

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Центр \_\_\_\_\_ Післядипломної освіти \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

\_\_\_\_\_ Дослідження спектральної конволюції графових нейронних мереж \_\_\_\_\_  
(тема)

Виконав:  
студент 2 курсу, групи \_\_\_\_\_ СШІЗДМ-20-2 \_\_\_\_\_  
Осьмачко В. Є. \_\_\_\_\_  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва спеціальності)

Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Системи штучного інтелекту \_\_\_\_\_  
(повна назва спеціалізації)

Керівник \_\_\_\_\_ к.т.н., доцент, Шевченко О. Ю. \_\_\_\_\_  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_ В.О. Філатов \_\_\_\_\_  
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Центр \_\_\_\_\_ Післядипломної освіти \_\_\_\_\_  
(повна назва)  
Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_  
(повна назва)  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)  
Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)  
Освітня програма \_\_\_\_\_ Системи штучного інтелекту (СШІ) \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Осьмачко Всеволоду Євгеновичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Дослідження спектральної конволюції \_\_\_\_\_  
графових нейронних мереж \_\_\_\_\_

затверджена наказом університету від 21 березня 2022 р. № 30 Стз

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 20 \_\_\_\_ р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет та відомих наукових  
проектів, електронні документації, тестові набори даних \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області та постановка  
задачі дослідження, графові нейронні мережі, спектральна теорія графів, труднощі та  
особливості відомих методів конволюції графових нейронних мереж, метрики якості  
класифікації, середовище реалізації, опис алгоритму, аналіз отриманих результатів \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – два графи зі схожими тенденціями зав'язків, Рисунок 2 – приклади графів, Рисунок 3 – Аналоговий сигнал, Рисунок 4 – семпльовані графи, Рисунок 5 – застасовані фільтри, Рисунок 6 – інтерполяція до аналогового сигналу, Рисунок 7 – графічне відображення власних векторів лапласіана графа, Рисунок 8 – інші приклади власних векторів, Рисунок 9 – Акуратність логістичної регресії, Рисунок 10 – 15 відсотків збільшених даних, Рисунок 11 – 85 відсотків збільшених даних, Рисунок 12 – Співвідношення класів 13 – 15 відсотків збільшених даних, Рисунок 14 – Співвідношення класів 4:1 – 85 відсотків збільшених даних, Рисунок 15 – Співвідношення класів 1:4 – 50 відсотків збільшених даних, Рисунок 16 – Граф вхідних даних (співвідношення інтенсивності зав'язків – 6:1), Рисунок 17- Акуратність GCN, Рисунок 18 – Граф вхідних даних (однакове співвідношення інтенсивності зав'язків), Рисунок 19 – GCN при однаковому співвідношенні інтенсивності зав'язків, Рисунок 20 – Граф вхідних даних (співвідношення інтенсивності зав'язків – 1:6 у бік між-класових зав'язків), Рисунок 21 - Граф вхідних даних - співвідношення інтенсивності зав'язків – 4:1 (розмір вузлів однаковий)

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. каф. ШІ Шевченко О. Ю.		

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Отримання завдання на кваліфікаційне проектування	22.03.2022	виконано
2.	Аналіз завдання та пошук літератури за темою	23.03-18.04	виконано
3.	Опрацювання літератури та аналіз об'єкту	26.03-04.04	виконано
4.	Вибір програмних засобів для розробки системи	05.04-07.04	виконано
5.	Розробка програмного засобу	07.04-17.04	виконано
6.	Аналіз отриманих результатів	18.04-19.04	виконано
7.	Оформлювання пояснювальної записки	20.04-29.04	виконано
8.	Оформлення презентаційних матеріалів	02.05.2022	виконано
9.	Представлення на рецензування	03.05.2022	виконано
11.	Представлення кваліфікаційної роботи		

Дата видачі завдання 22 березня 2022 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Шевченко О. Ю.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 78 с., 30 рис., 2 табл., 31 формул, 2 дод., 92 джерела.

ГРАФ, ГРАФОВА НЕЙРОННА МЕРЕЖА, СПЕКТРАЛЬНА КОНВОЛЮЦІЯ, СПЕКТРАЛЬНА ТЕОРІЯ ГРАФІВ.

Об'єктом дослідження є процес класифікації графових даних, що пов'язані між собою у граф, за допомогою графових нейронних мереж.

Предметом дослідження є властивості згорткової архітектури графових нейронних мереж, зокрема спектральної згорткової архітектури, її вплив на точність класифікації графових нейронних мереж, здатність згорткових фільтрів бити перенесеними на інші дані.

Метою даного дослідження є розробка графової нейронної мережі з, що включають шари спектральної конволюції (згортки) для вирішення задачі класифікації графових даних.

Методи дослідження. Теорія штучних нейронних мереж, яка дозволила синтезувати нові архітектури нейронних мереж, спектральна теорія графів, що дала можливість використати властивості власних векторів і власних значень матриць, пов'язаних з графом, теорія оптимізації, яка забезпечила розробку методів налаштування вагових коефіцієнтів з підвищеною швидкістю і стійкістю до зашумлених даних.

На сьогоднішній день актуальною є задача розробки нових методів навчання систем обчислювального інтелекту, що здатні використовувати не лише вектор вхідних даних, але і мережу зав'язків між цими даними яка об'єднується у граф.

## РЕФЕРАТ

Пояснительная записка: 78 с., 30 рис., 2 табл., 31 формул, 2 прил., 92 источника.

ГРАФ, ГРАФОВАЯ НЕЙРОННАЯ СЕТЬ, СПЕКТРАЛЬНАЯ КОНВОЛЮЦИЯ, СПЕКТРАЛЬНАЯ ТЕОРИЯ ГРАФОВ.

Объектом исследования является процесс классификации графовых данных, связанных между собой, с помощью графовых нейронных сетей.

Предметом исследования являются свойства сверточной архитектуры графовых нейронных сетей, в частности, спектральной сверточной архитектуры, ее влияние на точность классификации графовых нейронных сетей, способность сверточных фильтров быть перенесенными на другие данные.

Целью данного исследования является разработка графовой нейронной сети, включающей слой спектральной конволюции (свертки) для решения задачи классификации графовых данных.

Методы исследования. Теория искусственных нейронных сетей, позволившая синтезировать новые архитектуры нейронных сетей, спектральная теория графов, позволившая использовать свойства собственных векторов и собственных значений матриц, связанных с графом, теория оптимизации, обеспечившая разработку методов настройки весовых коэффициентов с повышенным быстродействием и устойчивостью к зашумленным данным.

На сегодняшний день актуальна задача разработки новых методов обучения систем вычислительного интеллекта, которые способны использовать не только вектор входных данных, но и сеть связей между этими данными, которая объединяется в граф.

## ABSTRACT

Explanatory note: 78 p., 30 fig., 2 tabl., 31 formulas, 2 ann., 92 sources.

GRAPH, GRAPHIC NEURAL NETWORK, SPECTRAL CONVOLUTION, SPECTRAL GRAPH THEORY.

The object of research is the process of classifying graph data, using graph neural networks.

The subject of research is the properties of the convolutional architecture of graph neural networks, in particular, the spectral convolutional architecture, its impact on the classification accuracy of graph neural networks, the ability of convolutional filters to be transferred to other data.

The purpose of this research is to develop a graph neural network that includes layers of spectral convolution to solve the problem of classifying graph data.

Research methods. The theory of artificial neural networks, which made it possible to synthesize new architectures of neural networks, spectral graph theory, which made it possible to use the properties of eigenvectors and eigenvalues of matrices associated with a graph, optimization theory, which provided the development of weight tuning methods with increased speed and resistance to noisy data.

Today, the task of developing new methods for training computational intelligence systems that are able to use not only the vector of input data, but also the network of links between these data, which is combined into a graph, is relevant.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів..	9
Вступ.....	10
1 Аналіз предметної галузі та постановка завдання дослідження.....	11
1.1 Застосування графових згорткових мереж .....	13
1.1.1 Застосування в комп'ютерному зору.....	14
1.1.2 Зображення.....	14
1.1.3 Відео.....	15
1.1.4 Хмари точок.....	15
1.1.5 Застосування в обробці природної мови.....	16
1.1.6 Застосування в науці.....	17
1.1.6.1 Фізика.....	17
1.1.6.2 Хімія, біологія та матеріалознавство.....	17
1.1.7 Аналіз соціальних мереж.....	18
1.2 Класичні просторово згорткові графові мережі на основі CNN....	19
1.3 Просторові згорткові мережі графа на основі поширення .....	20
2 Спектральна конволюція графових нейронних мереж .....	25
2.1 Спектральна ConvGNN.....	25
2.2 Просторова ConvGNN.....	27
2.3 Просторові згортки зі спектральним коренем.....	30
2.4 ChebNet.....	31
2.5 CayleyNet.....	32
2.6 GCN.....	33
2.7 DSGCN .....	34
2.8 Переносимість фільтрів спектральних графів.....	35
3 Імітаційне моделювання спектральної конволюції графових нейронних мереж .....	46
Висновки.....	64

Перелік джерел посилання.....	65
Додаток А Вихідний код програми.....	73
Додаток Б Відомість кваліфікаційної роботи.....	78

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

CNN – Convolution Neural Network – згорткова нейронна мережа;

ConvGNN – Convolution Graph Neural Network – згорткова графова нейронна мережа;

ConvNets – Convolution Neural Network – згорткова нейронна мережа;

GCN – Graph Convolution Network – згорткова графова нейронна мережа;

GNN – Graph Neural Network – графова нейронна мережа.

## ВСТУП

Протягом останнього десятиліття глибоке навчання, а точніше згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN), мали сильний вплив на різні програми машинного навчання, такі як розпізнавання зображень та аналіз мовлення. Ці успіхи здебільшого були досягнуті на послідовностях або зображеннях, тобто на даних, визначених у сіткових структурах, які виграють від операцій лінійної алгебри в евклідових просторах. Однак є багато областей, де дані (наприклад, соціальні мережі, молекули, графік знань) не можуть бути тривіально закодовані в евклідовий простір, але можуть бути природним чином представлені у вигляді графів. Мета полягає в тому, щоб повернутися до нейронних мереж для роботи з даними графа, щоб отримати користь від можливості навчання за ознаками. У цьому контексті в літературі було запропоновано багато графових нейронних мереж. GNN – це нейронні мережі, які покладаються на обчислення прихованих ознак вузлів, використовуючи інформацію, яку несе весь граф. На відміну від звичайної нейронної мережі, де архітектура мережі пов'язана з відомою та інваріантною топологією даних (наприклад, двовимірна сітка для зображень), особливості вузлів GNN поширюються відповідно до топології графа.

Робота зосереджена на спектральних фільтрах на графах, а саме фільтрах, визначених як поелементне множення частотного спектру графа. Одним із прикладів є графові згорткові нейронні мережі (ConvNets), де набір даних складається із сигналів, визначених на багатьох різних графах, а вивчені фільтри мають узагальнювати сигнали на нових графах, яких немає у навчальному наборі.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

Графи, природно, з'являються в багатьох областях застосування, починаючи від соціального аналізу, біоінформатики і закінчуючи комп'ютерним баченням. Унікальна здатність графів дозволяє фіксувати структурні зв'язки між даними і, таким чином, дає змогу отримати більше ідей у порівнянні з аналізом даних ізольовано. Однак часто дуже складно вирішувати проблеми навчання на графах, оскільки багато типів даних спочатку не структуровані як графи, такі як зображення та текстові дані, і для даних, структурованих у графі, основна зв'язність візерунки часто складні й різноманітні. З іншого боку, репрезентаційне навчання досягло великих успіхів у багатьох сферах. Таким чином, потенційним рішенням є вивчення представлення графів у маловимірному евклідовому просторі, щоб можна було зберегти властивості графа. Незважаючи на те, що було докладено величезних зусиль для вирішення проблеми навчання графічного представлення, багато з них все ще страждають від неглибоких механізмів навчання. Моделі глибокого навчання на графах (наприклад, графіві нейронні мережі) нещодавно з'явилися в машинному навчанні та інших суміжних областях і продемонстрували чудову продуктивність у різних проблемах.

Графи природно виникають у багатьох реальних програмах, включаючи соціальний аналіз [1], виявлення шахрайства [2], [3], прогнозування трафіку [4], комп'ютерний зір [5] та багато іншого. Представляючи дані у вигляді графів, структурну інформацію можна закодувати, щоб моделювати відносини між сутностями та надати більш багатообіцяюче розуміння, що лежить в основі даних. Наприклад, у транспортній мережі вузли часто є датчиками, а ребра представляють просторову близькість між датчиками. На додаток до тимчасової інформації, яку надають самі датчики, структура графа, змодельована

просторовими кореляціями, призводить до помітного покращення проблеми прогнозування трафіку [4]. Більше того, моделюючи транзакції між людьми у вигляді графа, складні шаблони транзакцій можна видобути для виявлення синтетичної особи [3] та виявлення відмивання грошей [6].

Однак складна структура графів [7] часто перешкоджає можливості отримати справжнє розуміння, що лежить в основі графів. Така складність, наприклад, полягає в неевклідовій природі даних, структурованих у графі. Потенційним рішенням для роботи зі складними шаблонами є вивчення представлень графів у низьковимірному евклідовому просторі за допомогою методів вбудовування, включаючи традиційні методи вбудовування графів [8], [9], [10], а також новітні методи вбудовування мереж [11], [12]. Після вивчення низьковимірних представлень можна легко вирішити багато проблем, пов'язаних з графами, наприклад, класичну класифікацію вузлів і передбачення зв'язків [12]. Існує багато ретельних оглядів як традиційного вбудовування графів, так і останніх методів мережевого вбудовування.

Незважаючи на певні успіхи цих методів вбудовування, багато з них страждають від обмежень неглибоких механізмів навчання [11], [12] і можуть не виявити більш складні закономірності за графами. З іншого боку, моделі глибокого навчання були продемонстровані в багатьох програмах. Наприклад, згорткові нейронні мережі (CNN) досягають багатообіцяючої продуктивності в багатьох програмах комп'ютерного зору [13] та обробки природної мови [14]. Однією з ключових причин такого успіху є те, що моделі CNN можуть в значній мірі використовувати властивості стаціонарності та композиційності певних типів даних. Зокрема, завдяки сітчастій природі зображень, згорткові шари в CNN дозволяють використовувати переваги ієрархічних шаблонів і витягувати високорівневі характеристики зображень для досягнення великої виразної здатності. Основні моделі CNN мають на меті вивчити набір локалізованих фільтрів фіксованого розміру, які можна тренувати, які сканують кожен піксель у

зображеннях та комбінують навколишні пікселі. Основні компоненти включають шари згортки та об'єднання, якими можна оперувати даними з евклідовою або сітковою структурою.

Однак неевклідова характеристика графів (наприклад, неправильна структура) робить згортки та фільтрацію на графах не такими чіткими, як на зображеннях. В останні десятиліття дослідники працювали над тим, як проводити згорткові операції над графами. Одним з основних напрямків дослідження є визначення згорток графа зі спектральної точки зору, і, таким чином, обробка сигналів графа, така як фільтрація графів і вейвлети графа, привернула багато дослідницьких інтересів. Є огляд обробки сигналів графа, включаючи загальні операції та аналіз на графах [15]. Коротко кажучи, згортки спектрального графа визначаються в спектральній області на основі перетворення Фур'є графа, аналогії одновимірного сигналу перетворення Фур'є. Таким чином, згортки графа на основі спектру можна обчислити, взявши обернене перетворення Фур'є множення між двома перетвореними Фур'є сигналами графа. З іншого боку, згортка графа також може бути визначена в просторовій області (тобто в області вершин) як агрегації представлень вузлів з околиць вузлів. Поява цих операцій відкриває двері для побудови графових згорткових мереж. Взагалі кажучи, моделі згорткових мереж з графами – це тип архітектур нейронних мереж, які можуть використовувати структуру графа та згортати інформацію про вузли з околиць згортковим способом. Графові згорткові мережі мають велику виразну силу для вивчення представлень графів і досягли чудової продуктивності в широкому діапазоні завдань і застосувань.

### 1.1 Застосування графових згорткових мереж

Графові згорткові мережі також можна класифікувати відповідно до областей їх застосування. Можна представити застосування згорткових

мереж графів у комп'ютерному баченні, обробці природної мови, науці та інших областях.

### 1.1.1 Застосування в комп'ютерному зору

Комп'ютерний зір був одним з найпопулярніших напрямків досліджень за останні десятиліття. Багато існуючих архітектур глибокого навчання, що використовуються в проблемах комп'ютерного зору, побудовані на основі класичних згорткових нейронних мереж (CNN). Незважаючи на великі успіхи CNN, їм важко закодувати внутрішні структури графів у конкретних навчальних завданнях. На відміну від цього, для вирішення деяких проблем комп'ютерного зору були застосовані мережі згорткових графів і показали порівнянну або навіть кращу продуктивність.

### 1.1.2 Зображення

Класифікація зображень має велике значення в багатьох реальних програмах. За допомогою деяких ретельно розроблених методів побудови графів (наприклад, графів подібності kNN) або інших контрольованих підходів неструктуровані зображення можна перетворити в дані структурованого графа і, таким чином, застосувати до згорткових мереж графів. Існуючі моделі класифікації зображень включають, але не обмежуються ними [5], [16], [17], [18], [19]. Інша програма для зображень – це візуальні відповіді на запитання, які досліджують відповіді на запитання на зображеннях. Крім того, оскільки зображення часто містять кілька об'єктів, розуміння взаємозв'язків (тобто візуальних зв'язків) між об'єктами допомагає охарактеризувати взаємодію між ними, що робить візуальні міркування гарячою темою в комп'ютерному баченні.

### 1.1.3 Відео

Одним із ефективних застосувань відео є розпізнавання дій, яке може допомогти зрозуміти відео. У [20] згорткова модель просторово-часового графа розроблена для того, щоб усунути необхідність присвоєння деталей, виготовлених вручну, і може досягти більшої виразної сили. Іншим методом, заснованим на скелеті, де пропонується процес побудови узагальненого графа, щоб фіксувати варіації в скелетних послідовностях, а потім узагальнений граф подається до згорткової мережі графа для навчання варіацій.

