University of Padova

Department of Mathematics "Tullio Levi-Civita"

Ph.D. School in Brain, Mind & Computer Science
Computer Science and Innovation for Societal Challenges
XXXVII Series

Leveraging Deep Learning in Human Resources: A Systematic Investigation of Graph Neural Networks for Candidate-Job Matching

Supervisor
Prof. Alessandro Sperduti

Co-supervisor
Prof. Anna Spagnolli

Ph.D. Candidate
Paolo Frazzetto

Academic Years
2021-2024

Ringraziamenti

Giungere al completamento di una tesi di dottorato rappresenta un momento straordinario, che segna il culmine di un percorso tanto intenso quanto difficile da descrivere a parole. È la realizzazione di un sogno ambizioso, un traguardo che non avrei potuto raggiungere senza il sostegno e il contributo di numerose persone, a cui desidero esprimere la mia più sincera gratitudine.

Desidero innanzitutto ringraziare il mio supervisore Prof. Alessandro Sperduti, per avermi guidato in questo percorso con pazienza e dedizione. La sua capacità di accendere la curiosità scientifica, i suoi consigli indispensabili e le conoscenze trasmesse sono stati fondamentali per intraprendere questa carriera nel mondo della ricerca.

Vorrei esprimere la mia gratitudine ad Amajor, in particolare a Flavia, Eros, Mattia, Andi e tutto il team, per aver creduto in questo progetto fin dall'inizio. Il loro supporto e la loro fiducia hanno reso possibile trasformare un'idea ambiziosa in una realtà concreta.

Un sentito ringraziamento va ai colleghi dell'Università di Padova: Uzair, Luca, Davide, Flavio, Alessandro, Tommaso ed Enrico, per aver trascorso insieme questo percorso di dottorato, rendendo più leggere anche le giornate più intense.

A special thanks goes to Prof. Barbara Hammer and the research group at Bielefeld, where I found not only a stimulating professional environment but also a warm welcome that made my stay an unforgettable experience. In particular, I would like to thank Riza, Fabian, Luca, and all the members of the group with whom I shared pleasant moments and discussions.

Ai miei genitori, Lia e Mario, va un ringraziamento che le parole non possono esprimere pienamente. Mi avete sempre ispirato a inseguire i miei sogni e a lavorare sodo, vi sento vicini anche quando sono lontano. Estendo la mia gratitudine ai miei familiari: Riccardo, Silvia, Teresa, Learco, Benito ed Elviretta, che hanno sempre creduto in me e in quello che faccio.

Un pensiero sentito va a Elena e Luca, che durante questo percorso di studi mi hanno fatto sentire sempre accolto e supportato in ogni modo possibile. La loro disponibilità e il loro sostegno sono stati per me un grande aiuto, li ringrazio per la loro generosità e vicinanza.

I also want to thank Eddy, who taught me the importance of not taking anything for granted and to think for myself.

Infine, il ringraziamento più profondo e sentito va a Ludovica, che ha illuminato ogni giorno di questo percorso e mi ha motivato a dare il meglio. Il suo sostegno nei momenti più difficili, la sua presenza nelle gioie più grandi e la sua capacità di dare un significato ad ogni istante sono indescrivibili. Senza di lei, nulla di tutto questo sarebbe stato possibile. Essere con te è un dono di cui sarò eternamente grato, e guardo con immensa gioia ed emozione al nostro futuro, pronti a scrivere insieme le prossime pagine della nostra storia. Grazie per tutta la pazienza e comprensione che hai avuto e grazie per avermi dato forza e prospettiva quando ne avevo più bisogno.
Grazie davvero.

I am grateful to many other people who have crossed my path during these years of doctoral studies. Although I cannot name them all, each one has contributed in a unique way to this journey, leaving an indelible mark on my personal and professional growth.

Padova, Dicembre 2024 *Paolo Frazzetto*

“Humans should do zero percent of the hard and boring work, computers the rest.”
— Jürgen Schmidhuber

Abstract

This dissertation explores the application of Graph Neural Networks (GNNs) to Human Resource Management (HRM), focusing on the challenge of candidate-job matching (CJM). Through theoretical contributions and applied studies, it demonstrates the potential of graph-based approaches to enhance HR analytics and personnel selection. The research addresses key questions on representing HR data using deep learning, translating it into graph structures, identifying effective GNN architectures, and applying them to HR tasks. Novel methods are developed to convert various HR data types into graphs, including Likert-scale questionnaires, candidate profiles, and candidate-job pairs. Theoretical contributions include a topological-based aggregation for GNNs using Generative Topographic Mapping, multiplicative integration as a graph convolution operation, and an efficient algorithm for computing Shapley Interactions in GNNs. These advancements improve GNN performance and interpretability. Applied case studies leverage Large Language Models for feature extraction from CVs and job descriptions, combining them with GNNs for predictive modeling. A comprehensive pipeline is developed to process real-world HR data, construct purpose-built graphs for each candidate-job pair, and perform inductive learning for CJM. Results show that graph-based approaches outperform traditional methods in capturing complex relationships in HR data. The research highlights key considerations for applying GNNs to HR, including handling class imbalance, ensuring interpretability, and incorporating domain knowledge. This work bridges cutting-edge machine learning with real-world HR challenges, offering new perspectives for addressing modern recruitment complexities. It emphasizes the importance of keeping humans central in the recruitment process while leveraging AI to augment decision-making, pointing towards a future of more sophisticated, fair, and effective AI-driven recruitment systems.

Contents

III Applied Contributions 91

List of Figures

List of Tables

Listing of Acronyms

HR	Human Resources
HRM	Human Resource Management
CV	Curriculum Vitae or Resume
JD	Job Description
JP	Job Posting
CJM	Candidate-Job Matching
PE	Person-Environment fit
PJ	Person-Job fit

ML	Machine Learning
NN	Neural Network
GNN	Graph Neural Network
MLP	Multilayer Perceptron
GC	Graph Convolution
GCN	Graph Convolutional Network
GIN	Graph Isomorphism Network
GAT	Graph Attention Network
GTM	Generative Topographic Mapping
RBF	Radial Basis Function
MI	Multiplicative Integration
LLM	Large Language Model

XAI	Explainable AI
SV	Shapley Value
SI	Shapley Interaction
MI	Möbius Interactions

1
Introduction

The field of Human Resource Management (HRM)—the strategic management of people in an organization—is undergoing a significant transformation driven by the rapid advancements in Artificial Intelligence (AI) and Machine Learning (ML). As companies strive to attract, select, and retain top talent in an increasingly competitive landscape, there is a growing need for more sophisticated and data-driven approaches to recruitment and personnel selection. This dissertation explores the multi-disciplinary intersection of Deep Learning techniques, particularly Graph Neural Networks (GNNs), with real-world HR data to address one of the fundamental challenges in recruitment: finding the best candidate for an open job position or Candidate-Job Matching (CJM).

HR data is inherently complex, multifaceted, and often unstructured. It covers an extensive spectrum of information categories and types, such as structured data from application forms and assessment tests, semi-structured data from resumes, and unstructured text from cover letters, job descriptions, or interview transcripts. Traditional approaches to CJM often struggle to integrate and analyze this diverse data, relying primarily on keyword matching or statistical methods. These conventional techniques frequently fall short of capturing the intricate relationships within HR data, the contextual nuances of candidates' experiences and ambitions, and the subtle requirements of job roles within its broader organizational framework. In contrast, this research tries to address these challenges by leveraging graph structures and GNNs, which are particularly well-suited for such tasks due to their ability to model complex relationships and hierarchical structures. The intent is to capture and learn the rich interconnections that exist in the recruitment ecosystem, all by representing candidates, job postings, and their various attributes as nodes and edges in a graph.

This work is carried out in partnership with *Amajor SB S.p.A.*, an innovative benefit corporation based in Padua, Italy. Amajor's mission is to guide small and medium-sized enterprises in improving their business models, with a focus on entrepreneurial personal values. Their unique consulting approach, encapsulated in their *"4H Method"* (Heart, Head, Hands, and High-Value Results), emphasizes keeping people, their values, and their needs at the center of every business decision. This dissertation is the outcome of close collaboration with Amajor's team and HR recruiters—who provided irreplaceable expertise and proprietary data—and explores this boundary between HR expertise and AI capabilities. We all strongly assert that AI's role should be to enhance and support human decision-making, not to replace it. Our ultimate wish is to equip HR practitioners with richer insights and streamlined workflows, allowing them to dedicate their efforts to the intricate and human-centered elements of hiring, which require empathy, intuitive understanding, and sophisticated reasoning.

Throughout this research endeavor, we seek to answer several key questions:

1. How can real-world HR data, including candidate profiles, job descriptions, and assessment questionnaires, be effectively represented and processed using Deep Learning?
2. How can HR data be translated into graph-based structures? What additional additional insights and benefits would this approach provide?
3. What are then the most effective GNN architectures and training methodologies for learning such HR graphs?
4. What are the key considerations and best practices for applying GNNs to HR data, taking into account the unique characteristics and constraints of this domain?
5. How accurate and efficient are the AI-based methods that have been examined? What are the limitations of these methodologies, and in what ways can they assist HR recruiters in personnel selection?

This dissertation is structured in three parts. The first part provides valuable context to frame the research domain: Chapter 2 provides the necessary background on HRM, the recruiting process at Amajor, and an introduction to Deep Learning for graphs, while Chapter 3 reviews the state-of-the-art in AI for recruiting and GNNs. The subsequent two parts first elaborate on the theoretical contributions about GNNs, and then present the applied case studies. Namely:

- Chapters 4 to 6 illustrates the development of novel GNN architecture, aggregation method and explainability technique;
- Chapters 7 and 8 develop approaches to create graph-based representation for HR data, uch as candidate similarity networks based on profile or resume similarity.

- Chapter 9 showcases the application of GNN to real-world candidate-job matching, proving their potential and usefulness for HR recruiters;

In these chapters, we also offer insights and best practices for implementing ML and GNNs in the HR field, addressing the specific challenges and limitations encountered in practical applications, while also covering ethical considerations and biases in AI-driven recruitment. Finally, Chapter 10 summarizes and concludes the thesis.

