérés. Les tests se sont concentrés sur la recherche de motifs fréquents de taille petite à moyenne allant de 4 à 10. Les autres expériences présentées dans le Tableau suivent le même protocole de test que celui décrit précédemment.

Pour gérer le nombre exponentiel de combinaisons, 14 tests ont été réalisés en faisant varier des paramètres tels que la taille des graphes, la taille des motifs et le nombre de motifs. Cela nous a permis d'examiner l'influence de ces paramètres sur le ratio de motifs obtenus.

D'après les résultats présentés dans le tableau 4.1, nous observons que notre approche, Multi-SPMiner, a obtenu des performances stables dans l'extraction de motifs fréquents à partir de graphes spatio-temporels. Le taux de motifs trouvés varie de 60 % à 68 %, ce qui peut être considéré comme de bons résultats étant donné la complexité de la tâche. Le plus haute taux de motifs trouvés de 68,3 % a été atteinte lors du test où une seul taille de motif a été inséré dans le graphe spatio-temporel. En revanche, le taux de motifs trouvés la plus faible a été obtenue avec des graphes ayant une plus

TABLE 4.2 – Comparaison des résultats pour différentes tailles de motifs dans les expériences d'entraînement et de test.

Nombre de nœuds	Tailles des motifs dans les graphes d'entraînement	Tailles des motifs dans les graphes de test	Ratio
1000	40 de taille 8	40 de taille 6	0.653
1000	40 de taille 10	40 de taille 6	0.642
1000	40 de taille 4	40 de taille 7	0.662
1000	40 de taille 6	40 de taille 9	0.648

grande densité de motifs. Néanmoins, les autres paramètres testés ne semblent pas influencer le taux de motifs trouvés de la méthode.

En comparant les expériences, nous observons des variations dans le ratio des motifs obtenus. Par exemple, dans les expériences avec un nombre fixe de nœuds de graphe et une distribution de relations cohérente, le ratio de motifs obtenus varie de 59,2% à 66,4%. De plus, les expériences impliquant différentes temporalités et distributions de relations mettent en avant la robustesse de la méthode. Par exemple, dans l'expérience avec la distribution des relations [2,2,2] et des motifs de taille 4 et 6, Multi-SPMiner atteint un ratio de motifs obtenus de 66,1%. Ce résultat suggère que Multi-SPMiner peut efficacement capturer des motifs même dans des scénarios avec moins de temporalités et des distributions de relations plus uniformes.

Capacité de généralisation

Afin d'examiner la capacité de généralisation de notre modèle et sa capacité à apprendre les relations structurelles entre les nœuds facilitant la découverte de motifs, nous avons mené une série d'expériences où l'entraînement et les tests ont été effectués sur des graphes avec des tailles de motifs variées. Ces expériences visaient à examiner la capacité du modèle à gérer différentes tailles de motifs lors des tests tout en maintenant des performances cohérentes. L'objectif était d'évaluer si le modèle pouvait transférer efficacement les connaissances apprises à des graphes avec des tailles de motifs différentes. Le tableau 4.2 présente les résultats de ces expériences. Dans la première expérience, nous avons entraîné le modèle sur des graphes contenant des motifs de taille 8, puis nous l'avons testé sur des graphes avec des motifs de taille 6. De même, dans la deuxième expérience, le modèle a été entraîné sur des motifs de taille 10 et testé sur des graphes contenant des motifs de taille 6. En outre, nous avons mené des expériences supplémentaires où nous avons entraîné le modèle sur des motifs de plus petites tailles (4 et 6) et évalué ses performances sur des graphes avec des motifs de plus grande taille (7 et 9). Ce changement dans l'approche de test nous a permis d'examiner la capacité du modèle à généraliser à partir de plus petits motifs vers des plus grands. Il est remarquable que les résultats des tests obtenus, présentés dans le tableau 4.2, s'alignent étroitement avec les résultats précédemment obtenus dans le tableau 4.1. Cela indique que le Réseau Convolutionnel Multi-Graphes (MGCN) a appris

avec succès les relations entre les graphes et les sous-graphes, indépendamment des tailles spécifiques des motifs impliqués. Ces expérimentations ont démontré la capacité du modèle à gérer efficacement des graphes avec des tailles de motifs variées lors de l'entraînement et des tests, soulignant sa robustesse et sa capacité de généralisation. Ces expériences fournissent des informations précieuses sur la polyvalence de notre modèle et sa capacité à découvrir des motifs dans des graphes avec des distributions de tailles de motifs diverses. Cette flexibilité améliore son applicabilité dans des scénarios réels où les motifs peuvent présenter des variations significatives de taille. La capacité démontrée de notre approche à apprendre et exploiter efficacement ces relations ouvre la voie à des tâches de découverte de motifs plus larges et plus précises dans divers domaines, avec la possibilité d'entraîner et de tester sur différents types de données.

Robustesse face aux nœuds manquants

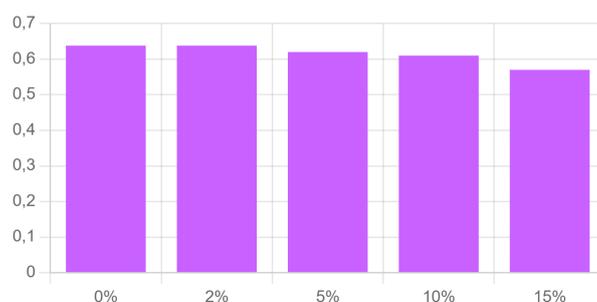


FIGURE 4.6 – Effet de la suppression des nœuds sur la découverte de motifs dans le graphe : Les pourcentages indiqués sous les barres représentent la proportion de nœuds supprimés, tandis que les valeurs à gauche représentent le ratio correspondant de motifs découverts.

Pour évaluer la robustesse de notre approche face à des données bruitées où certains nœuds seraient manquants, nous avons réalisé une expérience impliquant la suppression aléatoire de nœuds du multigraphe. Nous avons reproduit la cinquième expérience de la table 4.1 afin d'assurer la cohérence de notre évaluation. Dans la figure 4.6, nous présentons les résultats obtenus après la suppression de 2%, 5%, 10% et 15% des nœuds du multigraphe. Les résultats expérimentaux révèlent que le taux de motifs trouvés de notre modèle reste en grande partie inchangée lorsque 2% et 5% des nœuds sont supprimés. Cependant, une légère diminution de le taux de motifs trouvés est observée lorsque 10% et 15% des nœuds sont supprimés. Cette baisse peut être attribuée à l'impact des nœuds supprimés sur la fréquence de certains motifs. En conséquence, certains motifs peuvent ne plus satisfaire le seuil de support s , ce qui entraîne leur exclusion de la liste des motifs fréquents par l'algorithme.

Néanmoins, il est important de noter que notre approche ne présente aucune baisse significative de taux de motifs trouvés malgré la suppression de nœuds. Cette découverte souligne la résilience de notre méthode face à la suppression de nœuds et met en évidence sa capacité à maintenir des performances robustes même en présence de perturbations ou de données manquantes.

Comparaison avec SPMiner :

Multi-SPMiner a un objectif différent de celui de SPMiner, car il vise à extraire des motifs de tailles variées en itérant sur différentes tailles de motifs, tandis que SPMiner se concentre sur l'extraction des N motifs les plus fréquents d'une taille fixe. Notre approche peut être considérée comme une généralisation de SPMiner, car en ne considérant qu'une seule connexion et en ignorant le reste, Multi-SPMiner converge vers le modèle de base, SPMiner, aboutissant ainsi à un résultat identique. Pour examiner notre hypothèse, nous avons réalisé une expérience en utilisant le générateur de graphes spatio-temporels, en nous concentrant spécifiquement sur des graphes avec uniquement des connexions spatio-temporelles. En ignorant les étiquettes des nœuds et des arêtes, nous avons obtenu des graphes à lien unique et inséré des motifs de taille 6. Pour comparer notre approche, Multi-SPMiner, avec le modèle pré-entraîné SPMiner, nous avons entraîné notre modèle à partir de zéro (en utilisant le même protocole d'entraînement que dans les expériences précédentes).

Les résultats ont révélé des performances comparables, avec un taux de motifs trouvés de 78,7 % pour SPMiner et de 74,2 % pour Multi-SPMiner. La légère différence peut être attribuée aux différences dans la taille des données d'entraînement (des milliers de graphes pour SPMiner contre 200 graphes pour notre modèle) et à l'intégration des orientations et des étiquettes dans notre modèle, que SPMiner ne prend pas en compte. Dans l'ensemble, Multi-SPMiner a montré des résultats prometteurs, démontrant son potentiel pour l'analyse de multigraphes, comme initialement prévu.

4.5 Conclusion

Ce chapitre a présenté la méthode Multi-SPMiner avec une évaluation approfondie. Les résultats obtenus démontrent la robustesse et l'efficacité de notre approche, avec des performances stables dans l'extraction de motifs fréquents, un taux de motifs trouvés variant de 60% à 68% sur une variété de configurations de graphes. Nous avons également mis en évidence la capacité de généralisation du modèle, sa résilience face aux nœuds manquants, et sa performance comparable à SPMiner dans des conditions

similaires. Ces résultats prometteurs valident notre approche de base, mais cette approche peut être améliorée. Dans le chapitre suivant, nous explorerons des techniques avancées pour optimiser davantage notre modèle. Nous nous concentrerons sur l'intégration de modèles GNN basés sur l'attention, qui pourraient capturer plus efficacement les relations complexes dans les graphes spatio-temporels. De plus, nous viserons à améliorer le processus d'apprentissage de Multi-SPMiner, dans le but d'accroître le taux de motifs trouvés et son efficacité dans la découverte de motifs fréquents dans des graphes spatio-temporels de grande taille et de complexité croissante.

5

Méthode Multi-SPMiner améliorée

Sommaire

5.1	Extension vers des Modèles Basés sur l'Attention	104
5.2	Amélioration techniques	108
5.3	Tests des modèles améliorés Multi-SPMiner sur des données synthétiques	110
5.3.1	Réseau de neurones basé sur l'attention :	110
5.3.2	Amélioration sur la fonction du perte et l'entraînement des GNN :112	
5.4	Conclusion	114

5.1 Extension vers des Modèles Basés sur l'Attention

L'approche initiale de Multi-SPMiner repose sur l'utilisation exclusive de réseaux de convolution multigraphique. Cependant, pour améliorer les performances et le taux de motifs trouvés du modèle, nous avons exploré également l'intégration de modèles basés sur l'attention, compatibles avec les multigraphes. L'objectif principal de cette exploration est d'améliorer le taux de motifs trouvés obtenue par le modèle de référence sur les données synthétique, tout en appliquant notre méthodologie aux données du monde réel. Pour évaluer de manière exhaustive notre approche, nous introduisons divers modèles de Graph Neural Network (GNN) pour une comparaison avec le modèle MGCN de référence. En particulier, nous mettons l'accent sur l'incorporation de modèles basés sur l'attention, qui, selon des études récentes [128], [129], ont démontré des performances supérieures par rapport aux approches GNN traditionnelles. Ces modèles d'attention sont conçus pour pondérer les différentes arêtes et nœuds en fonction de leur importance, permettant ainsi une meilleure représentation des structures complexes dans les graphes.

Toutes les architectures GNN utilisées dans cette étude sont spécifiquement conçues pour gérer les données structurées en graphes caractérisées par des arêtes multiples reliant les nœuds, englobant les relations spatiales, temporelles, et de filiation. Cette conception permet au modèle de capturer et d'exploiter efficacement les diverses relations intégrées dans la structure du graphe.

La recherche d'un taux de motifs trouvés amélioré est une pierre angulaire de notre travail, en particulier lors de l'application de notre méthodologie aux données du monde réel. Dans la section suivante, nous allons citer les différents modèles testés.

Graph Attention Network (GAT) Le premier modèle que nous avons employé est un modèle de GAT, choisi pour son efficacité dans les scénarios où les nœuds sont impliqués dans des interactions diversifiées. La capacité dynamique des GAT à sélectionner de manière sélective les voisins pertinents améliore leurs performances, en particulier dans les tâches nécessitant une prise en compte nuancée de la structure sous-jacente du graphe. Pour étendre l'applicabilité du modèle GAT aux multigraphes, nous avons

adapté l'architecture de base du GAT pour prendre en compte tous les plans distincts de graphes au sein du multigraphe. Cette modification assure que les GAT peuvent gérer les multiples interactions présentes dans les données de multigraphes.

Relational Graph Convolutional Networks (R-GCNs) [130] Ces modèles sont spécialement conçus pour apprendre des représentations dans des graphes multi-relationnels, où différentes arêtes représentent divers types de relations. Contrairement aux GCN traditionnels, les R-GCN intègrent le type de relation entre les nœuds lors de l'agrégation des informations. Cela leur permet de capturer les interactions complexes et diversifiées dans les données multi-relationnelles. Pour chaque nœud i , les messages agrégés sont calculés en tenant compte du type de relation :

$$\alpha_i^{(l)} = \sum_{e \in E_i} \sum_{j \in N_i^e} a_{ij}^{(l,e)} \mathbf{m}_{ij}^{(l,e)},$$

où E_i est l'ensemble de toutes les relations impliquant le nœud i , N_i^e représente l'ensemble des nœuds voisins j du nœud i qui sont reliés à i par la relation e , $\alpha_i^{(l)}$ est l'information agrégée au nœud i à la couche l , $a_{ij}^{(l,e)}$ est le poids d'attention assigné au message du nœud j à travers la relation e , et $\mathbf{m}_{ij}^{(l,e)}$ représente le message du nœud j au nœud i le long de la relation de type e à la couche l . La représentation du nœud est ensuite mise à jour en utilisant :

$$\mathbf{h}_i^{(l)} = \sigma \left(\mathbf{W}^{(l)} \alpha_i^{(l)} + \mathbf{b}^{(l)} \right),$$

où $\mathbf{W}^{(l)}$ est une matrice de poids entraînable et $\mathbf{b}^{(l)}$ est un biais pour la couche l .

Attention Gated Graph Convolutional Networks (GGCN) [131] Ces modèles améliorent le cadre traditionnel des GCN en introduisant des mécanismes de portes qui agrègent sélectivement les informations des nœuds voisins. Cette approche répond aux limitations des GCN à filtrer les informations non pertinentes et à capturer des relations complexes. L'adaptation des GGCN aux multigraphes implique l'utilisation de matrices d'adjacence distinctes pour chaque type d'arête et l'incorporation de plongements de types d'arêtes dans le calcul des scores d'attention. Les plongements à chaque couche et pour chaque nœud i sont calculés comme suit :

$$\begin{aligned}
 m_i^e &= \sum_{j \in N_i^e} \alpha_{ij}^e \cdot (W^{(e)} h_j^{(t-1)}) \\
 u_i^e &= \sigma(U^{(e)} h_i^{(t-1)} + V^{(e)} m_i^e) \\
 r_i^e &= \sigma(R^{(e)} h_i^{(t-1)} + S^{(e)} m_i^e) \\
 \tilde{h}_i^e &= \tanh(W^{(e)} h_i^{(t-1)} + U^{(e)}(r_i^e \odot m_i^e)) \\
 h_i^{(t)} &= (1 - u_i^e) \odot h_i^{(t-1)} + u_i^e \odot \tilde{h}_i^e,
 \end{aligned}$$

où m_i^e agrège les représentations des nœuds voisins par la relation e pondérées par leurs scores d'attention α_{ij}^e , u_i^e et r_i^e agissent comme des portes/grille (gates) de mise à jour et de réinitialisation, respectivement. \tilde{h}_i^e combine l'état caché précédent et les messages agrégés pour générer une représentation candidate, $h_i^{(t)}$ est l'état caché final après incorporation de la porte de mise à jour. Ici, \odot désigne la multiplication élément par élément.

Heterogeneous Attention Networks (HANs) [132] Ces modèles sont conçus pour des graphes hétérogènes, qui comprennent plusieurs types de nœuds et d'arêtes, permettant de modéliser des relations complexes. Les HANs utilisent des mécanismes d'attention à la fois au niveau des nœuds et des types pour se concentrer sur les informations les plus pertinentes. Dans les multigraphes, les HANs utilisent plusieurs matrices d'adjacence, une pour chaque type d'arête e . Le score d'attention et le passage de messages sont calculés pour chaque type d'arête séparément :

$$h_i = \sigma \left(W_t^i x_i + \sum_{e \in E_i} \sum_{j \in N_i^e} a_{ij}^e (W_e^j h_j) \right),$$

où N_i^e est l'ensemble des voisins du nœud i connectés par des arêtes de type e , E_i est l'ensemble des types d'arêtes connectées au nœud i , W_e^j est une matrice de poids spécifique pour le nœud voisin j , W_t^i est une matrice de poids spécifique qui se réfère au type t du nœud i et a_{ij}^e est le score d'attention entre le nœud i et son voisin j connecté par l'arête de type e . Le score d'attention est calculé comme suit :

$$a_{ij}^e = \text{softmax} \left(f(W_t^i x_i, W_e^j h_j, W_r^e) \right),$$

où f est un perceptron multicouche (MLP) et W_r^e est une matrice de poids entraînable pour les plongements de types d'arêtes.

Multi-view Graph Attention Networks (MGATs)[133] Ces modèles étendent le modèle GAT en utilisant le concept de vues multiples, où chaque vue correspond à un type différent de relation capturée par une matrice d'adjacence distincte. Les MGATs agrègent l'information de plusieurs vues simultanément, en utilisant l'attention pour apprendre l'importance de chaque vue pour chaque nœud. Cette approche permet au modèle d'intégrer divers types de relations au sein du graphe. Dans notre travail, nous considérons chaque plan du graphe spatiotemporel comme une vue distincte, avec des matrices de poids d'importance égale. Pour chaque nœud i , les MGATs apprennent une représentation combinée en agrégeant les informations de ses représentations à travers toutes les vues :

$$\beta_i = \text{softmax}(\mathbf{t}^T \mathbf{h}_i), \quad \mathbf{z}_i = \sum_{k=1}^K \beta_{ik} \mathbf{h}_i^k,$$

où β_i est le vecteur de poids d'attention pour le nœud i à travers différentes vues, k indexe les différentes vues ou types de relations dans le graphe à vues multiples, \mathbf{t} est un vecteur de poids entraînable, \mathbf{h}_i représente le vecteur concaténé de toutes les représentations cachées par vue pour le nœud i , et \mathbf{z}_i est la représentation combinée finale pour le nœud i .

Multi-Mask Aggregators for GAT (MMA-GAT) [134] Ces derniers introduisent un mécanisme d'agrégation flexible dans le cadre du GAT en apprenant plusieurs masques pondérés pour chaque nœud en fonction de ses voisins. Ces masques permettent au modèle de capturer différents schémas d'importance pour divers types de relations entre les nœuds. Pour chaque nœud i et type d'arête e , MMA-GAT apprend un "vecteur de masque" $\mathbf{m}_{ie} \in \mathbb{R}^{N_i^e}$, où N_i^e représente l'ensemble des voisins du nœud i connectés par le type d'arête e . Les éléments de ce vecteur, notés m_{iej} , représentent les poids assignés à chaque voisin j du nœud i pour le type d'arête e . Le modèle agrège les informations des voisins en utilisant ces poids :

$$\hat{h}_i^e = \alpha_e \cdot \sum_{j \in N_i^e} m_{iej} \cdot \mathbf{W}_e \mathbf{h}_j^{t-1},$$

où α_e est un paramètre scalaire entraînable associé à chaque vecteur de masque \mathbf{m}_{ie} , contrôlant l'influence du vecteur de masque correspondant pendant le processus d'agrégation et t représente le pas de temps ou la couche dans un cadre temporel ou itératif. Plus précisément, \mathbf{h}_j^{t-1} désigne la représentation cachée du voisin j provenant du pas de temps ou de la couche précédente $t - 1$.

5.2 Amélioration techniques

En plus de modifier le modèle GNN pour l'approche Multi-SPMiner, nous proposons des mises à jour techniques du processus d'entraînement. Dans ce qui suit, nous présentons un ensemble d'hypothèses visant à contrôler le processus d'entraînement en ajoutant des contraintes supplémentaires, ce qui rend l'entraînement plus sensible aux motifs fréquents. Une mise à jour majeure que nous introduisons dans les tests suivants consiste, en plus de décomposer aléatoirement le graphe dans le processus d'entraînement, à inclure un ensemble de motifs fréquents dont nous savons qu'ils sont présents dans le graphe.

1. Perte de Marge Pondérée

Les modèles SPMiner et Multi-SPMiner de référence utilisent une perte à marge maximale pour imposer le plongement ordonné, où la perte pénalise lorsque le plongement d'un sous-graphe ne se trouve pas en "bas à gauche" de son supergraphe.

Nous proposons l'introduction d'une perte de marge pondérée qui met l'accent sur l'importance de plongements corrects pour les motifs plus fréquents. Par exemple, si certains motifs sont connus ou supposés se produire fréquemment, la fonction de perte pourrait attribuer des pénalités plus élevées lorsque le modèle ordonne incorrectement ces motifs. Cela peut être fait en incorporant un facteur de pondération basé sur la fréquence estimée ou connue des motifs :

$$loss_w = \sum_{(A,B) \in P} w_B \cdot E(A, B) + \sum_{(A',B') \in N} w_{B'} \cdot \max(0, \alpha - E(A', B'))$$

où w_B est un poids entraînable associé au motif B , reflétant son importance ou sa fréquence.

2. Intégration d'une Perte d'Estimation de Fréquence

En plus de la perte de plongement d'ordre, on introduit un objectif secondaire qui estime explicitement la fréquence des motifs.