### 1.1.4 Хмари точок

Хмари точок забезпечують гнучке геометричне представлення для багатьох додатків у комп'ютерній графіці та комп'ютерному баченні. Після інноваційної PointNet [21], найсучасніші глибокі нейронні мережі розглядають локальні особливості хмар точок [21], [22]. Однак у цих роботах ігноруються геометричні співвідношення між точками. EdgeConv [23], з іншого боку, пропонується фіксувати локальну геометричну структуру, зберігаючи властивість інваріантності перестановок, і перевершує інші існуючі підходи в задачі сегментації хмари точок. Для сегментації на хмарах точок була запропонована регуляризована модель згорткової мережі графа, в якій лапласіан графа динамічно оновлюється, щоб охопити зв'язність вивчених ознак. FeaStNet [24], побудований на основі згорткових мереж графів, динамічно визначає зв'язок між ваговими вагами фільтрів та околицями графа, показуючи порівнянну продуктивність у маркуванні частин.

### 1.1.5 Застосування в обробці природної мови

Класифікація тексту є однією з найбільш класичних проблем обробки природної мови. З документами як вузлами і зв'язками цитування між ними як ребрами можна побудувати мережу цитування, в якій атрибути вузла часто моделюються пакетом слів. У цьому сценарії простим способом класифікації документів за різними категоріями є класифікація вузлів. Було запропоновано багато моделей згорткових мереж. Інший спосіб — переглядати документи на рівні графа (тобто кожен документ моделюється як граф) і класифікувати тексти за класифікацією графів. Крім того, TextGCN [25] моделює весь корпус у вигляді гетерогенного графа і одночасно вивчає вбудовування слів і документ, за яким слід softmax класифікатор для класифікації тексту.

Вилучення інформації часто є наріжним каменем багатьох програм, пов'язаних з НЛП, і згорткові мережі графів широко застосовуються в ньому та його варіантах проблем. Наприклад, GraphIE [26] спочатку використовує рекурентну нейронну мережу для створення локальних контекстно-залежних прихованих уявлень слів або речень, а потім вивчає нелокальні залежності між текстовими одиницями, а потім декодер для маркування на рівні слова. GraphIE можна застосувати до вилучення інформації, наприклад вилучення іменованих об'єктів. Для виділення відношень між словами [27], [28] та вилученням подій були розроблені згорткові мережі графів [29], [30].

Для семантичного машинного перекладу згорткові мережі графів можна використовувати для введення семантичного упередження в кодувальники речень і досягнення покращення продуктивності [31]. Крім того, модель згорткової мережі з розширеним ітерованим графом призначена для аналізу залежностей [32].

## 1.1.6 Застосування в науці

### 1.1.6.1 Фізика

У фізиці елементарних частинок струмені називають колімованими бризками енергійних адронів, і багато завдань пов'язані зі струменями, включаючи проблеми класифікації та регресії, пов'язані з початковими частинками, які породжують струмені. Нещодавно були розроблені варіанти нейронної мережі з передачі повідомлень [33] для класифікації струменів на два класи: струмені на основі квантової хромодинаміки та струмені на основі W-бозонів [34]. ParticleNet, побудований на основі згортки ребер [23], є налаштованою архітектурою нейронної мережі, яка працює безпосередньо з хмарами частинок для позначення струменя [35]. Крім того, для класифікації сигналів IceCube також застосована модель згорткової мережі графів [36]. Іншим цікавим застосуванням є передбачення фізичної динаміки, наприклад, як куб деформується при зіткненні з землею. Запропоновано ієрархічне представлення об'єкта на основі графів, яке розкладає об'єкт на частинки та з'єднує частинки в межах однієї групи або з предками та нащадками [37]. Можна використати згорткову мережу з ієрархічним графом, щоб дізнатися про фізичні передбачення.

### 1.1.6.2 Хімія, біологія та матеріалознавство

Вивчення молекул привернуло велику увагу в хімії, відкритті ліків і матеріалознавстві. Наприклад, для прогнозування молекулярних відбитків пальців використовувалися графові згорткові мережі [38], [39]. У відкритті ліків DeepChemStable [40], режим згорткової мережі, що базується на увазі, використовується для прогнозування хімічної стабільності сполуки. Крім того, моделюючи взаємодію білок-білок, взаємодію ліки-білок-мішень у мультимодальний граф, згортки графа можна застосувати для

прогнозування побічних ефектів поліпрагмазії [41]. Іншим важливим застосуванням в хімії є передбачення молекулярних властивостей. Нейронні мережі, що передають повідомлення (MPNN) [33], загальний каркас нейронної мережі графа, можна використовувати для прогнозування квантових властивостей молекули. PotentialNet [42] спочатку передбачає згортки графів над хімічними зв'язками для вивчення особливостей атомів, потім тягне за собою поширення як на основі зв'язку, так і на основі просторової відстані і, нарешті, проводить збір графів над атомами ліганду, а потім повністю зв'язаний шар для прогнозування молекулярних властивостей.

### 1.1.7 Аналіз соціальних мереж

Крім застосувань у класичних проблемах соціальної науки, таких як виявлення спільноти [42], [43] та передбачення зв'язків [21], [44], [45], згорткові мережі графів застосовуються в багатьох інших проблемах. DeepInf [46] має на меті передбачити соціальний вплив шляхом вивчення прихованих функцій користувачів. Можна використовувати графові згорткові мережі для прогнозування кількості ретвітів [47]. Крім того, фейкові новини можна також виявити за допомогою згортки графів [48]. Згорткові мережі графіків широко використовуються для соціальних рекомендацій, які мають на меті використовувати взаємодію між користувачем та/або користувачем для підвищення ефективності рекомендацій. Пропонують модель дифузії нейронного впливу, яка враховує те, як на користувачів впливають їхні надійні друзі, для кращих соціальних рекомендацій.

## 1.2 Класичні просторово згорткові графові мережі на основі CNN

Класичні моделі CNN щодо даних, подібних до сітки, таких як зображення, продемонстрували великий успіх у багатьох пов'язаних програмах, включаючи класифікацію зображень [49], [50], [51], виявлення об'єктів [13], [52], семантичну сегментацію [53], [54], тощо. Основні властивості даних, подібних до сітки, які використовуються архітектурами згортки, включають: кількість сусідніх пікселів для кожного пікселя є фіксованою, і просторовий порядок сканування зображень визначається природним чином, тобто зліва праворуч і зверху вниз. Однак, на відміну від зображень, у довільних даних графа не фіксується ні кількість сусідніх одиниць, ні просторовий порядок серед них.

Для вирішення цих проблем було запропоновано багато робіт для побудови графових згорткових мереж безпосередньо на класичних CNN. Пропонують вирішити вищезгадані проблеми шляхом виділення локально зв'язаних областей з графів [55]. Запропонована модель PATCHY-SAN спочатку визначає впорядкування вузлів за даним підходом позначення графа, таким як методи на основі центральності (наприклад, ступінь, PageRank, міжутворення тощо), і вибирає послідовність вузлів фіксованої довжини. По-друге, щоб вирішити проблему довільного розміру сусідства вузлів, для кожного вузла створюється околиця фіксованого розміру. Нарешті, граф околиць нормалізується відповідно до процедур позначення графів, так що вузлам подібних структурних ролей призначаються подібні взаємні позиції, після чого відбувається навчання представлення за допомогою класичних CNN. Однак, оскільки просторовий порядок вузлів визначається даним підходом до позначення графів, який часто ґрунтується виключно на структурі графа, PATCHY-SAN не має гнучкості навчання та загальності для більш широкого кола застосувань.

На відміну від PATCHY-SAN, що впорядковує вузли за структурною інформацією [55], модель LGCN [56] пропонується для перетворення даних

неправильного графа в дані, подібні до сітки, використовуючи як структурну інформацію, так і вхідну карту ознак  $p$ -го шару. Зокрема, для вузла  $u \in \mathcal{V}$  в  $\mathcal{G}$  він укладає вхідну карту ознак сусідів вузла  $u$  в одну матрицю  $M \in \mathbb{R}^{|\mathcal{N}(u)| \times d_p}$ , де  $|\mathcal{N}(u)|$  являє собою кількість сусідніх вузлів з 1 стрибком вузла  $u$ . Для кожного стовпця  $M$  перші  $r$  найбільших значень зберігаються і утворюють нову матрицю  $\tilde{M} \in \mathbb{R}^{r \times d_p}$ . Таким простим способом вхідну карту ознак разом зі структурною інформацією графіка можна перетворити на одновимірні сіткові дані  $\tilde{X}_p \in \mathbb{R}^{n \times (r+1) \times d_p}$ . Потім класичну 1-D CNN можна застосувати до  $\tilde{X}_p$  і вивчити нові представлення вузлів  $X^{p+1}$ . Зауважте, що метод навчання на основі підграфів також пропонується для масштабування моделі до великомасштабних графіків.

Оскільки згортка в класичних CNN може керувати лише даними з однаковими топологічними структурами, інший спосіб розширити класичні CNN на дані графіка – це розробити операцію згортки з усвідомленням структури як для евклідових, так і для неевклідових даних. Оскільки ця структурно-свідома згортка вимагає вивчення нескінченних параметрів, для апроксимації використовується поліном Чебишева [58]. Також класичну CNN можна перебудувати, розробляючи набір фільтрів фіксованого розміру, які можна вивчати (наприклад, від розміру-1 до розміру- $K$ ) і показує, що ці фільтри адаптивні до топології графа.

### 1.3 Просторові згорткові мережі графа на основі поширення

У цьому підрозділі ми зосередимося на згортках просторового графа, які поширюють та об'єднують представлення вузлів із сусідніх вузлів у вершинній області. Розглянемо згортку графіка для вузла  $u$  на  $p$ -му шарі, яка оформлена як:

$$x_{\mathcal{N}(u)}^p = X^p(u, :) + \sum_{v \in \mathcal{N}(u)} X^p(v, :), \quad (1.1)$$

$$X^{p+1}(u, :) = \sigma\left(x_{\mathcal{N}(u)}^p \Theta_{|\mathcal{N}(u)|}^p\right), \quad (1.2)$$

де  $\Theta_{|\mathcal{N}(u)|}^p$  є ваговою матрицею для вузлів з таким же ступенем, як  $|\mathcal{N}(u)|$  на  $p$ -му шарі.

Однак для довільно великих графіків кількість унікальних значень ступеня вузла часто є дуже великим числом. Отже, на кожному шарі потрібно буде вивчати багато вагових матриць, що, можливо, призведе до проблеми перенавчання.

Запропонована згорткова мережу на основі дифузії (названу DCNN), яка викликає поширення та агрегацію представлень вузлів за допомогою процесів дифузії графів [59].  $K$ -ступінчаста дифузія здійснюється за  $k$ -ою потужністю матриці переходів  $P^k$ , де  $P = D^{-1}A$ . Тоді операція дифузії-згортки формулюється так:

$$Z(u, k, i) = \sigma\left(\Theta(k, i) \sum_{v=1}^n P^k(u, v) X(v, i)\right), \quad (1.3)$$

де  $Z(u, k, i)$  –  $i$ -я вихідна характеристика вузла  $u$ , агрегована на основі  $P^k$ , а нелінійна функція активації  $\sigma(\cdot)$  обрана як гіперболічна дотична функція.

Припустимо, що розглядається дифузія  $K$  стрибків, і тоді матриця переходу  $K$ -ї потужності вимагає обчислювальної складності  $O(n^2K)$ , що неприпустимо особливо для великомасштабних графів.

Монті запропонував загальну мережу згортки графів під назвою MoNet [5] шляхом розробки універсального оператора виправлення, який інтегрує сигнали в околиці вузла. Зокрема, для вузла  $i$  та його сусіднього

вузла  $j \in \mathcal{N}(i)$  вони визначають  $d$ -вимірні псевдокоординати  $u(i, j)$  і подають його в  $P$  функцій ядра, які можна вивчати  $(w_1(u), \dots, w_P(u))$ .

Тоді оператор виправлення формулюється як

$$D_p(i) = \sum_{j \in \mathcal{N}(i)} w_p(u(i, j)) x(j), p = 1, \dots, P, \quad (1.4)$$

де  $x(j)$  – сигнал значення у вузлі  $j$ .

Тоді згортка графіка в просторовій області базується на операторі виправлення як:

$$(x *_s y)(i) = \sum_{l=1}^P g(p) D_p(i) x. \quad (1.5)$$

Показано, що шляхом ретельного вибору  $u(i, j)$  та функції ядра  $w_p(u)$  багато існуючих моделей згорткових мереж [60], [59] можна розглядати як окремий випадок MoNet. SplineCNN [61] дотримується тієї ж рамки, але використовує інше ядро згортки на основі В-сплайнів.

Для графів, що супроводжуються інформацією про атрибути ребра, вагові параметри фільтрів часто обумовлюються конкретними атрибутами краю в околиці вузла. Для використання атрибутів краю, операція з обумовленої краєм згортки розроблена на основі запозичення ідеї мережі динамічних фільтрів [62]. Для краю між вузлом  $v$  і вузлом  $u$  на  $p$ -му рівні ЕСС з відповідною мережею  $F^p$ , що генерує фільтри:  $\mathbb{R}^s \rightarrow \mathbb{R}^{d_{p+1} \times d_p}$ , яка генерує специфічну для краю матрицю ваг  $\Theta_{v,u}^p$ , операція згортки математично формалізується як:

$$X^{p+1}(u, :) = \frac{1}{|\mathcal{N}(u)|} \sum_{v \in \mathcal{N}(u)} \Theta_{v,u}^p X^p(v, :) + b^p, \quad (1.6)$$

де  $b^p$  є зміщенням, яке можна вивчати, а мережа фільтрації-генерування  $F^p$  реалізована багатошаровими перцептронами.

Модель навчання індуктивного представлення на основі агрегації – GraphSAGE [63]. Повна пакетна версія алгоритму проста: для вузла  $u$  шар згортки в GraphSAGE по-перше об'єднує вектори представлення всіх його безпосередніх сусідів у поточному шарі через деякий агрегатор, який можна навчатися, по-друге об'єднує вектор представлення вузла  $u$  з його агрегованим представленням, а потім по-третє подає конкатенований вектор до повністю зв'язаного шару з деякою нелінійною функцією активації  $\sigma(\cdot)$ , з подальшим кроком нормалізації. Формально  $p$ -й згортковий шар у GraphSAGE містить:

$$x_{\mathcal{N}(u)}^p \leftarrow \text{AGGREGATE}_p(\{X^p(v, :), \forall v \in \mathcal{N}(u)\}), \quad (1.7)$$

$$X^{p+1}(u, :) \leftarrow \sigma(\text{CONCAT}(X^p(u, :), x_{\mathcal{N}(u)}^p)\Theta^p). \quad (1.8)$$

Є кілька варіантів функцій агрегатора, включаючи агрегатор середнього значення, агрегатор LSTM та агрегатор об'єднання.

Використовуючи агрегатори середніх, рівняння (1.7) можна спростити до:

$$X^{p+1}(u, :) \leftarrow \sigma(\text{MEAN}(\{X^p(u, :)\} \cup \{X^p(v, :), \forall v \in \mathcal{N}(u)\})\Theta^p), \quad (1.9)$$

що приблизно нагадує модель GCN [60].

Крім того, агрегатор об'єднання складається так:

$$\text{AGGREGATE}_p^{\text{pool}} = \max(\{\sigma(X^p(v, :))\Theta^p + b^p, \forall v \in \mathcal{N}(u)\}). \quad (1.10)$$

Для забезпечення міні-пакетного навчання автори також пропонують варіант шляхом рівномірної вибірки фіксованого розміру сусідніх вузлів для кожного вузла [63].

Однак продуктивність навчання представлення вузлів часто погіршується, оскільки згорткові моделі графів стають глибшими. На практиці було показано, що двошарова модель згортки графа часто досягає найкращої продуктивності в GCN [60] та GraphSAGE [63]. Згідно з [64], згортка в GCN [60] пов'язана з лапласівським згладжуванням [65], і більше шарів згортки призводить до менш розрізнених уявлень навіть для вузлів з різних кластерів. З іншої точки зору, проаналізувавши різну поведінку розширення для двох типів вузлів, включаючи вузли в ядрі, подібній до розширювача, і вузли в деревоподібній частині графіків та однакова кількість кроків поширення може призвести до різних ефектів [66]. Наприклад, для вузлів всередині основної частини вплив їхніх характеристик поширюється набагато швидше, ніж вузлів у частині дерева, і, таким чином, це швидке середнє робить представлення вузлів нерозрізненими. Щоб пом'якшити цю проблему та зробити згорткові моделі графів глибшими, запозичивши ідею залишкової мережі [67] у комп'ютерному баченні. Jumping Knowledge Network може адаптивно вибирати агрегації з різних шарів згортки. Іншими словами, останній рівень моделі може вибірково агрегувати проміжні уявлення для кожного вузла незалежно. Пошарові агрегатори включають агрегатор конкатенації, агрегатор максимального об'єднання та агрегатор уваги LSTM. Крім того, модель Jumping Knowledge Network допускає комбінацію з іншими існуючими моделями нейронної мережі графів, такими як GCN [60], GraphSAGE [63] та GAT [68].

## 2 СПЕКТРАЛЬНА КОНВОЛЮЦІЯ ГРАФОВИХ НЕЙРОННИХ МЕРЕЖ

### 2.1 Спектральна ConvGNN

Спектральні ConvGNN спираються на теорію спектральних графів [69]. У цьому контексті сигнал на графах фільтрується за допомогою власного розкладання графа Лапласа [70]. Лапласіан графа визначається  $L = D - A$  (або  $L = I - D^{-1/2}AD^{-1/2}$  для нормалізованої версії), де  $A$  – матриця суміжності,  $D \in \mathbb{R}^{n_k \times n_k}$  – діагональна матриця ступенів з записами  $D_{i,i} = \sum_j A_{j,i}$  і  $I$  є тотожною матрицею. Оскільки лапласіан є позитивно напіввизначеним, його можна розкласти на  $L = U\Sigma U^T$ , де  $U$  – матриця власних векторів, а  $\Sigma = \text{diag}(\lambda)$ , де  $\lambda$  позначає вектор додатних власних значень. Графове перетворення Фур'є будь-якого одновимірного сигналу на графіку визначається як  $x_{ft} = U^T x$ , а його обернене задається як  $x = Ux_{ft}$ . Транспонуючи теорему згортки на графіки, спектральну фільтрацію в частотній області можна визначити як

$$x_{filtered} = U \text{diag}(F(\lambda)) U^T x, \quad (2.1)$$

де  $F(\lambda)$  – шукана функція фільтра, застосована до власних значень  $\lambda$ .