1.1 Publications and Contributions

The research presented in this dissertation has resulted in several publications and contributions to the literature, namely:

1. **Frazzetto, Paolo**, Luca Pasa, Nicolò Navarin, and Alessandro Sperduti [2023a]. "Topology preserving maps as aggregations for Graph Convolutional Neural Networks". In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC 2023, Tallinn, Estonia, March 27-31, 2023*. ACM, pp. 536–543. DOI: 10.1145/3555776.3577751

   This paper introduces a novel graph aggregation method based on Generative Topographic Mapping, enhancing the expressiveness of GNNs (Chapter 4);

2. **Frazzetto, Paolo**, Luca Pasa, Nicolò Navarin, and Alessandro Sperduti [2024a]. "Beyond the Additive Nodes' Convolutions: a Study on High-Order Multiplicative Integration". In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC 2024, Avila, Spain, April 8-12, 2024*. ACM, pp. 474–481. DOI: 10.1145/3605098.3636016

   This work explores the use of Multiplicative Integration in graph convolution operations, improving the performance of GNNs on various benchmark datasets (Chapter 5);

3. Fumagalli, Fabian, Maximilian Muschalik, **Paolo Frazzetto**, Janine Strotherm, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara Hammer [2025]. "Exact Computation of Any-Order Shapley Interactions for Graph Neural Networks". In: *The International Conference on Learning Representations (ICLR)*. Under review.

   This international collaboration investigates and explains graph predictions by exactly quantifying the contributions and interactions among multiple nodes (Chapter 6).

4. **Frazzetto, Paolo**, Muhammad Uzair-Ul-Haq, and Alessandro Sperduti [2023b]. "Enhancing Human Resources through Data Science: a Case in Recruiting". In: *Proceedings of the 2nd Italian Conference on Big Data and Data Science (ITADATA 2023), Naples, Italy, September 11-13, 2023*. Vol. 3606. CEUR Workshop Proceedings. CEUR-WS.org. URL: https://ceur-ws.org/Vol-3606/paper71.pdf

This paper presents initial experiments on applying graph-based methods based on assessment distances for candidate classification (Chapter 7);

5. **Frazzetto**, **Paolo**, Muhammad Uzair Ul Haq, Flavia Fabris, and Alessandro Sperduti [2024b]. “From Text to Talent: A Pipeline for Extracting Insights from Candidate Profiles”. In: *The 3rd Italian Conference on Big Data and Data Science, (ITADATA 2024), Pisa, Italy, September 17-19, 2024*. Proceedings not yet available.

   This work introduces a thorough pipeline for leveraging language models and GNNs to improve candidate-job matching using heterogeneous graph structures (Chapter 8);

On the other hand, Chapter 9 introduces a novel approach to match individual applicants to a job position based on a specific graph structure, which represents a significant departure from previously explored methods. While this chapter has not yet been peer-reviewed or published, I believe it further demonstrates an exciting new direction for AI in HR, offering both theoretical and practical advancements.

All these investigations are also raising even more questions and opening up new research directions, pointing out some ways for more sophisticated solutions for AI-driven recruitment. Yet, we should never lose sight of the highest goal: AI must assist recruiters in connecting people with meaningful careers, allowing them to thrive within their organizations.

Part I

Background

2
Context

This section provides the necessary background and context for this research case. We start with an overview of human resources management and its importance in organizations and employee well-being. We then focus on the recruitment process and the challenges involved in the candidates-vacancy alignment. Next, the company that enabled this research is presented—Amajor SB S.p.A.. Then, we discuss how artificial intelligence techniques have been employed in this specific domain. Finally, we provide a technical overview of Graph Neural Networks, that will be leveraged to address the candidates-job matching.

2.1 Human Resource Management

Human Resource Management (HRM) is a vital function in organizations concerned with effectively managing its people—its Human Resources—to achieve organizational goals. It involves developing and implementing strategies, policies, and practices to optimize human capital and ensure that the organization has the right people with the right skills in the right roles [Armstrong, 2020]. HRM covers a wide range of activities, including:

Human Capital Management: obtaining, analyzing and reporting on data that informs the direction of value-adding people management strategic, investment, and operational decisions;

Corporate Social Responsibility: a commitment to managing the business ethically in order to make a positive impact on society and the environment;

Knowledge management: creating, acquiring, capturing, sharing, and using knowledge to enhance learning and performance;

Resourcing: attracting and retaining high-quality people to ensure the organization has the people it needs;

Engagement: the development and implementation of policies designed to increase the level of employees' engagement with their work and the organization;

Organization development: the planning and implementation of programs designed to enhance the effectiveness with which an organization functions and responds to change;

Talent management: the systematic attraction, identification, development, engagement, retention, and deployment of individuals who are of particular value to an organization

Learning and development: providing an environment in which employees are encouraged to learn and develop;

Employee relations: defining the organization's intentions about what needs to be done and what needs to be changed in how the organization manages its relationships with employees and their trade unions.

Employee well-being: meeting the needs of employees for a healthy, safe, and supportive work environment.

The overall aims of HRM are to support the organization in achieving its objectives by developing and implementing HR strategies that are integrated with the business strategy, contribute to the development of a high-performance culture, ensure the organization has the talented, skilled, and engaged people it needs, and build positive employment relationships [Bratton, 2021].

The Harvard framework of HRM [Beer, 1984, 2015] initially emphasizes the human aspect and considers people as assets rather than costs. It stresses the importance of congruence between HR and business strategy and gaining commitment through participation and informed choice. The framework also recognizes the impact of situational factors on HR policy choices. This approach views employees as critical resources that give organizations a competitive advantage through their commitment, adaptability, and high-quality skills and performance. In fact, effective HRM can positively impact organizational performance in several ways. Firms with high-performance HR systems have significantly higher levels of firm performance [Becker, 1996; Guest, 2000]. These studies suggest that HR practices, particularly those focused on acquiring, developing, and motivating employees, can lead to improved individual and organizational performance and outcomes. For example, specific HRM practices, such as job design, direct participation, and information provision, are associated with higher levels of performance and work satisfaction [Guest, 2002]. Still, while research has shown an association between HRM and performance, the causal links are not definitive [Guest, 2011].

Alongside their effect on organizational performance, the impact of HRM practices on *employee well-being* has emerged as a crucial area of research and development. These practices aim to create a supportive work environment that enables employees to fulfill their potential and feel valued within the organization. However, the relationship between company performance and employee well-being is not always straightforward [Vanhala, 2006]. While effective HRM practices can be relatively good predictors of company performance, their direct impact on employee well-being is less clear and often better explained by work-related factors such as job content, workload, and social support. Organizations are beginning to develop a holistic approach to HRM, addressing not only performance-related practices but also factors that directly influence employee well-being [Peccei, 2019; Salas-Vallina, 2021].

The recent report of De Neve [2023] is based on data from over 25 million surveys, representing the largest global study on work well-being. They demonstrated that employees who are happy, have a sense of purpose, are satisfied with their jobs, and experience low stress levels help increase firm performance—outperforming stock-market benchmarks. Surprisingly, they also reveal that 78% of employees are not thriving.

As organizations face ongoing challenges to be agile, innovative, and competitive, the role of HRM will only continue to grow in strategic importance. Implementing HRM practices that promote both employee well-being and organizational success in a sustainable manner is, therefore, fundamental in today's society.

2.1.1 Resourcing

People resourcing is a crucial function within HRM that focuses on ensuring an organization has the right people with the necessary skills and capabilities to achieve its strategic objectives. It encompasses a variety of activities, including workforce planning, recruitment and selection, talent management, and managing employee turnover and absence [Armstrong, 2020]. This work is positioned within *recruitment and selection*—the task of finding and hiring the best-qualified candidates for precise job openings.

Recruitment is the process of attracting a pool of qualified candidates. It involves analyzing the requirements of a job, considering the competencies and skills necessary for success, and then using various sourcing strategies to identify suitable candidates [Breaugh, 2013]. Common recruitment methods are, for instance:

- Advertising job postings on the organization's website, job boards, or social media;
- Employee referrals;
- Recruitment agencies or executive search firms;
- Job fairs or campus recruitment events;
- Networking and headhunting.

The choice of recruitment methods depends on factors such as the nature of the job, the urgency of the hiring need, and the available budget.

On the other hand, the *selection* involves assessing the suitability of candidates attracted through the recruitment process and choosing the best person for the vacant job [Sekiguchi, 2004]. It typically consists of several stages:

1. Screening applications and resumes/CVs to identify candidates who meet the minimum requirements;
2. Conducting interviews (phone, video, or in-person) to assess candidates' competencies, experience, and fit with the organization;
3. Administering assessments such as personality tests, cognitive ability tests, or work samples;
4. Making a job offer to the selected candidate(s).

The selection process aims to gather relevant information about candidates and make a fair and objective decision based on their merits and suitability for the role.

Role of the HR Recruiter

HR recruiters play a vital role in the recruitment and selection process. They work closely with hiring managers to understand the job requirements and develop an effective sourcing strategy. Key responsibilities of an HR recruiter include:

- Conducting job analysis and preparing job descriptions and person specifications;
- Identifying appropriate sourcing channels and advertising job openings;
- Screening applications and shortlisting candidates;
- Conducting interviews and assessments;
- Coordinating the hiring process and communicating with candidates;

- Extending job offers and negotiating terms of employment;
- Onboarding new hires and ensuring a smooth integration into the organization.

HR recruiters need to have a good understanding of the organization's culture, values, and talent needs. They should also possess strong interpersonal and communication skills as they interact with a wide range of stakeholders throughout the recruitment and selection process.

Job Descriptions and Postings

Job descriptions and postings are essential tools in the recruitment process. A *job description* (JD) is a written report that outlines the essential functions, responsibilities, qualifications, and reporting relationships of a job. It serves as a basis for creating *job postings* (JP), which are advertisements used to attract candidates to apply for the vacant position. An effective JP usually includes:

- Job title and location;
- Overview of the organization and its culture;
- Key responsibilities and duties;
- Required qualifications, skills, and experience;
- Desired competencies and attributes;
- Compensation and benefits information;

Well-crafted job descriptions and postings help attract the right candidates, set clear expectations, and provide a framework for the selection process.

2.1.2 Challenges of Screening and Candidate-Job Matchin

While recruitment and selection processes aim to identify and hire the most suitable candidates for job openings, HR professionals often face challenges in effectively screening applicants and ensuring a good fit between candidates and job vacancies—the *Candidate-Job Matching* (CJM) [Nikolaou, 2015].