Nous ajoutons un module d'estimation de fréquence, qui prédit directement le nombre d'occurrences d'un motif. La fréquence prédite peut être comparée à la fréquence réelle (ou à une fréquence estimée pour l'entraînement) en utilisant une fonction de perte telle que l'erreur quadratique moyenne (MSE) ou l'erreur absolue moyenne (MAE) :

$$loss_f = \frac{1}{n} \sum_{i=1}^n (\hat{f}_i - f_i)^2$$

où \hat{f}_i est la fréquence prédite et f_i est la fréquence réelle, et i est le motif en question, et n est le nombre total des motifs.

Entraînement Conjoint : La combinaison de cette perte d'estimation de fréquence avec la perte de plongement d'ordre dans un cadre d'apprentissage multi-tâches, peut être fait en minimisant une somme pondérée des deux pertes :

$$loss_{total} = \lambda \cdot loss + (1 - \lambda) \cdot loss_f$$

où λ contrôle l'équilibre entre le maintien du plongement ordonné correct et la prédiction précise de la fréquence des motifs.

3. Apprentissage à partir d'exemples négatifs difficiles

Ici nous cherchons à concentrer le processus d'entraînement sur les exemples négatifs les plus difficiles (ceux que le modèle a du mal à différencier des exemples positifs), ce qui pourrait améliorer la discrimination et le taux de motifs trouvés.

Pendant l'entraînement, on identifie et intègre périodiquement les échantillons négatifs les plus difficiles dans l'ensemble d'entraînement. Cela peut être intégré à la fonction de perte à marge maximale en sélectionnant dynamiquement les paires négatives avec la marge la plus faible, c'est-à-dire celles qui sont les plus proches de ne pas respecter les contraintes de plongement d'ordre.

Pour construire de telles paires, au lieu de créer un ensemble de paires négatives N en sélectionnant aléatoirement des paires (A', B') où A' n'est pas un sous-graphe de B' , nous sélectionnons d'abord un graphe B' , puis générons un négatif proche A' en supprimant un nombre aléatoire de nœuds de B' et en les remplaçant par des nœuds absents du graphe original B' .

4. Apprentissage Par Curriculum

Nous proposons également d'augmenter progressivement la complexité des motifs utilisés pour l'entraînement, permettant au modèle d'apprendre d'abord des distinctions faciles avant de passer à des distinctions plus difficiles. L'objectif était d'entraîner progressivement un modèle sur des motifs de plus en plus grands, puis de les rendre plus difficiles, donnant ainsi au modèle la capacité d'apprendre à composer d'abord des motifs plus petits, puis des motifs plus grands, ce qui constitue un processus plus complexe. Nous commençant l'entraînement avec des motifs simples (par exemple, de taille plus petite ou moins complexes), puis nous introduisons progressivement des motifs plus

complexes à mesure que le taux de motifs trouvés du modèle s'améliore. Cette approche de curriculum pourrait aider le modèle à mieux comprendre les caractéristiques structurelles des motifs, conduisant à une détection de fréquence plus précise.

5.3 Tests des modèles améliorés Multi-SPMiner sur des données synthétiques

5.3.1 Réseau de neurones basé sur l'attention :

Le tableau 5.1 récapitule les différentes expériences menées avec différents modèles GCN et divers paramètres. Les expériences détaillées dans le tableau 5.1 suivent le protocole défini dans le tableau 4.1. Les huit tests introduisent des variations dans des paramètres tels que la taille des graphes, la taille des motifs et le nombre de motifs. Cette stratégie de test complète garantit une évaluation approfondie des performances du modèle dans des conditions variées. Les résultats des tests sont les valeurs moyennes obtenues à partir de 10 itérations de chaque test.

Dans les résultats présentés dans le tableau 5.1, il est évident que tous les modèles GNN basés sur l'attention ont surpassé le taux de motifs trouvés de la méthode de référence, démontrant des performances constantes à travers les différents tests. Notamment, le modèle MMA-GAT a affiché le taux de motifs trouvés globale la plus élevée à 72% (soit +8% par rapport au modèle de base).

De plus, les observations indiquent que les tests impliquant des graphes avec une densité de motifs plus faible ont donné lieu à un taux de motifs trouvés relativement plus faible. Ce résultat est attribué à l'utilisation de graphes avec les mêmes propriétés de densité de motifs pour l'entraînement, car avec une densité de motifs plus faible, la méthode n'avait pas suffisamment d'exemples de motifs pour entraîner les modèles de GNN. Fait intéressant, l'augmentation de la taille des graphes n'a pas eu d'impact significatif sur le taux de motifs trouvés, ce qui confirme que ces modèles peuvent être entraînés sur de petits graphes et testés sur des graphes beaucoup plus grands.

En outre, le nombre d'arêtes par nœud a montré une influence minimale sur le taux de motifs trouvés, avec une légère augmentation de l'écart-type notée dans les graphes à densité d'arêtes plus élevée. Le modèle MMA-GAT a atteint le taux de motifs trouvés la plus élevée (75%) sur des graphes fortement denses en motifs (188 motifs de taille 4, 125 motifs de taille 6) caractérisés par une densité d'arêtes plus faible [2,2,2], ainsi qu'avec (120 motifs de taille 5, 100 motifs de taille 6) avec une densité de motifs plus forte.

TABLE 5.1 – Résultats des tests obtenus pour les différents modèles GNN avec différents paramètres. **Nombre de nœuds du graphe** : le nombre de nœuds dans un seul graphe, **Nombre de nœuds par temporalité**, **Nombre de relations** : une liste définissant le nombre maximal de liens (spatial, spatiotemporel, filiation) dans le motif.

Nombre de nœuds	Nœuds/Temp	Distribution des relations	Motifs	Modèle de base	GAT	RGCN	GGCN	HAN	MGAT	MMAGAT
200	40	[5,5,2]	16 de taille 5, 13 de taille 6	0.63 ± 0.03	0.65 ± 0.02	0.67 ± 0.02	0.67 ± 0.02	0.68 ± 0.02	0.66 ± 0.02	0.69 ± 0.01
1000	40	[5,5,2]	90 de taille 5, 75 de taille 6	0.61 ± 0.01	0.69 ± 0.03	0.72 ± 0.02	0.69 ± 0.02	0.72 ± 0.01	0.68 ± 0.03	0.72 ± 0.01
1000	40	[5,5,2]	75 de taille 8, 60 de taille 10	0.63 ± 0.03	0.70 ± 0.03	0.71 ± 0.02	0.71 ± 0.01	0.70 ± 0.02	0.70 ± 0.02	0.71 ± 0.02
1000	40	[5,5,2]	120 de taille 5, 100 de taille 6	0.62 ± 0.01	0.70 ± 0.03	0.70 ± 0.01	0.71 ± 0.02	0.74 ± 0.02	0.71 ± 0.02	0.75 ± 0.03
1000	40	[5,5,2]	80 de taille 4, 100 de taille 6	0.65 ± 0.02	0.68 ± 0.02	0.69 ± 0.03	0.70 ± 0.02	0.73 ± 0.02	0.70 ± 0.02	0.73 ± 0.02
1000	40	[8,8,3]	80 de taille 5	0.66 ± 0.02	0.67 ± 0.02	0.69 ± 0.05	0.68 ± 0.03	0.69 ± 0.04	0.70 ± 0.04	0.71 ± 0.05
1000	40	[2,2,2]	188 de taille 4, 125 de taille 6	0.66 ± 0.01	0.70 ± 0.02	0.72 ± 0.03	0.73 ± 0.02	0.74 ± 0.01	0.71 ± 0.03	0.75 ± 0.02
2000	40	[5,5,2]	10 motifs de 4, 5, 6, 8, 10 chacun	0.64 ± 0.03	0.67 ± 0.01	0.69 ± 0.03	0.70 ± 0.02	0.72 ± 0.02	0.70 ± 0.03	0.72 ± 0.01
			Moyenne	0.64 ± 0.02	0.68 ± 0.02	0.70 ± 0.03	0.70 ± 0.02	0.71 ± 0.02	0.69 ± 0.02	0.72 ± 0.02

5.3.2 Amélioration sur la fonction de perte et l'entraînement des GNN :

Dans cette section, nous évaluons les améliorations décrites dans la section 5.2. Les améliorations proposées incluent deux modifications de la fonction de perte (Marge Pondérée et Perte de Fréquence) et deux ajustements au processus d'entraînement (mining d'exemples négatifs difficiles et apprentissage aar curriculum). Afin d'évaluer l'impact individuel de chacune de ces propositions, nous avons réalisé des tests pour mesurer leur influence sur les résultats des expériences précédentes. Pour simplifier les tests initiaux, nous avons limité la portée aux meilleurs modèles, à savoir MMA-GAT, et appliqué les modifications à trois cas de test issus du tableau 5.2.

Le tableau 5.2 présente les résultats de performance individuels du modèle MMA-GAT avec l'approche de base, ainsi que les modifications techniques proposées. Tout d'abord, nous observons que la Marge Pondérée produit systématiquement une amélioration moyenne de 1% dans les tests. Similairement, l'inclusion du terme Perte de Fréquence dans la fonction de perte entraîne une augmentation moyenne de moins de 2%. La technique de Mining de Négatifs Difficiles montre une amélioration significative pouvant atteindre 3%, avec une augmentation moyenne de 2%. Enfin, aucune amélioration notable n'a été observée dans l'ensemble des modèles et tests lors de l'application de l'Apprentissage Par Curriculum, ce qui indique une efficacité limitée de cette technique dans le cadre actuel.

Sur la base de ces résultats, nous prenons la décision de reproduire tous les résultats du tableau 5.1 et d'inclure une combinaison des trois améliorations (étant donné que l'Apprentissage Par Curriculum n'a pas montré d'amélioration, nous l'éliminons). Les nouveaux ensembles d'entraînement pour l'approche incluront un ensemble d'exemples négatifs difficiles, et la fonction de perte finale pour l'entraînement est définie comme suit :

$$\text{Perte Totale} = \lambda \cdot \text{Perte de Marge Pondérée} + (1 - \lambda) \cdot \text{Perte de Fréquence}$$

pour les tests suivants, λ est fixé à une valeur plus élevée et ajusté afin de donner plus d'influence à la Perte de Marge Pondérée.

Le résumé des performances des différents modèles GCN avec les modifications apporté est présenté dans le tableau 5.3. Comme dans les résultats précédents, tous les modèles GNN basés sur l'attention surpassent systématiquement le taux de motifs trouvés du modèle de base. Les modèles présentent des améliorations moyennes notables de 3%, tandis que le modèle de base montre une amélioration plus modeste de 2%, confirmant une progression globale dans les performances tout en restant en accord avec les expériences initiales.

TABLE 5.2 – Résultats des tests obtenus pour MMA-GAT avec modifications.

Nombre de nœuds	Nœuds/Temp	Distribution des relations	Motifs	Modèle de base	Marge Pondérée	Perte de Fréquence	Mining d'exemples Négatifs Difficiles	Apprentissage Par Curriculum
1000	40	[5,5,2]	90 de taille 5, 75 de taille 6	0.72 ± 0.01	0.79 ± 0.01	0.73 ± 0.01	0.74 ± 0.01	0.72 ± 0.02
1000	40	[5,5,2]	80 de taille 4, 100 de taille 6	0.73 ± 0.02	0.75 ± 0.02	0.75 ± 0.02	0.77 ± 0.02	0.75 ± 0.02
2000	40	[5,5,2]	10 motifs de 4, 5, 6, 8, 10 chacun	0.72 ± 0.01	0.72 ± 0.01	0.73 ± 0.01	0.75 ± 0.01	0.72 ± 0.01
			Amélioration Moyenne	0.01	0.02	0.03	0.03	0.00

Il est intéressant de noter que les modèles continuent à bien fonctionner même lorsque la taille des graphes augmente, confirmant que ces architectures GNN peuvent être entraînées sur de petits graphes et généralisées efficacement à des graphes beaucoup plus grands. Toutes les améliorations ajoutées reposent sur l'architecture de base, de sorte que les propriétés du modèle multi-SPMiner de base ne sont pas modifiées. En revanche, les graphes à faible densité de motifs intégrés, comme observé précédemment, ont tendance à montrer une légère baisse de taux de motifs trouvés, bien que ces modèles s'adaptent généralement bien aux variations de structure.

Concernant la variance, les résultats montrent une cohérence générale avec les tests antérieurs, avec des valeurs de variance restant stables dans la plupart des cas. Toutefois, dans quelques tests spécifiques, des variations mineures de ± 0.01 ont été observées, notamment dans les graphes à densité plus élevée. Ces fluctuations semblent refléter la variabilité naturelle due à la complexité des motifs intégrés et des relations entre les nœuds. Néanmoins, ces écarts ne remettent pas en question les conclusions globales concernant les performances des modèles.

5.4 Conclusion

Ce chapitre a exploré diverses améliorations de l'approche Multi-SPMiner pour l'extraction de sous-graphes fréquents dans les multigraphes spatio-temporels. Nous avons étudié plusieurs modèles de réseaux de neurones graphiques (GNN) basés sur l'attention, notamment GAT, RGCN, GGCN, HAN, MGAT et MMA-GAT, comme alternatives au modèle convolutif de référence. Ces modèles basés sur l'attention ont systématiquement surpassé le modèle de base, MMA-GAT affichant le taux de motifs trouvés globale la plus élevée à 72% (une amélioration de 8% par rapport au modèle de référence). Nous avons également introduit des améliorations techniques au processus d'entraînement, notamment la perte de marge pondérée, la perte de fréquence, le mining d'exemples négatifs difficiles et l'apprentissage par curriculum. Parmi celles-ci, le mining d'exemples négatifs difficiles a montré l'amélioration la plus significative, avec une augmentation de taux de motifs trouvés allant jusqu'à 3%. La combinaison de la perte de marge pondérée, de la perte de fréquence et du mining d'exemples négatifs difficiles a conduit à des améliorations supplémentaires pour tous les modèles, MMA-GAT atteignant jusqu'à 78% de taux de motifs trouvés dans certains scénarios.

TABLE 5.3 – Résultats des tests obtenus pour les différents modèles améliorés avec différents paramètres.

Nombre de nœuds	Nœuds/Temp	Distribution des relations	Motifs	Modèle de base						
				GAT	RGCN	GGCN	HAN	MGAT	MMAGAT	
200	40	[5.5.2]	16 de taille 5, 13 de taille 6	0.68 ± 0.02	0.70 ± 0.02	0.69 ± 0.02	0.71 ± 0.02	0.69 ± 0.02	0.72 ± 0.01	
1000	40	[5.5.2]	90 de taille 5, 75 de taille 6	0.72 ± 0.03	0.75 ± 0.02	0.72 ± 0.02	0.75 ± 0.01	0.70 ± 0.03	0.75 ± 0.02	
1000	40	[5.5.2]	75 de taille 8, 60 de taille 10	0.73 ± 0.03	0.74 ± 0.02	0.73 ± 0.01	0.73 ± 0.02	0.72 ± 0.02	0.74 ± 0.02	
1000	40	[5.5.2]	120 de taille 5, 100 de taille 6	0.65 ± 0.01	0.73 ± 0.01	0.74 ± 0.02	0.77 ± 0.02	0.74 ± 0.01	0.77 ± 0.03	
1000	40	[5.5.2]	80 de taille 4, 100 de taille 6	0.68 ± 0.02	0.72 ± 0.03	0.73 ± 0.02	0.76 ± 0.02	0.73 ± 0.02	0.76 ± 0.02	
1000	40	[8.8.3]	80 de taille 5	0.69 ± 0.02	0.71 ± 0.05	0.71 ± 0.03	0.72 ± 0.04	0.73 ± 0.05	0.74 ± 0.05	
1000	40	[2.2.2]	188 de taille 4, 125 de taille 6	0.73 ± 0.02	0.74 ± 0.03	0.76 ± 0.02	0.77 ± 0.01	0.74 ± 0.02	0.78 ± 0.02	
2000	40	[5.5.2]	10 motifs de 4, 5, 6, 8, 10 chacun	0.70 ± 0.01	0.72 ± 0.03	0.73 ± 0.02	0.74 ± 0.02	0.72 ± 0.03	0.74 ± 0.01	
			Moyenne	0.71 ± 0.02	0.73 ± 0.03	0.73 ± 0.02	0.74 ± 0.02	0.72 ± 0.02	0.75 ± 0.02	

6

Méthode Deep QMiner

Sommaire

6.1	Introduction	117
6.2	Formulation du problème	118
6.3	Configuration de l'Apprentissage par Renforcement	118
6.3.1	Système de récompense intermédiaire pour l'extraction de motifs	120
6.3.2	Système de récompense progressive pour l'ajout de nœuds .	120
6.3.3	Pénalités pour Étapes Invalides	121
6.4	Phase d'Entraînement	122
6.4.1	Exploration Epsilon-Greedy	122
6.4.2	Découverte de motifs multiples au cours des épisodes	124
6.5	Phase de test	126
6.6	Tests de l'approche Multi Agent Deep-QMiner sur des données syn- thétiques	126
6.7	Conclusion	130

6.1 Introduction

L'apprentissage par renforcement a montré un grand potentiel pour résoudre des problèmes complexes liés aux multigraphes (Chapitre 2), où les approches algorithmiques traditionnelles rencontrent des difficultés en termes d'évolutivité et de flexibilité. En exploitant l'apprentissage par renforcement, en particulier le Deep Q-learning (DQN) [135], nous permettons aux agents d'explorer de manière autonome la structure du graphe et de découvrir des motifs fréquents sans avoir recours à des techniques de recherche exhaustive. La nature dynamique de l'apprentissage par renforcement permet un apprentissage adaptatif basé sur le retour d'information, ce qui le rend particulièrement adapté aux données de graphes à haute dimension et à grande échelle.

Le Deep Q-Learning, en particulier, offre la possibilité d'apprendre des représentations complexes des structures de graphes et de prendre des décisions séquentielles optimales pour la découverte de motifs. Cette approche vise à surmonter les limitations des méthodes traditionnelles en termes de temps de calcul et de capacité à traiter des graphes de grande taille. L'objectif principal est de développer un système capable d'identifier efficacement des motifs fréquents dans des multigraphes complexes, tout en s'adaptant dynamiquement à différentes structures de graphes. Les avancées potentielles de cette approche incluent une meilleure généralisation à des graphes non vus, une réduction significative du temps de calcul pour les grands graphes, et pourrait ouvrir une nouvelle voie de recherche mettant en évidence l'utilisation de l'apprentissage par renforcement dans ce type de problèmes de fouille de motifs dans les graphes.

Dans ce chapitre, nous présentons une approche détaillée utilisant le Deep Q-Learning Multi-Agent pour découvrir des motifs fréquents dans de grands multigraphes. La méthode utilise des agents multiples qui explorent le multigraphe en utilisant le Deep Q-Learning pour apprendre implicitement quelles structures de sous-graphes (motifs) sont fréquentes, sans compter explicitement les occurrences de motifs. L'approche tire parti de l'exploration epsilon-greedy [136] et d'un système de récompense basé sur la curiosité pour encourager les agents à explorer de nouvelles régions du multigraphe. Pendant la phase d'entraînement, les agents apprennent à partir d'un ensemble prédéfini de motifs fréquents (appelé le sac de motifs), et lors de la phase de test, ils

généralisent les politiques(stratégies) apprises pour découvrir des motifs fréquents de manière autonome.

6.2 Formulation du problème

Pour l'approche Deep-QMiner, nous suivons la même formulation du problème que celle utilisée précédemment. Étant donné un graphe spatio-temporel représenté en format multigraphe $G = (V, E)$, où V représente l'ensemble des nœuds et E l'ensemble des arêtes, avec plusieurs arêtes entre une même paire de nœuds (c'est-à-dire un multigraphe), l'objectif est de découvrir tous les motifs fréquents de taille K , où "fréquent" est défini comme des motifs apparaissant plus souvent qu'un seuil de fréquence donné σ . Maintenant que les données sur lesquelles nous travaillons et la tâche sont spécifiées, nous présentons dans ce qui suit la configuration de la méthode proposé : Deep-QMiner.

6.3 Configuration de l'Apprentissage par Renforcement

Tout d'abord, nous introduisons le concept de 'sac', qui représente l'ensemble des motifs fréquents que les agents tentent de trouver :

Sac de motifs Afin d'entraîner et tester notre approche, un **sac de motifs** qui contient tous les motifs fréquents qui apparaissent dans le multigraphe est introduit. Le sac de motifs sert d'ensemble cible de motifs que les agents doivent apprendre à trouver. L'objectif des agents est d'apprendre une politique optimale leur permettant de découvrir des motifs dans le graphe et de les associer aux motifs du sac. Au fil du temps, les agents devraient construire un sac de motifs identique à celui fourni pendant l'entraînement, ce qui implique qu'ils ont réussi à identifier tous les motifs fréquents dans le graphe.

La tâche est formulée comme un problème de **renforcement multi-agent**, où les agents visent à apprendre une politique optimale pour construire des motifs fréquents à partir de différentes régions du multigraphe. Leur apprentissage est guidé par la qualité de la correspondance des motifs qu'ils découvrent avec ceux du sac de motifs.

Dans ce contexte, le problème est formalisé comme un **Processus de Décision Markovien (MDP)**, où les composants sont définis comme suit :

- **État (S)** : L'état s_t à l'instant t représente le motif partiel actuel (sous-graphe) en cours de construction par l'agent. Cet état est encodé à l'aide de plongements de graphes obtenus par un GNN, qui capturent l'information structurelle du sous-graphe actuel. Le plongement inclut également des informations sur le voisinage

du sous-graphe actuel, indiquant quels nœuds (y compris les arêtes multiples dans le cas des multigraphes) peuvent être ajoutés pour l'étendre.