Як наслідок, шар згортки графіка в спектральній області може бути записаний сумою відфільтрованих сигналів з наступною функцією активації, як у [71], а саме

$$H_j^{(l+1)} = \sigma \left( \sum_{i=1}^{f_l} U \text{diag}(F_{i,j,l}) U^T H_i^{(l)} \right), \quad (2.2)$$

для всіх  $j \in \{1, \dots, f_{l+1}\}$ ,

де  $\sigma$  – функція активації, така як RELU (RECTified Linear Unit);

$H_i^{(l)}$  – вектор  $i$ -ї характеристики  $l$ -го шару;

$F_{i,j,l} \in \mathbb{R}^n$  – відповідний ваговий вектор, розмір якого є числом власних векторів (також  $n$ , кількість вузлів).

Спектральна ConvGNN, заснована на (2.2), намагається налаштувати параметри, які можна навчати,  $F_{i,j,l}$ , як запропоновано в [72] для задачі з одним графом.

Першим недоліком є необхідність перетворення Фур'є та зворотного перетворення Фур'є шляхом множення матриць  $U$  та  $U^T$ . Інший недолік виникає при узагальненні підходу до задач навчання з кількома графами. Дійсно,  $k$ -й елемент вектора  $F_{i,j,l}$  зважає внесок  $k$ -го власного вектора у вихідні дані. Ці ваги не можна поділити між графіками різного розміру, що означає, що потрібна різна довжина  $F_{i,j,l}$ . Більше того, незважаючи на те, що граф має однакову кількість вузлів, їхні власні значення будуть різними, якщо їх структури відрізняються. Як наслідок, задана вага  $F_{i,j,l}$  може відповідати різним власним значенням у різних графах.

Щоб подолати ці проблеми, було визначено декілька просторово-локалізованих фільтрів, таких як кубічна параметризація B-сплайна [71] та поліноміальна параметризація [73]. За допомогою таких підходів параметри, які можна навчати, визначаються:

$$F_{i,j,l} = B \left[ W_{i,j}^{(l,1)}, \dots, W_{i,j}^{(l,S)} \right]^T, \quad (2.3)$$

де  $B \in \mathbb{R}^{n \times S}$  – початкова розроблена матриця;

$W^{(l,s)}$  – матриця, яку можна навчати для  $s$ -го ядра згортки  $l$ -го шару;

$W_{i,j}^{(l,s)}$  –  $(i, j)$ -й запис  $W^{(l,s)}$ ;

$S$  – бажана кількість ядер згортки.

Кожен стовпець у  $V$  розроблено як функція власних значень, а саме  $V_{i,j} = (F_j(\lambda_i))$ . У поліноміальному випадку кожен стовпець  $V$  є степенем власних значень, починаючи з 0-го і закінчуючи  $(S - 1)$ -им степенем. У випадку кубічного  $V$ -сплайна матриця  $V$  кодує коефіцієнти кубічного  $V$ -сплайна [71]. Зовсім нещодавній ConvGNN під назвою CayleyNet параметризує коефіцієнти, які можна навчати, за допомогою

$$F_{i,j,l} = [g_{i,j,l}(\lambda_1, h), \dots, g_{i,j,l}(\lambda_n, h)], \quad (2.4)$$

де  $h$  – параметр масштабу;

$\lambda_n$  – це  $n$ -е власне значення;

$g$  – функція спектрального фільтра, визначена наступним чином у [74]:

$$g(\lambda, h) = c_0 + 2Re \left( \sum_{k=1}^r c_k \left( \frac{h\lambda - i}{h\lambda + i} \right)^k \right), \quad (2.5)$$

де  $i^2 = -1$ ,  $Re(\cdot)$  – функція, яка повертає дійсну частину;

$c_0$  – дійсний навчальний коефіцієнт;

$k = 1, \dots, r, c_k$  – комплексні навчальні коефіцієнти.

Параметризація CayleyNet також приймає форму (2.3)

## 2.2 Просторова ConvGNN

Просторові ConvGNN можна узагальнити як поширення ознак вузла до вузлів сусідства з подальшою функцією активації у вигляді

$$H^{(l+1)} = \sigma \left( \sum_s C^{(s)} H^{(l)} W^{(l,s)} \right), \quad (2.6)$$

де  $H^{(l)} \in \mathbb{R}^{n \times f_l}$  – матриця ознак  $l$ -го шару з  $n$  вузлами та  $f_l$  об'єктами,  $s$  індексує ядра згортки;

$C^{(s)}$  – ядро згортки, яке визначає, як ознаки вузла поширюються на сусідні вузли;

$W^{(l,s)} \in \mathbb{R}^{f_l \times f_{l+1}}$  – це навчальна вагова матриця, яка відображає  $f_l$ -вимірні об'єкти у вимірі  $f_{l+1}$ .

Вибір ядер згортки визначає метод в літературі. Ванільна версія використовує одне ядро згортки з  $C = A + I$ . Така непросторова ConvGNN має ефект фільтрації низьких частот, оскільки застосовує однакові коефіцієнти до всіх сусідів і до самого вузла. Фільтри високих частот можна отримати шляхом диференціювання вагових матриць, що використовуються для обчислення сусідів і власних внесків [75]. У такому випадку процес згортки визначається як  $C^{(1)} = A$  і  $C^{(2)} = I$ .

Запропоновано деякі рішення для подолання обмежень використання лише фільтрів низьких і високих частот. Якщо вузли мають дискретні мітки (якщо ступінь вузла не може використовуватися як дискретна ознака), ваги можуть розділятися сусідами, чиї мітки однакові [76]. Інший метод полягає у визначенні впорядкування на вузлах, що входять у рецептивне поле згортки, і розподілі коефіцієнтів відповідно до цього переупорядкування [69]. Процес переупорядкування називається канонічним переупорядкуванням вузлів. Подібний підхід спільного використання, заснований на переупорядкованих сусідах, був представлений у [70].

Відмінність полягає в тому, що переупорядкування обчислюється відповідно до абсолютної кореляції ознак із центральним вузлом. Інший просторово-проектований метод, запропонований у [77], розглядає процес дифузії на графіку з використанням випадкових блукань. Це дозволяє викликати мінливість вихідного сигналу шляхом застосування випадкових блукань різної довжини до різних об'єктів. Усі вищезгадані згортки просторового графіка використовують матриці  $C^{(s)}$  із фіксованим

проектуванням, а змінність індукується  $W^{(l,s)}$  у (2.6). Інші методи використовують ядра згортки, що піддаються навчанню, щоб зробити згортки більш продуктивними з точки зору частотних профілів вихідного сигналу, такі як мережі уваги графів [78, 79], MoNet [80] та SplineCNN [81]. Механізм уваги налаштовує кожен елемент ядра згортки 1-го шару  $C^{(l,s)}$ , який визначається як функція властивостей зв'язаних вузлів і деякого параметра, який можна навчати

$$C_{i,j}^{(l,s)} = f \left( H_i^{(l)}, H_j^{(l)}, W_{AT}^{(l,s)} \right), \quad (2.7)$$

де  $H_i^{(l)} \in \mathbb{R}^{f_l}$  – вектор ознак  $i$ -го вузла для шару  $l$ ;

$W_{AT}^{(l,s)}$  кодує параметр, який можна навчати, ядра  $s$ -ї згортки для шару  $l$ ;

$f$  – деяка поелементна функція, яку потрібно вибрати.

У цьому випадку, оскільки ядра згортки вивчаються через  $W_{AT}^{(l,s)}$  з (2.7), навчальні параметри  $W^{(l,s)}$  із (2.6) можна визначити як ідентичність або інші параметри, які можна навчати, які можуть бути спільними з  $W_{AT}^{(l,s)}$ . Найвпливовіший механізм уваги, застосований до графічних даних, званий GAT [78], використовує вагові коефіцієнти мульти-уваги (позначені як ядра згортки з багатьма опорами), з

$$f \left( H_i^{(l)}, H_j^{(l)}, W_{AT}^{(l,s)} \right) = \text{softmax}_j \left( \sigma \left( a \left[ W H_i^{(l)} || W H_j^{(l)} \right] \right) \right), \quad (2.8)$$

де два лінійних перетворення розглядаються елементами загального набору параметрів, які можна навчати,  $W_{AT}^{(l,s)} = \{a, W\}$ , де  $a$  є ваговим вектором. Оператор  $||$  – оператор конкатенації,  $\sigma$  відповідає функції LeakyReLU [82], а  $\text{softmax}_j$  – нормована експоненціальна функція, яка використовує всіх сусідів  $i$ -го вузла для нормалізації краю  $i$ -го до  $j$ -го вузла. У обчисленні вихідних даних шару згортки (2.6) GAT пропонує використовувати ті самі

параметри  $W$ . Основним обмеженням цього методу є використання дуже маленького контексту, обмеженого особливостями пари вузлів, для визначення інтенсивності уваги. Dual-Primal Graph CNN (DPGCNN) [83] розширює цей підхід, визначаючи увагу за допомогою нових функцій, обчислених з околиці кожного вузла пари, отже, з використанням більшого контексту. Оскільки методи, згадані вище, визначені в просторовій області, вони роблять не надають жодного аналізу їх частотного спектру фільтрів. Більш того, їх частотні характеристики будуть різними для різних графіків. Крім того, їм потрібна більша мультипідтримка (увага чи підшарові ваги), щоб отримати вихід із високою мінливістю, що різко збільшує кількість параметрів моделі, які можна навчати.

### 2.3 Просторові згортки зі спектральним коренем

Як було сказано раніше, нещодавно було запропоновано деякі методи, щоб позбутися обчислювального навантаження графа Фур'є та оберненого графа перетворення Фур'є, в той же час покладаючи свої основи в спектральній області. Ці рішення спираються на апроксимацію згортки спектрального графа, засновану на поліноміальному розкладанні Чебишева для масштабованого лапласіана графа. Відповідно, першими двома ядрами Чебишева є  $C^{(1)} = I$  і  $C^{(2)} = 2L/\lambda_{max} - I$ , а інші ядра визначаються як

$$C^{(k)} = 2C^{(2)}C^{(k-1)} - C^{(k-2)}. \quad (2.9)$$

Дослідники показали, що будь-який бажаний фільтр можна записати як лінійну комбінацію цих ядер [84]. ChebNet є першим методом, який використав ці ядра в ConvGNN [73]. Одним з основних розширення та спрощення методу поліноміального розширення Чебишева є мережа згортки графів (GCN) [85]. GCN використовує віднімання других ядер Чебишева з першого за припущенням  $\lambda_{max} = 2$  і  $L$  є нормованим графіком

лапласіана. Однак замість того, щоб використовувати це віднімене ядро, вони використали прийом повторної нормалізації та визначили остаточне єдине ядро за допомогою:

$$C = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}, \quad (2.10)$$

де  $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$  і  $\tilde{A} = (A + I)$  матриця суміжності з доданими самозв'язками.

Цей підхід вплинув на багато інших внесків. Метод безпосередньо використовує цю згортку, але змінює архітектуру мережі, додаючи повністю підключений шар як останній шар. Алгоритм MixHop [86] використовує 2-ю або 3-ю степені однієї і тієї ж згортки. Наступні методи значно відрізняються від чистих просторових і чистих спектральних згорток. Вони не розроблені з використанням власних значень, а неявно розроблені як функція структурної інформації (суміжність, лапласів) і виконують згортку в просторовій області, як і всі просторові згортки. Однак їхні частотні профілі стабільні для різних довільних графіків, як і спектральні згортки.

## 2.4 ChebNet

Після обчислення ядер ChebNet за (2.9), можна використовувати для отримання їх частотних профілів. Перші два частотні профілі ядра ChebNet є  $F_1(\lambda) = 1$  і  $F_2(\lambda) = 2\lambda/\lambda_{max} - 1$ , де  $\mathbf{1}$  є вектором одиниць. Оскільки  $\lambda_{max} = 2$  для всіх трьох графіків, ми отримуємо  $F_2(\lambda) = \lambda - 1$ . Третій і наступні частотні профілі ядра також можна обчислити за допомогою  $F_k(\lambda) = 2F_2(\lambda)F_{k-1}(\lambda) - F_{k-2}(\lambda)$ , що призводить до  $F_3(\lambda) = \lambda^2 - 4\lambda + 1$ , наприклад, для третього ядра. Аналізуючи частотний профіль ChebNet, можна стверджувати, що згортки здебільшого охоплюють спектр. Однак

жодне з ядер не фокусується на певних частинах спектру. Як приклад, друге ядро є переважно низькочастотним і високочастотним фільтром і зупиняє середню смугу, тоді як третє пропускає дуже високі, дуже низькі та середні смуги, але зупиняє майже першу та третю чверть спектру. Отже, якщо зв'язок між парами введення-виводу можна визначити за допомогою лише фільтрів низьких частот, високих частот або певного смугового фільтра, потрібна велика кількість ядер згортки. Однак у літературі для експериментів зазвичай використовуються лише 2 або 3 ядра [73], [85].

## 2.5 CayleyNet

CayleyNet використовує згортки спектрального графа, частотні профілі яких можна змінити шляхом масштабування власних значень [74]. Профіль частоти визначається складною раціональною функцією власних значень, масштабованою за параметром  $h$ , який можна навчати в (2.5). CayleyNet можна визначити за допомогою матриці частотного профілю  $V$ . Використовуючи це уявлення, CayleyNet можна розглядати як кількома – згортки ядра з дійсними коефіцієнтами навчання. Згідно з цим аналізом, CayleyNet використовує  $2r + 1$  ядра згортки графа, при цьому  $r$  є кількістю комплексних коефіцієнтів [74]. Параметр масштабу  $h$  впливає на масштабування осі  $x$ , але не змінює глобальну форму. Коли  $h = 1$ , частотні профілі можна визначити в діапазоні  $[0,2]$  (оскільки  $\lambda_{max} = 2$  на всіх трьох тестових графіках). Якщо  $h = 1.5$ , профіль частоти можна визначити до  $1.5\lambda_{max} = 3$  і змінити масштаб мітки осі з  $[0,3]$  до  $[0,2]$  у вихідному діапазоні. Вивчення масштабування власних значень може здатися вигідним. Однак це викликає додаткові витрати на обчислення для обчислення нового ядра згортки. Щоб обмежити цю вартість, апроксимація обчислюється з використанням фіксованої кількості Jacobi iterations [74]. Крім того, як і ChebNet, CayleyNet не має жодних специфічних для смуги згорток, навіть якщо врахувати різні коефіцієнти масштабування.

## 2.6 GCN

Що стосується ChebNet, то теоретичний аналіз частотних профілів згортки GCN показує, що частотний профіль GCN можна апроксимувати відповідно до  $F(\lambda) \approx 1 - \lambda \bar{d} / (\bar{d} + 1)$ , де  $d$  – середнє ступінь вузла. Отже, гранична частота згортки GCN дорівнює  $\lambda_{cut} \approx (1 + \bar{d}) / \bar{d}$ . Теоретично, якщо всі вузли ступеня відрізняються, стандартний частотний профіль не буде гладким і включатиме деякі збурення. Крім того, повночастотний профіль буде складатися з ненульових компонентів. Експериментальний аналіз поведінки GCN [85] у спектральній області спочатку передбачає обчислення ядра згортки, як зазначено в (2.10). Тоді спектральне представлення отриманої матриці згортки можна розрахувати назад. Три стандартні частотні профілі мають майже однакову форму фільтра низьких частот, що відповідає функції, що складається із спадної частини в трьох перших чвертях діапазону власних значень, за якою слідує зростаюча частина в решті діапазону. Це спостереження узгоджується з теоретичним аналізом. Що стосується повних частотних профілів, то для звичайного лінійного графа немає внеску за межі діагоналі. І навпаки, деякі значення не по діагоналі не є нульовими для. Знову ж таки, це спостереження підтверджує теоретичний аналіз. Оскільки частотний профіль GCN не охоплює весь спектр, такий підхід не може вивчити відносини, які можуть бути представлені високочастотною або смуговою фільтрацією. Таким чином, навіть незважаючи на те, що він дає дуже хороші результати для задачі класифікації одного графового вузла [85], він може виявитися невдалим для задач, де дискримінантна інформація міститься в певних діапазонах частот. Тому такий підхід можна вважати проблемним.

## 2.7 DSGCN

Замість проектування ядер просторової згортки  $C^{(s)}$  з (2.6) за допомогою функцій суміжності графа та/або графа Лапласа, пропонуємо використовувати ядра згортки  $S$ , які мають спеціально розроблені стандартні частотні профілі. Ці розроблені частотні профілі є функцією власних значень, таких як  $[F_1(\lambda), \dots, F_S(\lambda)]$ . У цій пропозиції кількість ядер та їх частотні профілі є гіперпараметрами. Потім ми можемо розрахувати відповідні просторові матриці згорток. Щоб отримати згортки графа, що не залежать від проблеми, сума частотних профілів усіх розроблених згорток повинна охоплювати більшу частину можливого спектру, а частотний профіль кожного ядра має зосередитися на деякі певні діапазони частот. Щоб з'ясувати довільні співвідношення пар введення-виведення, необхідно ефективно спроектувати кілька ядер згортки. Однак збільшення числа  $S$  ядер згортки лінійно збільшує кількість параметрів, які можна навчати. Отже, загальна кількість мультипідтримуючих ConvGNN визначається як  $S \sum_{i=0}^L f_i f_{i+1}$ , де  $L$  – кількість шарів, а  $f_i$  – довжина ознак  $i$ -го шару.

Щоб подолати цю проблему, пропонується використовувати мережу згортки графів із роздільною глибиною – DSGCN (Depthwise Separable Graph Convolutions). Структура Depthwise Separable Convolution вже використовувалася в задачах комп'ютерного зору для зменшення розміру моделі та її складності [87], [88]. Замість того, щоб фільтрувати всі вхідні характеристики для кожної вихідної функції, DSGCN полягає у фільтрації кожної вхідної функції один раз. Потім відфільтровані сигнали об'єднуються в потрібну кількість вихідних об'єктів за допомогою згорток  $1 \times 1$  з різними коефіцієнтами внеску. Математично, пряме обчислення кожного шару DSGCN визначається:

$$H^{(l+1)} = \sigma \left( \left( \sum_{s=1}^S w^{(s,l)} \odot (C^{(s)} H^{(l)}) \right) W^{(l)} \right). \quad (2.11)$$

У цьому виразі позначення  $\odot$  позначає оператор поелементного множення. Зауважте, що в кожному шарі є лише одна навчальна матриця  $W$ . Інші навчальні змінні  $w^{(s,l)} \in \mathbb{R}^{1 \times f_l}$  кодують внесок ознак для кожного ядра згортки та шару. Кількість параметрів, які можна навчати, для цього випадку стає  $\sum_{i=0}^L S f_i + f_i f_{i+1}$ . Раніше додавання нового ядра збільшує кількість параметрів на  $\sum_{i=0}^L f_i f_{i+1}$ . Використовуючи роздільні згортки, це число лише збільшується на  $\sum_{i=0}^L f_i$ . Ця модифікація особливо цікава, коли кількість функцій велика. З іншого боку, мінливість моделі також зменшується. Якщо дані мають меншу кількість функцій, використання цього підходу може бути неоптимальним.