Information Overload One common challenge is dealing with a large volume of applications, particularly for highly sought-after positions. Screening a large number of resumes and applications can be time-consuming and resource-intensive. HR recruiters must efficiently review the information candidates provide and quickly identify those who meet the minimum requirements for the role. This can be particularly challenging when candidates have diverse backgrounds or when the quality of applications varies significantly.

Limited Information Another challenge is the limited information available in candidates' resumes and applications. Although these documents provide an overview of a candidate's qualifications and experience, they may not give a complete picture of their skills, competencies, and potential fit with the organization's culture. HR recruiters often need to rely on interviews and assessments to gather more comprehensive information about candidates' suitability for the required role.

Assessing Fit Ensuring a good fit between candidates and vacancies is crucial for successful hiring outcomes, but it can be challenging to assess. Fit involves not only aligning candidates' technical skills and qualifications with the job requirements but also their compatibility with the organization's culture, values, and work environment [Kristof, 1996]. Assessing fit often requires going beyond the information provided in resumes and applications and using techniques such as behavioral interviewing and cultural fit assessments.

Biases in Screening Unconscious biases can also pose challenges in screening and candidate-job matching. HR recruiters may inadvertently be influenced by factors such as a candidate's name, age, gender, or educational background, leading to biased decision-making. To mitigate these biases, organizations can use structured screening processes, blind resume reviews (by removing identifying information), or diverse interview panels.

Skill Gaps In some cases, there may be a mismatch between the skills and qualifications of available candidates and the job requirements. This can be particularly challenging in rapidly evolving industries or for highly specialized roles. HR recruiters may need to consider alternative sourcing strategies, such as targeting passive candidates or partnering with educational institutions to develop talent pipelines.

To address these challenges, HR professionals can leverage various strategies and technologies, for example:

- Using Applicant Tracking Systems (ATS) to efficiently manage and screen large volumes of applications;
- Developing structured screening criteria and interview guides to ensure consistency and fairness in evaluating candidates;
- Conducting skills assessments and work sample tests to gather objective data on candidates' abilities;
- Providing interview training to hiring managers to minimize biases and improve the quality of candidate evaluations;
- Analyzing data on past hiring successes and failures to continuously refine and improve screening and selection processes.

Finding the best candidate is not a trivial task and requires time and experience. For these reasons, along with the above-mentioned strategies, novel technological solutions are emerging aiming to support HR recruiters [Kuncel, 2014; Strohmeier, 2022].

2.2 Amajor SB

This section introduces the consulting firm that enabled and founded this research project, *Amajor SB*.[1] It is a benefit corporation—based in Padova, Italy—whose main activities concern proprietary consulting services to guide small- and medium-sized enterprises and improve their business model through the entrepreneurs' values. Authentic and diverse datasets were obtained through the cooperation with Amajor and its team, enabling the analysis and development of models that closely mirror the challenges and complexities faced by HR practitioners. The relevance, applicability, and alignment of the research findings with the practical needs of the industry are ensured through this applied research approach, leading to more effective and impactful HR recruitment strategies. Amajor provides different services, and to better frame the scope of this work, they are presented in the remaining part of this section.

2.2.1 Entrepreneurial Assessment

Amajor's first service is an analysis that begins with the Entrepreneurial Assessment. This assessment analyzes the operational habits of entrepreneurs, highlighting both the challenges they face in managing their businesses and the untapped potential they possess as leaders within their companies.

2.2.2 A+ Questionnaire

One of the tools used for this assessment is the proprietary *A+ Questionnaire* (QST), which provides feedback on each person's potential and how they can best harness it (see Fig. 2.1 for an example). After working with more than 120 clients over a period of five years, Amajor's team has found that understanding entrepreneurs' habits is essential to helping both them and their companies realize their full potential—the entrepreneur's habits have a significant impact on the entire organization and its members. The A+ Questionnaire was specifically designed for this purpose: to provide concrete insights into how each person's habits influence their behavior, efficiency, and well-being. The A+ Questionnaire is first administered to the entrepreneur

[1] Corporate website: `https://www.amajorsb.com/`

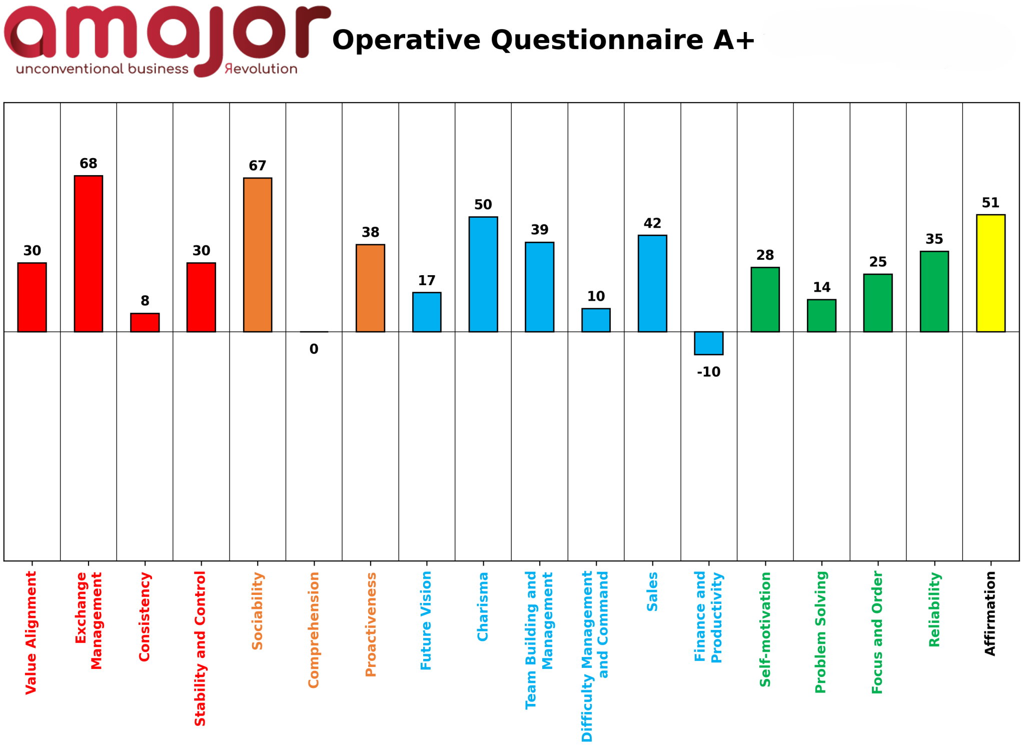


Figure 2.1: Example of the A+ Questionnaire. The outcomes of the questions yield 18 numerical traits obtained through a proprietary algorithm. The colors group these traits based on their affinity (self-fulfilment, interpersonal skills, management, execution, self-affirmation). Each trait takes numerical values in the range $[-100, 100]$. This visualization, currently employed by the HR team, was also a side-project of this research.

and then to the company's first line of management. This process allows the design of a growth plan that aligns with the company's needs and desires while identifying potential strengths to leverage for business development, as well as areas for improvement that may require targeted actions to prosper. More detailed information about the A+ Questionnaire is provided in Section 7.2.4.

2.2.3 Business Assessment

Following the Entrepreneurial Assessment, two types of business analysis services can be developed: *Entrepreneurial Line* and *Business Analyses*.

The former is based on the "4H Method" (Heart, Head, Hands, and High-Value Results) [Peronato, 2022]. This method involves several stages, depicted in Fig. 2.2: Heart, defining the company's core values and vision; Head, developing the mission and a business expansion plan; Hands designing the organizational chart, defining each role's purpose, and establishing Key Performance Indicators (KPIs); High-Value Results, implementing development plans and monitor outcomes. This strategic service identifies the company's core values as a driver for growth. It starts with discovering the values of entrepreneurs or top management, followed by strategy formulation and role identification, culminating in organizational growth. It aligns employees with the entrepreneur's vision, fostering motivation and engagement, leading to increased performance and sustainable growth.

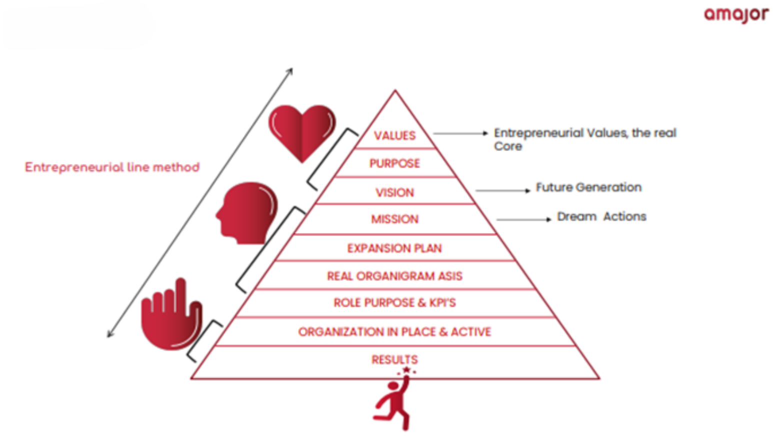


Figure 2.2: Amajor 4H Method. Starting from the entrepreneur's values, mission, and vision, business processes are implemented to achieve the company's growth and employees' well-being and engagement.

Other business analysis services offer various depths of investigation through specific questionnaires, administered anonymously or not, depending on the company's needs.

Personnel Assessment: This involves analyzing human resources, starting with top management. It identifies employees' strengths, potential roles, and hidden potential within the company. This helps determine if individuals are optimally positioned within the organization.

Climate Assessment: Conducted anonymously across departments, this survey measures the organizational climate, reflecting how employees perceive the work environment and leadership. It evaluates communication, the extent to which employees feel valued, and the alignment with company values. This assessment provides a clear picture of the company's internal culture and is a critical tool for improving employee well-being and cohesion.

Organizational Assessment: Through questionnaires and interviews, this analysis identifies inefficient processes or communication barriers. It helps understand how employees experience and contribute to the company's organization and its evolution.

Innovation and Business Development Assessment: This analysis evaluates the company's ability to innovate and remain competitive, identifying the key drivers that fuel its effectiveness.

By combining all these areas with the Entrepreneurial Assessment, Amajor helps companies identify their true purpose, which is a key outcome of the entire analysis process.