- **Action (A)** : L'action a_t est la décision prise par l'agent à l'instant t pour étendre le motif. À chaque instant, l'agent peut choisir d'ajouter un nouveau nœud au motif actuel. L'ensemble des actions disponibles regroupent les visites des nœuds voisins (tenant compte des arêtes multiples entre les nœuds) qui peuvent être ajoutés au sous-graphe actuel.
- **Récompense (R)** : La fonction de récompense est conçue pour encourager la découverte de motifs correspondant à ceux du sac de motifs prédéfini pendant l'entraînement. Les récompenses sont structurées comme suit :
 - Une récompense positive R_{match} est donnée si l'agent construit un motif correspondant à un motif du sac.
 - Une pénalité R_{invalid} est donnée si le motif est invalide (par exemple, taille ou structure incorrecte).
 - Une **récompense basée sur la curiosité** $R_{\text{curiosity}}$ est fournie pour la découverte de nouveaux nœuds qui n'ont pas été fréquemment visités, cette récompense encourage les agents à explorer de nouvelles régions du multigraphe en fournissant de petites récompenses positives lorsqu'ils découvrent ces nœuds. La récompense de curiosité est donnée comme suit :

$$R_{\text{curiosity}} = \begin{cases} 1 & \text{si un nouveau nœud est visité} \\ 0 & \text{sinon} \end{cases}$$

Ce mécanisme garantit que les agents ne se contentent pas de motifs déjà découverts ou de parties déjà explorées du multigraphe, favorisant ainsi une exploration plus large.

- **Politique (π)** : La politique $\pi(s)$ représente la stratégie de l'agent pour sélectionner des actions en fonction de l'état actuel. L'objectif de l'agent est d'apprendre une politique optimale $\pi^*(s)$ qui maximise la récompense cumulative espérée au fil du temps :

$$\pi^*(s) = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^T \gamma^t R(s_t, a_t) \right]$$

où γ est le facteur d'actualisation qui détermine l'importance des récompenses futures.

À travers l'exploration et l'exploitation du multigraphe, les agents visent à construire des motifs correspondant à ceux du sac de motifs, et l'objectif final est que chaque agent (ou tous les agents collectivement) produisent un sac de motifs contenant exactement tous les motifs fréquents présents dans le graphe original.

6.3.1 Système de récompense intermédiaire pour l'extraction de motifs

En fait, les agents ne reçoivent des récompenses que lorsqu'ils réussissent à trouver un motif à la fin d'un épisode. Dans l'apprentissage par renforcement, les récompenses viennent généralement après les actions, mais les agents ont besoin de retours pendant le processus de construction d'un motif pour décider de la meilleure action suivante (par exemple, quel nœud ajouter).

Pour répondre à cette préoccupation, nous affinons la structure de la récompense afin de fournir des récompenses intermédiaires à chaque étape, guidant l'agent vers la construction d'un motif valide et fréquent.

6.3.2 Système de récompense progressive pour l'ajout de nœuds

Pour s'assurer que les agents prennent de bonnes décisions à chaque étape (c'est-à-dire, quel nœud ajouter), nous pouvons introduire un système de récompense progressive qui fournit un retour pendant le processus de construction d'un motif.

Récompense de Motif Partiel R_{partial} : À chaque étape de la construction d'un motif, l'agent peut recevoir des récompenses intermédiaires pour la création de motifs partiels valides susceptibles de conduire à des motifs fréquents dans le sac de motifs. Par exemple, à mesure que l'agent ajoute des nœuds à un motif partiel, il peut être récompensé en fonction de la proximité du motif actuel avec la formation d'un motif fréquent valide.

A mesure que les agents ajoutent des nœuds/arêtes, ils reçoivent des récompenses pour la construction de sous-graphes faisant partie de motifs fréquents dans le sac de motifs. Plus le sous-graphe se rapproche d'un motif fréquent connu, plus la récompense est élevée. La récompense de Motif Partiel peut être définie par : $R_{\text{partial}} = +1$ pour chaque étape valide vers la construction d'un motif fréquent

Par exemple, si l'agent a construit un sous-graphe correspondant partiellement à un motif fréquent (par exemple, 3 nœuds sur K), il reçoit une petite récompense pour cela ($3/K$ pour cet exemple).

6.3.3 Pénalités pour Étapes Invalides

À chaque étape, les agents doivent également être pénalisés pour les actions qui conduisent à des constructions de motifs invalides ou sous-optimales. Ces pénalités découragent les agents d'explorer des directions non productives.

Pénalité pour Sous-graphe Déconnecté $R_{\text{disconnected}}$: Si un agent sélectionne un nœud qui brise la connectivité du motif partiel ou ne contribue pas à la formation d'un motif valide, il reçoit une pénalité : $R_{\text{disconnected}} = -1$

Construction de la fonction de récompense complète : Avec cette structure de récompense, l'agent reçoit un retour après chaque action qu'il entreprend, le guidant vers la construction d'un motif valide et fréquent. La **Q-value** $Q(s, a)$ pour chaque action incorpore désormais ces récompenses intermédiaires, permettant à l'agent d'évaluer quelle action conduira à la récompense cumulative attendue la plus élevée.

À chaque étape, l'agent :

1. **Évalue les actions possibles** (c'est-à-dire ajouter différents nœuds au sous-graphe actuel).
2. **Calcule la Q-value attendue** pour chaque action, qui prend désormais en compte :
 - Les **récompenses intermédiaires** pour la formation d'un motif partiel valide.
 - Les **pénalités** pour les actions menant à des sous-graphes invalides ou des étapes redondantes.
 - La **récompense finale** pour la construction d'un motif complet correspondant à un du sac de motifs.
3. **Sélectionne l'action** qui maximise la Q-value.

Cela permet à l'agent de prendre des décisions éclairées sur quel nœud ou arête ajouter ensuite, en se basant sur les récompenses qu'il s'attend à recevoir à la fois immédiatement (grâce aux récompenses intermédiaires) et dans le futur (en complétant un motif valide).

Le système de récompense final comprend les éléments suivants :

1. **Récompense de Correspondance de Motif** $R_{\text{match}} = +5$: donnée pour avoir complété un motif correspondant à un du sac de motifs.
2. **Récompense de Motif Partiel** $R_{\text{partial}} = +1$: récompensant chaque étape valide vers la construction d'un motif fréquent.

3. **Récompense Basée sur la Curiosité** $R_{\text{curiosity}} = +1$: pour renforcer l'exploration de nouveaux nœuds ou arêtes.
4. **Pénalité de Motif Invalide** $R_{\text{invalid}} = -K - 3$: pour la construction de motifs invalides. Cette pénalité empêche les agents de perdre du temps à construire des motifs non viables, les guidant vers des actions correctes qui mènent à des motifs valides.
5. **Pénalité pour Sous-graphe Déconnecté** $R_{\text{disconnected}} = -1$: donnée pour avoir brisé la connectivité.

6.4 Phase d'Entraînement

6.4.1 Exploration Epsilon-Greedy

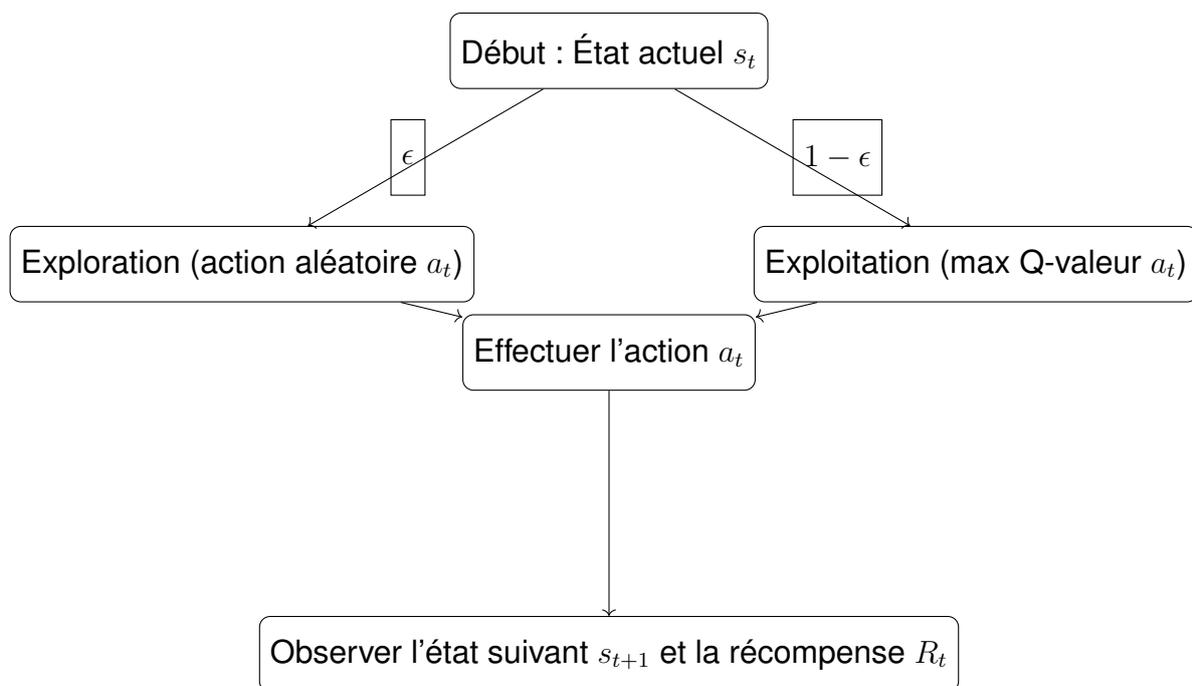


FIGURE 6.1 – Mécanisme Epsilon-Greedy pour la sélection de nœuds

Pendant l'entraînement, chaque agent suit une **stratégie epsilon-greedy** pour équilibrer l'exploration et l'exploitation. À chaque étape, l'agent choisit une action a_t soit en :

- **Exploitant** ses valeurs Q apprises (avec une probabilité $1 - \epsilon$), en sélectionnant l'action qui maximise la valeur Q pour l'état actuel, ou
- **Explorant** de manière aléatoire (avec une probabilité ϵ), en choisissant une action aléatoire pour explorer de nouvelles zones du multigraphe.

Le paramètre **epsilon** diminue avec le temps pour passer de l'exploration à l'exploitation à mesure que l'agent apprend la structure du multigraphe :

$$\epsilon_t = \max(\epsilon_{\min}, \epsilon_{\text{start}} \cdot \epsilon \cdot \epsilon_{\text{decay}})$$

Le paramètre **epsilon** est initialement défini à une valeur ϵ_{start} et diminue progressivement vers une valeur minimale ϵ_{\min} selon un facteur de déca ϵ_{decay} .

Cela garantit qu'au début de l'entraînement, les agents explorent plus largement le multigraphe, tandis qu'à mesure qu'ils progressent, ils exploitent les connaissances acquises pour se concentrer sur les actions plus susceptibles de conduire à des motifs fréquents (figure 6.1).

Fonction Q et l'apprentissage profond

Les agents utilisent un réseau Q , un réseau de neurones qui approxime la fonction de valeur $Q(s, a)$, qui estime la récompense cumulative attendue pour prendre l'action a à l'état s :

$$Q(s, a) = \mathbb{E} \left[\sum_{t=0}^T \gamma^t R(s_t, a_t) \right]$$

La fonction Q est mise à jour en utilisant l'équation de **Bellman** suivante :

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[R(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

où α est le taux d'apprentissage, γ est le facteur de réduction (actualisation), et $\max_{a'} Q(s', a')$ est la valeur Q maximale pour l'état s' suivant s , après l'action a .

L'agent apprend à approximer les valeurs Q à l'aide d'un MGCN. Cela se fait par interaction avec le multigraphe, où l'agent construit des motifs et reçoit des récompenses basées sur la correspondance des motifs découverts avec ceux du sac de motifs.

Pendant l'entraînement, les agents ne sont pas obligés de compter explicitement les occurrences de motifs. Au lieu de cela, ils apprennent à faire correspondre implicitement les motifs qu'ils découvrent avec ceux du sac de motifs. Si un motif trouvé correspond à un motif inclus dans le sac de motifs, l'agent reçoit une récompense positive R_{match} . Cette structure de récompense guide les agents pour apprendre quelles actions mènent aux motifs fréquents.

6.4.2 Découverte de motifs multiples au cours des épisodes

Comme le multigraphe contient plusieurs motifs fréquents, les agents travaillent à travers plusieurs **épisodes** pour découvrir différents motifs. Chaque épisode représente une nouvelle tentative de l'agent pour construire un motif. Au cours de chaque épisode, l'agent :

- **Commence par un nouvel état initial** : L'agent commence par sélectionner un nœud ou un sous-graphe aléatoire dans le multigraphe. Cela garantit que l'agent explore différentes parties du multigraphe au cours de plusieurs épisodes.
- **Construit un motif** : L'agent construit un motif étape par étape en sélectionnant des nœuds ou des arêtes à ajouter au sous-graphe actuel. L'objectif est de compléter un motif de taille K qui correspond à un motif dans le sac de motifs.
- **Terminaison de l'épisode** : L'épisode se termine lorsque l'agent a soit complété un ou plusieurs motifs de taille K , soit épuisé sa capacité à étendre le sous-graphe (en raison d'actions invalides ou de contraintes du multigraphe). Une fois qu'un motif complet est formé, l'agent ne s'arrête pas mais réinitialise plutôt le processus de construction de motif au sein du même épisode en sélectionnant un nouveau nœud et en recommençant la construction.

Exploration au cours des épisodes

Chaque épisode permet à l'agent de repartir de zéro depuis une nouvelle partie du multigraphe, ce qui aide à découvrir des motifs diversifiés. Cette approche épisodique garantit que les agents ne sont pas confinés à une seule région du multigraphe, et évite de découvrir de manière répétée les mêmes motifs. Après avoir terminé un épisode, l'agent commence un nouvel épisode, explorant potentiellement une partie complètement différente du multigraphe.

Les agents peuvent être entraînés conjointement ou séparément. Dans ce travail, nous avons choisi un entraînement conjoint : tous les agents partagent un réseau Q global et apprennent une politique commune. Comme les agents explorent différentes parties du graphe, leurs expériences partagées aident le réseau Q à apprendre plus efficacement et à couvrir plus largement le graphe. Cela signifie que les valeurs Q sont mises à jour collectivement sur la base des expériences de tous les agents. Les agents coopèrent implicitement, apprenant des explorations de chacun individuelle du multigraphe.

Chaque agent explore une partie différente du graphe, et leurs expériences (c'est-à-dire les états qu'ils visitent, les actions qu'ils prennent et les récompenses qu'ils reçoivent) sont regroupées dans un tampon d'expérience commun. Le réseau Q est entraîné sur cette expérience combinée, bénéficiant de l'exploration de tous les agents. Les agents

apprennent les actions des autres, ce qui accélère le processus d'apprentissage en recevant des exemples d'entraînement diversifiés.

Étant donné que le graphe est grand, il est important que les agents couvrent efficacement différentes régions. L'entraînement conjoint permet aux agents de partager leurs apprentissages, améliorant l'exploration globale et réduisant la redondance. De plus, en regroupant les expériences, le réseau Q partagé peut apprendre plus rapidement et plus efficacement, utilisant mieux l'exploration collective des agents. Pour tenir compte de l'évitement de la redondance : une mémoire globale est utilisée dans l'entraînement conjoint pour s'assurer que les agents ne découvrent pas à plusieurs reprises les mêmes motifs, améliorant ainsi l'efficacité de l'apprentissage.

Le nombre de réseaux Q est fixé en fonction de la tailles de motifs de motifs recherchés dans les graphes. Par exemple, si la tâche consiste à rechercher des motifs de tailles 4 et 6, deux réseaux Q sont configurés séparément, et des instances d'agents sont créées pour entraîner chacun de ces deux réseaux de manière indépendante. Cela permet à l'agent de ne pas prendre la décision de continuer à développer le motif ou de s'arrêter à un stade précoce, une tâche qui s'est avérée difficile lors de la phase de test.

Mémoire pour les motifs découverts

Pendant l'entraînement, les agents maintiennent une **mémoire locale** et une **mémoire partagée globale** des motifs déjà découverts. Cela évite la redécouverte redondante des mêmes motifs au cours des épisodes. En gardant une trace des motifs découverts précédemment, les agents sont incités à rechercher de nouveaux motifs dans des parties inexplorées du multigraphe.

- **Mémoire locale** : Chaque agent stocke les motifs qu'il a déjà découverts au cours de l'épisode en cours. Si l'agent redécouvre un motif, il peut éviter une exploration redondante.
- **Mémoire globale** : Une mémoire globale partagée entre les agents permet la collaboration, garantissant que si un agent découvre un motif, les autres agents évitent de le redécouvrir.

Par ailleurs, il est important que la comparaison et l'ajout à la mémoire globale ne soient effectués qu'à la fin des épisodes afin de limiter le nombre d'accès (figure 6.2).

6.5 Phase de test

Pendant la phase de test, les agents ne reçoivent plus de récompenses ni n'ont accès au sac de motifs. Les agents doivent se fier aux valeurs Q apprises pour découvrir des motifs de manière autonome dans un nouveau multigraphe ou dans des régions inexplorées du multigraphe original.

Exploitation en phase de test

Dans la phase de test, les agents se concentrent sur l'**exploitation**. Comme aucune récompense ou pénalité n'est fournie, les agents s'appuient entièrement sur la politique $\pi(s)$ qu'ils ont apprise pendant l'entraînement. Les valeurs Q dictent quelles actions sont les plus susceptibles de conduire à la découverte de motifs fréquents.

À chaque étape, l'agent sélectionne l'action a qui maximise la valeur Q pour l'état actuel :

$$a_t = \arg \max_a Q(s_t, a)$$

Cette sélection déterministe des actions garantit que l'agent utilise les connaissances acquises pour maximiser les chances de découvrir des motifs fréquents.

Pendant la phase de test, nous proposons d'ajouter un mécanisme pour ajouter les motifs au sac des motifs trouvés. Ce mécanisme consiste en un mécanisme de consensus que nous appelons 'match and drop'. Les agents peuvent collaborer pour atteindre un consensus sur la question de savoir si un motif doit être ajouté à la mémoire globale. Si plusieurs agents découvrent indépendamment le même motif à travers différents épisodes, cela fournit un signal plus fort indiquant que le motif est important.

Seuls les motifs qui ont été découverts par plusieurs agents ou à travers plusieurs épisodes sont ajoutés à la mémoire globale. Le nombre de rencontres par différents motifs est fixé en fonction du seuil de la valeur de support de la fréquence des motifs σ .

6.6 Tests de l'approche Multi Agent Deep-QMiner sur des données synthétiques

Afin d'établir une base de référence pour tester l'approche du multi-agent Deep-QMiner et de pouvoir la comparer à Multi-SPMiner, nous avons effectué les mêmes tests sur les mêmes données de graphe. Au cours de ces tests, plusieurs hyperparamètres ont été évalués afin de parvenir à la meilleure configuration du modèle.