## 2.8 Переносимість фільтрів спектральних графів

Успіх згорткових нейронних мереж (ConvNets) в евклідових доменах викликав інтерес до розширення цих методів для створення графів структурованих даних. graphConvNet – це відображення, яке отримує сигнал, визначений на вершинах графа, і повертає значення в деякому вихідному просторі. ConvNet складається з багатьох рівнів обчислень, де кожен шар обчислює набір фільтрів вихідних даних попереднього рівня, за яким слідує нелінійність і, за бажанням, групування кроків і шарів без згортки. У налаштуваннях машинного навчання загальна архітектура ConvNet є фіксованою, але конкретні фільтри, які використовуються на кожному рівні, є вільними параметрами. Під час навчання коефіцієнти фільтра оптимізовані, щоб мінімізувати деякі функції втрат. У деяких ситуаціях і граф, і сигнал, визначені на графу, є змінними у вхідному просторі ConvNet. У цих ситуаціях, якщо два графи представляють одне й

те саме основне явище, а два сигнали, подані на двох графах, у певному сенсі подібні, вихід ConvNet для обох сигналів також має бути подібним. Цю властивість зазвичай називають переносимістю, і вона є важливою вимогою, якщо ми хочемо, щоб мережа ConvNet добре узагальнювала (рисунок 2.1) тестовий набір.

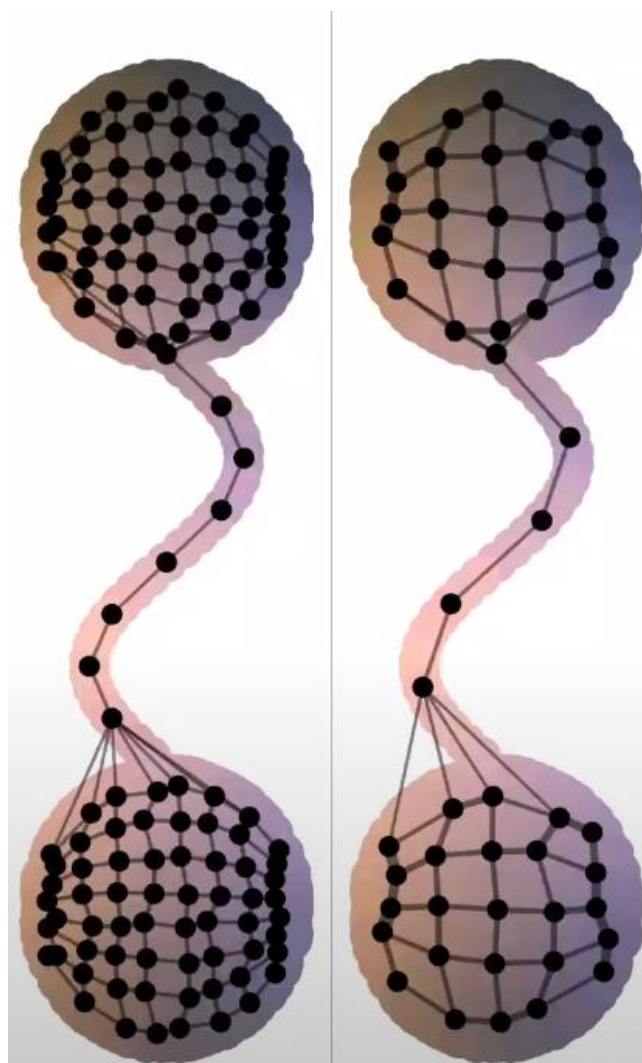


Рисунок 2.1 – Два графи зі схожими тенденціями зав'язків

Необхідною умовою будь-якого розумного визначення переносимості є стабільність. А саме, за умови фільтра, якщо топологія графа збурена, то фільтр на збуреному графі близький до фільтра на незбуреному графі. Без стабільності неможливо навіть перенести фільтр з графа на інший дуже

близький граф, тому стабільність необхідна для передачі. Числові результати щодо стійкості фільтрів поліноміального графа до адитивних гаусових збурень власних векторів лапласіана графа. Оскільки власний розклад не є стійким до збурень у топології графа, цей результат не доводить стійкість до таких збурень. Показано, що очікуваний графічний фільтр при випадкових втратах ребр дорівнює точному виходу. Однак не обмежується похибка на виході з точки зору похибки в топології графа (рисунок 2.2 – 2.6). Покажемо лінійну стійкість графічних фільтрів до загальних збурень у топології.

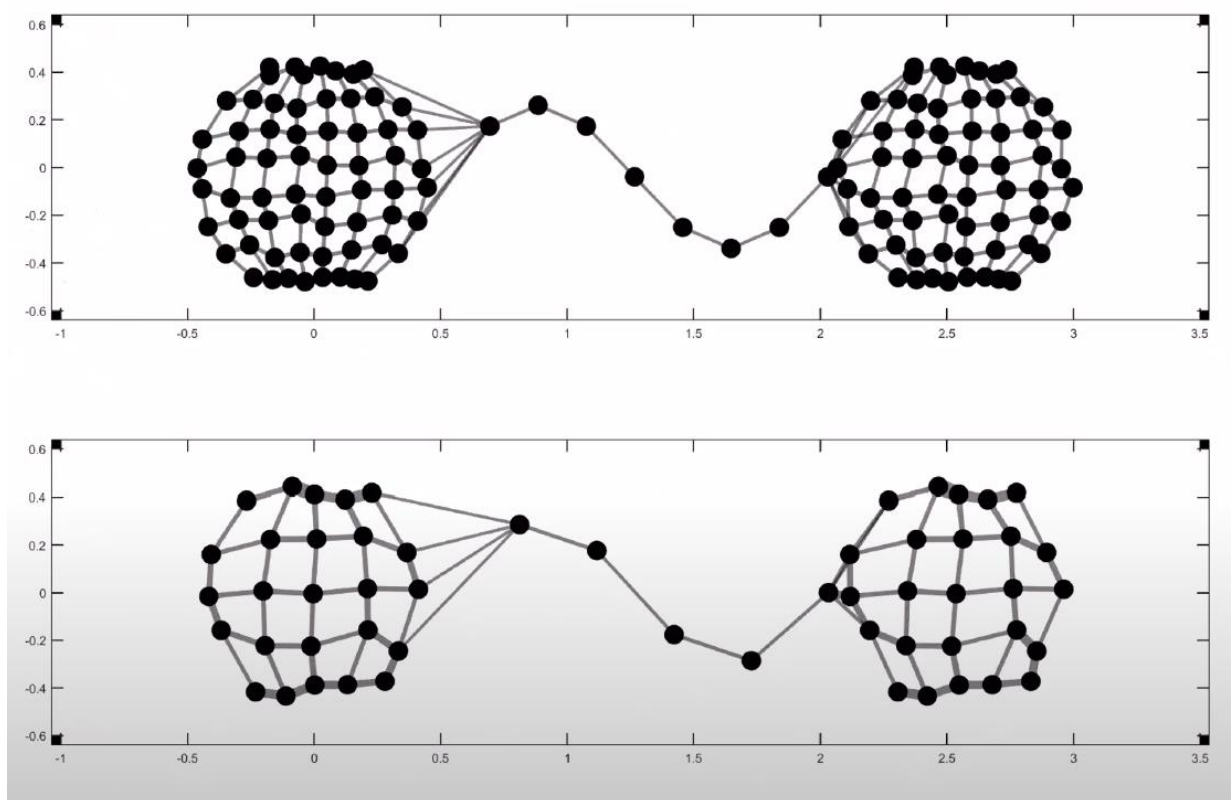


Рисунок 2.2 – Приклади графів

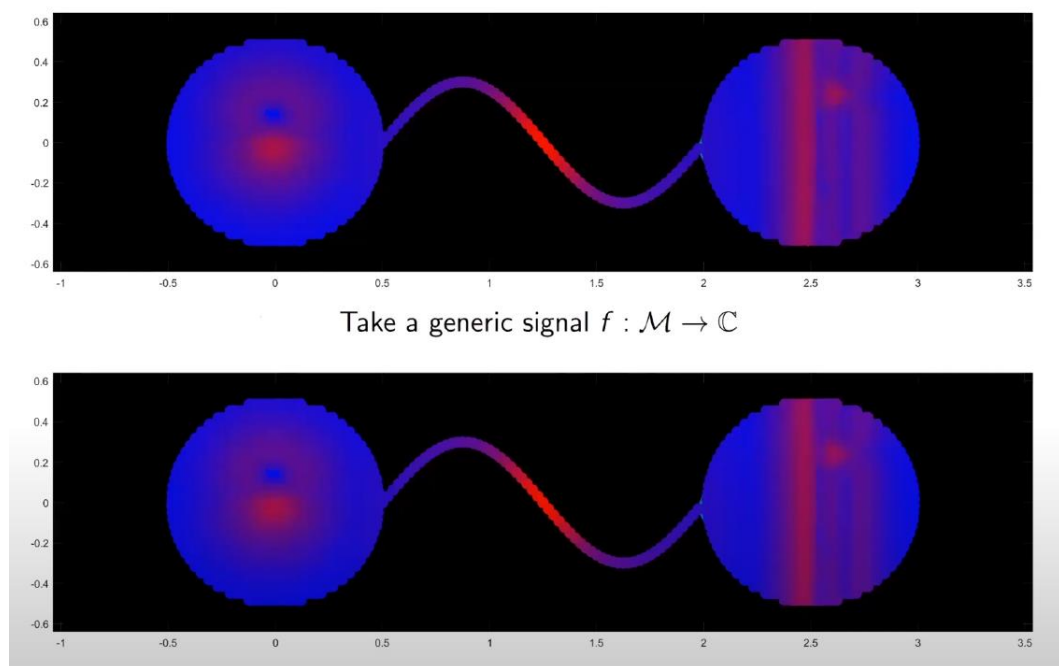


Рисунок 2.3 – Аналоговий сигнал

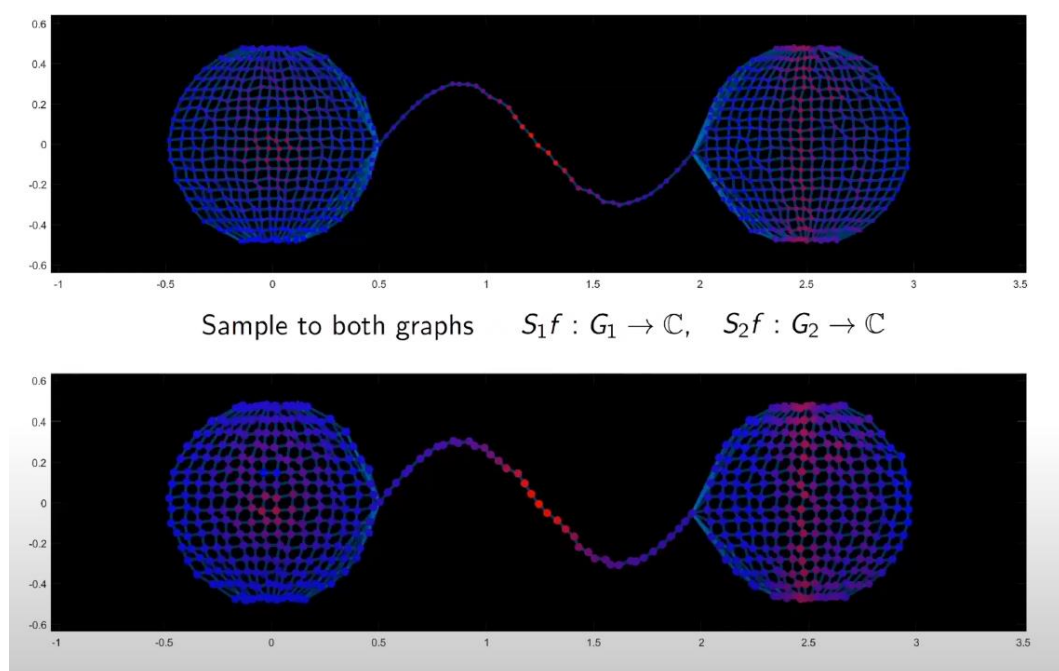


Рисунок 2.4 – Семпльовані графи

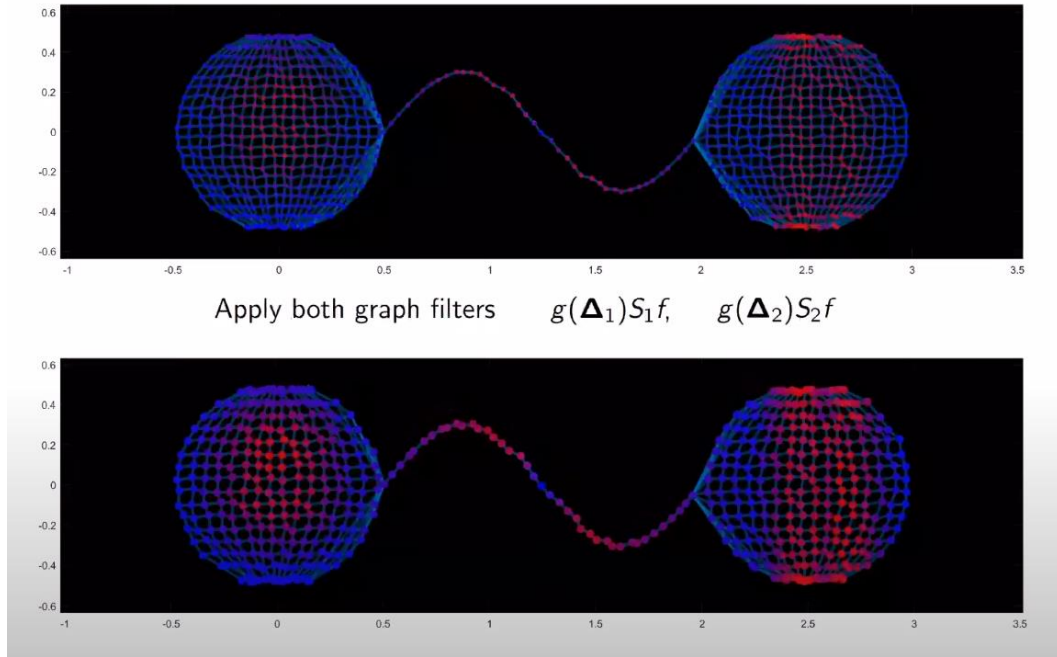


Рисунок 2.5 – Застасовані фільтри

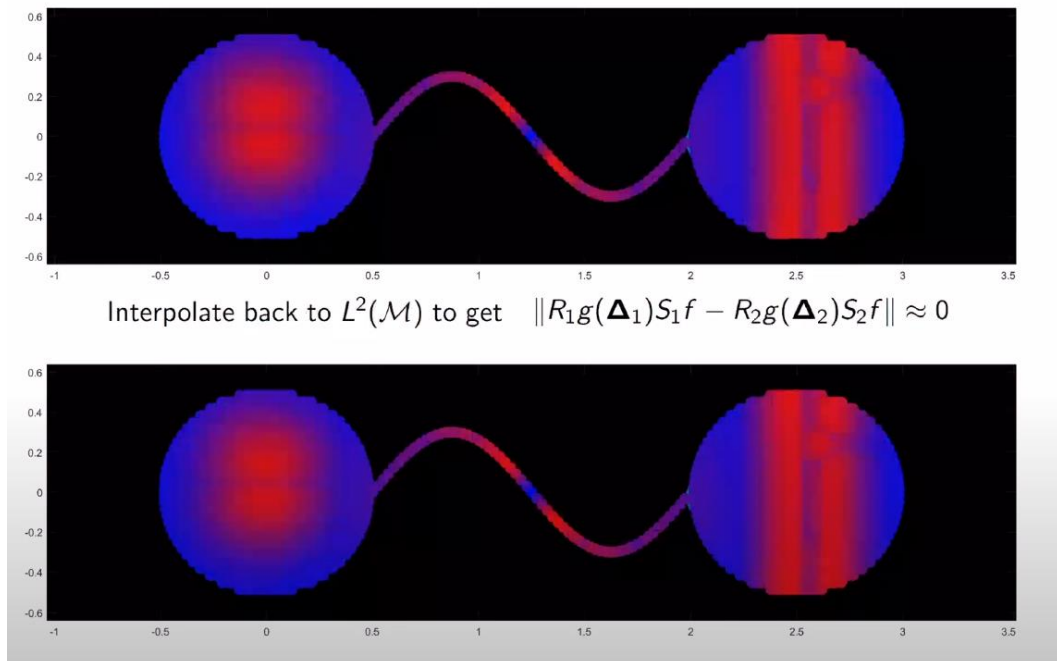


Рисунок 2.6 – Інтерполяція до аналогового сигналу

Загалом існує два підходи до визначення згортки на графах, обидва узагальнюють стандартну згортку на евклідові області [89], [90]. Просторові підходи узагальнюють ідею ковзного вікна на графі. Тут основним завданням є визначення способу трансляції ядра фільтра вздовж вершин графа. Деякі популярні приклади просторових методів [91], [92], [71]. Спектральні методи засновані на теоремі згортки в евклідових областях, яка стверджує, що згортка в просторовій області еквівалентна точковому множенню в частотній області. Завдання тут полягає у визначенні частотної області та перетворення Фур'є графів. Основна ідея полягає у визначенні лапласіана графа або іншого оператора графа, який ми інтерпретували як оператор зсуву, і використання його власних значень як частот, а його власних векторів як відповідних чистих гармонік [73]. Розкладання графічного сигналу до його чистих гармонічних коефіцієнтів за визначенням є перетворенням Фур'є на графі, а фільтри визначаються множенням різних частотних компонентів на різні значення.