2.2.4 Resourcing Services

The goal of the recruitment and selection processes at Amajor is to help entrepreneurs find individuals who not only possess the necessary skills and knowledge required by the company but are also aligned with the company's expansion plans, values, and vision. When providing recruitment services, the objective is to identify candidates who can both thrive within the company's framework and contribute to its growth. This approach ensures that new hires are not only qualified for the role but also motivated by the company's mission and vision, allowing them to grow alongside the company. The recruitment and selection process has become a strategic service, as it allows companies to find individuals who align with specific roles and the company's broader vision, thus unlocking the full potential of their employees. The resourcing process follows these stages, graphically summarized in Fig. 2.3:

1. **Collection of client needs for the role:** The Recruiters conduct a job analysis with the client, defining the role's requirements and characteristics (Job Analysis).

2. **Validation of the Job Analysis:** The Recruiters study whether the required role fits the client's needs, particularly verifying alignment between required skills and the offered salary based on market data.

3. **Analysis of the internal database:** The Recruiters screen existing profiles in the company's database to identify potential candidates that match the role.

4. **Creation and publication of job postings:** The Recruiters create job postings and determine the most appropriate channels for publication based on the role's specifics and the company's location.

5. **Active search or Talent Attraction activities:** If there are insufficient candidates from the postings, the Recruiter initiates an active search through online job portals or contacts relevant networks like universities, training centers, and schools (also known as *head hunting*).

6. **Management of applications:** The Back-Office area organizes incoming applications, files them into individual folders, and creates a candidate profile in the database with their respective information.

7. **Sending of A+ Questionnaire and data update:** The back office area sends the A+ Questionnaire to all candidates and constantly updates the database with the results.

8. **Screening:** The Recruiters evaluate whether candidates' CVs and A+ Questionnaire results align with the client's requirements before proceeding with an initial individual interview.

9. **Interview scheduling and feedback:** The back office area schedules interviews with candidates who passed the screening phase and sends feedback emails to those who were not selected.

10. **Selection interviews:** The Recruiters conduct the first interview to assess the candidate's suitability to move forward in the selection process.

11. **Scheduling technical interviews and feedback:** The back office area organizes any necessary technical interviews with specialized staff for suitable candidates and sends feedback emails to those not selected.

12. **Feedback management and organization of candidate presentation:** The back office area communicates feedback to candidates who were not selected in the final phase, while the Recruiters contact candidates to present them to the client and arrange the meeting.

13. **Candidate presentation:** The Recruiters oversee the presentation of candidates to the client and assist in the final selection process.

14. **Support in negotiation:** Once the suitable candidate is identified, the Recruiters facilitate smooth negotiations regarding economic terms.

15. **Onboarding plan:** The Recruiters support the client in defining an effective onboarding plan aligned with the company's expansion project.

16. **Monitoring of the onboarding process:** The Recruiters monitor the new employee's progress for three months, addressing any needs that arise during this period.

The services and process described above are undeniably complex, costly, and time-consuming, requiring a significant investment of both human and financial resources. To address these challenges and enhance the efficiency of their business model, Amajor is committed to investing in research and development activities. Indeed, this research covers the process from the initial application screening to the shortlisting of candidates and, ultimately, the final hiring decisions, leveraging HR data collected at every stage of this extensive process. However, Amajor firmly believes that while AI can optimize processes, the core of its mission remains unchanged: keeping people, their values, and their needs at the center of every decision and action. This human-centric approach continues to guide the company's strategic growth and ensures that technology serves to elevate, rather than replace, the personal touch in its services.

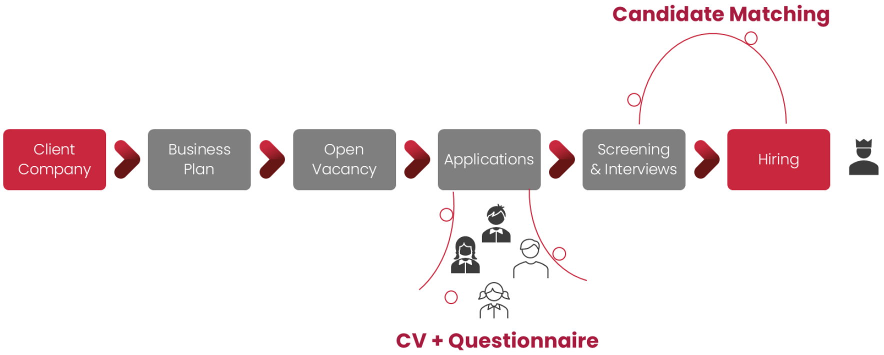


Figure 2.3: Overview of Amajor's resourcing services process. The scope of this research encompasses the stages from the initial screening of applications to the shortlisting of candidates and subsequently the final hiring decisions, while using HR data gathered at each step of this comprehensive process.

2.3 AI in HR

In a nutshell, Machine Learning (ML) is a subset of Artificial Intelligence (AI) that enables systems to learn from data, identify patterns, and make decisions with minimal human intervention [Russell, 2016]. Such technology has emerged as a transformative force in many domains [Van Esch, 2019], and HRM is no exception [Hmoud, 2019; Strohmeier, 2022; Vrontis, 2023]. ML offers broad and promising potential in the HR discipline while also presenting some serious challenges. Broadly speaking, the process of implementing ML methods typically follows several key steps: domain understanding, data understanding, data preparation, model generation, model evaluation, and model application. This process can be translated into the HR field and executed either explicitly by HR professionals using ML software or embedded within HR-specific software by existing vendors and providers. ML algorithms in HR can be broadly categorized based on their functions, including prediction, classification, anomaly detection, segmentation, and association. These algorithms can utilize various types of data, ranging from structured static data (e.g., payroll information) to unstructured data (e.g., employee documents) and even stream data (e.g., social media data).

The application of ML in HR offers several potential benefits. It can enhance decision-making processes, improve problem-solving capabilities, and even automate certain HR tasks. ML is seen as a key enabler of concepts such as big HR data analytics and algorithmic HR decision-making [Bondarouk, 2017; Marler, 2017], and organizations leveraging data-driven approaches have long demonstrated improved financial and operational performance [McAfee, 2012]. However, the implementation of ML in HR is not without challenges. Technical challenges include the potential for spurious correlations, conservatism in suggestions due to limited historical data, lack of publicly available datasets, and replication of existing biases or errors. Ethical and social threats are also significant, including concerns about discrimination, privacy violations, and the opacity of ML processes [Tambe, 2019; Burrell, 2021].

Research on ML in HR is currently divided between two main streams: methodical-technical research focusing on the development of ML artifacts and managerial-behavioral research examining the application and implications of ML in HR contexts [Strohmeier, 2022]. Future research directions include the need for more interdisciplinary collaboration, improved theoretical foundations, and increased empirical investigations into the real-world impacts of ML in HR—as in the scope of this investigation.

It is clear that ML has the potential to transform HR practices significantly. However, realizing this potential will require careful navigation of both technical and ethical challenges, as well as continued research and development in this rapidly advancing field. For a more in-depth overview of the state-of-the-art of AI in HR, we refer to Section 3.1.

2.3.1 AI Act

It is worth mentioning the European Union's approved *Artificial Intelligence Act* (AI Act) [AI Act, 2024], that represents a significant development in the regulation of AI systems and it has a particular relevance to the domain of human resources as well. The AI Act categorizes certain AI applications in HR as "high-risk", reflecting the potential for these systems to significantly impact individuals' lives and opportunities. Namely, AI applications in HR are mentioned in Annex III.4.(a):

> "AI systems intended to be used for the recruitment or selection of natural persons, in particular to place targeted job advertisements, to analyse and filter job applications, and to evaluate candidates".

Such applications are thus subject to stringent requirements to ensure fairness, transparency, and accountability. These requirements emphasize the importance of implementing measures to evaluate AI systems, ensure traceability, properly inform all stakeholders, and keep human oversight on the whole process. They pose significant challenges for HR departments and AI developers, and organizations must invest in robust data management practices to comply with these regulations. Nevertheless, the AI Act also offers new opportunities and promotes responsible AI development and use. It could help address some of the ethical and social concerns associated with AI in HR and avoid blunders committed in the past [Dastin, 2018]. It may also drive innovation in explainable AI and fair ML algorithms, potentially leading to more trustworthy and effective AI systems in HR, ultimately shaping the future development and application of AI in HR, and encouraging a more responsible and human-centric approach to AI-driven HR practices.

It is important to note that the research presented in this thesis is purely exploratory and conceptual in nature. This work has not been implemented in any production setting and is not intended for immediate practical application. The study utilizes historical data and records exclusively, serving as a proof-of-concept to explore potential applications of graph neural networks in candidate-job matching. The findings and methodologies discussed herein are meant to contribute to the academic discourse and provide a foundation for future research. They should not be interpreted as ready-for-deployment solutions in real-world HR processes without further development, testing, and ethical considerations.

2.3.2 Privacy Regulations

Privacy is a paramount concern in the field of HR, especially when dealing with the sensitive personal data of job applicants. This sensitive data falls under specific legal frameworks, most notably the General Data Protection Regulation (GDPR) in the European Union [GDPR, 2016]. This regulation mandates that organizations handle personal data with care, ensuring its confidentiality, security, and lawful processing. Therefore, HR professionals must navigate these specific legal requirements by implementing robust data protection measures, obtaining informed consent from candidates, and ensuring secure storage and transmission of personal information.

In the scope of this work, we have taken extensive measures to handle data in a compliant manner. Prior to data collection and questionnaire administration, candidates were duly informed about the purpose and scope of data processing and then provided their informed consent. In addition, we ensure anonymity by removing any personally identifiable information from the datasets. We processed the data in its aggregate structure without any additional bias. Besides, we adhere to data retention policies and will promptly delete the original records once they are no longer necessary for the intended purposes. In this fashion, we aim to maintain the confidentiality and trust of the candidates' personal data, along with promoting open science and productive discussions in this field.

2.4 Deep Learning for Graphs

Graphs are mathematical objects effectively used to represent entities and relations thereof. Graph-structured data appears in many domains and real-world applications [M. Newman, 2018], such as molecular chemistry [Gilmer, 2017; Stärk, 2022], water distribution networks [Ashraf, 2023], sociology [Borgatti, 2009], power grids [Hansen, 2023], finance [Easley, 2010], physics [Sanchez-Gonzalez, 2020], or epidemiology [Pastor-Satorras, 2001; Navarin, 2024].