Nous avons principalement ajusté nos modèles selon les configurations suivantes : Si les motifs cibles sont de 2 tailles différentes, deux réseaux Q sont initialisés comme

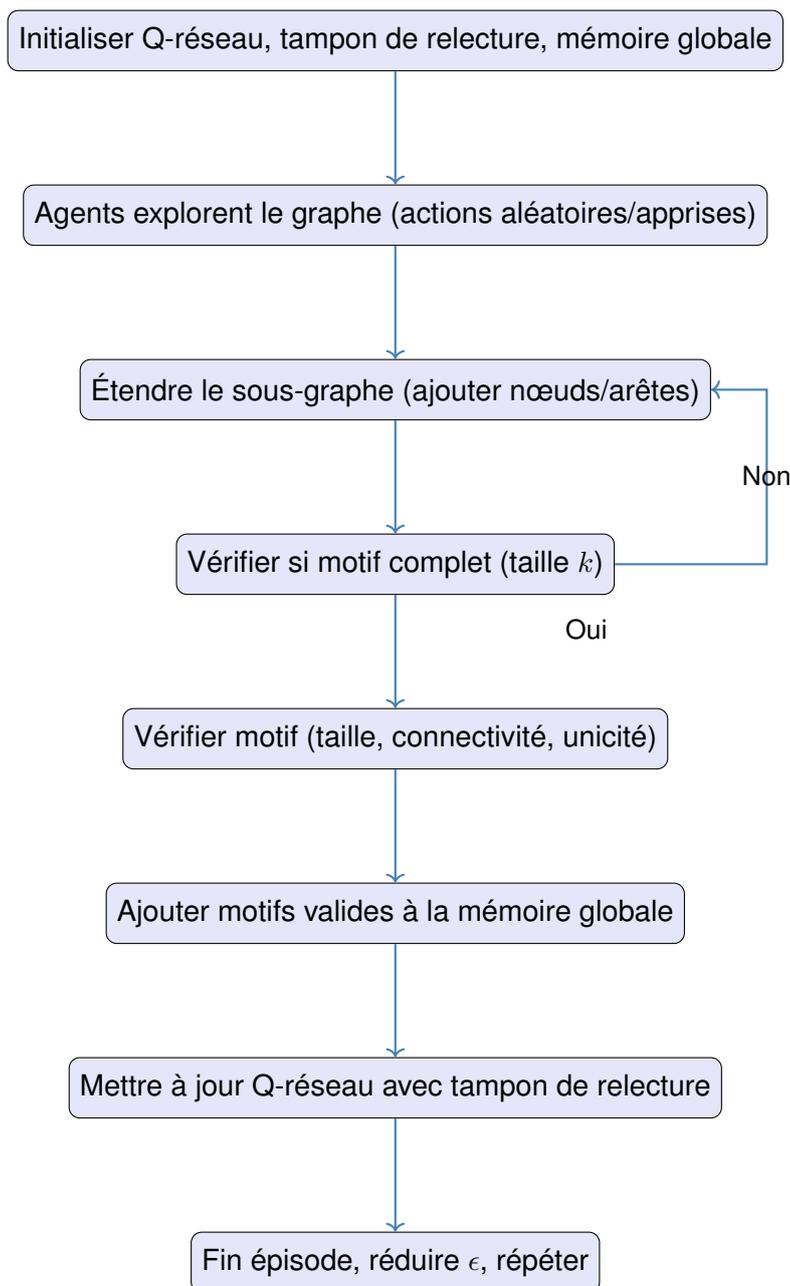


FIGURE 6.2 – Approche d’apprentissage profond par Q multi-agents pour la découverte de motifs

illustré dans la figure 6.3, chacun pour explorer une taille de motif spécifiée. D’autres paramètres importants qui ont été testés sont la taille de la mémoire tampon et la taille du batch pour les mises à jour du réseau Q, qui déterminent la quantité d’expériences passées que les agents utilisent pendant l’apprentissage. Des mémoires tampons plus grandes et des tailles de batch plus importantes peuvent améliorer la stabilité, mais vont ralentir l’apprentissage en contrepartie, tandis que des tailles plus petites

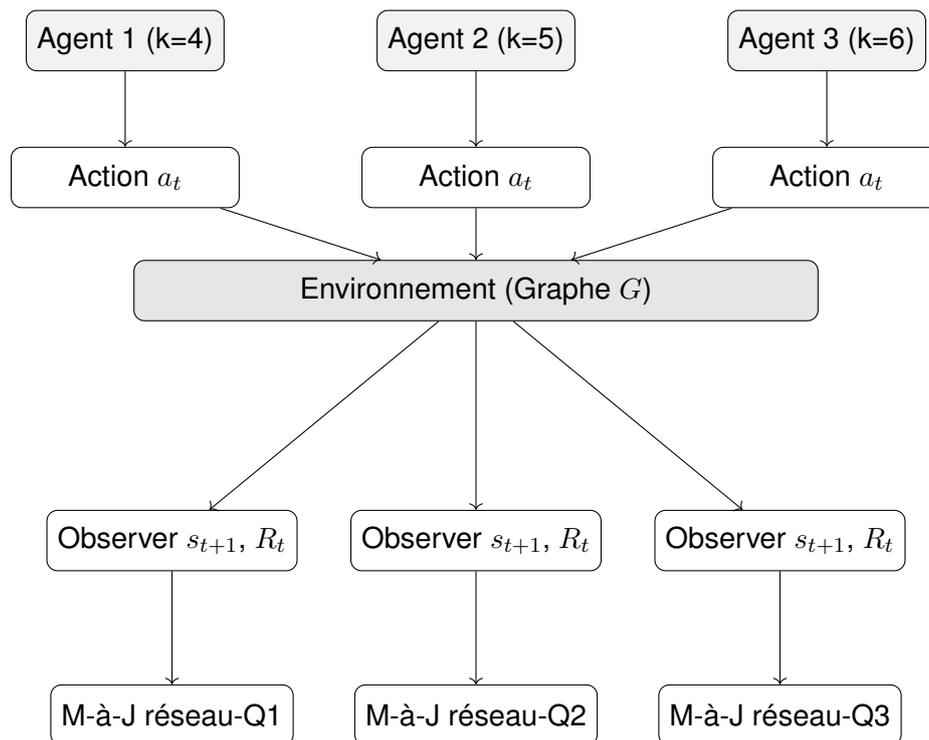


FIGURE 6.3 – Interaction des agents avec l’environnement dans chaque épisode

peuvent accélérer les mises à jour mais risquent de provoquer un sur-ajustement aux expériences récentes.

Un autre paramètre important qui a été testé est la stratégie epsilon-greedy, qui contrôle l’équilibre entre exploration et exploitation. Tester différentes stratégies de décroissance d’epsilon peut avoir un impact significatif sur la façon dont les agents construisent le motif. En début d’entraînement, nous proposons une exploration plus élevée ($\epsilon = 0.7$), puis, plus tard dans l’entraînement, à mesure que ϵ diminue, l’exploitation devient plus dominante et l’agent se repose sur les valeurs Q apprises. En outre, différentes architectures basées sur les réseaux MGCN ont été testées avec des tailles variées. Dans nos tests, nous comparons l’influence du nombre d’agents utilisés pour explorer le graphe. Le tableau 6.1 présente les résultats obtenus avec l’approche Multiagent Deep-QMiner sur le même ensemble de tests présenté précédemment, avec un nombre d’agents varié. L’évaluation est effectuée en termes de taux de motifs trouvés et du pourcentage moyen de nœuds visités par les agents.

Le tableau présenté offre un aperçu des performances du modèle de fouille multi-agent par Deep Q-Learning (Deep-QMiner) pour la découverte de motifs fréquents à travers des graphes de tailles variées, ainsi que les résultats obtenus avec différents nombres d’agents (30, 60 et 100). Une observation clé est la couverture constante, où les graphes sont assez bien visités par tous les agents, en particulier avec 100 agents, qui atteignent une couverture presque parfaite (environ 99%). La configuration à 60 agents

représente un juste milieu, maintenant une couverture élevée (97%) sans sacrifier trop de ressources computationnelles (discuté dans le dernier chapitre) ni de performances.

Performance de l'Évolutivité : l'un des points les plus remarquables du Tableau est que, contrairement à MultiSPMiner, le Deep-QMiner multi-agent ne s'adapte pas aussi efficacement aux graphes de grande taille. Cela se manifeste par une baisse progressive des performances en termes de taux de motifs trouvés à mesure que la taille du graphe augmente. Bien que Deep-QMiner obtienne une taux de motifs trouvés plus élevée dans les petits graphes (200 nœuds), surpassant SPMiner, ses performances se dégradent significativement pour les graphes avec plus de nœuds (1000 et 2000 nœuds). Cela suggère que le modèle est mieux adapté à la découverte de motifs dans les petits graphes, où il peut identifier efficacement les motifs fréquents.

Configurations d'agents et performance :

- **100 agents** : La taux de motifs trouvés dans la plupart des expériences avec 100 agents est constamment élevé, dépassant souvent celui de 60 agents, mais au détriment de la performance générale. Cela pourrait résulter du grand nombre d'agents explorant l'espace simultanément. Ce phénomène suggère un compromis : bien qu'un plus grand nombre d'agents augmente la probabilité de découvrir des motifs véritables, il augmente également le risque d'en identifier des incorrects.
- **60 agents** : La configuration avec 60 agents semble offrir le meilleur équilibre entre **ratio** et **coût computationnel**. Elle atteint une taux de motifs trouvés relativement élevée tout en nécessitant moins de ressources informatiques que la configuration à 100 agents. Par exemple, pour des motifs de taille 5 ou 6, la configuration à 60 agents parvient à maintenir une bonne taux de motifs trouvés (0,58), ce qui en fait un choix pratique pour les graphes de taille moyenne.
- **30 agents** : Cette configuration montre systématiquement une taux de motifs trouvés plus faible. Avec moins d'agents, la couverture et la taux de motifs trouvés sont compromis en raison du nombre réduit d'agents explorant l'espace du graphe.

Impact de la taille du graphe et de la complexité des motifs :

- Pour les **motifs plus petits (tailles 4, 5 et 6)**, la taux de motifs trouvés est relativement élevée dans l'ensemble, en particulier pour la configuration à 100 agents, ce qui suggère que le modèle excelle dans l'identification de motifs de ces tailles dans des graphes plus denses ou plus petits.
- En revanche, la taux de motifs trouvés pour les **motifs plus grands (tailles 8 et 10)** diminue dans toutes les configurations d'agents, indiquant une difficulté à découvrir des motifs plus grands ou plus complexes. Cette baisse du taux de motifs trouvés est particulièrement perceptible dans les graphes de plus grande taille (1000 nœuds), où le modèle n'obtient pas le même niveau de taux de motifs trouvés observée avec les motifs plus petits.

En résumé, le Deep-QMiner multi-agent démontre de solides performances dans les petits graphes qui excède même les performances de Multi-SPMiner, en particulier pour les motifs de taille 4 à 6, mais rencontre des défis d'évolutivité dans les graphes plus grands. La configuration à 60 agents offre une approche équilibrée en termes de couverture, de taux de motifs trouvés et de coût computationnel (exploré dans le chapitre 7) ce qui en fait un choix adapté pour les graphes de taille moyenne. Cependant, la configuration à 100 agents, bien qu'elle permette un taux de motifs trouvés plus élevée, entraîne un coût computationnel plus élevé (jusqu'à $\times 2$ en terme de l'espace mémoire, et $\times 1.3$ en terme de temps d'exécution). Les résultats globaux suggèrent que l'optimisation du nombre d'agents et de la taille des graphes est cruciale pour obtenir les meilleures performances en matière de découverte de motifs.

6.7 Conclusion

Dans ce chapitre, nous proposons une nouvelle approche : l'approche de Deep Q-Learning multi-agent pour la découverte de motifs fréquents dans les multigraphes. Cette approche équilibre efficacement l'exploration et l'exploitation grâce à une sélection d'actions epsilon-greedy et à des récompenses basées sur la curiosité. Les agents apprennent à faire correspondre implicitement les motifs dans un sac de motifs prédéfini pendant l'entraînement, et pendant la phase de test, ils appliquent les politiques apprises pour découvrir de manière autonome des motifs fréquents sans avoir besoin de compter explicitement les fréquences. En exploitant plusieurs épisodes, les agents peuvent explorer différentes parties du multigraphe, garantissant la découverte d'une large gamme de motifs. L'approche se généralise bien aux nouveaux multigraphes et aux régions inconnues, ce qui en fait une méthode évolutive et robuste pour la découverte de motifs à grande échelle dans les multigraphes. L'algorithme 2 résume le processus proposé. Dans la section suivante, nous testerons cette approche et la comparerons en termes de taux de motifs trouvés à Multi-SPMiner, sur les données environnementales.

Input: Multigraphe $G = (V, E)$, Sac de motifs B (motifs fréquents), Nombre maximal d'épisodes M , Nombre maximal d'étapes par épisode T

Output: Motifs fréquents découverts

Initialiser(*Q-network* $Q(s, a|\theta)$, *tampon de réexpérience* \mathcal{D} , *mémoire globale* M_g , *pour les motifs découverts*, *paramètres d'apprentissage* $\alpha, \gamma, \epsilon_{start}, \epsilon_{decay}, \epsilon_{min}$);

```

for chaque épisode  $e = 1$  à  $M$  do
  for chaque agent  $i = 1$  à  $N$  do
    Initialiser un état de départ aléatoire  $s_0$ ;
    for chaque étape  $t = 1$  à  $T$  do
      if valeur aléatoire  $\leq \epsilon$  then
        Choisir une action aléatoire  $a_t$  (exploration);
      else
        Choisir l'action  $a_t = \arg \max_a Q(s_t, a|\theta)$  (exploitation);
      end
      Exécuter l'action  $a_t$  et observer l'état suivant  $s_{t+1}$  et la récompense  $R_t$ ;
      if le sous-graphe est un motif complet de taille  $K$  then
        if le motif correspond à un motif dans  $B$  then
          Attribuer la récompense  $R_{match}$ ;
        else
          Attribuer la pénalité  $R_{invalid}$ ;
        end
      else
        Attribuer les récompenses intermédiaires  $R_{partial}$ ;
        Attribuer la récompense de curiosité  $R_{curiosity}$ ;
        Attribuer la pénalité  $R_{disconnected}$  si le sous-graphe est invalide;
      end
      Stocker la transition  $(s_t, a_t, R_t, s_{t+1})$  dans le tampon de réexpérience  $\mathcal{D}$ ;
      MettreÀJourRéseauQ (Échantillonner depuis  $\mathcal{D}$  et mettre à jour  $Q(s_t, a_t|\theta)$ 
        en utilisant l'équation de Bellman);
       $s_t \leftarrow s_{t+1}$ ;
    end
    if le motif correspond à un inclus dans la  $B$  then
      Stocker le motif dans la mémoire globale  $M'_g$ ;
    end
  end
  Diminuer  $\epsilon$  :  $\epsilon = \max(\epsilon_{min}, \epsilon \cdot \epsilon_{decay})$ ;
end

```

Algorithm 2: Apprentissage Profond Multi-Agent Q-Learning pour la Découverte de Motifs dans les Multigraphes (Entraînement Conjoint)

TABLE 6.1 – Résultats des tests obtenus pour Deep-QMiner Multi-agents. C : la couverture

Nombre de nœuds	Nœuds/Temp	Distribution des relations	Motifs	30 Agents	60 Agents	100 Agents
200	40	[5,5,2]	16 de taille 5, 13 de taille 6	0.80 ± 0.02, C=0.98	0.83 ± 0.02, C=0.99	0.82 ± 0.03, C=1.00
1000	40	[5,5,2]	90 de taille 5, 75 de taille 6	0.53 ± 0.01, C=0.95	0.59 ± 0.02, C=0.97	0.58 ± 0.02, C=0.99
1000	40	[5,5,2]	75 de taille 8, 60 de taille 10	0.48 ± 0.01, C=0.96	0.53 ± 0.02, C=0.97	0.53 ± 0.03, C=0.99
1000	40	[5,5,2]	120 de taille 5, 100 de taille 6	0.52 ± 0.01, C=0.97	0.58 ± 0.02, C=0.97	0.58 ± 0.03, C=0.99
1000	40	[5,5,2]	80 de taille 4, 100 de taille 6	0.51 ± 0.02, C=0.92	0.55 ± 0.01, C=0.97	0.56 ± 0.03, C=0.99
1000	40	[8,8,3]	80 de taille 5	0.50 ± 0.02, C=0.94	0.56 ± 0.01, C=0.97	0.56 ± 0.03, C=0.99
1000	40	[2,2,2]	188 de taille 4, 125 de taille 6	0.49 ± 0.01, C=0.93	0.58 ± 0.02, C=0.97	0.59 ± 0.02, C=0.99
2000	40	[5,5,2]	10 motifs de 4, 5, 6, 8, 10 chacun	0.31 ± 0.02, C=0.91	0.32 ± 0.01, C=0.97	0.32 ± 0.02, C=0.99

7

Évaluation des méthodes sur les données environnementales et discussion

Sommaire

7.1	Données environnementales	134
7.1.1	Registre Parcellaire Graphique	134
7.1.2	Corine Land Cover	135
7.1.3	Génération des graphes spatio-temporels	136
7.1.4	Vérité terrain	137
7.2	Tests sur les données environnementales	137
7.3	Comparaison de la performance computationnelle des approches proposées :	140

Ce chapitre présente une évaluation approfondie des approches Multi-SPMiner et Deep Q-Miner sur des données environnementales réelles, en utilisant deux jeux de données essentiels : le Registre Parcellaire Graphique (RPG) et Corine Land Cover (CLC). Nous commençons par une description détaillée des jeux de données RPG et CLC. Ensuite, nous expliquons la méthodologie employée pour générer des graphes spatiotemporels à partir de ces données. La partie centrale de ce chapitre est consacrée à une série de tests. Nous évaluons les performances de différentes variantes de modèles GNN de Multi-SPMiner, ainsi que les performances de Deep Q-Miner avec différentes configurations d'agents. Ces tests sont menés sur des graphes générés à partir de données réelles et synthétiques, permettant une comparaison approfondie des approches. Enfin, nous analysons les résultats obtenus en termes de taux de motifs trouvés et de temps d'exécution, comparant nos approches entre elles et avec les MuGram. Cette analyse nous permet de mettre en lumière les forces et les limites de Multi-SPMiner et Deep Q-Miner dans le contexte de données environnementales réelles.

7.1 Données environnementales

7.1.1 Registre Parcellaire Graphique

Registre Parcellaire Graphique (RPG) ¹ a été créé en 2002 par la France en réponse aux exigences de l'Union Européenne, et recense les parcelles agricoles en échange des aides de la politique agricole commune (PAC). Il est mis à jour chaque année par les agriculteurs, qui y déclarent leurs îlots de culture et les types de cultures pratiquées. Avec plus de 7 millions de parcelles, il constitue une base de données géographique précieuse. Dans le RPG, les parcelles sont définies sous forme de polygones, représentés par les coordonnées géoréférencées de leurs sommets. Chaque polygone est considéré comme une entité connexe, sans trou, avec une frontière distincte et un intérieur bien défini. La version diffusée dans le framework du service public est anonymisée, avec certains détails retirés. Chaque parcelle contient des informations sur la culture principale, les groupes de cultures, et les cultures intermédiaires entre deux cultures annuelles. Les attributs contenus dans RPG et associés à chacune des parcelles correspondent à :

1. <https://www.data.gouv.fr/fr/datasets/registre-parcellaire-graphique-rpg-contours-des-parcelles-et-ilots-cultureaux-et-leur-groupe-de-cultures-majoritaire/>



FIGURE 7.1 – Extrait du millésime 2017 du RPG sur la plaine d'Alsace

- un identifiant unique de la parcelle,
- un code permettant de connaître la culture principale annuelle sur la parcelle,
- un code permettant de connaître à quel groupe de cultures appartient la culture de la parcelle,
- deux codes permettant de connaître les cultures dérochées, c'est-à-dire les cultures intercalées entre deux cultures annuelles.

7.1.2 Corine Land Cover

Le jeu de données Corine Land Cover (CLC)² est une ressource essentielle pour la recherche environnementale et l'analyse de l'utilisation des sols. Développé par l'Agence Européenne pour l'Environnement (AEE), il offre une représentation complète et à haute résolution de la couverture des sols dans les pays européens. Le jeu de données présente une structure hiérarchique à trois niveaux de détail, délimitant méticuleusement des catégories spécifiques telles que les zones urbaines, les zones

2. <https://land.copernicus.eu/en/products/corine-land-cover>

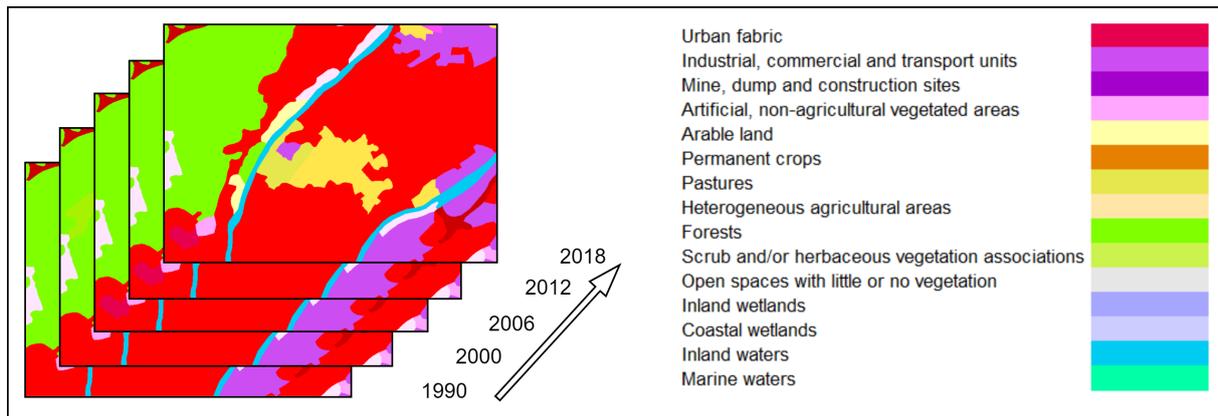


FIGURE 7.2 – Un extrait d'exemple du jeu de données Corine Land Cover (CLC) illustrant l'évolution de la région parisienne, où différents types de terres sont codés par couleur. Chaque segment se voit attribuer un identifiant unique (ID) essentiel pour le suivi des identités, tandis que les types de terres servent de caractéristiques des nœuds dans le processus de création du graphe spatiotemporel. A droite, les différents types de zones utilisant le 2ème niveau de hiérarchie

agricoles, les forêts et les plans d'eau. La granularité cartographique utilisée est de 25 hectares, et le jeu de données CLC couvre 39 pays, offrant des informations précieuses sur les tendances et les modèles de couverture des sols à l'échelle continentale. La série chronologique à long terme du jeu de données CLC, comprenant cinq itérations (1990, 2000, 2006, 2012, 2018), permet une analyse historique et la détection de tendances. Le système de classification détaillé, qui a été appliqué de manière cohérente au fil des différentes périodes temporelles, fait du jeu de données CLC un outil indispensable pour surveiller les changements d'utilisation des sols, évaluer les impacts environnementaux et élaborer des stratégies de gestion durable des terres.

7.1.3 Génération des graphes spatio-temporels

Pour évaluer l'efficacité des algorithmes proposés sur les jeux de données RPG et CLC, nous suivons la méthodologie décrite dans [137]. Une approche est adoptée pour générer des graphes spatiotemporels (graphes ST), en sélectionnant aléatoirement diverses villes européennes. Ensuite, des graphes spatiotemporels sont construits en se basant sur les temporalités distinctes disponibles dans les jeux de données (figure 7.3). Pendant la phase de construction, l'accent est mis sur l'expansion progressive de la région géographique sélectionnée à travers toutes les périodes temporelles, afin d'obtenir des graphes de taille fixe, sans dépasser 1000 nœuds. Les identifiants fournis dans les jeux de données sont utilisés pour identifier les nœuds, tout en prenant en compte le second niveau des classes pour identifier les types d'utilisation des sols (15 types différents) pour la base CLC. Les relations spatiales, spatiotemporelles et de

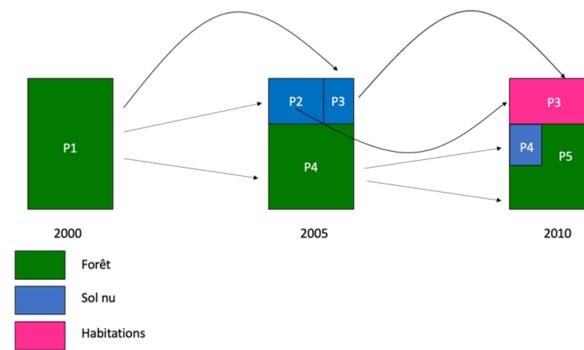


FIGURE 7.3 – Exemple de construction d'un graphe spatio-temporel. Les données représentent une région spatiale évoluant dans le temps, où P1, P2, P3, P4 et P5 correspondent, par exemple, à des parcelles agricoles.

filiation entre les nœuds sont calculées conformément à la méthodologie présentée dans [137]. Afin de mieux représenter les phénomènes sur les parcelles agricoles, deux types de proximité ont été définis pour les relations spatiales, et un modèle de relations floues a été proposé pour prendre en compte l'incertitude liée à la localisation des mêmes parcelles au fil du temps. Les relations de filiation, construites grâce aux identités des parcelles fournies par le RPG, permettent ainsi d'obtenir un graphe spatiotemporel complet, conformément à la définition donnée par Del Mondo et al. [138].