Переважає більшість дослідників із спільноти ConvNet зараз зосереджується на розробці просторових методів. Однією з типових мотивів переваги просторових методів є твердження, що спектральні методи не підлягають передачі, а отже, вони погано узагальнюють графи, невидимі в навчальному наборі. Покажемо, що найсучасніші методи фільтрації спектральних графів можна передати. Це не заперечує проти просторових методів, але показує потенціал спектральних підходів для роботи з наборами даних, які мають різні графи. Цікаво, що дослідники отримують найсучасніші результати, використовуючи фільтри спектрального графіка на змінних графіках, без будь-яких модифікацій для компенсації «непередаваності».

Розглянемо неорієнтований зважений граф  $\mathcal{G} = \{E, V, W\}$  з вершинами  $V = \{1, \dots, N\}$ , ребра  $E$  та матриця суміжності  $W$ . Матриця суміжності  $W$  є симетричною і представляє ваги ребер, де  $w_{n,m}$  відмінний від нуля, лише якщо вершина  $n$  з'єднана з вершиною  $m$ . Розглянемо матрицю ступенів  $D$ ,

визначену як діагональну матрицю з елементами  $d_{n,n} = \sum_{m=1}^N w_{n,m}$ . Частотна область графіка визначається вибором оператора зсуву, а саме самосупряженого оператора  $\Delta$ , що враховує зв'язність графа. Як прототиповий приклад розглянемо ненормований лапласіан  $\Delta = D - W$ , який лінійно залежить (афінно) від  $W$ . Іншими прикладами поширених операторів зсуву є нормований лапласіан  $\Delta_n = I - D^{-1/2}WD^{-1/2}$ , і сама матриця суміжності. Будемо називати загальний самосупряжений оператор зсуву лапласіаном і позначимо його  $\Delta$ . Позначимо власні значення  $\Delta$  через  $\{\lambda_n\}_{n=1}^N$ , а власні вектори через  $\{\phi_n: V \rightarrow \mathbb{C}\}_{n=1}^N$ . Перетворення Фур'є сигналу графіка  $f: V \rightarrow \mathbb{C}$  задається вектором інтенсивностей частоти

$$\mathcal{F}f = (\langle f, \phi_n \rangle)_{n=1}^N, \quad (2.12)$$

де  $\langle u, v \rangle$  – внутрішній добуток в  $\mathbb{C}^N$ , наприклад, стандартний добуток.

Обернене перетворення Фур'є вектора  $(v_n)_{n=1}^N$  задається як

$$\mathcal{F}^*(v_n)_{n=1}^N = \sum_{n=1}^N v_n \phi_n. \quad (2.13)$$

Оскільки  $\{\phi_n\}_{n=1}^N$  є ортонормованим базисом,  $\mathcal{F}^*$  є оберненим до  $\mathcal{F}$ . Фільтр спектрального графіка  $G$  на основі коефіцієнтів  $(g_n)_{n=1}^N$  визначається як

$$Gf = \sum_{n=1}^N g_n \langle f, \phi_n \rangle \phi_n. \quad (2.14)$$

Будь-який спектральний фільтр, визначений (2.14), є еквіваріантом перестановки, а саме, не залежить від індексації вершин. Переіндексація

вершин у вхідних даних призводить до тієї ж переіндексації вершин у вихідних даних (рисунки 2.7 – 2.9).

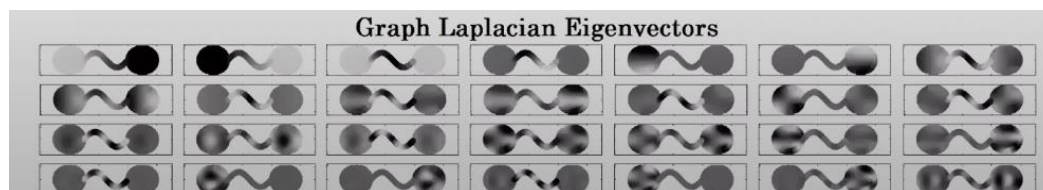


Рисунок 2.7 – Графічне відображення власних векторів лапласіана графа

eigenvalues	eigenfunctions
2	
6	
12	
20	
30	

Рисунок 2.8 – Інші приклади власних векторів

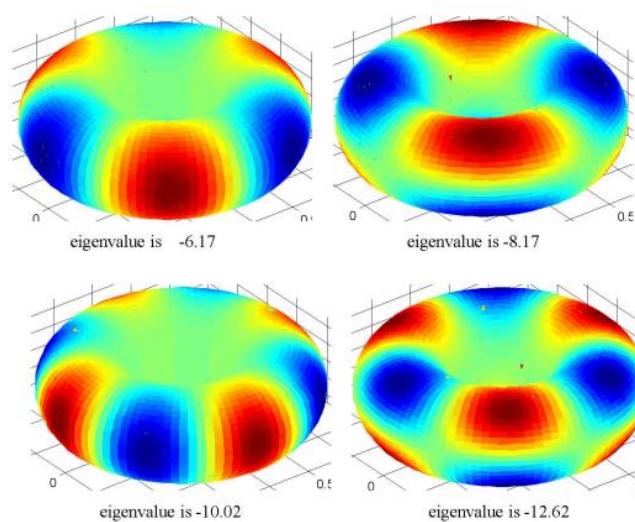


Рисунок 2.9 – Інші приклади власних векторів

Спектральні фільтри, визначені (2.14), мають два недоліки. По-перше, вони не підлягають передачі. По-друге, вони потребують високої обчислювальної складності. Формула (2.14) вимагає обчислення власного розкладання лапласіана  $\Delta$ , яке вимагає обчислень і може бути нестабільним, коли кількість вершин  $N$  велика. Більше того, не існує загального алгоритму «графічного FFT» для обчислення перетворення Фур'є сигналу  $f \in L^2(V)$ , і (2.14) вимагає обчислення частотних складових  $\langle f, \phi_n \rangle$  та їх підсумовування безпосередньо.

Щоб подолати ці два обмеження, найсучасніші методи, такі як [85], [69], [78], [79], засновані на функціональному обчисленні. Функціональне обчислення – це теорія застосування функцій  $g: \mathbb{C} \rightarrow \mathbb{C}$  про нормальні оператори в гільбертових просторах. У особливому випадку самосупряженого або унітарного оператора  $T$  з дискретним спектром  $g(T)$  визначається як

$$g(T)f = \sum_n g(\lambda_n) \langle f, \phi_n \rangle \phi_n \quad (2.15)$$

для будь-якого вектора  $f$  у просторі Гільберта, де  $\{\lambda_n, \phi_n\}$  – власний розклад оператора  $T$ . Оператор  $g(T)$  є нормальним для загального  $g: \mathbb{C} \rightarrow \mathbb{C}$ , самосупряженим для  $g: \mathbb{C} \rightarrow \mathbb{R}$  та унітарним для  $g: \mathbb{C} \rightarrow e^{i\mathbb{R}}$  (де  $e^{i\mathbb{R}}$  – одиничне комплексне коло). Визначення (2.15) є канонічним у наступному сенсі. В особливому випадку, коли

$$g(\lambda) = \frac{\sum_{l=0}^L c_l \lambda^l}{\sum_{l=0}^L d_l \lambda^l} \quad (2.16)$$

є раціональною функцією,  $g(T)$  можна визначити двома способами. По-перше, за (2.15), а по-друге за композиціями, лінійними комбінаціями та інверсіями, як

$$g(T) = \left( \sum_{l=0}^L c_l T^l \right) \left( \sum_{l=0}^L d_l T^l \right)^{-1}. \quad (2.17)$$

Можна показати, що (2.15) і (2.17) еквівалентні. Більше того, визначення (2.15) також канонічне щодо нераціональних функцій. Взагалі кажучи, якщо раціональна функція  $q$  апроксимує функцію  $g$ , то оператор  $q(T)$  апроксимує оператор  $g(T)$ . Реалізація (2.17) долає обмеження визначення (2.14), де тепер фільтри визначаються через (2.15) з поліномом або раціональною функцією  $g$ . Покладаючись на просторові операції композицій, лінійних комбінацій та інверсій, обчислення спектрального фільтра здійснюється повністю в просторовій області, ніколи не вдаючись до спектральних обчислень. Таким чином, власне розкладання та перетворення Фур'є завжди обчислюються. Інверсії в  $g(T)f$  включають розв'язування систем лінійних рівнянь, які можна обчислити безпосередньо, якщо  $N$  мале, або за допомогою якогось методу ітераційного наближення для великого  $N$ . Методи, використовують поліноміальні фільтри, або використовують фільтри раціональних функцій. Ми називаємо спектральними методами, заснованими на функціональних *фільтрах функціонального обчислення*.

Заява про непередаваність формулюється на основі чутливості власного розкладання Лапласа до малих збурень у  $W$  або еквівалентно в  $\Delta$ . А саме, мале збурення  $\Delta$  може призвести до великого збурення власного розкладання  $\{\lambda_n, \phi_n\}_{n=1}^N$ , що призводить до великої зміни фільтра, визначеного (2.14). Це твердження було сказано лише для спектральних фільтрів, визначених через (2.14), для яких воно вірне. Аргумент нестабільності не доводить неможливості передачі, оскільки сучасні спектральні методи не використовують явно власні вектори і не параметризують коефіцієнти фільтра  $g_n$  через індекс  $n$  власних значень. Натомість найсучасніші методи базуються на функціональному обчисленні

та визначають коефіцієнт фільтра за допомогою функції  $g : \mathbb{R} \rightarrow \mathbb{C}$ , як  $g(\lambda_n)$ . Параметризація коефіцієнтів фільтра за  $g$  не залежить від особливостей того, як індексується спектр, і натомість представляє загальний відгук у частотній області, де значення кожної частоти визначає її відгук, а не її індекс. Нефункціональні фільтри обчислення, визначені (2.15), мале збурення  $\Delta$ , яке призводить до збурення  $\lambda_n$ , також призводить до апертурбації коефіцієнтів  $g(\lambda_n)$ . Виявляється, що збурення в  $g(\lambda_n)$  неявно компенсує нестабільність власного розкладання, а спектральні фільтри функціонального обчислення є стабільними.

Розглянемо приклади моделей.

**ChebNets.** Розглянемо нормований перекладений лапласіан  $\Delta_n - I$ . У ChebNets [85]  $g$  є поліномом, і оскільки спектр  $\Delta_n - I$  знаходиться в  $[-1, 1]$ , значення  $g$  поза  $[-1, 1]$  не впливають на фільтр  $g(\Delta_n - I)$ . Таким чином, ми можемо припустити, що  $g$  є поліномом у  $[-1, 1]$  і заповненим поза  $[-1, 1]$ , щоб отримати гладку функцію з компактним підтримкою. Легко помітити, що такий  $g$  знаходиться в  $Cay^1(\mathbb{R})$ .

Таким чином, для двох перекладених нормованих лапласів  $\Delta_n - I$  і  $\Delta'_n - I$  двох графів  $\|g(\Delta_n - I) - g(\Delta'_n - I)\| = O(\|\Delta_n - \Delta'_n\|)$ .

Загальні раціональні функції. Наведене вище твердження справедливе також для загальних раціональних функцій, якщо припустити, що спектр  $\Delta$ ,  $\Delta'$  міститься в деякій заздалегідь визначеній зоні  $[0, M]$ . Таким чином, поліноміальні фільтри та фільтри раціональної функції ARMA також можна перенести за умови рівномірно обмежених лапласів.

**CayleyNets.** CayleyNets [78] завжди можна передати, оскільки фільтр Cayley за визначенням знаходиться в  $Cay^1(\mathbb{R})$  із кінцевим розширенням.

Експериментальні результати узгоджуються з теоретичною властивістю лінійної стійкості.

### 3 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СПЕКТРАЛЬНОЇ КОНВОЛЮЦІЇ ГРАФОВИХ НЕЙРОННИХ МЕРЕЖ

В даний час методи машинного навчання активно використовуються для вирішення наукових та виробничих завдань. Класифікація – одне з груп алгоритмів. Найбільш популярною метрикою з метою оцінки ефективності роботи класифікаційного алгоритму є акуратність (accuracy). Але на незбалансованих наборах даних ця метрика часто не є показовою і навіть помилковою.

Як правило, результати вирішення проблеми двійкової класифікації позначені як позитивні та негативні. Ці рішення можуть бути представлені в матриці помилок, що містить 4 значення:

- вірно-позитивні (TP), об'єкти, які були класифіковані як позитивні і є дійсно позитивними (що належать до даного класу);

- вірно-негативні (TN) об'єкти, які були класифіковані як негативні та дійсно негативні (не належать до даного класу);

- хибно-позитивні (FP) об'єкти, які були класифіковані як позитивні, але фактично негативні;

- хибно-негативні (FN) об'єкти, які були класифіковані як негативні, але фактично позитивні.

На основі матриці помилок та її значень розраховуються різні метрики класифікаційної здатності алгоритму. Наступні зустрічаються метрики найчастіше: Акуратність, Точність (precision), Повнота (recall), F1 (F-measure).

Акуратність – широко використовується і легка для розуміння метрика. Це відношення всіх правильних прогнозів до загальної кількості всіх передбачуваних зразків. У ряді завдань акуратність може бути неінформативною. Наприклад, припустимо, що лише 3% усіх пацієнтів мають деякі захворювання. Створимо алгоритм, який позначатиме всіх пацієнтів, як здорових. У підсумку він буде помилковим тільки в 3 випадках

зі 100. Акуратність дорівнюватиме 0,97 або 97% і це висока оцінка, але насправді алгоритм нічого не робить і абсолютно марний.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \cdot \quad (3.1)$$

Точність (precision) – це частка прогнозованих позитивних результатів, які є справді вірно-позитивними результатами всім позитивно передбачених об'єктів. Іншими словами, точність дає нам відповідь на запитання "З усіх об'єктів, які класифіковані як такі, що належать класу, скільки насправді належить йому?"

$$precision = \frac{TP}{TP + FP} \cdot \quad (3.2)$$

Повнота (recall) – пропорція всіх вірно-позитивно передбачених об'єктів до загальної кількості справді позитивних. Тобто, повнота показує, скільки зразків з усіх позитивних прикладів були класифіковані правильно. Чим вище значення повноти, тим менше позитивних прикладів пропущено класифікації.

$$recall = \frac{TP}{TP + FN} \cdot \quad (3.3)$$

F-мера (F1) – зважена гармонійна середня повнота та точності. Цей показник демонструє, як багато випадків прогнозується моделлю правильно, і скільки дійсних екземплярів модель не пропустить. F-мера поєднує в собі інформацію про повноту та точність використовуваної моделі.

$$F1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \quad (3.4)$$

Акуратність активно застосовується у дослідженнях, але, як правило, не існує сама по собі, а йде у поєднанні з іншими метриками. У більшості випадків показники якості: повнота, точність і F-мера також наводяться разом для отримання повнішої картини про класифікуючу здатність моделі.

Нажаль більшість публічних датасетів не має таких характеристик, що б дозволили продемонструвати здатність спектральної конволюції виділяти структурні особливості графа, та враховувати їх при навчанні та роботі. В той час як саме такі структурні особливості проявляються в дуже складних системах на виробництві та мають ключове значення для передбачень у графових нейронних мережах.

Тож зробимо синтетичний набір даних, який може бути поєднані у граф. Для простоти обмежимо розмірність вхідних даних до 1 значення. Для виконання задачі класифікації нам знадобиться принаймні 2 класи. Згенеруємо 100 екземплярів по 50 кожного класу з нормально розподіленим значенням. Але середнє значення половини екземплярів одного з класів нехай буде більшим за інші (рисунок 3.1).

Для наглядної картини ми можемо уявити, що це задача ідентифікування спільнот. Нехай дані будуть людьми, класи будуть спільнотами-прихильниками певного політичного напрямку, а розмір, або значення самих даних – дохід цих людей. Пізніше ми додамо зв'язки (ребра графа) між людьми для аналізу графових даних.

Тепер поставимо задачу класифікувати політичні погляди людей за тими даними, що в нас є – зараз це доходи. Це буде задача класифікації вузлів графа. Нехай модель навчається на окремих графах однакової розмірності. Таким чином ми зможемо перевірити не тільки працездатність моделі, але ще й здатність параметрів навченої моделі переноситися на інші графи и бути стабільною до збурень самого графу та його структури. Для

цього нам необхідно створити не просто один граф, а створити алгоритм випадкової генерації таких графів, де можна задавати різну кількість даних для кожного класу, різну кількість даних збільшеного розміру, різне співвідношення інтенсивності зав'язків всередині та між класами, та інші параметри.

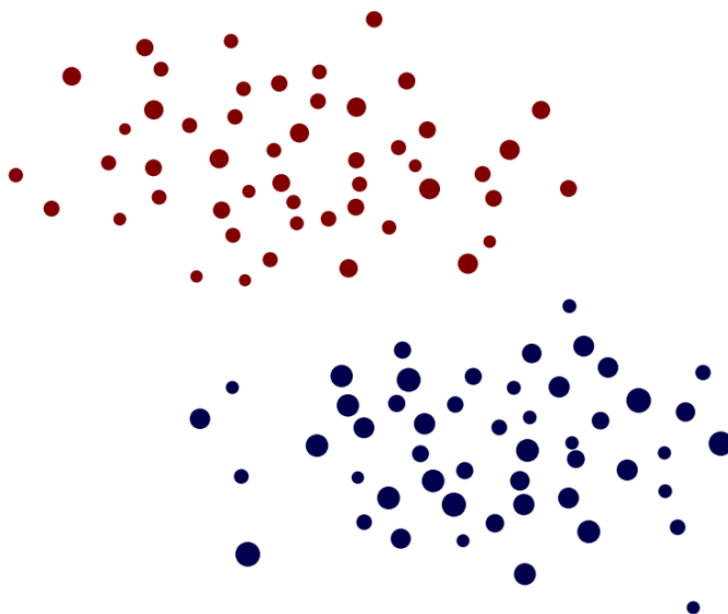


Рисунок 3.1 – 2 класи – крапки різними кольорами, та розмір, що відповідає вхідному значенню. Половина синіх крапок – у середньому більша за всі інші. Положення у просторі на даний момент не має значення.

Будемо навчатися в 50 епох на повному датасеті, який складається з 10 наборів даних згенерованих як описано вище. 6 наборів використаємо для тренування, а 4 для тестування нашої моделі.