Deep learning has shown astounding results on tasks for non-structured data, such as images or texts, so it is not surprising that deep learning models for graph data have been developed in recent years [Bacciu, 2020; L. Wu, 2022]. The initial definition of neural networks designed for graphs was introduced several years ago [Sperduti, 1997], paving the way for the concept of Graph Neural Networks (GNNs) [Scarselli, 2008; Micheli, 2009]. Lately, motivated by CNNs, RNNs, autoencoders, transformers, and new advancements in deep learning, researchers are rapidly developing new techniques and approaches to handle the complexity of graph data.

2.4.1 Notation and Preliminaries

In mathematical terms, a graph is defined as a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathcal{Y})$ where $\mathcal{V}$ denotes the set of N nodes (or vertices), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges (or links). If nodes v and u are connected, then $(v, u) \in \mathcal{E}$. All edges can be represented as adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, whose elements $a_{uv} = 1 \iff (u, v) \in \mathcal{E}$, otherwise $a_{uv} = 0$. This work deals with undirected graphs, i.e. $a_{uv} = a_{vu} \; \forall \; v, u \in \mathcal{V}$. The matrix $\mathbf{X} \in \mathbb{R}^{N \times S}$ encodes the node attributes (or features), and $\mathbf{x}_v \in \mathbb{R}^S$ represents the features of node v. The set of nodes linked to node v, also known as neighborhood, is denoted as $\mathcal{N}_v$. The amount of edges belonging to a node defines its degree, $d_v \in \mathbb{N}$. Each graph, or its nodes, can have labels denoted with the set $\mathcal{Y}$. Their predictions are represented as $\tilde{\mathcal{Y}}$.

In the context of GNNs, the *message-passing* layer is crucial for propagating information across the graph [Bronstein, 2021]. GNNs iteratively learn node representations $\mathbf{h}_v$ by aggregating information from neighboring nodes. Namely, the ℓ-th layer of a message-passing GNN can be formulated as:

$$\mathbf{h}_v^{(\ell)} = \phi^{(\ell)}\left(\mathbf{h}_v^{(\ell-1)}, \oplus\left(\left\{\psi^{(\ell)}(\mathbf{h}_v^{(\ell-1)}, \mathbf{h}_u^{(\ell-1)}) \colon u \in \mathcal{N}_v\right\}\right)\right) \tag{2.1}$$

where

- $\mathbf{h}_v^{(\ell)} \in \mathbb{R}^{N \times D^{(\ell)}}$ is the updated feature vector for node v;
- $\mathbf{h}_v^{(\ell-1)} \in \mathbb{R}^{N \times D^{(\ell-1)}}$ is previous feature vector for node v, with $\mathbf{h}_v^{(0)} = \mathbf{x}_v$;
- $\phi^{(\ell)}$ is a learnable *update* function;
- $\mathcal{N}_u$ is the neighbourhood of node u;
- $\oplus$ represents a differentiable, permutation-invariant *aggregation* function (e.g., sum, mean, max) of messages of the neighborhood of node u;
- $\psi^{(\ell)}$ is a learnable *message* function, computing the message vector sent from node u to node v.

The general idea of message passing is depicted in Fig. 2.4, and the above formulation allows for various implementations of GNNs for different ϕ, ψ, and $\oplus$, as presented in Section 3.2.1. After ℓ iterations of message passing, the feature vector $\mathbf{h}_v^{(\ell)}$ encapsulates both the structural information and the content of the ℓ-hop neighborhood of node v. This rich representation enables both node-level and graph-level tasks. For tasks that require predictions at the node level, we define a node classification function $f_v \colon \mathcal{G} \rightarrow \tilde{\mathcal{Y}}$. This function typically employs a

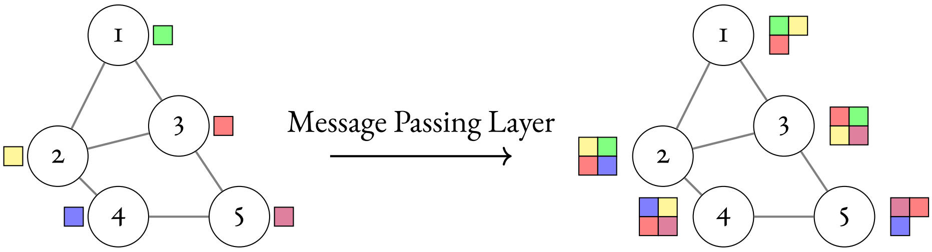


Figure 2.4: Illustration of feature propagation in Graph Neural Networks. Left: Initial graph with each node having a unique feature (represented by a colored block). Right: The same graph after one message passing layer, where each node's features now include information from its neighbors and will be processed by ϕ, ψ, and $\oplus$. This process allows local information to spread across the graph structure, enabling the network to learn both local and global patterns.

Multi-Layer Perceptron (MLP) that operates directly on the node embeddings:

$$f_v(\mathcal{G}) = \mathrm{MLP}(\mathbf{h}_v^L) \tag{2.2}$$

where L is the final layer of message passing. For tasks that require a single prediction for the entire graph, we define a graph classification function $f\colon \mathcal{G} \to \tilde{\mathcal{Y}}$. This function first aggregates node embeddings using a global pooling operation over all its nodes $\bigoplus$, then applies an MLP (also known as readout):

$$f(\mathcal{G}) = \mathrm{MLP}\left(\bigoplus\left(\{\mathbf{h}_v^L\colon v \in V\}\right)\right) \tag{2.3}$$

The pooling operation $\bigoplus$ is crucial for graph-level tasks as it must transform a set of node embeddings of variable cardinality into a fixed-size graph embedding, while being permutation invariant or equivariant. For a more in-depth description of pooling, refer to Section 3.2.2. After the message passing layers and potential pooling operations, the final layer of a GNN typically projects the node or graph representations into the desired output space. For a classification task with C classes, this often involves a linear transformation followed by a softmax activation:

$$\tilde{y} = \mathrm{softmax}\left(f(\mathcal{G})\right). \tag{2.4}$$

The model is then trained by minimizing a loss function $\mathcal{L}$, which measures the discrepancy between the predicted outputs and the true labels. For multi-class classification, a common choice is the cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\tilde{y}_i) \tag{2.5}$$

where y_i is the true label (typically one-hot encoded) and $\tilde{y}_i$ is the predicted probability for class i. For clarity of exposition, several important components that are typically part of GNNs and neural networks in general have been omitted. These comprise *activation functions*, *dropout*, *regularization*, *normalization* and *residual connections*.

In conclusion, the specific choices and design configurations of a GNN often depend on the particular task and dataset at hand, and this will consequently be discussed.

3
Related Works

This chapter provides a comprehensive review of the existing literature relevant to our research on GNNs for candidate-job matching (CJM). We begin by examining the current state of the art in recruitment and CV matching techniques, highlighting both traditional approaches and recent advancements in machine learning-based methods. Subsequently, we present the foundational works on GNNs, with a particular focus on three critical aspects: graph convolutions, pooling strategies, and explainability. These areas form the theoretical backbone of our proposed approaches and are further developed as novel contributions presented in Chapters 4 to 6. With these understandings from the recruitment sector and with the latest technical advances in GNNs, we set the stage for our investigation on GNNs applied to HR data in Chapters 7 to 9.

3.1 AI for Resourcing

The previous chapter illustrated how the application of AI to personnel selection represents a significant development in HRM. Laumer [2022] and König [2022] provide a comprehensive overview of this emerging area, highlighting both its potential and challenges. This section expands their key insights and provides relevant references for the context of this thesis. First, some open challenges are presented; then, the concept of Person-Environment Fit is introduced to adequately frame the literature. This allows us to present the related works, leaving some future research directions in the end.

3.1.1 Challenges and Opportunities of AI in Personnel Selection

As presented in Section 2.1.2, HR recruiters often struggle with several limitations, including applicants' tendency to present an overly favorable image, human biases in decision-making, cognitive limitations of assessors, and inadequate utilization of potentially valuable data. These challenges create an opportunity for AI and ML approaches to potentially improve the efficiency and effectiveness of selection processes.

ML techniques can be applied to various aspects of the selection process, from analyzing video interviews to evaluating CV. The use of ML in personnel selection offers several potential advantages—it allows for the efficient processing of large, unstructured datasets from various sources and the integration of multi-modal information for more comprehensive evaluations. ML approaches may also reduce certain human biases, increase speed and efficiency in screening large applicant pools, and discover novel insights and patterns in applicant data.

Despite its potential, the application of ML in personnel selection faces several significant setbacks [Goretzko, 2021]. One major concern is the potential for ML models to inadvertently perpetuate existing biases present in training data or human decision-making. Legal and ethical concerns arise from the use of certain applicant characteristics in ML models, which may raise discrimination issues. Applicants often have negative perceptions of automated evaluation systems, which could impact their willingness to engage with organizations using such technologies. Additionally, HR professionals may be reluctant to rely on ML systems, preferring to trust their own judgment. The "black box" nature of many ML models also poses challenges for transparency and explainability, making it difficult to understand and justify their decisions. Another significant issue is the generalizability of ML models: systems trained on data from one context may not perform well when applied to different organizations or job roles, limiting their broad applicability.

3.1.2 Survey Based on the Person-Environment Fit

Resourcing is complex and multi-faceted, and so are its applications of AI. Their foundations and scope are built upon the concept from industrial-organizational psychology of *Person-Environment* (PE) fit, which describes an individual's relationship to their work environment [Jansen, 2006]. Accordingly, there is no one-dimensional view of this fit; instead, it consists of and depends on various multidimensional factors: the person–vocation (PV) fit, the person–

job (PJ) fit, the person-organization (PO) fit, the person-person (PP) fit, and the person-team (PT) fit. In this work, the PJ fit, i.e. the match of a person's characteristics and skills with those of a specific job, is primarily addressed. However, it is relevant to note that Amajor's recruiting services are part of a broader business development plan. Candidates are selected and hired also based on their PO fit, considering a supplementary fit between the organization's and candidates' characteristics and values. A good PO fit positively influences the individual's satisfaction, willingness to perform, and commitment to the organization. Indeed, the purpose of recruitment is to identify the most appropriate candidate for an open role. In this context, "best" refers not to being literally the top individual but rather the most suitable person in relation to the various attributes of the PE fit and its specific characteristics of each dimension.