7.1.4 Vérité terrain

Pour établir une vérité terrain fiable pour l'évaluation du modèle, et compte tenu de l'impossibilité d'obtenir les motifs au sein des graphes spatiotemporels construits, nous avons opté pour l'implémentation de l'algorithme Mugram [112]. Mugram génère une liste exhaustive des motifs spatiotemporels fréquents disponibles pour un seuil donné. L'algorithme adopte une approche exhaustive, testant tous les motifs fréquents potentiels en l'occurrence on ne peut pas vérifier. Bien que Mugram soit gourmand en ressources et nécessite un temps d'exécution important, il constitue une solution appropriée pour obtenir la vérité terrain.

7.2 Tests sur les données environnementales

En commençant par Multi-SPMiner, deux évaluations principales ont été effectuées pour chaque modèle GNN. Lors du premier test, les modèles ont été entraînés sur des graphes synthétiques contenant des motifs de tailles variées. Ensuite, les modèles ont été évalués sur des graphes spatiotemporels générés à partir de 20 régions européennes sélectionnées aléatoirement pour le jeu de données CLC, et de régions

TABLE 7.1 – Résultats Multi-SPMiner obtenus sur le jeu de données CLC, après un entraînement sur des graphes générés synthétiquement et des graphes de données réelles.

Taille du motif cible	Données d'entraînement	Baseline	GAT	RGCN	GGCN	HAN	MGAT	MMAGAT	Moyenne
6	Synthétique	0.67 ± 0.01	0.69 ± 0.02	0.70 ± 0.02	0.70 ± 0.01	0.73 ± 0.03	0.72 ± 0.01	0.73 ± 0.02	0.70 ± 0.02
	CLC	0.68 ± 0.01	0.69 ± 0.02	0.71 ± 0.01	0.71 ± 0.02	0.74 ± 0.03	0.73 ± 0.02	0.74 ± 0.01	0.72 ± 0.02
8	Synthétique	0.66 ± 0.03	0.70 ± 0.02	0.69 ± 0.03	0.69 ± 0.01	0.71 ± 0.02	0.68 ± 0.03	0.71 ± 0.02	0.69 ± 0.02
	CLC	0.66 ± 0.02	0.70 ± 0.03	0.71 ± 0.01	0.72 ± 0.03	0.74 ± 0.01	0.73 ± 0.02	0.73 ± 0.03	0.71 ± 0.02
10	Synthétique	0.65 ± 0.01	0.68 ± 0.03	0.70 ± 0.03	0.67 ± 0.02	0.68 ± 0.01	0.68 ± 0.03	0.69 ± 0.02	0.68 ± 0.02
	CLC	0.64 ± 0.03	0.67 ± 0.02	0.68 ± 0.03	0.68 ± 0.01	0.70 ± 0.02	0.69 ± 0.03	0.69 ± 0.01	0.68 ± 0.02

TABLE 7.2 – Résultats Multi-SPMiner sur le jeu de données RPG, après un entraînement sur des graphes générés synthétiquement et des graphes de données réelles.

Taille du motif cible	Données d'entraînement	Baseline	GAT	RGCN	GGCN	HAN	MGAT	MMAGAT	Moyenne
6	Synthétique	0.69 ± 0.02	0.71 ± 0.03	0.72 ± 0.03	0.68 ± 0.02	0.74 ± 0.04	0.70 ± 0.02	0.75 ± 0.03	0.71 ± 0.03
	RPG	0.69 ± 0.01	0.70 ± 0.02	0.73 ± 0.02	0.69 ± 0.02	0.76 ± 0.03	0.75 ± 0.02	0.72 ± 0.01	0.73 ± 0.02
8	Synthétique	0.64 ± 0.02	0.72 ± 0.03	0.70 ± 0.02	0.71 ± 0.03	0.72 ± 0.04	0.66 ± 0.02	0.70 ± 0.03	0.69 ± 0.03
	RPG	0.67 ± 0.02	0.73 ± 0.03	0.70 ± 0.01	0.74 ± 0.02	0.76 ± 0.01	0.71 ± 0.01	0.74 ± 0.03	0.72 ± 0.02
10	Synthétique	0.66 ± 0.02	0.67 ± 0.02	0.69 ± 0.04	0.66 ± 0.03	0.70 ± 0.03	0.67 ± 0.02	0.68 ± 0.03	0.68 ± 0.03
	RPG	0.66 ± 0.03	0.70 ± 0.02	0.69 ± 0.02	0.70 ± 0.01	0.72 ± 0.03	0.70 ± 0.03	0.68 ± 0.01	0.69 ± 0.02

aléatoires en France pour le jeu de données RPG. Les graphes synthétiques utilisés pour l'entraînement, ont été soigneusement conçus pour inclure des motifs de haute fréquence des tailles spécifiées. De plus, afin de se rapprocher des données réelles, ces graphes synthétiques incorporaient cinq temporalités. Ce test visait à évaluer la capacité du modèle à être entraîné une seule fois puis appliqué à n'importe quel graphe cible provenant de domaines divers.

La deuxième évaluation a consisté à entraîner les modèles sur des graphes dérivés de différentes localisations au sein des jeux de données CLC et RPG. L'objectif principal de cette évaluation était l'identification de motifs de tailles 6, 8 et 10. Pour chaque scénario de test, nous avons généré 200 graphes pour l'entraînement et un ensemble supplémentaire de 20 graphes pour les tests.

Les tableaux 7.1 et 7.2 résumant les résultats de taux de motifs trouvés obtenus par Multi-SPMiner lors des tests réalisés sur les jeux de données CLC et RPG. L'accent a été mis sur l'évaluation de le taux de motifs trouvés dans l'identification des motifs de tailles 6, 8 et 10. Conformément aux résultats précédents, les modèles basés sur l'attention ont démontré un léger avantage en termes de performances. On peut aussi observer que tous les modèles ont obtenu des taux de motifs trouvés légèrement supérieures pour les motifs de plus petite taille. En examinant l'impact de la source des données d'entraînement, les résultats de taux de motifs trouvés montrent une variation négligeable. Cette variation est probablement dû aux différences dans les caractéristiques des nœuds entre les jeux de données, mais sa faible valeur souligne l'adaptabilité du modèle à différents types de données après l'entraînement. Ce premier test montre notamment la polyvalence de Multi-SPMiner à gérer des ensembles de données divers. Par ailleurs, aucune différence significative n'a été observée entre les taux de motifs obtenues sur les deux jeux de données, confirmant ainsi le fort potentiel de généralisation des modèles.

TABLE 7.3 – Résultats obtenus sur le jeu de données RPG, après un entraînement sur des graphes générés synthétiquement et des graphes de données réelles, utilisant Deep-Qminer

Taille du motif cible	Données d'entraînement	30 Agents	60 Agents	100 Agents
6	Synthétique	0.48 ± 0.02, C=0.92	0.57 ± 0.01, C=0.98	0.55 ± 0.02, C=0.99
	RPG	0.52 ± 0.01, C=0.93	0.59 ± 0.02 , C=0.98	0.58 ± 0.03, C=0.99
8	Synthétique	0.42 ± 0.02, C=0.92	0.51 ± 0.01, C=0.98	0.50 ± 0.02, C=0.98
	RPG	0.53 ± 0.01, C=0.93	0.53 ± 0.02, C=0.97	0.54 ± 0.03, C=0.99
10	Synthétique	0.40 ± 0.01, C=0.92	0.53 ± 0.02, C=0.97	0.51 ± 0.03, C=0.98
	RPG	0.47 ± 0.02, C=0.93	0.53 ± 0.01, C=0.98	0.56 ± 0.02, C=0.99

TABLE 7.4 – Résultats obtenus sur le jeu de données CLC, après entraînement sur des graphes générés de manière synthétique et des graphes de données réelles, en utilisant Deep-Qminer

Taille du motif cible	Données d'entraînement	30 Agents	60 Agents	100 Agents
6	Synthétique	0.50 ± 0.02, C=0.93	0.55 ± 0.02, C=0.97	0.57 ± 0.03, C=0.98
	CLC	0.54 ± 0.01, C=0.92	0.60 ± 0.03 , C=0.99	0.59 ± 0.02 , C=1.00
8	Synthétique	0.44 ± 0.03, C=0.93	0.49 ± 0.02, C=0.97	0.53 ± 0.03, C=0.98
	CLC	0.55 ± 0.01, C=0.92	0.52 ± 0.02, C=0.98	0.58 ± 0.03, C=0.99
10	Synthétique	0.43 ± 0.02, C=0.93	0.50 ± 0.02, C=0.99	0.54 ± 0.03, C=0.99
	CLC	0.49 ± 0.01, C=0.92	0.54 ± 0.01, C=0.98	0.55 ± 0.02, C=0.99

Les tableaux 7.3 and 7.4 présentent les résultats obtenus avec Deep-QMiner sur les les même graphes. Ces résultats correspondent aux résultats obtenus sur des tests précédents, bien que l'on puisse observer un écart notable dans le taux de motifs trouvés lorsque le modèle s'est entraîné sur des graphes réels par rapport aux graphes synthétiques, en particulier avec un modèle de 60 agents. Globalement, le modèle parvient à atteindre un taux de motifs trouvés moyenne de 60% pour les motifs de taille 6, les meilleurs résultats obtenus avec cette approche dans tous les tests. Les variations de taux de motifs trouvés entre les agents (allant de 0,01 à 0,03) indiquent que les performances du modèle peuvent fluctuer légèrement d'un essai à l'autre, mais dans l'ensemble, les résultats restent relativement stables, les configurations avec un plus grand nombre d'agents obtenant de meilleures performances de manière constante, et une couverture plus élevés. Cependant, le bénéfice marginal d'une augmentation de 60 à 100 agents semble diminuer, comme le montrent les augmentations relativement faibles de le taux de motifs trouvés et de la couverture entre ces deux configurations. Les tableaux 7.3 and 7.4 révèlent que Deep-Qminer fonctionne légèrement mieux sur les ensembles de données réelles que sur les données générées synthétiquement. Cela pourrait indiquer que les graphes du monde réel offrent des motifs plus structurés et plus faciles à identifier, ou que les données synthétiques introduisent du bruit qui affecte le processus de recherche de motif.

7.3 Comparaison de la performance computationnelle des approches proposées :

L'analyse des coûts en mémoire de Multi-SPMiner, similaire à SPMiner, est influencée par le nombre de voisins et la dimension du plongement. Le coût mémoire estimé suit une expression de $O((N + K)d + Kn^2)$, prenant en compte le nombre de voisinages (N), la dimension de l'embedding (d), le nombre d'itérations du décodeur (K) et la taille de motif souhaitée (n). L'analyse du temps d'exécution prend en compte des facteurs tels que les voisinages, la taille des motifs, les nœuds et la dimension de l'embedding, avec une complexité de $O(Nb^2 + Kn(n^2 + Nnd))$. Comparé à SPMiner, Multi-SPMiner est plus complexe en raison de l'adoption d'un réseau de neurones convolutif multigraphe (MGCN) et de la prise en compte de multigraphes dirigés et étiquetés. Néanmoins, Multi-SPMiner présente un temps d'exécution et une utilisation de mémoire efficaces, sans augmenter significativement la complexité par rapport au modèle de base. En ce qui concerne la complexité temporelle du multi-agent Deep QMiner, elle est de

$$O(M \cdot N \cdot T \cdot (\Delta + B \cdot L \cdot n_l^2))$$

où M est le nombre d'épisodes, N est le nombre d'agents, T est le nombre maximal d'étapes par épisode (lié à la taille du motifs), Δ est le degré maximal des nœuds (voisins), B est la taille du batch pour les mises à jour du réseau Q, L est le nombre de couches du réseau Q, et n_l est le nombre de neurones par couche. La complexité spatiale est de

$$O(\mathcal{D} \cdot (S + A)) + O(L \cdot n_l^2) + O(v)$$

où \mathcal{D} est la taille de la mémoire tampon des expériences, S est la dimension de l'espace des états, A est le nombre d'actions possibles, et v est le nombre de motifs découverts. Cela reflète le coût computationnel du stockage des transitions, de l'entraînement du réseau Q et de la gestion de la mémoire pour les motifs découverts.

Les résultats présentés dans la figure 7.4 illustrent les temps d'exécution moyens pour différents modèles dans l'extraction de motifs fréquents de tailles variées (sur la base CLC). Il est à noter que tous les modèles ont affiché des temps d'exécution moyens inférieurs à une demi-minute lorsqu'ils sont appliqués aux graphes du jeu de données CLC. Il est également important de mentionner que l'algorithme Mugram, utilisé pour obtenir les motifs de référence, a obtenu un temps d'exécution moyen de 14 minutes, ce qui souligne l'efficacité de l'adoption de l'approche Multi-SPMiner. L'analyse de la figure 7.4 révèle que GGCN a enregistré le temps d'exécution le plus long, se classant derrière trois autres modèles en termes de performance. À l'inverse,

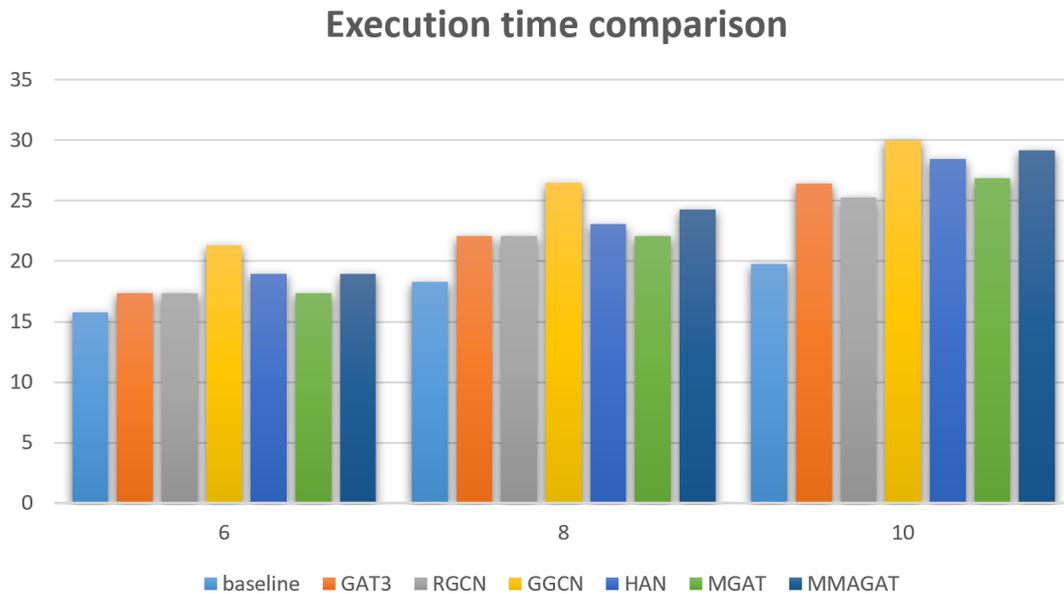


FIGURE 7.4 – Comparaison des temps d'exécution (en secondes) en utilisant différents modèles.

le modèle de base a réalisé le temps d'exécution le plus rapide, mais qu'avec le taux de motifs trouvés la plus faible.

Nos expériences démontrent que Multi-SPMiner avec les réseaux de convolution de graphes relationnels (R-GCNs) offrent un équilibre convaincant entre le taux de motifs trouvés et efficacité. Cependant, pour atteindre le plus grand taux de motifs trouvés possible, les réseaux Multi-Head Meta Attention (MMA-GATs) s'imposent comme le meilleur choix. Des recherches supplémentaires sont nécessaires pour étudier la façon dont ces modèles évoluent avec des graphes de plus en plus grands et complexes. La figure 7.5 met en évidence les performances temporelles et la précision des différents modèles de machine learning pour des motifs de tailles croissantes. Les modèles HAN, GGCCN et MGAT atteignent rapidement une précision élevée, ce qui montre leur efficacité. En revanche, le modèle Baseline est plus lent à atteindre sa précision maximale, ce qui reflète son inefficacité comparative face aux modèles avancés. De son côté, MMAGAT semble être le plus lent à converger dans les trois cas (motifs de tailles 6, 8, et 10). Globalement, la taille du motif semble affecter légèrement les courbes, mais les tendances générales restent similaires pour tous les modèles. Ces observations permettent de comparer les compromis entre précision et temps d'exécution selon les tailles des motifs et les modèles étudiés.

Pour l'approche multi-agent Deep Q-Miner, après avoir sélectionné les paramètres optimaux identifiés lors de la phase de test et en fixant le nombre d'épisodes à 10 000, l'approche nécessite jusqu'à 5 minutes pour s'exécuter. Cela est nettement plus long que l'approche Multi-SPMiner la plus lente. L'augmentation du temps de calcul peut

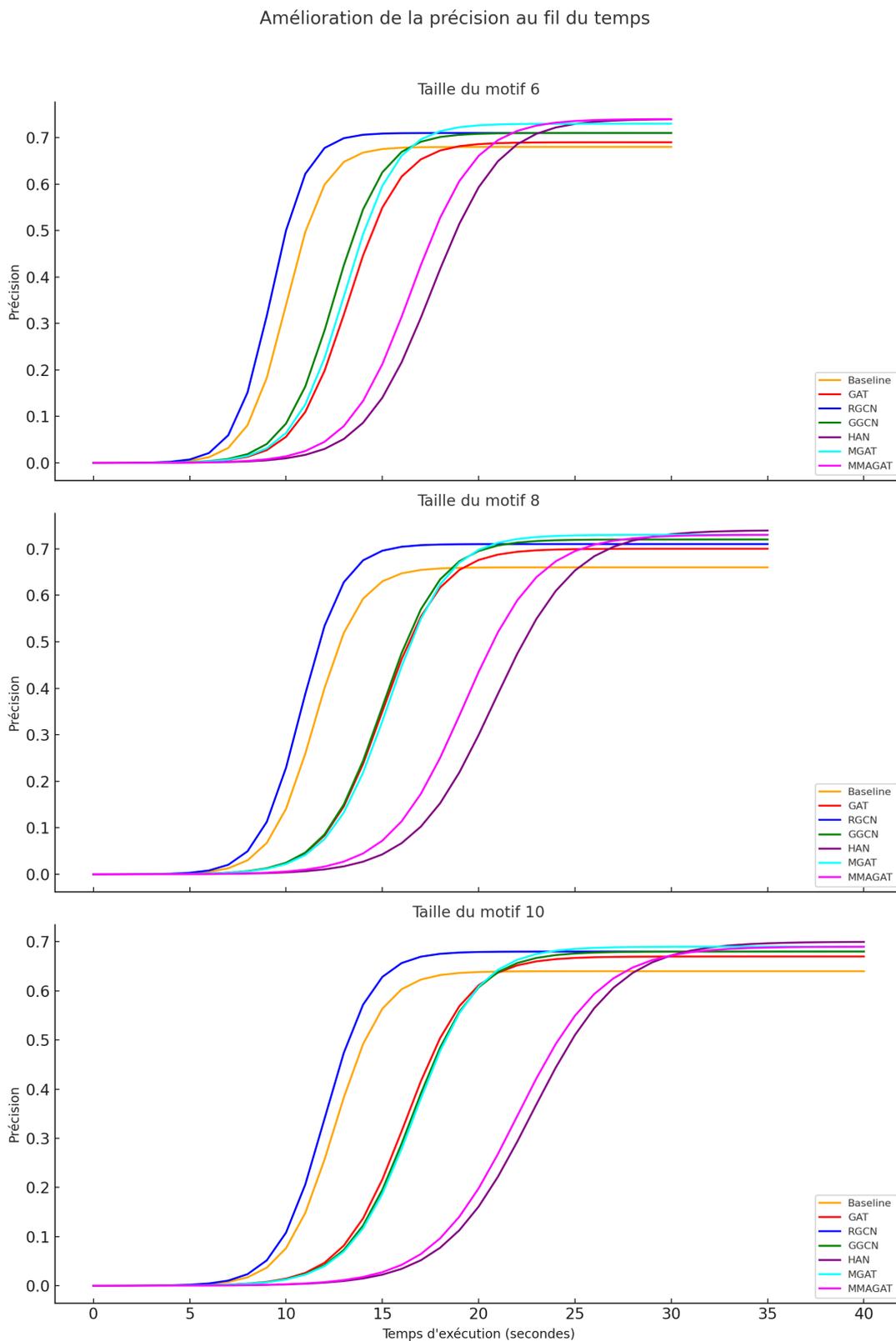


FIGURE 7.5 – L'évolution du taux de motifs trouvés des modèles en fonction du temps d'exécution pour différentes tailles de motifs (6, 8, et 10). Chaque courbe représente un modèle spécifique : Baseline, GAT, RGCN, GGCN, HAN, MGAT, et MMAGAT.

être attribuée à la complexité introduite par le cadre multi-agent et à la nature itérative du Deep Q-learning, qui implique non seulement la génération de motifs mais aussi l'apprentissage de politiques optimales à travers plusieurs agents.