Спробуємо використати модель Логістичної Регресії для цієї задачі. Оскільки тільки половина даних має більший за інших розмір (або дохід), то ми навряд отримаємо від цієї моделі акуратність вищу за 75 відсотків (рисунок 3.2).

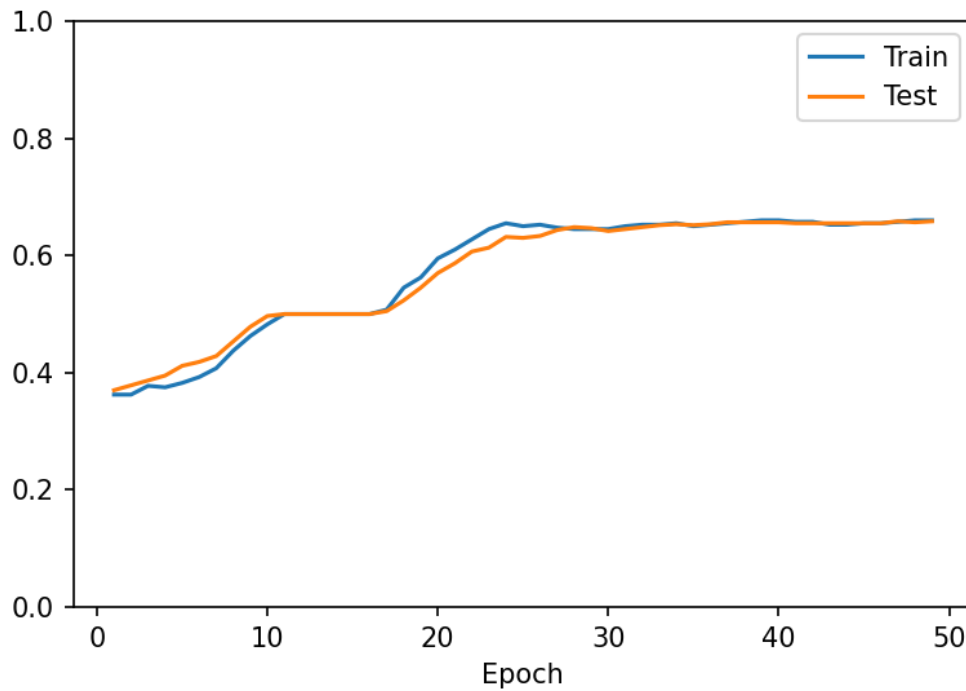


Рисунок 3.2 – Акуратність логістичної регресії досягла 66%

Спробуємо змінити співвідношення вхідних даних більшого розміру та проведемо експерименти з регресією. Що як таких даних більшого розміру буде не 50 відсотків на клас, а 85 або 15? (рисунок 3.3 та 3.4)

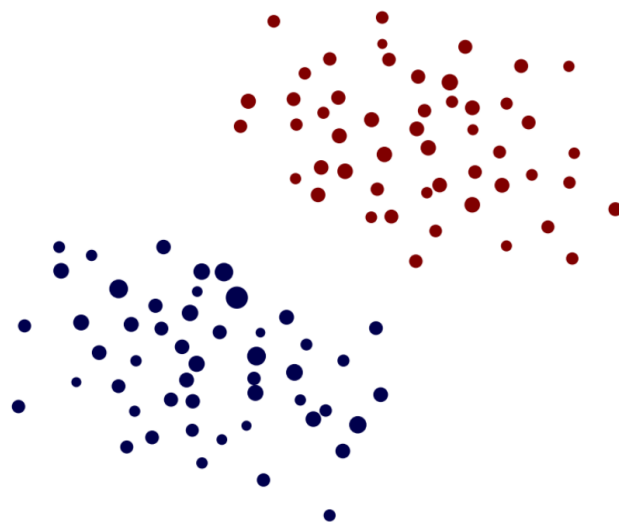


Рисунок 3.3 – 15 відсотків збільшених даних – акуратність – 53%

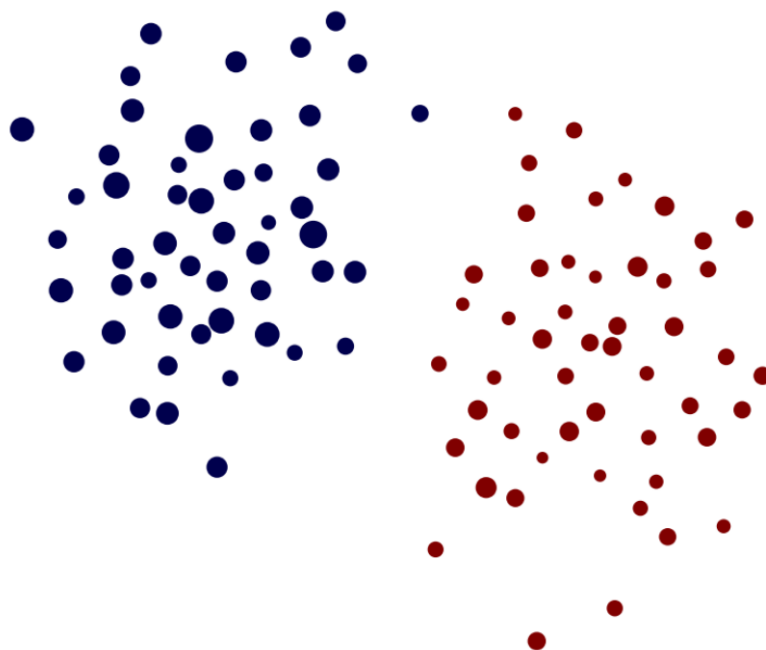


Рисунок 3.3 – 85 відсотків збільшених даних – акуратність – 78%

Також спробуємо створити дисбаланс класів (наприклад, рисунок 3.4, 3.5, 3.6), та вимірювати не лише акуратність, але також точність, повноту та метрику  $f1$ .

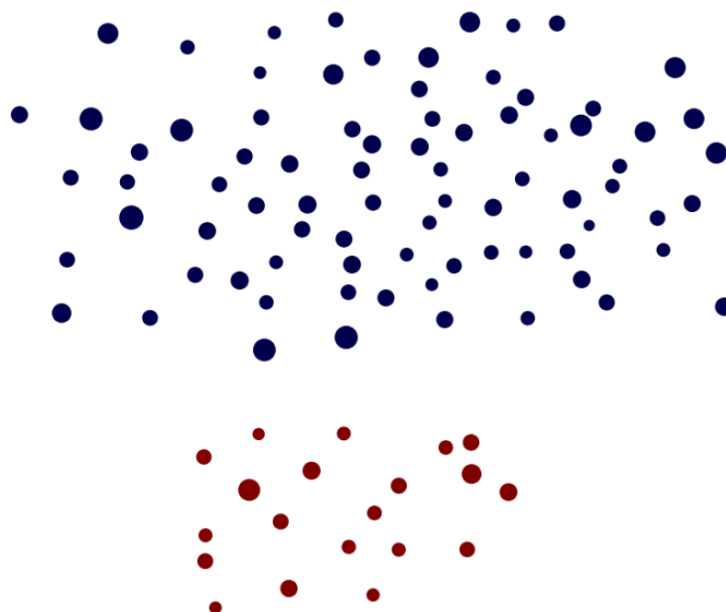


Рисунок 3.4 – Співвідношення класів 4:1 – 15 відсотків збільшених даних

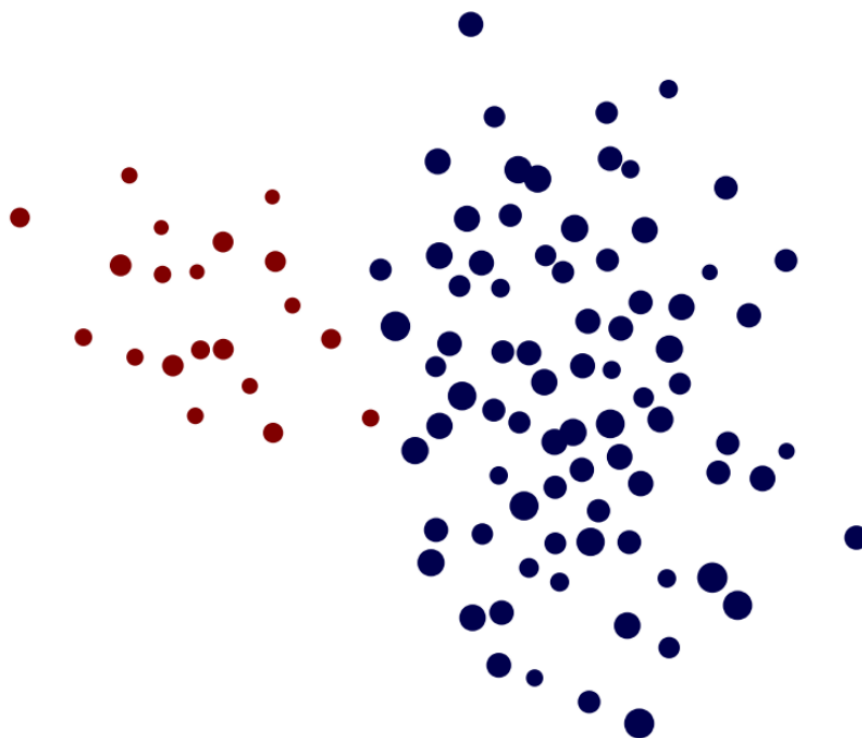


Рисунок 3.5 – Співвідношення класів 4:1 – 85 відсотків збільшених даних

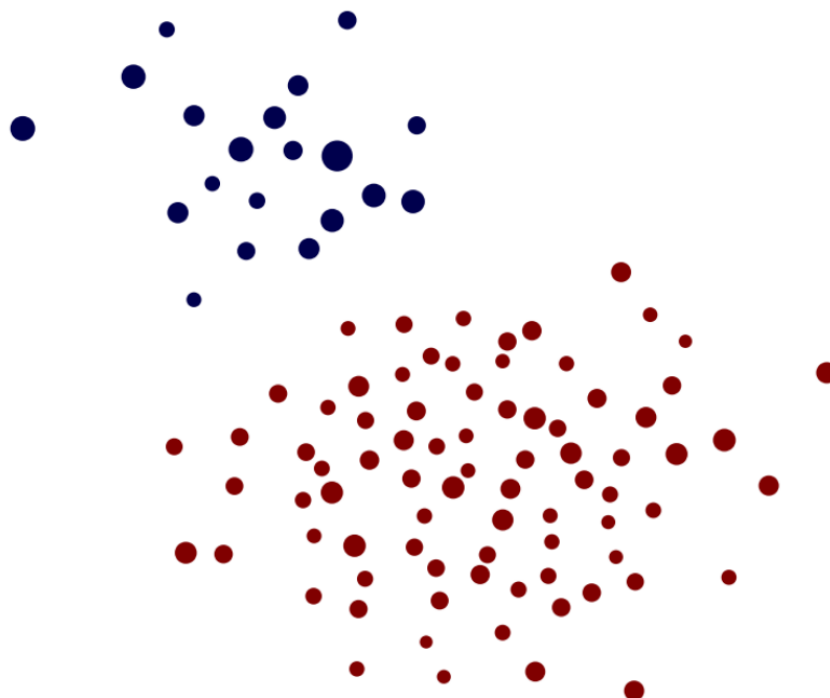


Рисунок 3.6 – Співвідношення класів 1:4 – 50 відсотків збільшених даних

Підрахуємо інші метрики моделі, такі як точність, повнота,  $f1$  (таблиця 3.1).

Таблиця 3.1 – Метрики логістичної регресії

Баланс класів	Кількість даних	Акуратність	Повнота	Точність	F1
1:1	50%	0.67	0.71	0.66	0.68
1:1	85%	0.78	0.79	0.78	0.78
1:1	15%	0.53	0.52	0.53	0.53
4:1	50%	0.79	0.08	0.3	0.13
4:1	85%	0.83	0.36	0.62	0.44
4:1	15%	0.8	0	0	0
1:4	50%	0.83	0.99	0.83	0.9
1:4	85%	0.88	0.96	0.9	0.93
1:4	15%	0.8	1	0.8	0.89

Тепер додамо до наших даних зв'язки між вершинами (рисунок 3.7). Це можуть бути, наприклад, контакти у соціальній мережі, або частота спілкування (або обміну меседжами) між цими людьми. Нехай частота зав'язків всередині класу буде у 6 разів інтенсивніше ніж між класами.

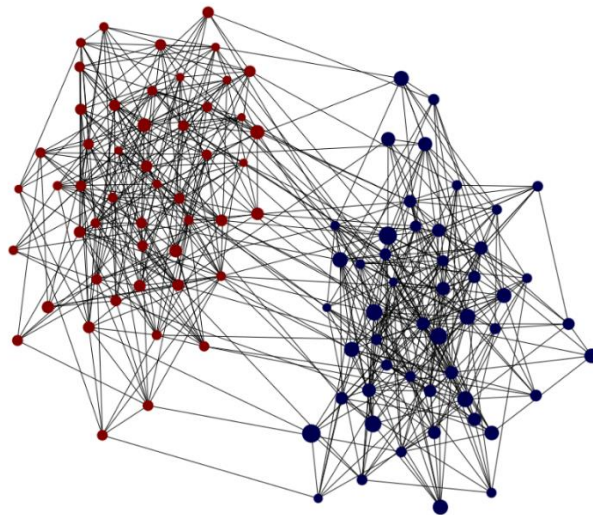


Рисунок 3.7 – Граф вхідних даних (співвідношення інтенсивності зав'язків – 6:1)

Модель регресії ніяким чином не враховувала зв'язки між вершинами. Але проста графова нейронна мережа (GCN) враховує: акуратність – 97% (рисунок 3.8), інші метрики вище за 96%.

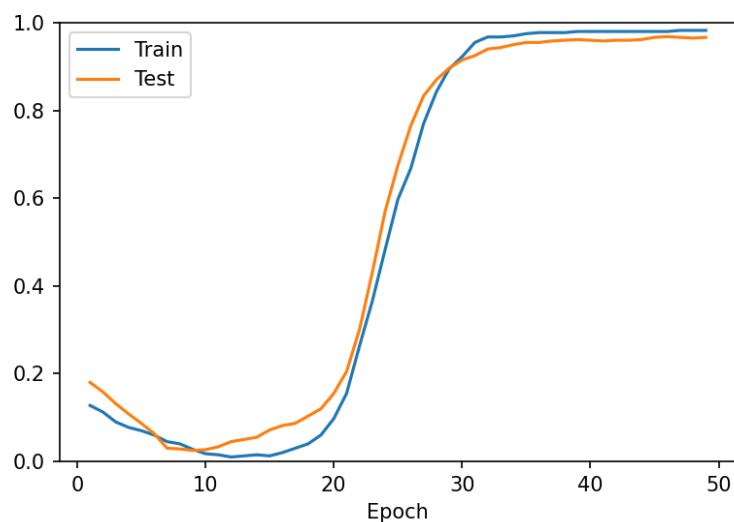


Рисунок 3.8 – Акуратність GCN досягла 97%

Тепер спробуємо змінити співвідношення інтенсивності класових та між-класових зав'язків, або ребр (рисунок 3.9).

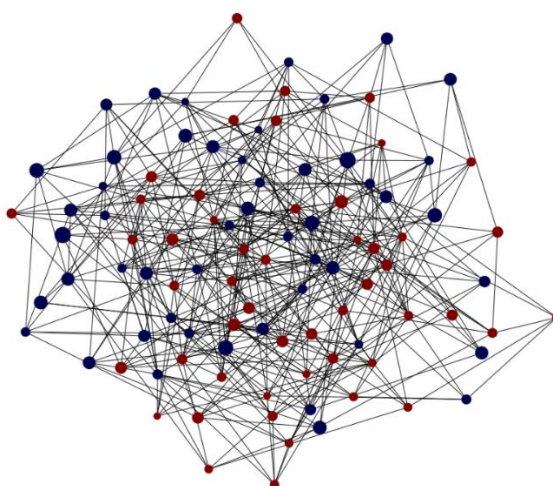


Рисунок 3.9 – Граф вхідних даних (однакове співвідношення інтенсивності зав'язків)

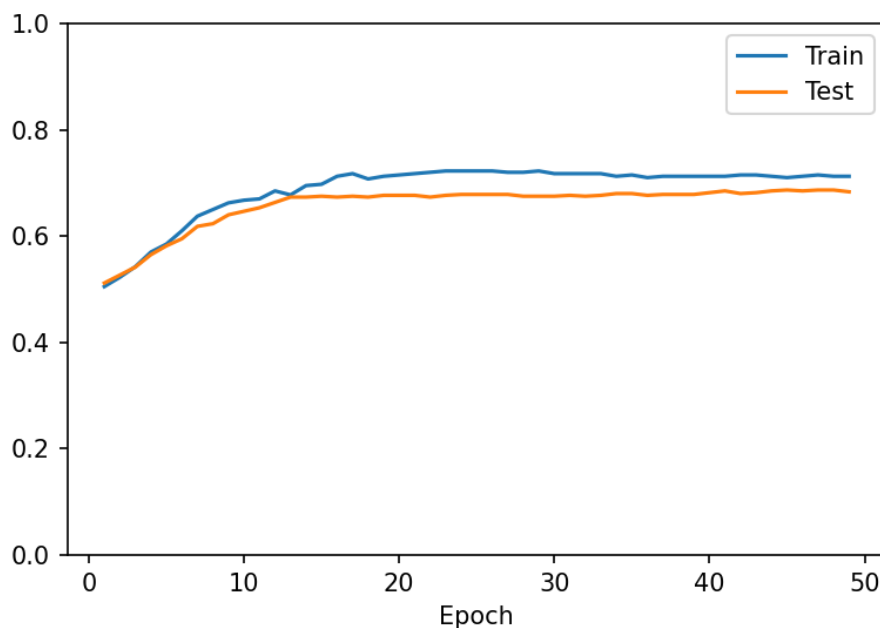


Рисунок 3.10 – GCN при однаковому співвідношенні інтенсивності зв'язків – акуратність: 0.68, повнота: 0.57, точність: 0.76, f1: 0.62 (тут і далі на графіках показані лише акуратність тренувального та тестових наборів даних)

Як бачимо (рисунок 3.10) результат роботи моделі не кращий за регресію, коли зв'язки між вершинами ніяк не допомагають виявленню класів.

Проаналізуємо результати роботи GCN, коли зв'язки між класами будуть інтенсивнішими за зв'язки всередині класу (рисунок 3.11). Наприклад, якщо ребра графу показують інтенсивність переписки, то цілком можлива ситуація, коли більше переписуються опоненти у поглядах ніж люди з однаковими поглядами.