Before going into the details of related works, Laumer [2022]'s literature review excellently encapsulates the majority of the current state-of-the-art, offering an overview of the primary areas of investigation and the methodologies employed. The authors identified 56 papers that report on AI-based approaches for predicting PE fit dimensions. These selected studies focused on three main areas of resourcing: identifying potential candidates (9 papers), pre-selection of candidates—the CJM—based on CVs or additional information (36 papers), and further assessment of candidates (15 papers). The authors observed that most of the research concentrated on particular aspects of PE fit, namely PJ fit (34 papers), PO fit (8 papers), and PT fit (3 papers). Notably, none of the identified papers explicitly focused on PV or PP fit. For what concerns the ML approaches employed in the reviewed studies, they have been grouped as follows:

- Decision Trees (18 papers)
- Deep Neural Networks (10 papers)
- Logistic Regression (9 papers)
- Random Forests (9 papers)
- Support Vector Machines (8 papers)
- Neural Networks (8 papers)
- K-Nearest Neighbors (8 papers)
- Naïve Bayes (5 papers)

Additionally, 19 papers focused on other AI techniques such as recommender systems, data mining, natural language processing (NLP), matching algorithms, fuzzy logic, network models, and sentiment analysis.

While this study primarily focuses on the widespread scenario of PJ fit and CJM, it is noteworthy that, to the best of our knowledge, this is the pioneering research utilizing GNN in the HR domain.

3.1.3 State of the Art

This section provides an overview of key studies and findings in the rapidly evolving field of AI in resourcing. Current research trends indeed demonstrate promising results in areas such as predicting personality traits and job performance from written materials using NLP, inferring personality traits and communication skills from audio or video data, and automating the CJM. However, it is important to note that much of the following research comes from computer science rather than traditional industrial-organizational psychology or management studies, and there is still a lack of robust multi-disciplinary validation studies in real-world hiring contexts.

Early works dealing with CJM are the ones by Yu [2005] and Yi [2007], in which they adopt information retrieval techniques to parse relevant sections of semi-structured resumes. On the other end, the work of Kessler [2008] implemented a k-nearest neighbors (kNN) model for CJM to identify key attributes from CVs and cover letters, such as skills, work history, salary expectations, and address. They developed a vector representation for both candidate profiles and job listings to determine the optimal model for ranking candidates based on job compatibility. The machine learning algorithm was trained on a dataset comprising 25 job postings and 2,916 candidate profiles, each labeled with a job fit score, achieving an accuracy of 0.64.

Faliagka [2014] used data from online social network profiles and blogs to predict person-job fit. The suggested system gathers objective criteria from the LinkedIn profiles of the applicants and semantically matches these with the job requirements. It further deduces their personality traits through linguistic analysis of their blog posts. Their decision tree algorithm performed well in predicting human experts' rankings of job fit, suggesting the potential of using digital footprints in selection processes.

Guo [2016] developed *RésuMatcher*, a personalized resumes-job matching system. It uses information extraction tools to parse resumes and job descriptions into structured models. It then calculates similarity scores between resumes and job models using an ontology-based similarity measure. Jobs are ranked based on these similarity scores, and their evaluation shows improvements over other information retrieval approaches, such as TF-IDF. NLP techniques have been increasingly exploited by Campion [2016], which conducted one of the first large-scale studies applying them to personnel selection. They analyzed approximately 40,000 accomplishment records from applicants to a large U.S. public employer. Using NLP, they constructed automatically scored ratings that correlated highly (0.60 to 0.65 range) with human ratings. This study first demonstrated the potential of AI to automate parts of the CV screening process. Sajjadiani [2019] utilized NLP approaches to analyze online applications for teaching positions in Minnesota. They estimated work experience relevance and categorized turnover history attributions. Their findings showed that certain attribution categories (e.g., "avoiding bad jobs") negatively predicted teacher performance and were linked to involuntary turnover.

Whereas NLP just uses text material, other researchers focused on other modalities (audio and visual) or on combining data from different modalities. For example, Naim [2016] focused on automating the analysis of job interviews. These interviews were recorded, transcribed, and then rated by nine Mechanical Turkers regarding the overall interview performance. They then used ML algorithms to evaluate 138 mock interviews based on facial, speech, and lexical features. Their results showed that speech and lexical features were particularly important in predicting overall interview ratings. Escalante [2020] developed a dataset of 10,000 short video clips, each annotated with a score by Mechanical Turkers. They used various ML approaches to predict these scores using audio and visual information. While primarily a proof-of-concept study, it demonstrated the feasibility of using ML for rapid assessment of video applications and candidates' personalities.

For what further concerns the PJ fit, deep-learning models were successfully used by Zhu [2018]. The authors proposed a convolutional neural network (CNN)-based model called Person-Job Fit Neural Network (PJFNN) to address the task of CJM. The model jointly learns representations of job requirements and candidate qualifications by mapping both resumes and job postings into a shared latent space. PJFNN identifies specific job requirements that candidates fulfill and measures the alignment between their experiences and the job's needs. Using historical job application data from a large tech company, the model significantly improves prediction performance for PJ fit compared to traditional methods. Likewise, Yan [2019] explored how latent preferences from past interview and application histories can enhance CJM in online recruitment platforms. The authors propose a specific deep learning architecture based on GRU encoders that uses a profiling memory module to model the preferences of both job seekers and recruiters. Using real-world data from an online platform, the authors demonstrate significant improvements in matching accuracy. While the latter works leverage recruiters' information and large annotated HR data, Martinez-Gil [2020] proposed a method that learns from past solved cases to predict how human experts would rank candidates for new job openings. Their approach computes transformation costs between profiles and job offers using background knowledge from recruitment taxonomies. It considers factors like replacement costs, insertion/deletion costs for skills, and weighting of different profile elements. The method was tested on real recruitment data from IT, legal, logistics, and marketing domains, outperforming a baseline information retrieval approach in most cases. On the other hand, Xiaowei Wang [2021] defined an approach called KG-DPJF (Knowledge Graph-based Deep-learning-inspired Person-Job Fitting model). Their model uses sentence vectors and subject-term graphs to represent job postings and candidate profiles; it incorporates a BERT-based encoder (Bidirectional Encoder Representations from Transformers) to generate sentence-level embeddings, a knowledge graph to capture domain knowledge and a multi-layer attention mechanism to model interactions between resume features and job requirements. The authors evaluate KG-DPJF on real recruitment data across multiple domains showing that incorporating knowledge graph and BERT embeddings improves performance over baselines. Pessach [2020] introduces the comprehensive analytics framework for HR recruiters to improve hiring and placement decisions. Their framework consists of two phases: a local prediction scheme for individual job placements and a global recruitment optimization model for organizational needs. The local prediction phase uses a Variable-Order Bayesian Network (VOBN) model applied to a large recruitment dataset based on numerical scores of the employees. The analysis reveals that the VOBN model can offer valuable and sometimes counter-intuitive insights. Y. Wang [2021] also utilized BERT to encode textual information in resumes and vacancy requirements, combining it with an attention interaction layer. Vanetik, 2023 extract neural sentence representations, keywords, and named entities using BERT from resumes and vacancies. The final ranking is based on the distance metric between their vector representation. The dataset that the author uses only consists of resumes of software developers.

Tian [2023] tackles the problem of resume classification. The authors proposed the use of Latent Semantic Analysis (LSA), BERT, and Support Vector Machine (SVM). LSA and BERT extract features from the resume text, and then SVM is used to classify resumes. However, the authors do not perform resume matching with the given vacancies. Similarly, Roy [2020] also proposed using ML to match candidates with job postings: the features are extracted from the CV, and JD using TF-IDF, and then matching between the two is done using cosine similarity, recommending candidates similar to the JD based on their words.Jain [2021] introduce CV and JD matching using topic modeling approaches such as LSA and Latent Dirichlet Allocation (LDA). They follow a two-step approach. First, they use TextRank to summarize the CVs by extracting the most relevant sentences, reducing the volume of information, and then employing LSA and LDA to match work experience and skills in the CVs with job descriptions.

Both methods involve creating document vectors and using cosine similarity to find the best matches. The author concludes that LSA outperforms LDA in terms of precision. Pudasaini [2022] also proposed the use of the word2vec algorithm using the CBOW (Continuous bag-of-words) model to create the embedding vectors and the cosine similarity for measuring the similarity score between CV and vacancy details. Schlippe [2023] introduced *Skill Scanner*, a framework for matching CVs with JP. The authors extracted skills from documents using BeautifulSoup, where 21.5k bullet points were extracted from 2,633 online job advertisements. The skills are then vectorized using Sentence-BERT. Then, the author uses K-means to group the 768-dimensional vectors using cosine distance as the metric. Clusters and their representative vectors were used to compare job market skills with CVs.

An ontological approach is the one chosen by Kavas [2023] to align the CVs and job descriptions based on European Skills, Competences, Qualifications and Occupations (ESCO) taxonomy, thus leveraging additional standardized information. They introduce ESCOXML-R, a model that facilitates the learning of the implicit relations between job titles, skills, descriptions, and keywords that define job experiences. On the other hand,Phan [2021] proposed the use of CSO (Computer Science Ontology) to classify documents based on their content automatically. It comprises two main components: the syntactic module, which maps n-grams to concepts using Levenshtein distance, and the semantic module, which identifies semantically related topics using Word2Vec embeddings. This classifier is applied to CVs and JDs within the IT domain. The proposed recruitment system integrates automated CV parsing, skill extraction, and matching modules to rank CVs based on criteria like education and skills. The system processes data from various sources and generates skill graphs for CJM.

3.1.4 Supplementary Relevant Research

Additional noteworthy works in this domain are those done by Langer [2018, 2021], who investigate applicant reactions to AI-based selection methods. Their studies revealed that applicants generally have negative reactions to being evaluated by automated systems, perceiving them as less fair and trustworthy than human evaluators. Interestingly, the absence of any information about their AI evaluation did not appear to cause significant issues, raising the question of which types of information would be most advantageous for applicant reactions. The authors also examined how applicants behave differently when they believe they are being evaluated by AI rather than humans, finding that applicants gave shorter answers and reported using less impression management when they thought AI was evaluating them. Considering applicant perceptions when implementing AI in selection processes is crucial, although it is often neglected.