Malgré son temps d'exécution plus long et le taux de motifs trouvés réduite par rapport à Multi-SPMiner, Les résultats présentés dans cette thèse démontrent la preuve de concept de Multi-Agent Deep Q-Miner et ouvrent une piste de recherche pour cette approche, en utilisant l'apprentissage par renforcement pour la découverte de motifs dans les multigraphes. La principale force de cette approche réside dans sa capacité à gérer des scénarios où la taille des motifs est minimale. Dans ces cas, le cadre multi-agent produit des résultats fiables de manière constante ,surpassant Multi-SPMiner, montrant un potentiel pour les tâches de découverte de motifs.

De plus, l'utilisation de plusieurs agents dans Deep Q-Miner permet au modèle d'explorer simultanément différentes régions du graphe, ce qui pourrait conduire à une meilleure généralisation dans des scénarios plus complexes (similaire aux tests effectués sur des données synthétiques). Bien que cette thèse se soit concentrée sur la démonstration de la viabilité du concept avec des motifs de petite taille, il existe une marge significative d'amélioration et d'expansion. Les travaux futurs pourraient se concentrer sur l'optimisation du processus d'entraînement et l'ajustement des paramètres pour réduire le temps de calcul, notamment lorsqu'il s'agit de graphes plus grands ou de motifs plus complexes.

Le principal compromis dans le multi-agent Deep Q-Miner réside entre le taux de motifs trouvés et le temps de calcul. L'apprentissage par renforcement offrent des avantages potentiels pour des problèmes de découverte de motifs plus complexes à l'avenir. De plus, les résultats de cette recherche établissent une base solide pour un développement ultérieur, montrant que l'approche peut évoluer et s'adapter à mesure qu'elle continue d'être perfectionnée.

8

Conclusion et perspectives

Sommaire

8.1	Résumé des contributions	145
8.2	Discussion des résultats	146
8.2.1	Performance de Multi-SPMiner	146
8.2.2	Efficacité de Deep-QMiner	147
8.2.3	Performance sur les données du monde réel	148
8.3	Objectifs et leur réalisation	148
8.4	Limitations et défis	149
8.5	Directions futures	150

8.1 Résumé des contributions

Cette thèse propose des avancées dans le domaine de l'extraction de motifs fréquents pour les graphes spatio-temporels, en exploitant la puissance des techniques d'apprentissage profond et d'apprentissage par renforcement. Notre travail a comblé une lacune dans la littérature existante en adaptant et en étendant des méthodes de pointe pour gérer les complexités des données spatio-temporelles représentées sous forme de multigraphes. Les principales contributions de cette recherche peuvent être résumées comme suit :

1. **Adaptation de SPMiner pour les multigraphes** : Nous avons réussi à adapter l'approche SPMiner, initialement conçue pour des graphes simples, pour fonctionner avec des multigraphes représentant des données spatio-temporelles. Cette adaptation, que nous avons nommée Multi-SPMiner, a impliqué des modifications significatives pour gérer plusieurs types d'arêtes et des informations temporelles, élargissant ainsi l'applicabilité des techniques d'extraction de sous-graphes fréquents par réseaux de neurones à des structures de données plus complexes. De plus, nous avons prouvé que notre modèle Multi-SPMiner est une généralisation du modèle de base SPMiner, et démontré sa capacité de généralisation en l'entraînant sur des données synthétiques et en le testant sur différents types de données.
2. **Architectures améliorées de réseaux de neurones de graphes** : Nous avons exploré et mis en œuvre diverses architectures de réseaux de neurones de graphes (GNN), notamment GAT, RGCN, GGCN, HAN, MGAT et MMA-GAT, pour améliorer les performances de Multi-SPMiner. Ces architectures ont été sélectionnées et adaptées pour capturer les relations spatiales et temporelles complexes présentes dans nos données. Nous avons également effectué une comparaison des plongements obtenus pour chaque modèle.
3. **Amélioration de la fonction de perte et du processus d'entraînement** : Nous avons introduit des améliorations à la fonction de perte et au processus d'entraînement de Multi-SPMiner, notamment la perte de marge pondérée, la perte de fréquence et l'extraction d'exemples négatifs difficiles. Ces améliorations ont

conduit à une découverte de motifs plus robuste et précise, comme le démontrent nos résultats expérimentaux.

4. **Nouvelle approche d'apprentissage par renforcement** : Nous avons proposé une nouvelle méthode, Deep-QMiner, qui utilise l'apprentissage par renforcement multi-agents pour la découverte de motifs fréquents dans les graphes spatio-temporels. Cette approche offre une perspective différente sur le problème et montre des résultats prometteurs, en particulier pour les graphes et les motifs de petite taille.
5. **Cadre d'évaluation complet** : Nous avons développé un protocole d'évaluation rigoureux, testant nos méthodes sur des ensembles de données synthétiques et des données environnementales réelles (ensembles de données CLC et RPG). Ce cadre permet une évaluation approfondie des performances et de la généralisabilité de nos approches.
6. **Capacité de généralisation sur les données** : Nous avons démontré que les modèles développés, entraînés sur un jeu de données synthétiques et testés sur un autre jeu de données aux caractéristiques similaires, obtiennent des résultats faiblement dégradés. Cette robustesse permet d'envisager l'utilisation d'un modèle unique pour différents graphes partageant des propriétés communes, sans nécessiter un réentraînement systématique.

8.2 Discussion des résultats

Les résultats obtenus à partir de nos expériences approfondies fournissent des informations précieuses sur l'efficacité de nos méthodes proposées et leurs applications potentielles dans des scénarios réels.

8.2.1 Performance de Multi-SPMiner

Multi-SPMiner a démontré des performances cohérentes et robustes dans divers scénarios de test, tant avec des données synthétiques que réelles. Le modèle a atteint des taux de motifs trouvés allant de 60% à 75%, selon l'architecture GNN spécifique utilisée et la complexité des graphes d'entrée. Les résultats notables incluent :

- **Comparaison des architectures GNN** : Les modèles basés sur l'attention, en particulier MMA-GAT, ont systématiquement surpassé le modèle de base, avec une amélioration moyenne de le taux de 3% à 5%. Cela souligne l'importance de capturer des relations complexes dans les graphes spatio-temporels grâce à des mécanismes d'attention sophistiqués. La méthode peut être sujette à identifier

des motifs qui ne sont pas fréquents dans le graphe en raison d'erreurs dans l'encodage. Son efficacité repose sur la précision du modèle utilisé avec la méthode. De plus, lors des tests précédents, un tel cas n'a pas été rencontré, mais nous ne pouvons pas théoriquement exclure sa possibilité.

- **Extensibilité** : Multi-SPMiner a montré une bonne extensibilité, maintenant des performances cohérentes à mesure que la taille des graphes augmentait de 200 à 2000 nœuds. C'est un avantage crucial pour les applications du monde réel où l'analyse de données à grande échelle est souvent nécessaire.
- **Capacité de généralisation** : Le modèle a démontré de fortes capacités de généralisation, obtenant de bons résultats sur des motifs de tailles différentes de celles utilisées pendant l'entraînement. Cela suggère que Multi-SPMiner apprend des modèles structurels fondamentaux plutôt que de s'ajuster à des tailles de motifs spécifiques.
- **Robustesse** : Les expériences avec suppression de nœuds ont montré que Multi-SPMiner est robuste aux données manquantes, maintenant ses performances même avec jusqu'à 10% des nœuds supprimés. Cette résilience est particulièrement précieuse pour les ensembles de données du monde réel où l'incomplétude des données est courante.

8.2.2 Efficacité de Deep-QMiner

L'approche Deep-QMiner, basée sur l'apprentissage par renforcement multi-agents, a montré des résultats prometteurs, en particulier pour les graphes et les motifs plus petits. Les observations clés incluent :

- **Performance sur les petits graphes** : Deep-QMiner a surpassé Multi-SPMiner sur les graphes de 200 nœuds, atteignant des taux allant jusqu'à 80%. Cela suggère que l'approche d'apprentissage par renforcement est particulièrement efficace pour des espaces de problèmes plus petits et plus gérables.
- **Défis d'extensibilité** : Les performances de Deep-QMiner se sont dégradées plus rapidement que celles de Multi-SPMiner à mesure que la taille des graphes augmentait. Cela indique qu'une optimisation supplémentaire est nécessaire pour améliorer son extensibilité pour les graphes plus grands.
- **Configuration des agents** : Les expériences avec différents nombres d'agents (30, 60 et 100) ont révélé que 60 agents offraient le meilleur équilibre entre couverture et efficacité computationnelle. Cette information est précieuse pour ajuster le modèle à différents scénarios d'application.

8.2.3 Performance sur les données du monde réel

L'application de nos méthodes aux ensembles de données CLC et RPG a donné des résultats encourageants :

- **Cohérence entre les ensembles de données** : Multi-SPMiner et Deep-QMiner ont montré des performances cohérentes sur ces deux ensembles de données, indiquant une bonne généralisation à différents types de données environnementales.
- **Impact de la taille des motifs** : Les performances étaient généralement meilleures pour les motifs de plus petite taille (6 nœuds) par rapport aux plus grands (8 et 10 nœuds). Cela suggère que nos méthodes sont particulièrement efficaces pour capturer des motifs locaux dans les données spatio-temporelles, ce qui est attribué au fait que les petits motifs sont plus fréquents.
- **Source des données d'entraînement** : De manière intéressante, il y avait peu de différence de performance entre les modèles entraînés sur des données synthétiques et ceux entraînés sur les ensembles de données du monde réel. Cela valide notre approche de génération de données synthétiques et suggère que nos modèles peuvent être efficacement pré-entraînés sur des données synthétiques avant d'être affinés sur des ensembles de données du monde réel.

8.3 Objectifs et leur réalisation

Dans l'introduction de cette thèse, nous avons établi plusieurs objectifs clés afin de répondre aux défis liés à la recherche de motifs fréquents dans les graphes spatio-temporels. Ces objectifs étaient les suivants :

- **Adapter et améliorer les méthodes existantes pour les graphes spatio-temporels** : Nous avons réussi à adapter des méthodes existantes et à les appliquer efficacement sur nos jeux de données. De plus, nous avons amélioré la fonction de perte et optimisé le processus d'entraînement, ce qui a permis de renforcer le taux de motifs trouvés et l'efficacité de la détection des motifs.
- **Développer une approche basée sur l'apprentissage par renforcement pour la découverte de motifs** : Cet objectif a été atteint par la mise en place d'un système multi-agents d'apprentissage par renforcement, qui a démontré sa capacité à explorer le graphe de manière efficace et à identifier des motifs fréquents, tant sur des données synthétiques que sur des données réelles.

- **Évaluer les performances des méthodes proposées sur des données synthétiques et réelles** : Nous avons mené des expérimentations sur des jeux de données synthétiques et réels, malgré certaines limitations telles que l’absence de vérité terrain pour les données réelles.

En conclusion, tous les objectifs fixés dans l’introduction ont été globalement atteints. Les méthodes développées ont permis d’améliorer significativement la recherche de motifs dans les graphes spatio-temporels, et les résultats obtenus ont confirmé la pertinence des approches proposées. Des pistes d’amélioration restent envisageables, notamment pour la mise à l’échelle, l’amélioration des taux de motifs trouvés et l’optimisation de la consommation de ressources.

Néanmoins, certaines limitations ont été rencontrées au cours de cette thèse, et elles ont été soit partiellement traitées, soit laissées pour des recherches futures. Dans la section suivante, nous discutons des limitations rencontrées et ouvrons des perspectives pour des travaux futurs qui découlent de cette thèse.

8.4 Limitations et défis

Bien que notre recherche ait réalisé des avancées significatives, il est important de reconnaître les limitations et les défis rencontrés :

1. **Manque de vérité terrain pour les données réelles** : L’un des principaux défis auxquels nous avons été confrontés était l’absence d’une vérité terrain définitive pour les motifs dans les ensembles de données CLC et RPG du monde réel. Cette limitation a rendu difficile l’évaluation précise de l’exactitude de nos méthodes sur ces ensembles de données. Pour atténuer cela, nous nous sommes appuyés sur des données synthétiques avec une vérité terrain connue et avons utilisé l’algorithme Mugram comme référence pour la comparaison. Cependant, l’absence de vérité terrain du monde réel reste une contrainte pour valider pleinement nos approches dans des scénarios pratiques.
2. **Ressources computationnelles** : Multi-SPMiner et Deep-QMiner, en particulier ce dernier, peuvent être intensifs en calcul pour les grands graphes, plus particulièrement lors de la phase d’entraînement. L’utilisation de la mémoire peut être substantielle, notamment lors du traitement de relations spatio-temporelles complexes et de plusieurs types d’arêtes. Cette limitation peut poser des défis pour le passage à l’échelle de très grands ensembles de données ou pour des applications disposant de ressources computationnelles limitées.

3. **Extensibilité de Deep-QMiner** : Bien que Deep-QMiner ait montré des résultats prometteurs pour les graphes plus petits, ses performances se sont dégradées plus rapidement que celles de Multi-SPMiner à mesure que la taille des graphes augmentait. Ce problème d’extensibilité limite son applicabilité aux réseaux du monde réel plus grands sans optimisation supplémentaire.
4. **Complexité des relations spatio-temporelles** : Capturer toute la complexité des relations spatio-temporelles, en particulier dans les données environnementales, reste un défi. Bien que nos méthodes montrent une amélioration par rapport aux approches existantes, il y a encore de la place pour une meilleure modélisation des dynamiques temporelles complexes et des dépendances spatiales.
5. **Interprétabilité** : Comme pour de nombreuses approches d’apprentissage profond, l’interprétabilité des motifs appris et des processus de prise de décision dans nos modèles peut être limitée. Cela peut être un inconvénient important dans les domaines où la compréhension du raisonnement derrière les motifs identifiés est cruciale.
6. **Sensibilité aux paramètres** : Multi-SPMiner et Deep-QMiner impliquent de nombreux hyperparamètres qui peuvent affecter significativement les performances. Trouver des configurations optimales pour différents ensembles de données et scénarios d’application peut prendre du temps et nécessiter une expertise du domaine.

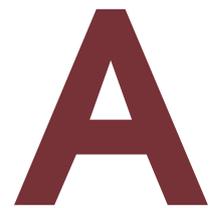
8.5 Directions futures

Sur la base des enseignements tirés de cette recherche et des limitations identifiées, plusieurs pistes prometteuses pour les travaux futurs se dégagent :

1. **Modélisation temporelle avancée** : L’incorporation de techniques de modélisation temporelle plus sophistiquées, telles que des mécanismes d’attention temporelle ou des architectures récurrentes spécifiquement conçues pour les dépendances à long terme, pourrait améliorer la capture de motifs temporels complexes dans les graphes spatio-temporels. Cela serait particulièrement intéressant à tester sur Deep-QMiner.
2. **Techniques d’IA explicable** : Développer des méthodes pour améliorer l’interprétabilité des motifs découverts et du processus de prise de décision de nos modèles serait précieux. Cela pourrait impliquer des techniques d’IA explicable ou l’intégration de connaissances du domaine pour guider le processus de découverte de motifs.

3. **La visualisation** : Nous avons manqué d'un outil viable pour visualiser les motifs et les graphes. Le développement d'un tel outil aiderait grandement dans le processus de compréhension des modèles ou des motifs trouvés. La visualisation des motifs découverts et de leur évolution au fil du temps pourrait offrir des acquis précieux sur le fonctionnement de nos modèles et sur la nature des patterns détectés dans les données spatio-temporelles.
4. **Application aux motifs approximatifs** : Bien que notre recherche se concentre principalement sur l'extraction de motifs exacts, nous reconnaissons l'importance potentielle des motifs approximatifs dans certains contextes. Les motifs approximatifs peuvent être particulièrement utiles dans des situations où les données sont bruitées ou incomplètes, comme dans l'analyse de données environnementales ou biologiques, ou lorsqu'une certaine flexibilité est nécessaire pour capturer des tendances générales plutôt que des correspondances exactes. Ces motifs peuvent révéler des similarités subtiles ou des tendances qui pourraient être manquées par une recherche de motifs exacts. Bien que cet aspect ne soit pas au cœur de notre travail actuel, il représente une piste intéressante pour des recherches futures.
5. **Améliorations de l'extensibilité** : Concernant Deep-QMiner, l'exploration de techniques telles que l'apprentissage par renforcement hiérarchique ou des représentations d'état plus efficaces pourrait améliorer son extensibilité aux graphes plus grands. De plus, une approche intéressante pourrait être l'ajout de plusieurs stratégies à la fois qui peuvent être apprises par les agents, et l'introduction d'un système de récompense plus complexe. Bien que cela puisse augmenter la complexité, le potentiel d'amélioration de le taux pourrait valoir la peine d'être testé. L'introduction d'approches d'échantillonnage qui donnent aux agents une information plus complète sur une partie des graphes ou du graphe entier et leur intégration pourrait également s'avérer utile pour un meilleur taux de motifs trouvé. Pour Multi-SPMiner, l'investigation de techniques d'échantillonnage de graphes ou d'approches de calcul distribué pourrait améliorer sa capacité à gérer de très grands ensembles de données.
6. **Apprentissage par transfert et apprentissage à peu d'exemples** : Développer des techniques d'apprentissage par transfert pour permettre aux modèles entraînés sur un type de données spatio-temporelles d'être efficacement adaptés à de nouveaux domaines pourrait grandement améliorer l'applicabilité pratique de nos méthodes. De plus, l'exploration d'approches d'apprentissage avec peu d'exemples pourrait aider dans les scénarios où les données étiquetées sont rares.

7. **Intégration avec des connaissances spécifiques au domaine** : Dans cette thèse, nous avons cherché à nous éloigner de l'extraction de motifs liée à un domaine spécifique, et visé à développer un modèle qui peut se généraliser. Il pourrait être intéressant d'essayer l'approche inverse. L'incorporation de connaissances spécifiques au domaine, en particulier pour les applications environnementales, pourrait conduire à des découvertes de motifs plus significatifs et pertinents. Cela pourrait impliquer le développement de modèles hybrides qui combinent l'apprentissage basé sur les données avec des systèmes à la base de règles dérivées de l'expertise du domaine.
8. **Découverte de motifs en temps réel** : Étendre nos méthodes pour prendre en charge des scénarios de données en temps réel ou en flux continu serait précieux pour les applications nécessitant une surveillance et une analyse continues des motifs spatio-temporels. L'apprentissage actif pourrait être ajouté pour guider le modèle sur les motifs à chercher, permettant une adaptation dynamique aux changements dans les données et aux besoins spécifiques des utilisateurs.
9. **Création d'ensembles de données de référence** : Collaborer avec des experts du domaine pour créer des ensembles de données de référence avec une vérité terrain annotée pour les motifs spatio-temporels serait inestimable pour la communauté de recherche. Cela faciliterait une évaluation et une comparaison plus rigoureuses des différentes approches d'extraction de motifs dans les graphes spatio-temporels.



Annexes

Sommaire

A.1	Ajustement des hyperparamètres pour Deep-QMiner	154
A.1.1	Hyperparamètres testés	154
A.1.2	Résultats et observations	154
A.2	Complexité temporelle et spatiale Deep Qminer	155
A.2.1	1. Complexité temporelle	155
A.2.2	2. Complexité spatiale	157

A.1 Ajustement des hyperparamètres pour Deep-QMiner

Dans cette annexe, nous décrivons les hyperparamètres testés dans l'approche Deep-QMiner, les valeurs optimales trouvées et les plages utilisées lors de l'expérimentation. L'ajustement des hyperparamètres est essentiel pour optimiser les performances et garantir que le modèle d'apprentissage par renforcement multi-agents découvre efficacement des motifs fréquents dans des graphes spatio-temporels.

A.1.1 Hyperparamètres testés

TABLE A.1 – Résumé des hyperparamètres testés pour Deep-QMiner

Hyperparamètre	Valeur optimale	Plage testée
Epsilon initial (ϵ_{start})	0.7	[0.5, 0.9]
Taux de déca y d'Epsilon (ϵ_{decay})	0.95	[0.9, 0.99]
Epsilon minimum (ϵ_{min})	0.03	[0.01, 0.1]
Facteur d'actualisation (γ)	0.99	[0.8, 0.99]
Taux d'apprentissage (α)	0.001	[0.0001, 0.01]
Taille du buffer de réexpé rience	50,000 transitions	[10,000, 100,000]
Longueur d'épisode (T)	300 étapes	[50, 500]

A.1.2 Résultats et observations

Grâce à l'ajustement systématique de ces hyperparamètres, nous avons observé des variations significatives dans l'efficacité de l'apprentissage et l'efficacité de la découverte des motifs. Notamment, un epsilon initial plus élevé combiné à un taux de déca y plus lent a donné de meilleurs résultats en matière d'exploration pendant les premières étapes de l'entraînement. De même, un facteur d'actualisation de 0.99 a permis d'obtenir des résultats favorables dans l'identification de motifs à long terme, soulignant l'importance d'une sélection stratégique des paramètres dans l'approche Deep-QMiner.

Une analyse plus approfondie de l'impact des structures de récompense a révélé que des récompenses de correspondance plus élevées augmentaient considérablement la motivation des agents à découvrir des motifs efficacement, tandis que les récompenses de curiosité équilibrent le besoin d'exploration.

A.2 Complexité temporelle et spatiale Deep Qminer

A.2.1 1. Complexité temporelle

La complexité temporelle globale de l'algorithme peut être décomposée en plusieurs éléments, principalement influencée par les facteurs suivants :

- Le nombre d'agents N
- Le nombre d'épisodes M
- Le nombre de pas par épisode T
- La taille du graphe $|V|$ (nombre de nœuds) et $|E|$ (nombre d'arêtes)
- Le coût des mises à jour du réseau Q (dépendant de la taille du réseau Q et de la taille du lot pour les mises à jour)

Analyse étape par étape de la complexité temporelle :

1. Boucle sur les épisodes :
 - L'algorithme s'exécute pour M épisodes.
 - Complexité temporelle : $O(M)$.
2. Boucle sur les agents :
 - Pour chaque épisode, l'algorithme s'exécute pour N agents.
 - Complexité temporelle : $O(N)$.
3. Boucle sur les pas par épisode :
 - Pour chaque agent, l'algorithme s'exécute pour jusqu'à T pas dans chaque épisode.
 - Complexité temporelle : $O(T)$.
4. Sélection d'action :
 - Le processus de sélection d'action implique soit l'exploration (action aléatoire), soit l'exploitation (choix de l'action qui maximise la valeur Q).
 - Exploration (action aléatoire) : Complexité temporelle $O(1)$ car une action aléatoire est choisie.
 - Exploitation (arg max des actions) : Cela implique l'évaluation des valeurs Q pour chaque action possible, qui dépend du nombre d'actions possibles A à chaque étape.