Тепер зв'язки графу не допомагають, а навіть заважають навчанню і модель показує гірші результати ніж звичайна регресія (рисунок 3.12).

Тепер спробуємо на тих самих даних використати іншу модель – спектральну конволюцію (згортку) графових нейронних мереж.

Для співвідношення інтенсивності зав'язків – 6:1, як і GCN модель показує дуже хороші результати (рисунок 3.13).

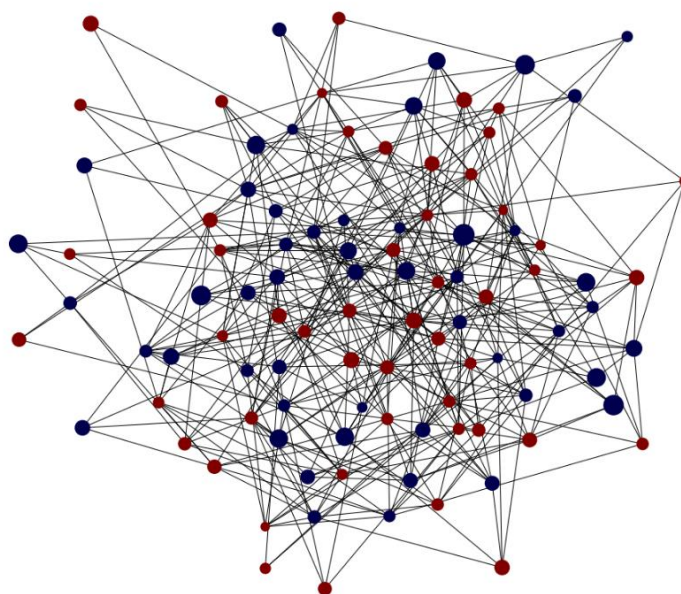


Рисунок 3.11 – Граф вхідних даних (співвідношення інтенсивності зав'язків – 1:6 у бік між-класових зав'язків )

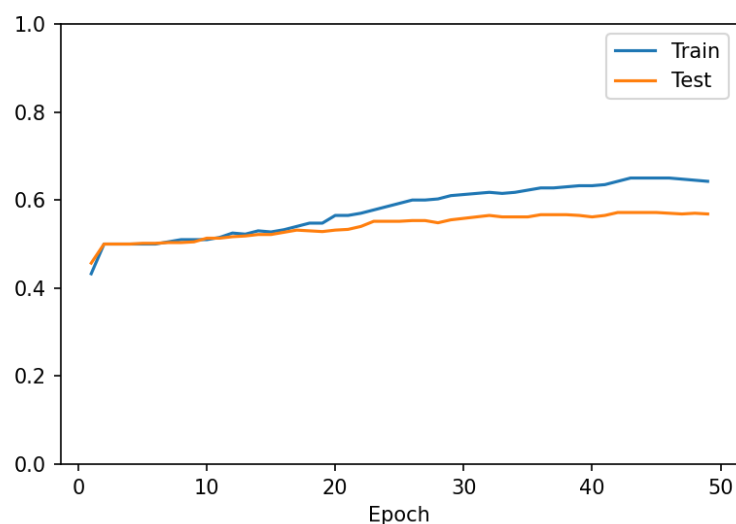


Рисунок 3.12 – GCN (співвідношення інтенсивності зав'язків – 1:6 у бік між-класових зав'язків) акуратність: 0.57, повнота: 0.35, точність: 0.64, f1: 0.42

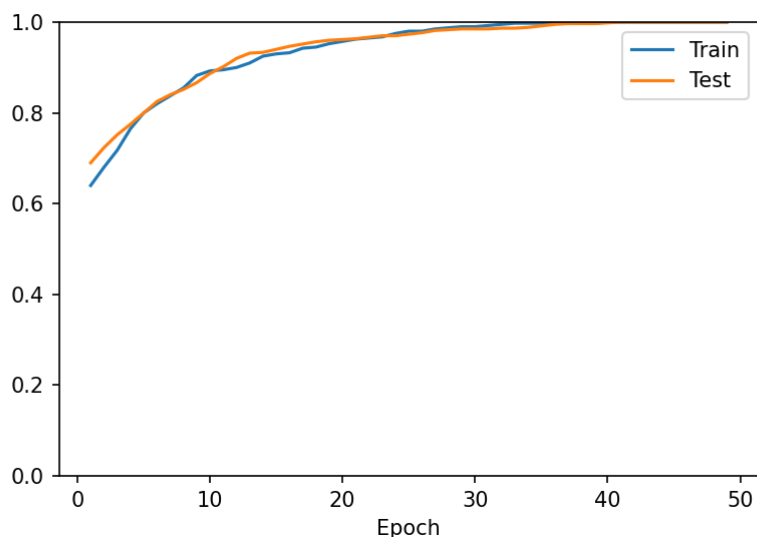


Рисунок 3.13 – ChebNet, співвідношення інтенсивності зав'язків – 6:1, всі метрики досягли значення: 1.0

Спробуємо однакову інтенсивність зав'язків.

На рисунку 3.14 бачимо, що модель не навчилася повністю, тож спробуємо збільшити кількість епох (рисунок 3.15).

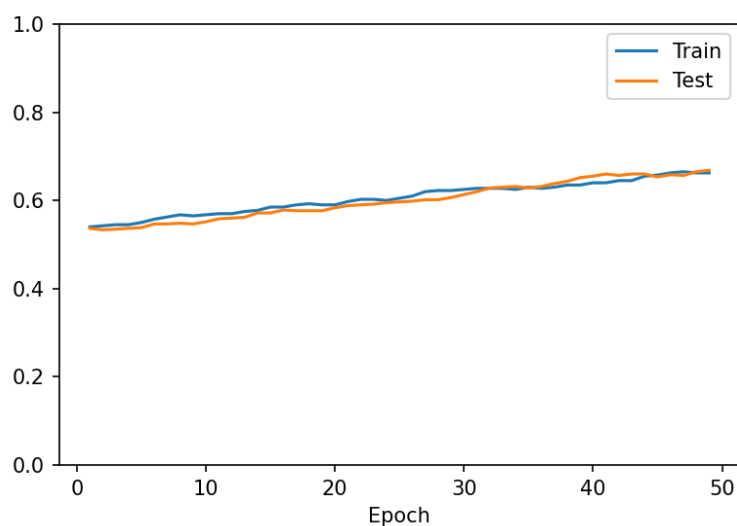


Рисунок 3.14 – ChebNet (однакове співвідношення інтенсивності зав'язків)  
50 епох – акуратність: 0.67, повнота: 0.71, точність: 0.65, f1: 0.68

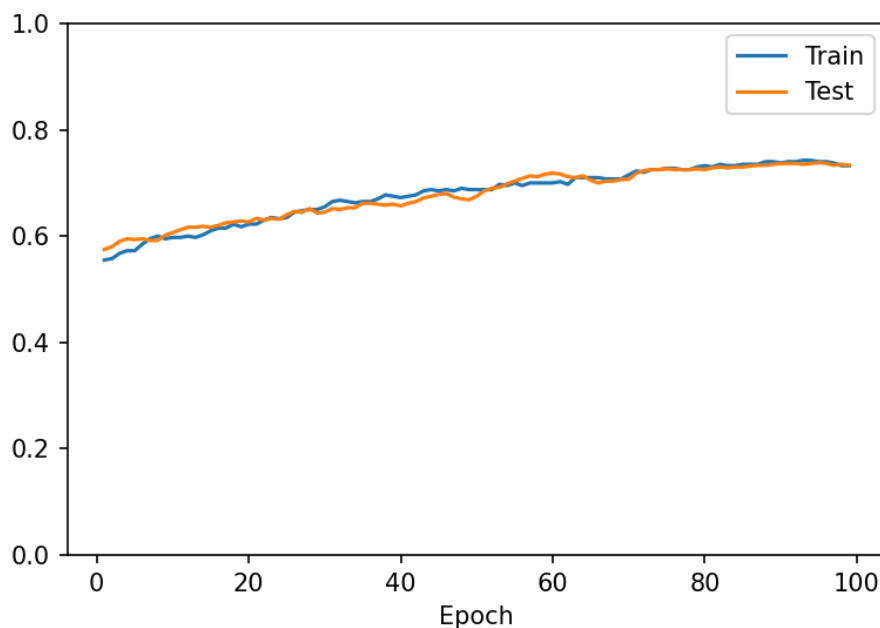


Рисунок 3.15 – ChebNet (однакове співвідношення інтенсивності зв'язків)  
100 епох – акуратність: 0.73, повнота: 0.72, точність: 0.74, f1: 0.73

Як бачимо модель ChebNet показує кращі результати.

А тепер порівняємо результати моделі для даних з співвідношенням інтенсивності зв'язків – 1:6 у бік між-класових зв'язків, де модель GCN показували показувала результати гірші за регресію.

Як бачимо на рисунку 3.16, метрики стали кращими за GCN модель і навіть кращими за регресію, з даними з однаковим співвідношенням інтенсивності зв'язків, хоча GCN модель – навпаки, погіршала результати з переходом до таких даних. Це пояснюється тим, що GCN здатна працювати лише по принципу low-pass фільтру, тобто агрегує значення сусідніх вершин. У той час, ChebNet знаходить найбільш ефективну комбінацію фільтрів, які можуть бути схожі і на high-pass фільтри. Розподіл даних, коли співвідношення інтенсивності зв'язків зміщене у бік між-класових зв'язків, як раз схоже на розподіл який робить high-pass фільтр, тому ChebNet і показує кращі результати.

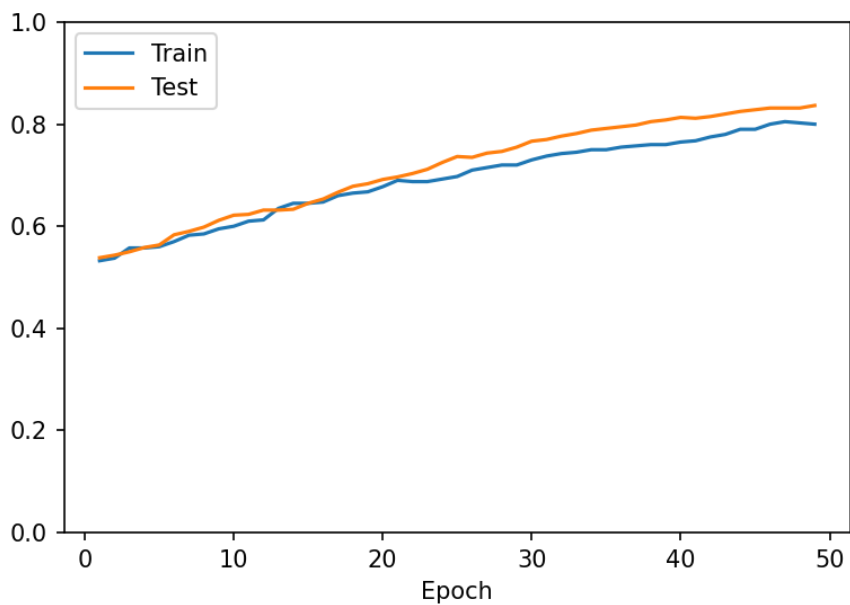


Рисунок 3.16 – ChebNet співвідношення інтенсивності зв'язків – 1:6, акуратність: 0.83, повнота: 0.84, точність: 0.84, f1: 0.84

Тапер розглянемо як GCN та ChebNet реагують на дисбаланс класів. Розглянемо випадок дисбалансу класів 4:1 (рисунок 3.17).

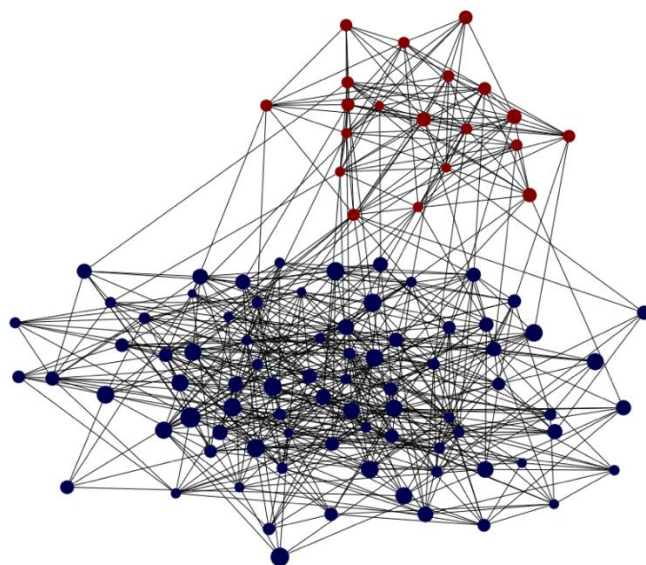


Рисунок 3.17 – Граф вхідних даних з дисбалансом класів 4:1

З таблиці 3.2 ми бачимо, що GCN працює краще ніж регресія, а ChebNet краще за GCN, особливо покращена метрика точності. Це ще раз доводить практичну застосовність моделей спектральної конволюції.

Таблиця 3.2 – Метрики різних моделей для графа з дисбалансом класів 4:1

Модель	Акуратність	Повнота	Точність	F1
Regression	0.79	0.08	0.3	0.13
GCN	0.84	0.2	0.5	0.23
ChebNet	0.91	0.55	0.99	0.64

Можна поставити питання, як саме ChebNet знаходить правильні передбачення класів нашого графу. Дослідимо, як саме працює спектральна згортка на наших даних. Ми можемо вичислити власні вектори нашого графу, та замість класів продемонструвати кольорове розподілення значень цього вектору. Це допоможе зрозуміти як саме працює модель та чому вона показує кращі метрики саме на цих даних.

Для того щоб приклад був більш наглядним трохи зменшимо різницю співвідношення інтенсивності зав'язків з 6:1 до 4:1 (рисунок 3.18).

Як бачимо на рисунку 3.19, один з власних векторів дуже добре розділяє «кластери» графу. Саме така властивість спектральної конволюції дозволяє виводити інформацію про структуру графа та допомагати моделі робити правильні передбачення.

Однією з причин чому ChebNet змогла надати, хоча і кращі, але не дуже високі результати, на графі з однаковим співвідношенням інтенсивності зав'язків (рисунок 3.15), є те, що в такому графі губиться будь яка структура. Змінимо трохи структуру вхідного графу так, щоб зв'язки між основними кластерами були не випадкові, а проходили через проміжний кластер при однаковому співвідношенні інтенсивності зав'язків. Потім подивимось які ще розподіли власних векторів можуть допомогти графовим нейронним мережам зі спектральною конволюцією.

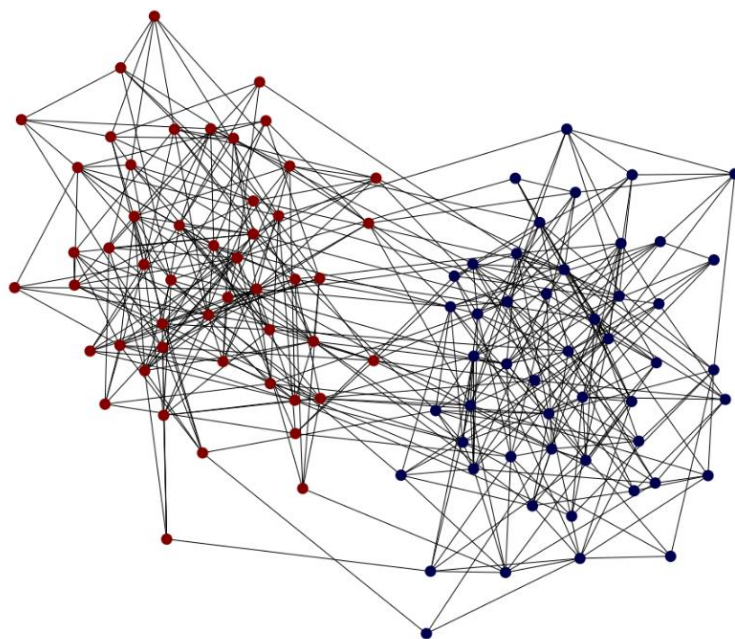


Рисунок 3.18 – Граф вхідних даних – співвідношення інтенсивності зав'язків – 4:1 (розмір вузлів однаковий)

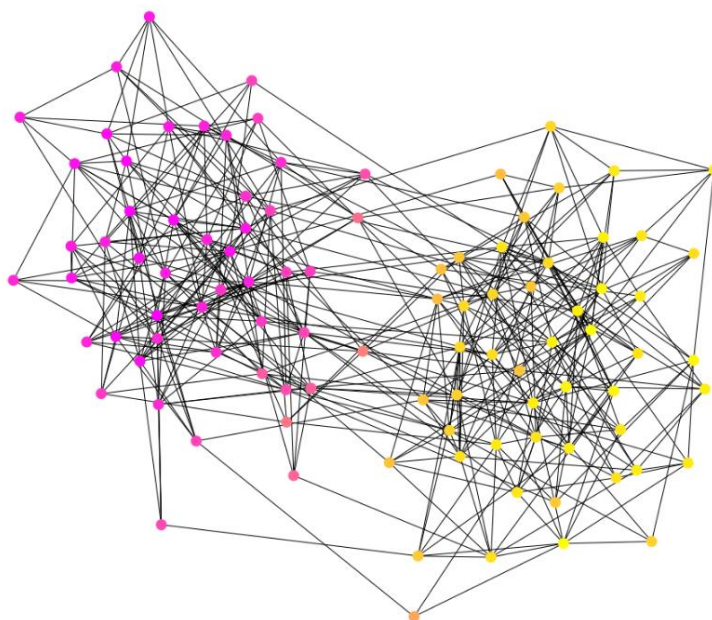


Рисунок 3.19 – Той самий граф – кольорове розподілення значень другого власного вектору (low-pass)

Додамо кластер меншого розміру між двома кластерами так, щоб всі ребра між ними проходили через цей кластер. Це придасть більше структурності до наших даних, та дозволить краще візуалізувати тенденцію будування різних власних векторів.

На рисунку 3.20 та 3.21 бачимо, як різні власні вектори здатні виділяти кластери різного розміру незважаючи на однакове співвідношення інтенсивності зав'язків між кластерами та всередині них, що і пояснює отримані результати якості роботи спектральної конволюції графових нейронних мереж.