The cornerstone research by Raghavan [2020] showcased a review of AI-based hiring tools offered by various vendors. They found that while many providers claim to address bias issues, there is often a lack of transparency about the methods used. Biases, accountability, and transparency are great problems when applying ML in this domain. Datasets may be skewed, algorithms may be optimized for a bias criterion, and additionally, data can mirror human biases. Removing identifying information from applicant data might mitigate this issue to a certain degree, as they still potentially correlate with sensitive applicant information [Fabris, 2023]. Solutions to this issue are far from trivial and may require a lot of human work (e.g., humans as data curators, humans monitoring systems for potential biases, humans updating systems).

The meta-scientific paper by König [2020] highlighted the challenges of integrating computer science approaches with traditional personnel selection research. They noted three differences between the two academic fields. Initially, computer scientists are often primarily focused on proof-of-concept studies. Consequently, their interest lies more in the potential to link specific input data to an outcome rather than in ensuring that the data were gathered under realistic, real-world scenarios. Secondly, the core objective of ML is prediction; therefore, ML-focused studies in computer science that examine issues in personnel selection are likely to emphasize how accurately an outcome can be predicted and which algorithm achieves the highest predictive accuracy. Yet, they are less likely to delve into the theoretical justification for including various predictors in a prediction model. Lastly, for computer scientists, personnel selection may be just one among many "use cases", so they might not place significant importance on the standards upheld by researchers in the personnel selection field, such as the psychometric properties of scales. Similarly, Oswald [2020] discussed the potential of big data and AI in industrial-organizational psychology and HRM, emphasizing the need for interdisciplinary collaboration and the importance of addressing ethical concerns in the development and implementation of AI-based selection tools.

These studies collectively demonstrate the growing interest in and potential of AI and ML in personnel selection. They highlight both the opportunities (such as increased efficiency and potentially reduced bias) and the challenges (including applicant reactions, ethical concerns, and the need for real-case validation) associated with these technologies. This study, while a proof-of-concept, distinguishes itself through its foundation on extensive collaboration with HR recruiters and the incorporation of data obtained from practical, real-world recruitment processes. Furthermore, this research utilizes psychometric instruments (the A+ Questionnaire) and is designed to assist recruiters by providing a graph structure that enhances the interpretability of the process data.

3.1.5 Open Research Directions

As the field of AI in personnel selection continues to evolve rapidly, several key areas emerge as critical for future research. One of the most pressing needs is for more studies demonstrating the real-world predictive validity of AI-based approaches in hiring contexts, which is also the scope of this investigation. While numerous proof-of-concept studies have shown promising results, there is a significant gap between laboratory findings and practical applications. As AI is rising more ad more to prominence in today's society, researchers should focus on conducting longitudinal studies that track the performance and tenure of employees selected using AI methods—only such research would provide crucial evidence for the effectiveness of these technologies and help build confidence in their use among HR professionals, candidates and managers.

Another critical area for future research lies in exploring optimal ways to integrate AI systems with human decision-making in resourcing. As König [2020] emphasize, fully automated selection is not legally defensible, not desirable in most contexts, as well as forbidden in the EU by the AI Act. Therefore, understanding how to create effective human-AI collaborative systems is paramount and needed in the upcoming years. With this research, Amajor is trying to build one such system.

Notably, addressing fairness and bias in AI-based recruiting models remains a critical challenge. While not being the focus of this work, future research should concentrate on developing and testing advanced methods for bias detection and mitigation. This work is essential not only for ethical reasons but also for legal compliance and to ensure that AI systems do not perpetuate or exacerbate existing inequalities in the hiring process. Researchers should explore techniques such as adversarial debiasing, fairness-aware machine learning, and the use of synthetic data to create more representative training datasets.

Enhancing the explainability and transparency of AI models used in personnel selection is another crucial area for future work. As the complexity of these models increases, so does the difficulty in understanding and interpreting their decisions. Research into explainable AI techniques specifically tailored for HR applications could help address this challenge. This research should aim to develop methods that can provide clear, understandable explanations for AI-driven selection decisions, which is also legally required. We explored this research direction for graph-based data in Chapter 6, yet more work needs to be done for HR data.

There is also a need to shift focus from mere automation to improving decision quality through aptitude-diagnostic approaches. Future research should explore how AI can not only replicate human decision-making but potentially surpass it by identifying novel predictors of job performance and success. This could involve leveraging big data analytics to uncover non-obvious relationships between candidate characteristics and job outcomes, or developing more sophisticated models of person-job fit that go beyond traditional criteria. Moreover, exploring AI applications for understudied person-environment fit dimensions, particularly Person-Vocation (PV) and Person-Person (PP) fit, represents another promising avenue for future research. Most current AI approaches in selection focus on PJ fit, but a more holistic understanding of how individuals fit within an organization requires consideration of these additional dimensions. Researchers could investigate how AI might be used to assess cultural fit, team dynamics, or long-term career compatibility. Ideally, such perspective would harness the full potential of AI, enabling candidates to find their *ideal* job and thrive in their profession.

The development of larger, high-quality datasets for AI research in personnel selection is ultimately necessary. The lack of publicly available HR dataset hinders research in this domain: many current studies rely on private, limited or potentially biased datasets, which can affect the generalizability and reliability of their findings. Although HR data is sensible and private in its nature, future work should focus on creating comprehensive, diverse, and ethically sourced datasets that reflect the complexity of real-world hiring scenarios. Potential methodologies may utilize anonymization techniques or adopt synthetic approaches. These datasets should be made available to the research community to facilitate more robust and comparable studies.

Comprehensive comparisons of different ML algorithms for specific resourcing tasks are also currently missing. While some studies have compared the performance of various algorithms, more systematic evaluations across a wide range of selection contexts and criteria would be valuable. Again, due to the sensitivity of HR data and commercial interests, many models are closed-sourced and thus with limited reproducibility.

Lastly, as AI technologies continue to advance, research into novel data sources and assessment methods for personnel selection is becoming increasingly of interest. This could include exploring the potential of virtual reality for job simulations, using wearable devices to assess stress and performance under pressure, or leveraging social media and digital footprints for personality assessment. However, such research must carefully consider the ethical implications and potential privacy concerns associated with these new data sources.

In conclusion, the possible future research directions in AI for personnel selection are diverse and challenging, reflecting the complexity and importance of this field. As we move forward, interdisciplinary collaboration between computer scientists, industrial-organizational psychologists, and HR professionals will be crucial in navigating the technical, ethical, and practical challenges of implementing AI in resourcing. This study wishes to be a step towards a future where AI serves as a powerful tool for creating more equitable, efficient, and effective hiring practices.

3.2 Graphs, Convolutions, Pooling, and XAI

In Section 2.4, we introduced GNNs as a powerful tool for analyzing and learning from graph-structured data, whose applications span across various domains [J. Zhou, 2020]. Nevertheless, the utilization of these approaches within the HR sector remains relatively unexplored, and this study positions itself as one of the pioneering efforts in this specific field of research.

As we explore the potential of GNNs for candidate-job vacancy matching, two fundamental operations come to the forefront: graph convolution and pooling. These operations form the backbone of many GNN architectures and play a crucial role in extracting meaningful features from graph-structured HR data. Finally, we already discussed how explaining AI predictions is vital in this domain of sensitive data. Therefore, we also present the recent advancement of explainable AI (XAI), with a particular focus on GNNs.

3.2.1 Graph Convolution

Graph Convolution (GC) is the cornerstone of GNNs, allowing them to aggregate information from neighboring nodes and capture the local structure of the graph. In the context of HR data, this operation will enable us to model complex relationships between candidates, job vacancies, and their attributes. For instance, a candidate's skills can be seen as nodes connected to both the candidate and relevant job postings, forming a rich, interconnected network of information. Several approaches to designing GCs have been proposed in the literature. In the following, three influential methods are presented: Graph Convolutional Networks, Graph Isomorphism Networks, Graph Attention Networks, and GraphConv. In Chapter 5 we will describe our contribution in this field.

Graph Convolutional Networks

The Graph Convolutional Network, introduced by Kipf [2017], is one of the most widely adopted GNN architectures. The GCN layer can be expressed as:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}\right) \tag{3.1}$$

where $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times D^{(\ell)}}$ is the matrix of node features at layer ℓ, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix with added self-loops, $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{D^{(\ell)} \times D^{(\ell+1)}}$ is the learnable weight matrix and $\sigma(\cdot)$ is a non-linear activation function. In the context of HR data, this convolution operation allows each candidate or job vacancy node to aggregate information from its immediate neighbors, effectively capturing local patterns in the graph structure.

Graph Isomorphism Network

The Graph Isomorphism Network (GIM), proposed by Xu [2019], aims to achieve maximum discriminative power among GNNs with a theoretically-grounded formulation. The GIN layer is defined as:

$$\mathbf{h}_v^{(\ell+1)} = \mathrm{MLP}^{(\ell)}\left((1+\epsilon^{(\ell)})\mathbf{h}_v^{(\ell)} + \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^{(\ell)}\right) \tag{3.2}$$

where $\epsilon^{(\ell)}$ is a learnable parameter and $\mathrm{MLP}^{(\ell)}$ is a multi-layer perception.

Graph Attention Network

The Graph Attention Network (GAT) [Velickovic, 2018], incorporates attention mechanisms into the graph convolution process. This allows the model to assign different importance to different neighbors when aggregating information. The GAT layer is defined as:

$$\mathbf{h}_v^{(\ell+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} \alpha_{vu}^{(\ell)}\mathbf{W}^{(\ell)}\mathbf{h}_u^{(\ell)}\right) \tag{3.3}$$

where $\mathbf{W}^{(\ell)}$ is a learnable weight matrix and $\alpha_{vu}^{(\ell)}$ is the attention coefficient between nodes v and u at layer ℓ. These are computed using a softmax function over the neighborhood:

$$\alpha_{vu}^{(\ell)} = \frac{\exp\left(f(\mathbf{W}^{(\ell)}\mathbf{h}_v^{(\ell)}, \mathbf{W}^{(\ell)}\mathbf{h}_u^{(\ell)})\right)}{\sum_{k \in \mathcal{N}(v) \cup \{v\}} \exp\left(f(\mathbf{W}^{(\ell)}\mathbf{h}_v^{(\ell)}, \mathbf{W}^{(\ell)}\mathbf{h}_k^{(\ell)})\right)} \tag{3.4}$$

where $f(\cdot, \cdot)$ is a learnable attention function, typically implemented as a single-layer feed-forward neural network.