- Complexité temporelle : $O(A)$, où A est le nombre d'actions disponibles (c'est-à-dire ajouter un nœud).
 - Dans le parcours de graphe, le nombre d'actions possibles A est proportionnel au degré du nœud courant. Pour les grands graphes, cela peut être approximé par $O(\Delta)$, où Δ est le degré maximum du graphe.
5. Mise à jour du réseau Q :
- La mise à jour du réseau Q nécessite l'échantillonnage d'un mini-lot de taille B depuis le tampon de répétition et l'exécution d'une passe avant et arrière à travers le réseau.
 - Passe avant (pour calculer les valeurs Q) : Si le réseau Q a L couches et que chaque couche a n_l neurones, la passe avant prend $O(L \cdot n_l^2)$.
 - Passe arrière (pour mettre à jour les poids du réseau) : Similaire à la passe avant, cela prend $O(L \cdot n_l^2)$.
 - Complexité temporelle de la mise à jour du réseau Q : $O(B \cdot L \cdot n_l^2)$.
6. Curiosité et autres attributions de récompenses :
- L'attribution de récompenses pour l'exploration guidée par la curiosité, les motifs partiels ou les motifs invalides implique généralement des vérifications en temps constant.
 - Complexité temporelle : $O(1)$ pour chaque étape.

Complexité temporelle totale : En prenant en compte tous les composants, la complexité temporelle totale de l'algorithme peut être exprimée comme :

$$O(M \cdot N \cdot T \cdot (A + B \cdot L \cdot n_l^2))$$

Où :

- M = Nombre d'épisodes
- N = Nombre d'agents
- T = Nombre maximum de pas par épisode
- A = Nombre d'actions disponibles (proportionnel au degré du nœud, $O(\Delta)$)
- B = Taille du lot pour les mises à jour du réseau Q
- L = Nombre de couches dans le réseau Q
- n_l = Nombre de neurones par couche

A.2.2 2. Complexité spatiale

La complexité spatiale de l'algorithme est principalement déterminée par les composants suivants :

- Taille du tampon de répétition \mathcal{D}
- Stockage du réseau Q $Q(s, a|\theta)$
- Mémoire globale pour les motifs découverts M_g

Analyse étape par étape de la complexité spatiale :

1. Tampon de répétition :

- Le tampon de répétition stocke les transitions (s_t, a_t, R_t, s_{t+1}) .
- Chaque transition se compose de l'état courant s_t , de l'action a_t , de la récompense R_t , et de l'état suivant s_{t+1} .
- Si S est la dimension de l'espace des états et A est le nombre d'actions possibles, chaque transition prend $O(S + A)$ d'espace.
- Pour un tampon de répétition de taille \mathcal{D} , la complexité spatiale totale est $O(\mathcal{D} \cdot (S + A))$.

2. Réseau Q :

- Le réseau Q se compose de poids répartis sur L couches, où chaque couche l a n_l neurones.
- L'espace requis pour stocker le réseau Q est la somme des poids de chaque couche :

$$O\left(\sum_{l=1}^L n_l^2\right)$$

- Pour un réseau simple avec des tailles similaires entre les couches, cela peut être approximé par $O(L \cdot n_l^2)$.

3. Mémoire globale pour les motifs découverts :

- La mémoire globale M_g stocke les motifs découverts jusqu'à présent.
- Si k motifs sont découverts et que chaque motif a une taille de 6, la complexité spatiale pour stocker ces motifs est $O(k \cdot 6) = O(k)$.

Complexité spatiale totale : La complexité spatiale totale est la somme de ces composants :

$$O(\mathcal{D} \cdot (S + A)) + O(L \cdot n_l^2) + O(k)$$

Où :

- \mathcal{D} = Taille du tampon de répétition
- S = Dimension de l'espace des états
- A = Nombre d'actions possibles
- L = Nombre de couches dans le réseau Q
- n_l = Nombre de neurones par couche
- k = Nombre de motifs découverts

B

Bibliographie

Sommaire

Références	160
Liste des travaux	174

Références

- [1] A. ZEGHINA, A. LEBORGNE, F. LE BER et A. VACAVANT, « Deep learning on spatiotemporal graphs : A systematic review, methodological landscape, and research opportunities, » *Neurocomputing*, t. 594, p. 127 861, 2024. adresse : <https://www.sciencedirect.com/science/article/pii/S0925231224006325>.
- [2] A. ZEGHINA, A. LEBORGNE, F. LE BER et A. VACAVANT, « Multi-SPMiner : A Deep Learning Framework for Multi-Graph Frequent Pattern Mining with Application to spatiotemporal Graphs, » *Procedia Computer Science*, t. 225, p. 1094-1103, 2023. adresse : <https://www.sciencedirect.com/science/article/pii/S1877050923012553>.
- [3] M. NASRI, M. BARATCHI, Y.-T. TSOU, S. GIEST, A. KOUTAMANIS et C. RIEFFE, « A novel metric to measure spatio-temporal proximity : a case study analyzing children’s social network in schoolyards, » *Applied Network Science*, t. 8, n° 1, août 2023. adresse : <http://dx.doi.org/10.1007/s41109-023-00571-6>.
- [4] B. KHEMANI, S. PATIL, K. KOTECHA et S. TANWAR, « A review of graph neural networks : concepts, architectures, techniques, challenges, datasets, applications, and future directions, » *Journal of Big Data*, t. 11, n° 1, jan. 2024. adresse : <http://dx.doi.org/10.1186/s40537-023-00876-4>.
- [5] L. CHE, Y. JIN, Y. SHI et al., « A drug molecular classification model based on graph structure generation, » *Journal of Biomedical Informatics*, t. 145, p. 104 447, 2023. adresse : <https://www.sciencedirect.com/science/article/pii/S1532046423001685>.
- [6] L. TANG et H. LIU, « Graph Mining Applications to Social Network Analysis, » in fév. 2010, t. 40, p. 487-513.
- [7] A. VAZQUEZ, A. FLAMMINI, A. MARITAN et A. VESPIGNANI, « Global protein function prediction from protein-protein interaction networks, » *Nature Biotechnology*, t. 21, n° 6, p. 697-700, mai 2003. adresse : <http://dx.doi.org/10.1038/nbt825>.

- [8] Y. QI, Q. LI, H. KARIMIAN et D. LIU, « A hybrid model for spatiotemporal forecasting of PM2.5 based on graph convolutional neural network and long short-term memory, » en, *Sci. Total Environ.*, t. 664, p. 1-10, mai 2019.
- [9] S. MIN, Z. GAO, J. PENG, L. WANG, K. QIN et B. FANG, « STGSN — A Spatial–Temporal Graph Neural Network framework for time-evolving social networks, » *Knowledge-Based Systems*, t. 214, p. 106 746, 2021. adresse : <https://www.sciencedirect.com/science/article/pii/S0950705121000095>.
- [10] S. GADGIL, Q. ZHAO, A. PFEFFERBAUM, E. V. SULLIVAN, E. ADELI et K. M. POHL, « Spatio-Temporal Graph Convolution for Resting-State fMRI Analysis, » in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, Lima, Peru : Springer-Verlag, 2020, p. 528-538. adresse : https://doi.org/10.1007/978-3-030-59728-3_52.
- [11] L. YU, B. DU, X. HU, L. SUN, L. HAN et W. LV, « Deep spatio-temporal graph convolutional network for traffic accident prediction, » *Neurocomputing*, t. 423, p. 135-147, 2021. adresse : <https://www.sciencedirect.com/science/article/pii/S092523122031451X>.
- [12] P. LU, W. BAI, D. RUECKERT et J. NOBLE, « Dynamic Spatio-Temporal Graph Convolutional Networks For Cardiac Motion Analysis, » in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021, p. 122-125.
- [13] Y. ZHANG, Y. LI, X. WEI et L. JIA, « Adaptive Spatio-Temporal Graph Convolutional Neural Network for Remaining Useful Life Estimation, » in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, p. 1-7.
- [14] B. YU, M. LI, J. ZHANG et Z. ZHU, « 3D Graph Convolutional Networks with Temporal Graphs : A Spatial Information Free Framework For Traffic Forecasting, » *CoRR*, t. abs/1903.00919, 2019. arXiv : 1903.00919. adresse : <http://arxiv.org/abs/1903.00919>.
- [15] M. LV, Z. HONG, L. CHEN, T. CHEN, T. ZHU et S. JI, « Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction, » *IEEE Transactions on Intelligent Transportation Systems*, t. 22, n° 6, p. 3337-3348, 2021.
- [16] G. DEL MONDO, M. RODRÍGUEZ, C. CLARAMUNT, L. BRAVO et R. THIBAUD, « Modeling consistency of spatio-temporal graphs, » *Data & Knowledge Engineering*, t. 84, p. 59-80, 2013. adresse : <https://www.sciencedirect.com/science/article/pii/S0169023X12001188>.
- [17] A. ROY et J. STELL, « A Qualitative Account of Discrete Space, » *GIScience 2002*, t. 2478, p. 276-290, sept. 2002.

- [18] A. LEBORGNE, M. KIRANDJISKA et F. L. BER, « Random generation of a locally consistent spatio-temporal graph, » in *Graph-Based Representation and Reasoning : 26th International Conference on Conceptual Structures, ICCS 2021, Virtual Event, September 20–22, 2021, Proceedings 26*, Springer, 2021, p. 155-169.
- [19] A. G. COHN, B. BENNETT, J. GOODAY et N. M. GOTTS, *Geoinformatica*, t. 1, n° 3, p. 275-316, 1997. adresse : <https://doi.org/10.1023/a:1009712514511>.
- [20] F. SCARSELLI, M. GORI, A. C. TSOI, M. HAGENBUCHNER et G. MONFARDINI, « The Graph Neural Network Model, » *IEEE Transactions on Neural Networks*, t. 20, n° 1, p. 61-80, 2009.
- [21] W. WANG, Y. HUANG, Y. WANG et L. WANG, « Generalized Autoencoder : A Neural Network Framework for Dimensionality Reduction, » in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, juin 2014, p. 490-497.
- [22] D. CHAI, L. WANG et Q. YANG, « Bike Flow Prediction with Multi-Graph Convolutional Networks, » in *Proceedings of the 26th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, sér. SIGSPATIAL '18, Seattle, Washington : Association for Computing Machinery, 2018, p. 397-400. adresse : <https://doi.org/10.1145/3274895.3274896>.
- [23] T. AZEVEDO, L. PASSAMONTI, P. LIO' et N. TOSCHI, « Towards a predictive spatio-temporal representation of brain data, » *ArXiv*, t. abs/2003.03290, 2020.
- [24] L. JUNE-GOO, J. SANGHOON, C. YOUNG-WON et al., « Deep Learning in Medical Imaging : General Overview, » *kjr*, t. 18, n° 4, p. 570-584, 2017. eprint : <http://www.e-sciencecentral.org/articles/?scid=1027354>. adresse : <http://www.e-sciencecentral.org/articles/?scid=1027354>.
- [25] J. EGGER, C. GSAXNER, A. PEPE et al., « Medical deep learning—A systematic meta-review, » *Computer Methods and Programs in Biomedicine*, t. 221, p. 106874, 2022. adresse : <https://www.sciencedirect.com/science/article/pii/S0169260722002565>.
- [26] T. AZEVEDO, A. CAMPBELL, R. ROMERO-GARCIA et al., « A deep graph neural network architecture for modelling spatio-temporal dynamics in resting-state functional MRI data, » *Medical Image Analysis*, t. 79, p. 102471, 2022. adresse : <https://www.sciencedirect.com/science/article/pii/S1361841522001189>.
- [27] H. CUI et al., « Braingb : a benchmark for brain network analysis with graph neural networks, » *IEEE Transactions on Medical Imaging*, t. 42, n° 2, p. 493-506, 2022.

- [28] S. HUANG et al., « Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation, » *Neuroimage*, t. 50, n° 3, p. 935-949, 2023.
- [29] B. NI, Q. SHEN, J. XU et H. QU, « Spatio-temporal flow maps for visualizing movement and contact patterns, » *Visual Informatics*, t. 1, n° 1, p. 57-64, 2017. adresse : <https://www.sciencedirect.com/science/article/pii/S2468502X17300098>.
- [30] M. KHODAYAR et J. WANG, « Spatio-Temporal Graph Deep Neural Network for Short-Term Wind Speed Forecasting, » *IEEE Transactions on Sustainable Energy*, t. 10, n° 2, p. 670-681, 2019.
- [31] R. D. HJELM, A. FEDOROV, S. LAVOIE-MARCHILDON et al., « Learning deep representations by mutual information estimation and maximization, » in *International Conference on Learning Representations*, 2019, p. 1-14. adresse : <https://openreview.net/forum?id=Bk1r3j0cKX>.
- [32] Z. PAWLAK, *Rough Sets*. New York, NY : Springer, jan. 2014.
- [33] A. KAPOOR, X. BEN, L. LIU et al., « Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks, » *CoRR*, t. abs/2007.03113, 2020. arXiv : 2007.03113. adresse : <https://arxiv.org/abs/2007.03113>.
- [34] Y. GU, Y. WANG et Y. LI, « A Survey on Deep Learning-Driven Remote Sensing Image Scene Understanding : Scene Classification, Scene Retrieval and Scene-Guided Object Detection, » *Applied Sciences*, t. 9, n° 10, 2019. adresse : <https://www.mdpi.com/2076-3417/9/10/2110>.
- [35] V. SHARMA, M. GUPTA, A. K. PANDEY, D. MISHRA et A. KUMAR, « A Review of Deep Learning-based Human Activity Recognition on Benchmark Video Datasets, » *Applied Artificial Intelligence*, t. 36, n° 1, p. 2 093 705, 2022.
- [36] S. YAN, Y. XIONG et D. LIN, « Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition, » *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 32, n° 1, avr. 2018. adresse : <https://ojs.aaai.org/index.php/AAAI/article/view/12328>.
- [37] B. LI, X. LI, Z. ZHANG et F. WU, « Spatio-Temporal Graph Routing for Skeleton-Based Action Recognition, » *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 33, n° 01, p. 8561-8568, juill. 2019. adresse : <https://ojs.aaai.org/index.php/AAAI/article/view/4875>.
- [38] P. GHOSH, Y. YAO, L. S. DAVIS et A. DIVAKARAN, « Stacked Spatio-Temporal Graph Convolutional Networks for Action Segmentation, » in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, p. 565-574.

- [39] P. DAS et A. ORTEGA, « Symmetric Sub-graph Spatio-Temporal Graph Convolution and its application in Complex Activity Recognition, » in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, p. 3215-3219.
- [40] B. YU, H. YIN et Z. ZHU, « Spatio-temporal Graph Convolutional Neural Network : A Deep Learning Framework for Traffic Forecasting, » *CoRR*, t. abs/1709.04875, 2017. arXiv : 1709.04875. adresse : <http://arxiv.org/abs/1709.04875>.
- [41] J. ZHANG, Y. ZHENG, J. SUN et D. QI, « Flow Prediction in Spatio-Temporal Networks Based on Multitask Deep Learning, » *IEEE Transactions on Knowledge and Data Engineering*, t. 32, n° 3, p. 468-478, 2020.
- [42] C. CHEN, K. LI, S. G. TEO et al., « Gated Residual Recurrent Graph Neural Networks for Traffic Prediction, » *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 33, n° 01, p. 485-492, juill. 2019. adresse : <https://doi.org/10.1609/aaai.v33i01.3301485>.
- [43] K. CHEN, F. CHEN, B. LAI et al., « Dynamic Spatio-Temporal Graph-Based CNNs for Traffic Flow Prediction, » *IEEE Access*, t. 8, p. 185 136-185 145, 2020.
- [44] J. KE, X. QIN, H. YANG, Z. ZHENG, Z. ZHU et J. YE, « Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network, » *Transportation Research Part C : Emerging Technologies*, t. 122, p. 102 858, jan. 2021.
- [45] X. ZHANG, C. WANG, J. CHEN et D. CHEN, « A deep neural network model with GCN and 3D convolutional network for short-term metro passenger flow forecasting, » *IET Intelligent Transport Systems*, mars 2023. adresse : <https://doi.org/10.1049/itr2.12352>.
- [46] Z. LU, W. LV, Z. XIE et al., « Graph Sequence Neural Network with an Attention Mechanism for Traffic Speed Prediction, » *ACM Transactions on Intelligent Systems and Technology*, t. 13, n° 2, p. 1-24, mars 2022. adresse : <https://doi.org/10.1145/3470889>.
- [47] Z. LV, J. XU, K. ZHENG, H. YIN, P. ZHAO et X. ZHOU, « LC-RNN : A Deep Learning Model for Traffic Speed Prediction, » in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, International Joint Conferences on Artificial Intelligence Organization, juill. 2018, p. 3470-3476. adresse : <https://doi.org/10.24963/ijcai.2018/482>.
- [48] L. HAN, B. DU, L. SUN, Y. FU, Y. LV et H. XIONG, « Dynamic and Multi-Faceted Spatio-Temporal Deep Learning for Traffic Speed Forecasting, » in *KDD'21 : Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &*

- Data Mining*, sér. KDD '21, Virtual Event, Singapore : Association for Computing Machinery, 2021, p. 547-555. adresse : <https://doi.org/10.1145/3447548.3467275>.
- [49] Z. DIAO, X. WANG, D. ZHANG, Y. LIU, K. XIE et S. HE, « Dynamic Spatial-Temporal Graph Convolutional Neural Networks for Traffic Forecasting, » in *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 33, juill. 2019, p. 890-897. adresse : <https://ojs.aaai.org/index.php/AAAI/article/view/3877>.
- [50] X. GENG, Y. LI, L. WANG et al., « Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting, » *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 33, n° 01, p. 3656-3663, juill. 2019. adresse : <https://ojs.aaai.org/index.php/AAAI/article/view/4247>.
- [51] J. GUANGYIN, X. ZHEXU, S. HENGYU, F. YANGHE et H. JINCAI, *Deep Multi-View Spatiotemporal Virtual Graph Neural Network for Significant Citywide Ride-hailing Demand Prediction*, 2020. adresse : <https://arxiv.org/abs/2007.15189>.
- [52] G. HONG, Z. WANG, T. HAN et H. JI, « Spatiotemporal Multi-Graph Convolutional Network for Taxi Demand Prediction, » in *2021 11th International Conference on Information Science and Technology (ICIST)*, 2021, p. 242-250.
- [53] G. CHEN, L. HU, Q. ZHANG, Z. REN, X. GAO et J. CHENG, « ST-LSTM : Spatio-Temporal Graph Based Long Short-Term Memory Network For Vehicle Trajectory Prediction, » in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, p. 608-612.
- [54] Y. ZHANG et T. CHENG, « Graph deep learning model for network-based predictive hotspot mapping of sparse spatio-temporal events, » *Computers Environment and Urban Systems*, t. 79, nov. 2019.
- [55] G. XIAO, R. WANG, C. ZHANG et A. NI, « Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks, » *Multimedia Tools and Applications*, t. 80, juin 2021.
- [56] Y. SHUGUAN, M. WEI, P. XIDONG et Q. SEAN, « A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources, » *Transportation Research Part C : Emerging Technologies*, t. 107, p. 248-265, oct. 2019. adresse : <https://doi.org/10.1016/j.trc.2019.08.010>.
- [57] G. JIN, H. SHA, Y. FENG, Q. CHENG et J. HUANG, « GSEN : An ensemble deep learning benchmark model for urban hotspots spatiotemporal prediction, » *Neu-*

- rocomputing*, t. 455, p. 353-367, 2021. adresse : <https://www.sciencedirect.com/science/article/pii/S0925231221007190>.
- [58] X. ZHANG, R. CAO, Z. ZHANG et Y. XIA, « Crowd Flow Forecasting with Multi-Graph Neural Networks, » in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, p. 1-7.
- [59] A. MOHAMED, K. QIAN, M. ELHOSEINY et C. CLAUDEL, « Social-STGCNN : A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction, » in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, p. 14 412-14 420.
- [60] P. GOMES, S. ROSSI et L. TONI, « Spatio-Temporal Graph-RNN for Point Cloud Prediction, » in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, p. 3428-3432.
- [61] X. QIAN, Y. ZHUANG, Y. LI, S. XIAO, S. PU et J. XIAO, « Video Relation Detection with Spatio-Temporal Graph, » in *Proceedings of the 27th ACM International Conference on Multimedia*, sér. MM '19, Nice, France : Association for Computing Machinery, 2019, p. 84-93. adresse : <https://doi.org/10.1145/3343031.3351058>.
- [62] B. YU, H. YIN et Z. ZHU, *ST-UNet : A Spatio-Temporal U-Net for Graph-structured Time Series Modeling*, 2019. adresse : <https://arxiv.org/abs/1903.05631>.
- [63] M. WANG, Y. LI, Y. ZHANG et L. JIA, « Spatio-temporal graph convolutional neural network for remaining useful life estimation of aircraft engines, » *Aerospace Systems*, t. 4, n° 1, p. 29-36, nov. 2020. adresse : <https://doi.org/10.1007/2Fs42401-020-00070-x>.
- [64] C. YING, A. KLEIN, E. CHRISTIANSEN, E. REAL, K. MURPHY et F. HUTTER, « NAS-Bench-101 : Towards Reproducible Neural Architecture Search, » in *Proceedings of the 36th International Conference on Machine Learning*, K. CHAUDHURI et R. SALAKHUTDINOV, éd., sér. Proceedings of Machine Learning Research, t. 97, PMLR, sept. 2019, p. 7105-7114. adresse : <https://proceedings.mlr.press/v97/ying19a.html>.
- [65] K. ZHOU, Q. SONG, X. HUANG et X. HU, « Auto-GNN : Neural Architecture Search of Graph Neural Networks, » *CoRR*, t. abs/1909.03184, 2019. arXiv : 1909.03184. adresse : <http://arxiv.org/abs/1909.03184>.
- [66] G. LI, G. QIAN, I. C. DELGADILLO, M. MÜLLER, A. THABET et B. GHANEM, « SGAS : Sequential Greedy Architecture Search, » in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