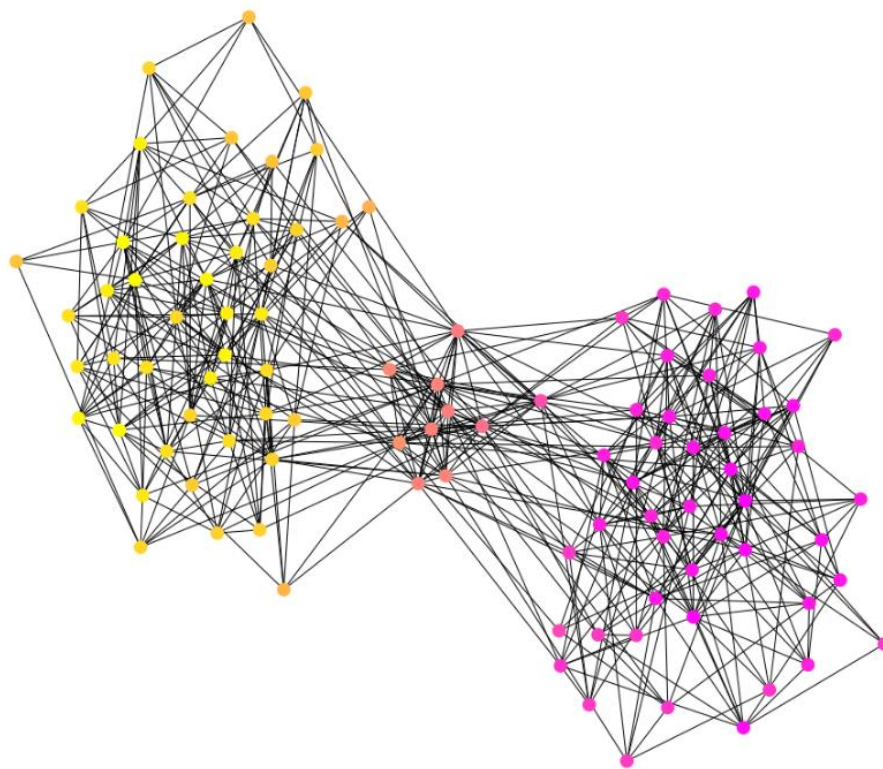


Рисунок 3.20 – Кольорове розподілення значень другого власного вектору

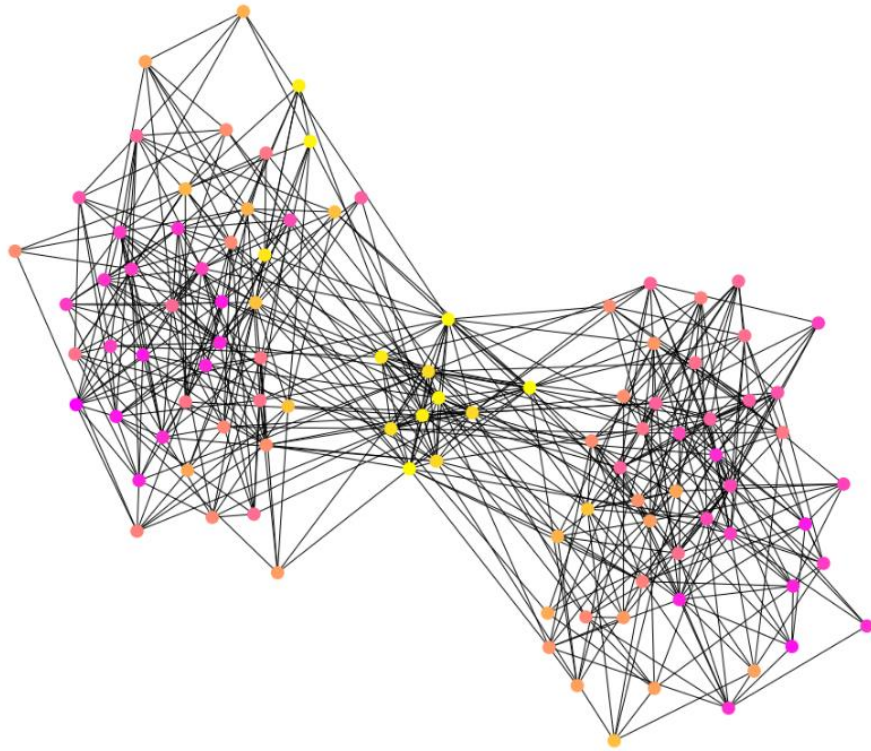


Рисунок 3.21 – Кольорове розподілення значень  
третього власного вектору

Таким чином, ми впевнилися у якості роботи моделі з архітектурою спектральної згортки, порівняли її зі звичайною графовою згорткою та з логістичною регресією, дослідили які саме характеристики вхідних даних можуть бути виявлені спектральною конволюцією та впевнилися у стабільності параметрів системи до збурень у структурі даних вхідного графа.

## ВИСНОВКИ

В роботі представлено результати, що відповідають меті дослідження, а саме – розробці графової нейронної мережі з, що включає шари спектральної конволюції (згортки) для вирішення задачі класифікації графових даних. Проведені дослідження дозволили зробити такі висновки:

1. розглянуто завдання класифікації графових даних з різними властивостями співвідношення інтенсивності зав'язків в середині та між класами;

2. проведено порівняльний аналіз якості роботи з різних моделей: логістичної регресії, простої графової нейронної мережі та графової мережі зі спектральною конволюцією;

3. доведено переваги якості роботи спектральної конволюції графових нейронних мереж для графових структурних вхідних даних;

4. виявлено та візуалізовано специфіку обробки структурної інформації вхідних даних спектральною згортковою мережею.

Аналіз показав стійкість спектральних фільтрів до збурень у структурі вхідного графу, що підтверджує конкурентоспроможні властивості переносимості спектральних методів на практиці. Також доведено, що ChebNet перевершує звичайні просторові методи конволюції графових нейронних мереж, особливо в умовах структурності вхідних даних (графу).

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Backstrom L., Leskovec J. Supervised random walks: predicting and recommending links in social networks. In: WSDM. New York: ACM; 2011. P. 44 – 635.
2. Akoglu L., Tong H., Koutra D. Graph based anomaly detection and description: a survey. *Data Min Knowl Discov.* 2015.
3. MathSciNet Article Google Scholar
4. Zhang S., Zhou D., Yildirim M.Y., Alcorn S., He J., Davulcu H., Tong H. Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In: SDM. Philadelphia: SIAM; 2017. P. 8 – 570.
5. Chapter Google Scholar
6. Zhou D., Zhang S., Yildirim M.Y., Alcorn S., Tong H., Davulcu H., He J. A local algorithm for structure-preserving graph cut. In: KDD. New York: ACM; 2017. P. 64 – 655.
7. Boccaletti S., Latora V., Moreno Y., Chavez M., Hwang D-U. Complex networks: structure and dynamics. *Phys Rep.* 2006.
8. Tenenbaum J.B., De Silva V., Langford J.C. A global geometric framework for nonlinear dimensionality reduction. *Science.* 2000.
9. Roweis S.T., Saul L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science.* 2000.
10. Belkin M., Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: NIPS. 2002. P. 91 – 585.
11. Perozzi B., Al-Rfou R., Skiena S. Deepwalk: online learning of social representations. In: KDD. New York: ACM; 2014. P. 10 – 701.
12. Grover A., Leskovec J. node2vec: scalable feature learning for networks. In: KDD. New York: ACM; 2016. P. 64 – 855.
13. Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014. P. 7 – 580.

14. Gehring J., Auli M., Grangier D., Dauphin Y.N. A convolutional encoder model for neural machine translation. 2016.
15. Shuman D.I., Narang S.K., Frossard P., Ortega A., Vandergheynst P. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process Mag.* 2013.
16. Bruna J., Zaremba W., Szlam A., LeCun Y. Spectral networks and locally connected networks on graphs. 2013.
17. Defferrard M., Bresson X., Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: *NIPS*. 2016. P. 52 – 3844.
18. Garcia V., Bruna J. Few-shot learning with graph neural networks. 2017.
19. Kampffmeyer M., Chen Y., Liang X., Wang H., Zhang Y., Xing EP. Rethinking knowledge graph propagation for zero-shot learning. 2018.
20. Johnson J., Gupta A., Fei-Fei L. Image generation from scene graphs. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. P. 28 – 1219.
21. Qi C.R., Su H., Mo K., Guibas L.J. Pointnet: deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. P. 60 – 652.
22. Shen Y., Feng C., Yang Y., Tian D. Neighbors do help: deeply exploiting local structures of point clouds. 2017.
23. Wang Y., Sun Y., Liu Z., Sarma S.E., Bronstein M.M., Solomon J.M. Dynamic graph cnn for learning on point clouds. 2018.
24. Verma N., Boyer E., Verbeek J. Feastnet: feature-steered graph convolutions for 3d shape analysis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. P. 606 – 2598.
25. Yao L., Mao C., Luo Y. Graph convolutional networks for text classification. 2018.

26. Qian Y., Santus E., Jin Z., Guo J., Barzilay R. Graphie: a graph-based framework for information extraction. 2018.
27. Zhang Y, Qi P, Manning C.D. Graph convolution over pruned dependency trees improves relation extraction. 2018.
28. Zhang N., Deng S., Sun Z., Wang G., Chen X., Zhang W., Chen H. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. 2019.
29. Liu X., Luo Z., Huang H. Jointly multiple events extraction via attention-based graph information aggregation. 2018.
30. Nguyen T.H., Grishman R. Graph convolutional networks with argument-aware pooling for event detection. In: Thirty-second AAAI conference on artificial intelligence. 2018.
31. Marcheggiani D., Bastings J., Titov I. Exploiting semantics in neural machine translation with graph convolutional networks. 2018.
32. Strubell E., McCallum A. Dependency parsing with dilated iterated graph cnns. 2017.
33. Gilmer J., Schoenholz S.S., Riley P.F., Vinyals O., Dahl G.E. Neural message passing for quantum chemistry. In: Proceedings of the 34th international conference on machine learning, vol. 70. 2017. P. 72 – 1263.
34. Henrion I., Brehmer J., Bruna J., Cho K., Cranmer K., Louppe G., Rochette G. Neural message passing for jet physics. 2017.
35. Qu H., Gouskos L. Particlenet: jet tagging via particle clouds. 2019.
36. Choma N., Monti F., Gerhardt L., Palczewski T., Ronaghi Z., Prabhat P., Bhimji W., Bronstein M., Klein S., Bruna J. Graph neural networks for icecube signal classification. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA). New York: IEEE; 2018. P. 91 – 386.
37. Mrowca D., Zhuang C., Wang E., Haber N., Fei-Fei L.F., Tenenbaum J., Yamins D.L. Flexible neural representation for physics prediction. In: Advances in neural information processing systems. 2018. P. 810 – 8799.

38. Duvenaud D.K., Maclaurin D., Iparraguirre J., Bombarell R., Hirzel T., Aspuru-Guzik A., Adams R.P. Convolutional networks on graphs for learning molecular fingerprints. In: Advances in neural information processing systems. 2015. P. 32 – 2224.
39. Kearnes S., McCloskey K., Berndl M., Pande V., Riley P. Molecular graph convolutions: moving beyond fingerprints. *J Comput Aided Mol Des.* 2016.
40. Li X., Yan X., Gu Q., Zhou H., Wu D., Xu J. Deepchemstable: chemical stability prediction with an attention-based graph convolution network. *J Chem Inf Model.* 2019.
41. Zitnik M., Agrawal M., Leskovec J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics.* 2018.
42. Feinberg E.N., Sur D., Wu Z., Husic B.E., Mai H., Li Y., Sun S., Yang J., Ramsundar B., Pande VS. Potentialnet for molecular property prediction. 2018.
43. Bruna J., Li X. Community detection with graph neural networks. 2017.
44. Harada S., Akita H., Tsubaki M., Baba Y., Takigawa I., Yamanishi Y., Kashima H. Dual convolutional neural network for graph of graphs link prediction. 2018.
45. Chen J., Xu X., Wu Y., Zheng H. Gc-lstm: graph convolution embedded lstm for dynamic link prediction. 2018.
46. Qiu J., Tang J., Ma H., Dong Y., Wang K., Tang J. Deepinf: social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. New York: ACM; 2018. P. 9 – 2110.
47. Vijayan R., Mohler G.. Forecasting retweet count during elections using graph convolution neural networks. In: 2018 IEEE 5th international conference on data science and advanced analytics (DSAA). New York: IEEE; 2018. P. 62 – 256.
48. Monti F., Frasca F., Eynard D., Mannion D., Bronstein MM. Fake news detection on social media using geometric deep learning. 2019.

49. Krizhevsky A., Sutskever I., Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. 2012.

50. Wang J., Yang Y., Mao J., Huang Z., Huang C., Xu W.. Cnn-rnn: a unified framework for multi-label image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 94 – 2285.

51. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014.

52. Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2015.

53. Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. P. 40 – 3431.

54. Chen L.-C., Papandreou G., Kokkinos I., Murphy K, Yuille AL. Semantic image segmentation with deep convolutional nets and fully connected crfs. 2014.

55. Niepert M., Ahmed M., Kutzkov K. Learning convolutional neural networks for graphs. In: International conference on machine learning. 2016. P. 23 – 2014.

56. Gao H., Wang Z., Ji S. Large-scale learnable graph convolutional networks. In: KDD. New York: ACM; 2018. P. 24 – 1416.

57. Chang J., Gu J., Wang L., Meng G., Xiang S., Pan C. Structure-aware convolutional neural networks. In: Advances in neural information processing systems. 2018. P. 11 – 20.

58. Hammond D.K., Vandergheynst P., Gribonval R. Wavelets on graphs via spectral graph theory. Appl Comput Harmon Anal. 2011. P. 50 – 129.

59. Atwood J., Towsley D. Diffusion-convolutional neural networks. In: NIPS. 2016.

60. Kipf T.N., Welling M. Semi-supervised classification with graph convolutional networks. 2016.

61. Fey M., Lenssen J.E., Weichert F., Müller H. Splinecnn: Ffast geometric deep learning with continuous b-spline kernels. In: CVPR. 2018. P. 77 – 869.
62. Jia X., De Brabandere B., Tuytelaars T., Gool L.V. Dynamic filter networks. In: Advances in neural information processing systems, 2016. P. 75 – 667.
63. Hamilton W., Ying Z., Leskovec J. Inductive representation learning on large graphs. NIPS, 2017. P. 34 – 1024.
64. Li Q., Han Z., Wu X.-M. Deeper insights into graph convolutional networks for semi-supervised learning, Thirty-second AAAI conference on artificial intelligence, 2018.
65. Taubin G. A signal processing approach to fair surface design. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques. New York: ACM, 1995. P. 8 – 351.
66. Xu K., Li C., Tian Y., Sonobe T., Kawarabayashi K., Jegelka S. Representation learning on graphs with jumping knowledge networks, 2018.
67. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. P. 8 – 770.
68. Velickovic P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y. Graph attention networks, 2017.
69. Chung F. Spectral graph theory. American Mathematical Society, 1997.
70. Shuman D. I., Narang S. K., Frossard P., Ortega A., Vandergheynst P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine, vol.30, no. 3, 2013. P. 83 – 98.
71. Bruna J., Zaremba W., Szlam A., LeCun Y. Spectral networks and locally connected networks on graphs, 2013.
72. Henaff M., Bruna J., LeCun Y. Deep convolutional networks on graph-structured data, 2015.

73. Defferrard M., Bresson X., Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 2016. P. 3844 – 3852.

74. Levie R., Monti F., Bresson X., Bronstein M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, vol. 67, no. 1, Jan 2019, P. 97 – 109.

75. Kearnes S., McCloskey K., Berndl M., Pande V., Riley P. Molecular graph convolutions: moving beyond finger-prints. *Journal of computer-aided molecular design*, vol. 30, no. 8, 2016. P. 595 – 608.

76. Wu Z., Pan S., Chen F., Long G., Zhang C., Yu P. S. A comprehensive survey on graph neural networks, 2019.

77. Atwood J., Towsley D. Diffusion-convolutional neural networks. *Advances in Neural Information Processing Systems*, 2016. P. 1993 – 2001.

78. Velickovic P., Cucurull G., Casanova A., Romero A., Lio P., Y. Bengio. Graph attention networks. *International Conference on Learning Representations (ICLR)*, 2018.

79. Yang Z., Cohen W. W., Salakhutdinov R. Revisiting semi-supervised learning with graph embeddings. *Proceedings of the 33rd International Conference on Machine Learning, ICML'16*, 2016.

80. Monti F., Boscaini D., Masci J., Rodola E., Svoboda J., Bronstein M. M. Geometric deep learning on graphs and manifolds using mixture model cnns. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. P. 5115 – 5124.

81. Fey M., Lenssen J. E., Weichert F., and Muller H. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. P. 869 – 877.

82. Xu B., Wang N., Chen T., Li M. Empirical evaluation of rectified activations in convolutional network, 2015.

83. Monti F., Shchur O., Bojchevski A., Litany O., Gunnemann S., Bronstein M. M.. Dual-primal graph convolutional networks, 2018.

84. Hammond D. K., Vandergheynst P., Gribonval R. Waveletson graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, 2011. P. 129 – 150.

85. Kipf T. N., Welling M. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.

86. Abu-El-Haija S., Perozzi B., Kapoor A., Harutyunyan H., Alipourfard N., Lerman K., Steeg G. V., Galstyan A.. Mixhop: Higher-order graph convolution architectures via sparsified neighborhood mixing. *International Conference on Machine Learning (ICML)*, 2019.

87. Chollet F. Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. P. 1251 – 1258.

88. Sandler M., Howard A. G., Zhu M., Zhmoginov A., Chen L. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR2018, Salt Lake City, UT, USA, June 18-22, 2018*. P. 4510 – 4520.

89. Bronstein M. M., Bruna J., LeCun Y., Szlam A., Vandergheynst P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, vol. 34, no. 4, July 2017. P. 18 – 42.

90. Wu Z., Pan S., Chen F., Long G., Zhang C., Yu P. S.. A comprehensive survey on graph neural networks. 2019.

91. Gori M., Monfardini G., Scarselli F.. A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, vol. 2, July 2005. P. 729 – 734.

92. Scarselli F., Gori M., Tsoi A. C., Hagenbuchner M., Monfardini G. The graph neural network model. *IEEE Transactions on Neural Networks*, vol. 20, no. 1, Jan 2009. P. 61 – 80.