- [67] T. LI, J. ZHANG, K. BAO, Y. LIANG, Y. LI et Y. ZHENG, « AutoST : Efficient Neural Architecture Search for Spatio-Temporal Prediction, » in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, août 2020. adresse : <https://doi.org/10.1145/3394486.3403122>.
- [68] Z. PAN, S. KE, X. YANG et al., « AutoSTG : Neural Architecture Search for Predictions of Spatio-Temporal Graph*, » in *Proceedings of the Web Conference 2021*, ACM, avr. 2021. adresse : <https://doi.org/10.1145/3442381.3449816>.
- [69] S. KE, Z. PAN, T. HE, Y. LIANG, J. ZHANG et Y. ZHENG, « AutoSTG+ : An automatic framework to discover the optimal network for spatio-temporal graph prediction, » *Artificial Intelligence*, t. 318, p. 103 899, mai 2023. adresse : <https://doi.org/10.1016/j.artint.2023.103899>.
- [70] D. CAO, Y. WANG, J. DUAN et al., « Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting, » in *Advances in Neural Information Processing Systems*, H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN et H. LIN, éd., t. 33, Curran Associates, Inc., 2020, p. 17 766-17 778. adresse : https://proceedings.neurips.cc/paper_files/paper/2020/file/cdf6581cb7aca4b7e19ef136c6e601a5-Paper.pdf.
- [71] J. XU et X. CAO, « Multigraph Neural Networks for Crowd Flow Forecasting, » *Journal of Urban Computing*, t. 15, n° 3, p. 245-259, 2022.
- [72] M. ABDULLAH et al., « Spatiotemporal Graph Convolutional Networks for Predicting Pedestrian Trajectories, » *Journal of Computer Vision and Pattern Recognition*, 2023.
- [73] S. THRUN et A. SCHWARTZ, « Finding Structure in Reinforcement Learning, » in *Advances in Neural Information Processing Systems*, G. TESAURO, D. TOURETZKY et T. LEEN, éd., t. 7, MIT Press, 1994. adresse : https://proceedings.neurips.cc/paper_files/paper/1994/file/7ce3284b743aefde80ffd9aec500e085-Paper.pdf.
- [74] Y. SUN, X. WU, W. ZHANG, P. ZHAO et S. Y. PHILIP, « Sugar : Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism, » in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, p. 3235-3245.
- [75] G. LAI, Y. ZHOU, T. YANG, X. LIU et X. H. LI, « Policy-GNN : Aggregation optimization for graph neural networks, » in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, p. 585-593.

- [76] S. HONG, S. PAN, L. YAO, C. LI et H. GUO, « Graph Transformer Networks, » *arXiv preprint arXiv :2012.09699*, 2020.
- [77] Z. YAN et Z. e. a. XU, « Automatic virtual network embedding : A deep reinforcement learning approach with graph convolutional networks, » *arXiv preprint arXiv :2004.05049*, 2020.
- [78] L. e. a. LI, « GDPNet : refining deep graph neural networks for heterogeneous graph representation learning, » *arXiv preprint arXiv :2003.02122*, 2020.
- [79] D. e. a. CHEN, « NetRL : Efficiently training deep neural networks using softmax approximations, » in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, p. 3149-3159.
- [80] K. WANG, T. GAO et Z. ZHANG, « CORL : Causal Discovery with Reinforcement Learning, » in *Advances in Neural Information Processing Systems*, 2020.
- [81] Q. e. a. CHEN, « Graph-to-Sequence Learning using Policy Gradient for Task-Oriented Dialogue Systems, » in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, p. 1051-1061.
- [82] S. J. e. a. RENNIE, « Self-critical sequence training for image captioning, » in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, p. 1179-1195.
- [83] S. e. a. ZHU, « Causal Discovery with Reinforcement Learning, » in *Proceedings of the 36th International Conference on Machine Learning*, t. 97, 2019, p. 7394-7402.
- [84] Y. e. a. JIANG, « Graph Neural Networks for Traffic Forecasting, » *Proceedings of the 2019 SIAM International Conference on Data Mining*, p. 498-506, 2019.
- [85] S. e. a. QIU, « Multi-Agent Reinforcement Learning for Dynamic Electronic Toll Collection Systems, » *arXiv preprint arXiv :1909.09806*, 2019.
- [86] J. e. a. LIN, « Efficient large-scale fleet management via multi-agent deep reinforcement learning, » *arXiv preprint arXiv :1802.06444*, 2018.
- [87] H. e. a. YU, « Policy Gradient Algorithms for Multi-agent Reinforcement Learning with Partial Observability, » *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, p. 2406-2408, 2019.
- [88] H. e. a. WEI, « IntelliLight : A reinforcement learning approach for intelligent traffic light control, » *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 2496-2505, 2019.
- [89] Z. e. a. YANG, « Deep generative models for relational data : A survey, » *arXiv preprint arXiv :2007.07219*, 2020.

- [90] E. e. a. MEIROM, « Controlling a dynamic process on a graph with limited interventions : Vertex cover, spreading processes and epidemics, » *Journal of Complex Networks*, t. 7, n° 5, p. 738-759, 2019.
- [91] W. e. a. MA, « Securing Personal Information Networks with Reinforcement Learning, » *arXiv preprint arXiv :1811.00891*, 2018.
- [92] L. e. a. PINTO, « Using degree centrality to evaluate the effectiveness of epidemic interventions, » *Journal of Statistical Mechanics : Theory and Experiment*, t. 2012, n° 3, P03015, 2012.
- [93] A. e. a. BRAUNSTEIN, « Network dismantling, » *Proceedings of the National Academy of Sciences*, t. 113, n° 44, p. 12 368-12 373, 2016.
- [94] T. e. a. GUO, « Deep Learning-based Dynamic Combinatorial Optimization Algorithms over Graphs, » 2020.
- [95] F. e. a. LACERDA, « DeepOpt : Learning to Optimize Multi-resource Placement in Distributed Systems, » in *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019, p. 1196-1206.
- [96] D. e. a. SILVER, « Dynamic Routing of Variable Paths Using Reinforcement Learning, » *arXiv preprint arXiv :2011.05017*, 2020.
- [97] J. e. a. SCHULMAN, « Proximal policy optimization algorithms, » in *Proceedings of the 34th International Conference on Machine Learning*, 2017, p. 511-526.
- [98] S. e. a. WANG, « Medical Pre-diagnosis Based on Graph Convolutional Neural Network Using Dynamic Electronic Health Records, » *IEEE Access*, t. 6, p. 75 458-75 469, 2018.
- [99] H. e. a. YUAN, « Brain network analysis : a survey, » *IEEE Access*, t. 8, p. 38 660-38 683, 2020.
- [100] Q. e. a. CHEN, « Neural Collaborative Filtering for Recommender Systems with Knowledge Graphs, » *arXiv preprint arXiv :2003.03734*, 2020.
- [101] Q. WANG, C. XU, W. ZHANG et J. LI, « GraphTTE : Travel Time Estimation Based on Attention-Spatiotemporal Graphs, » *IEEE Signal Processing Letters*, t. 28, p. 239-243, 2021.
- [102] R. DAI, S. XU, Q. GU, C. JI et K. LIU, « Hybrid Spatio-Temporal Graph Convolutional Network : Improving Traffic Prediction with Navigation Data, » in *KDD '20 : Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, août 2020, p. 3074-3082.
- [103] D. ZHANG, Y. PENG, Y. ZHANG, D. WU, H. WANG et H. ZHANG, « Train Time Delay Prediction for High-Speed Train Dispatching Based on Spatio-Temporal

- Graph Convolutional Network, » *IEEE Transactions on Intelligent Transportation Systems*, t. 23, n° 3, p. 2434-2444, 2022.
- [104] K. LEE et W. RHEE, « DDP-GCN : Multi-graph convolutional network for spatiotemporal traffic forecasting, » *Transportation Research Part C : Emerging Technologies*, t. 134, p. 103 466, jan. 2022.
- [105] Y. BAO, J. HUANG, Q. SHEN et al., « Spatial–Temporal Complex Graph Convolution Network for Traffic Flow Prediction, » *Engineering Applications of Artificial Intelligence*, t. 121, p. 106 044, mai 2023. adresse : <https://doi.org/10.1016/j.engappai.2023.106044>.
- [106] C. SHUAI, X. ZHANG, Y. WANG, M. HE, F. YANG et G. XU, « Online Car-Hailing Origin-Destination Forecast Based on a Temporal Graph Convolutional Network, » *IEEE Intelligent Transportation Systems Magazine*, p. 2-18, 2023.
- [107] H. LI, D. JIN, X. LI et al., « DMGF-Net : An Efficient Dynamic Multi-Graph Fusion Network for Traffic Prediction, » *ACM Transactions on Knowledge Discovery from Data*, t. 17, n° 7, p. 1-19, avr. 2023. adresse : <https://doi.org/10.1145/3586164>.
- [108] Z. WEI, H. ZHAO, Z. LI et al., « STGSA : A Novel Spatial-Temporal Graph Synchronous Aggregation Model for Traffic Prediction, » *IEEE/CAA Journal of Automatica Sinica*, t. 10, n° 1, p. 226-238, jan. 2023. adresse : <https://doi.org/10.1109/jas.2023.123033>.
- [109] P. ALMASAN, J. SUÁREZ-VARELA, K. RUSEK, P. BARLET-ROS et A. CABELLOS-APARICIO, « Deep reinforcement learning meets graph neural networks : Exploring a routing optimization use case, » *Computer Communications*, t. 196, p. 184-194, 2022. adresse : <https://www.sciencedirect.com/science/article/pii/S0140366422003784>.
- [110] M. KURAMOCHI et G. KARYPIS, « Finding frequent patterns in a large sparse graph, » *Data mining and knowledge discovery*, t. 11, n° 3, p. 243-271, 2005.
- [111] M. ELSEIDY, E. ABDELHAMID, S. SKIADOPOULOS et P. KALNIS, « Grami : Frequent subgraph and pattern mining in a single large graph, » *Proceedings of the VLDB Endowment*, t. 7, n° 7, p. 517-528, 2014.
- [112] V. INGALALLI, D. IENCO et P. PONCELET, « Mining frequent subgraphs in multi-graphs, » *Information Sciences*, t. 451, p. 50-66, 2018.
- [113] T. RAMRAJ et R. PRABHAKAR, « Frequent Subgraph Mining Algorithms – A Survey, » *Procedia Computer Science*, t. 47, p. 197-204, 2015, Graph Algorithms, High Performance Implementations and Its Applications (ICGHIA

- 2014). adresse : <https://www.sciencedirect.com/science/article/pii/S1877050915004664>.
- [114] X. YAN et J. HAN, « gSpan : Graph-based substructure pattern mining, » in *Proceedings of the 2002 IEEE International Conference on Data Mining, 2002*, p. 721-724.
- [115] X. YAN et J. HAN, « CloseGraph : Mining closed frequent graph patterns, » in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003*, p. 286-295.
- [116] C. BORGELT et M. BERTHOLD, « Mining molecular fragments : Finding relevant substructures of molecules, » in *Proceedings of the 2nd IEEE International Conference on Data Mining, 2002*, p. 51-58.
- [117] M. HEDEGAARD et L. JENSEN, « SPIN : a tool for efficient predicate correlation mining in biological networks, » *Nucleic Acids Research*, t. 42, n° 8, e55, 2014.
- [118] P. NIJKAMP et P. ROUSSEEUW, « Gaston : A graph-based substructure pattern mining algorithm for chemical compounds, » in *Proceedings of the 2005 SIAM International Conference on Data Mining, 2005*, p. 566-570.
- [119] L. HOLDER, D. COOK et S. DJOKO, « Substructure discovery in the SUBDUE system, » in *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, 1994*, p. 169-180.
- [120] B. BRINGMANN, S. NIJSSEN et N. TATTI, « Mining maximal frequent subgraphs in dynamic graph databases, » in *Proceedings of the 2010 SIAM International Conference on Data Mining, 2010*, p. 316-327.
- [121] G. DAMIAND, C. SOLNON, C. de LA HIGUERA, J.-C. JANODET et É. SAMUEL, « Polynomial Algorithms for Subisomorphism of nD Open Combinatorial Maps, » *Computer Vision and Image Understanding, 2011*. adresse : <https://hal.science/hal-00597483v1>.
- [122] C. SOLNON, G. DAMIAND, J.-F. COUCHOT et P.-E. PORTIER, « PLAGRAM : un algorithme de fouille de graphes plans efficace, » in *EGC, 2015*. adresse : https://www.researchgate.net/publication/278815641_PLAGRAM_un_algorithme_de_fouille_de_graphes_plans_efficace.
- [123] R. DEVILLE, E. FROMONT, B. JEUDY et C. SOLNON, « Mining Frequent Patterns in 2D+t Grid Graphs for Cellular Automata Analysis, » in *Proceedings of the 2017 SIAM International Conference on Data Mining, 2017*. adresse : <https://hal.science/hal-01494623v1>.

- [124] W. HAMILTON, R. YING et J. LESKOVEC, « Inductive Representation Learning on Large Graphs, » in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 1025-1035.
- [125] T. KIPF et M. WELLING, « Semi-Supervised Classification with Graph Convolutional Networks, » in *Proceedings of the 5th International Conference on Learning Representations*, 2016.
- [126] R. YING, A. WANG, J. YOU et J. LESKOVEC, « Frequent Subgraph Mining by Walking in Order Embedding Space., » *The International Conference on Machine Learning (ICML)*, 2020.
- [127] R. YING, T. FU, A. WANG, J. YOU, Y. WANG et J. LESKOVEC, *Representation Learning for Frequent Subgraph Mining*, <https://arxiv.org/abs/2402.14367>, 2024. arXiv : 2402.14367 [cs.LG].
- [128] C. PARK et C. LEE, « ST-GRAT : A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed, » in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020.
- [129] S.-Y. LOUIS, Y. ZHAO, A. NASIRI et al., « Graph convolutional neural networks with global attention for improved materials property prediction, » *Physical Chemistry Chemical Physics*, t. 22, n° 32, p. 18 141-18 148, 2020.
- [130] M. SCHLICHTKRULL, T. N. KIPF, P. BLOEM, R. van den BERG, I. TITOV et M. WELLING, *Modeling Relational Data with Graph Convolutional Networks*, 2017. arXiv : 1703.06103 [stat.ML].
- [131] L. WANG, H. CAO, H. XU et H. LIU, « A gated graph convolutional network with multi-sensor signals for remaining useful life prediction, » *Knowledge-Based Systems*, t. 252, p. 109 340, 2022.
- [132] X. WANG, H. JI, C. SHI et al., *Heterogeneous Graph Attention Network*, 2021. arXiv : 1903.07293 [cs.SI].
- [133] Z. HUANG, Y. REN, X. PU, S. HUANG, Z. XU et L. HE, « Self-Supervised Graph Attention Networks for Deep Weighted Multi-View Clustering, » *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 37, n° 7, p. 7936-7943, juin 2023. adresse : <http://dx.doi.org/10.1609/aaai.v37i7.25960>.
- [134] A. SARIGÜN et A. S. RIFAI OGLU, « Multi-Mask Aggregators for Graph Neural Networks, » in *The First Learning on Graphs Conference*, 2022. adresse : <https://openreview.net/forum?id=hZ3b8CskgC>.
- [135] S. S. MOUSAVI, M. SCHUKAT et E. HOWLEY, « Deep Reinforcement Learning : An Overview, » in *Proceedings of SAI Intelligent Systems Conference (IntelliSys)*

2016. Springer International Publishing, août 2017, p. 426-440. adresse : http://dx.doi.org/10.1007/978-3-319-56991-8_32.
- [136] A. DOS SANTOS MIGNON et R. L. DE AZEVEDO DA ROCHA, « An Adaptive Implementation of e-Greedy in Reinforcement Learning, » *Procedia Computer Science*, t. 109, p. 1146-1151, 2017, 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal. adresse : <https://www.sciencedirect.com/science/article/pii/S1877050917311134>.
- [137] A. LEBORGNE, A. MEYER, H. GIRAUD, F. LE BER et S. MARC-ZWECKER, « Un graphe spatio-temporel pour modéliser l'évolution de parcelles agricoles, » in *SAGEO 2019*, in French, IRSTEA - INRA, 2019.
- [138] G. DEL MONDO, P. PENG, J. GENSEL, C. CLARAMUNT et F. LU, « Leveraging Spatio-Temporal Graphs and Knowledge Graphs : Perspectives in the Field of Maritime Transportation, » *ISPRS International Journal of Geo-Information*, t. 10, n° 8, 2021. adresse : <https://www.mdpi.com/2220-9964/10/8/541>.

Liste des Travaux

- Zeghina, A. Leborgne, F. Le Ber, A. Vacavant : Deep learning on spatio-temporal graphs : a systematic review, methodological landscape, and research opportunities. *Neurocomputing*, 2023, NEUROCOMPUTING.
- Zeghina, A. Leborgne, F. Le Ber, A. Vacavant : Attention-based MultiSPMiner model for the robust extraction of frequent motifs in spatiotemporal graphs, 2024, *PATTERN RECOGNITION LETTERS* (Révision).
- A. Zeghina, A. Leborgne, F. Le Ber, and A. Vacavant : Multi-SPMiner : A Deep Learning Framework for Multi-Graph Frequent Pattern Mining with Application to spatiotemporal Graphs. *Procedia Computer Science*, 2023 (Conference).
- A. Zeghina, A. Leborgne, F. Le Ber, and A. Vacavant : Multi-SPMiner : un framework d'apprentissage profond pour l'extraction de motifs fréquents dans des graphes spatio-temporels, *EGC WORKSHOP GAST*, 2024 (conférence).
- A. Zeghina, A. Leborgne, F. Le Ber, and A. Vacavant : Multi-SPMiner : un framework d'apprentissage profond pour l'extraction de motifs fréquents dans des multi-graphes avec application aux graphes spatio-temporels, *EGC 2024* (Conférence).
- A. Zeghina, A. Leborgne, F. Le Ber, and A. Vacavant : Approche de Deep learning pour la recherche de motifs fréquents dans les graphes spatio-temporels, *Symposium du GDR MaDICS 2023* (Poster).
- Zeghina, A. Leborgne, F. Le Ber, A. Vacavant : Deep-QMiner : Multi-Agent Reinforcement Learning for Efficient Frequent Motif Discovery in Spatiotemporal Graphs, *Neurocomputing* (manuscrit en préparation).

Avertissement au lecteur / Warning to the reader

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition des membres de la communauté universitaire. Il est soumis à la propriété intellectuelle de l'auteur. Cela implique une obligation de citation et de référencement lors de l'utilisation de ce document. D'autre part, toute contrefaçon, plagiat, reproduction ou représentation illicite encourt une poursuite pénale.

This document is the result of a long process approved by the jury and made available to members of the university community. It is subject to the intellectual property rights of its author. This implies an obligation to quote and reference when using this document. Furthermore, any infringement, plagiarism, unlawful reproduction or representation will be prosecuted.

Code de la Propriété Intellectuelle

Article L122-4 :

Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la transformation, l'arrangement ou la reproduction par un art ou un procédé quelconque.

Any representation or reproduction in whole or in part without the consent of the author or his successors in title or assigns is unlawful. The same applies to translation, adaptation or transformation, arrangement or reproduction by any art or process whatsoever.

Articles L335-1 à L335-9. : Dispositions pénales / Penal provisions.

Assaad ZEGHINA

Recherche de motifs spatio-temporels fréquents dans un grand graphe via des méthodes d'apprentissage profond

Résumé

Les graphes sont essentiels pour représenter des relations complexes dans divers domaines. Cette thèse se concentre sur l'extraction de motifs fréquents dans les graphes spatio-temporels, un défi majeur en raison de la complexité algorithmique. Deux approches innovantes sont proposées : Multi-SPMiner, une adaptation de SPMiner pour multigraphes, utilisant un modèle de réseaux convolutionnels pour graphes, et Deep-QMiner, basé sur l'apprentissage par renforcement profond multi-agents. Ces méthodes, testées sur des données synthétiques et réelles, montrent leur efficacité pour extraire des motifs variés, avec une généralisation robuste entre ensembles de données, validant leur pertinence pour l'analyse de graphes complexes.

Mots clés : Extraction de motifs fréquents, graphes spatio-temporels, apprentissage profond, apprentissage par renforcement.

Abstract

Graphs are essential tools for representing complex relationships in various fields. This thesis focuses on the extraction of frequent motifs in spatiotemporal graphs, a major challenge due to the algorithmic complexity. Two innovative approaches are proposed: Multi-SPMiner, an adaptation of SPMiner for multigraphs, using a convolutional graph network model, and Deep-QMiner, based on multi-agent deep reinforcement learning. These methods, tested on synthetic and real-world data, demonstrate their effectiveness in extracting diverse motifs, with robust generalization across datasets, validating their relevance for analyzing complex graphs.

Keywords : Frequent Motif Mining, Spatiotemporal Graphs, Deep Learning, Reinforcement Learning.