GraphConv

GraphConv, introduced by Morris [2019], is a more general graph convolution operation that combines both local and global graph properties. The GraphConv layer is defined as:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\mathbf{D}^{-1}\mathbf{A}\mathbf{H}^{(\ell)}\mathbf{W}_1^{(\ell)} + \mathbf{H}^{(\ell)}\mathbf{W}_2^{(\ell)}\right) \tag{3.5}$$

where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$, $\mathbf{W}_1^{(\ell)}$ and $\mathbf{W}_2^{(\ell)}$ are learnable weight matrices. This formulation allows GraphConv to capture both the local neighborhood information (through $\mathbf{D}^{-1}\mathbf{A}\mathbf{H}^{(\ell)}\mathbf{W}_1^{(\ell)}$) and the node's own features (through $\mathbf{H}^{(\ell)}\mathbf{W}_2^{(\ell)}$).

3.2.2 Graph Pooling

While graph convolution excels at extracting local features, graph pooling operations are essential for capturing hierarchical structures and reducing the graph's size for more efficient processing [Hamilton, 2017; Xu, 2019; Bianchi, 2023; Grattarola, 2024].

For graph-level tasks, it is necessary to aggregate information from all nodes in the graph. This aggregation process, often referred to as readout, must be at least permutation invariant or equivariant to be independent of the ordering of the nodes.

Permutation Invariance and Equivariance

Permutation invariance ensures that the output of the pooling operation remains unchanged regardless of the order in which nodes are processed. This property is crucial in graph-level tasks, as usually there is no inherent ordering of nodes in a graph. Formally, for a pooling function $\bigoplus(\{\cdot\})$ and any permutation π, permutation invariance is defined as:

$$\bigoplus(\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}) = \bigoplus(\{\mathbf{h}_{\pi(1)}, \mathbf{h}_{\pi(2)}, \ldots, \mathbf{h}_{\pi(n)}\}) \tag{3.6}$$

Permutation equivariance, on the other hand, ensures that permuting the input results in an equivalent permutation of the output. This property is important for maintaining the structural information of the graph, and it is defined as:

$$\bigoplus(\{\mathbf{h}_{\pi(1)}, \mathbf{h}_{\pi(2)}, \ldots, \mathbf{h}_{\pi(n)}\}) = \pi(\bigoplus(\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\})) \tag{3.7}$$

Standard Aggregation Functions

Several simple—yet effective—permutation-invariant aggregation functions have been widely used in the GNN community [Xu, 2019; You, 2020]:

- **Sum Aggregation:** $f_{\text{sum}}(\{\mathbf{h}_1, \ldots, \mathbf{h}_n\}) = \sum_{i=1}^{n} \mathbf{h}_i$
- **Mean Aggregation:** $f_{\text{mean}}(\{\mathbf{h}_1, \ldots, \mathbf{h}_n\}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{h}_i$
- **Max Aggregation:** $f_{\text{max}}(\{\mathbf{h}_1, \ldots, \mathbf{h}_n\}) = \max_{i=1,\ldots,n} \mathbf{h}_i$

Each of these aggregation functions has its own strengths. Sum aggregation enables the learning of structural graph properties [Xu, 2019], making it useful for capturing overall patterns in the graph. Mean aggregation captures the distribution of elements, while Max aggregation is advantageous for identifying representative elements [Hamilton, 2017]. In practice, these aggregations are often used jointly [Corso, 2020].

Advanced Pooling Approaches

Recent research has explored more sophisticated pooling methods to enhance the expressive power of GNNs, enabling the network to learn increasingly abstract and coarser representations of the input graphs. For an overview of advanced pooling methods, we refer to Grattarola [2024], which proposes a formal characterization based on three main operations: selection, reduction, and connection (SRC). Such a framework unifies various pooling methods under a common structure, and they are characterized based on their trainability, density, adaptability, and hierarchy. The authors provide a systematic way to compare and analyze different pooling approaches.

Likewise, Bianchi [2023] focuses on the expressive power of pooling. They derive sufficient conditions for a pooling operator to fully preserve the expressive power of the message-passing layers preceding it, yielding a theoretically grounded criterion for evaluating and designing pooling operators. The authors analyze several existing pooling methods based on these expressiveness conditions, identifying which operators maintain or potentially compromise the GNN's original expressive power.

The choice of pooling operators is not trivial and can significantly affect the performance and capabilities of a GNN. They depend on the specific task, dataset characteristics, and computational constraints. As we progress in presenting our research, we will explore how different pooling strategies perform on common benchmark datasets and our specific HR graph structures. In Chapter 4, we present a topological-based pooling based on probability theory.

3.2.3 Explaining Graph Predictions

As already noted in the previous sections, the influence of ML models in critical decision-making processes across various domains, has brought the need for model interpretability to the forefront of AI research [Doshi-Velez, 2017]. Therefore, the past decade has witnessed a proliferation of methods aimed at explaining GNN predictions as well [Amara, 2022]. These approaches span a wide range of techniques, each with its own strengths and limitations.

Perturbation-based methods assess the relevance of graph components by observing changes in model output when elements are modified or removed [Ying, 2019; Luo, 2020; Schlichtkrull, 2021; Yuan, 2021]. These often employ masking techniques for nodes or edges to simulate the absence of certain information. Gradient-based methods, on the other hand, leverage the model's gradients to attribute importance to input features or graph structures [Pope, 2019; Sánchez-Lengeling, 2020; Schnake, 2022].

Surrogate model approaches aim to approximate the complex GNN with simpler, interpretable models [Vu, 2020; Y. Zhang, 2021; Q. Huang, 2023; Xiong, 2023]. These methods trade some fidelity for increased interpretability, offering a balance between accuracy and explainability.

Game theoretical approaches, particularly those based on the Shapley Value (SV) [Shapley, 1953], have found wide application in XAI. In the context of GNNs, the SV has been used to assess the quality of subgraph explanations, approximate feature importance, and explain pre-defined graph motifs [Duval, 2021; Yuan, 2021; Perotti, 2022; S. Zhang, 2022; Ye, 2023; Akkas, 2024]. Extensions and variants of the SV, such as the Myerson value [Myerson, 1977] for graph-restricted games, and Shapley-Taylor indices for higher-order interactions [Sundararajan, 2020a], have also been explored in the GNN context, offering more nuanced interpretations of model behavior.

Despite these advancements, several challenges remain in applying XAI methods to GNNs. Scalability is a significant concern, particularly for game-theoretical approaches when dealing with large graphs. The computational complexity of these methods can make them impractical for real-time applications, as in HR systems. Moreover, translating graph-based explanations into meaningful insights for HR professionals requires further research to bridge the gap between technical explanations and domain-specific interpretations. The ability to explain AI-driven decisions is not just a matter of transparency but often a legal requirement. Enhancing the explainability of AI models used in HR applications is crucial for providing clear, understandable explanations of selection decisions that can support recruiters, while addressing potential biases, ensuring fairness in the selection process, and building trust. Although our work in Chapter 6 proposes a novel game-theoretical approach for graph-based explanations, the field of XAI in HR contexts requires further development to meet all these specific needs.

Part II

Theoretical Contributions

4

Topological-based Aggregation for GNN

Graph Neural Networks offer a powerful means for learning the representations of graph nodes, which is particularly useful in graph analysis tasks like predicting node properties. Additionally, representations at the node level can be combined to form a comprehensive graph-level representation and predictor. This chapter investigates a novel method for defining the aggregation function, diverging from traditional approaches[1]. We introduce a graph aggregator leveraging Generative Topographic Mapping (GTM) to convert a collection of node-level representations into a single graph-level representation. The incorporation of GTM into a GNN architecture facilitates the estimation of probability densities for node representations, mapping them into a lower-dimensional space while preserving their inherent similarity and topological structure. This integration is supported by an innovative training procedure explicitly designed to learn from these dimensionally reduced representations, rather than the full initial data. Experimental evaluations across various graph classification datasets demonstrate that this methodology attains competitive predictive performance compared to conventional aggregation architectures, all while supported by a robust theoretical basis.

[1] This chapter is based on **Frazzetto, Paolo**, Luca Pasa, Nicolò Navarin, and Alessandro Sperduti [2023a]. "Topology preserving maps as aggregations for Graph Convolutional Neural Networks". In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC 2023, Tallinn, Estonia, March 27-31, 2023*. ACM, pp. 536–543. DOI: 10.1145/3555776.3577751

4.1 Motivation

Section 2.4 presents how a graph convolution layer receives in input a representation for each node in a graph, and it computes a new representation for each node that also considers the neighboring nodes. The core property of graph convolutions is that isomorphic graphs (i.e., graphs representing the same relationship among nodes) should produce the same node representations. To date, no polynomial-time algorithms can decide if two graphs are isomorphic. Thus, this property has to be verified by design. In the setting where the graph representation is exploited to represent samples (abstracted as nodes) that are not i.i.d., i.e., that are in relation one with each other (abstracted as edges), graph convolution is a powerful tool to generate node representations and node-level predictions. However, in the alternative, but not less common, setting in which each training example is represented as a distinct graph and the prediction has to be performed at the graph level (e.g., predicting properties of chemical compounds, each one represented as a different graph), another non-trivial representation issue arises: it is necessary to define an *aggregation*, or *pooling* operator associating a single representation for the whole graph (Eq. (2.3)).

The definition of the aggregation function is not trivial for three main reasons: first, it has to map a variable number of node representations into a single (preferably fixed-size) graph-level one; second, it should be independent of the node ordering, that is it should be a graph invariant (isomorphic graphs should produce the same representation), and third, we would like the representations of similar graphs (e.g., a graph $\mathcal{G}^{(1)}$ that is a subgraph of another graph $\mathcal{G}^{(2)}$) to be similar.

The simplest approach that is commonly adopted in literature is to consider commutative *global* aggregation functions such as the element-wise sum, mean, or maximum. However, it has been shown in [Navarin, 2019] that using such simple aggregations inevitably results in a loss of information, possibly impacting the predictive performance of the GNN architecture. More complex, non-linear aggregations have thus been proposed in the literature (Section 3.2.2).

Another approach consists in treating the node representations as elements belonging to an unordered set [Vinyals, 2016] and producing an order-invariant representation from them. In this setting, Deep Sets [Zaheer, 2017] is a general framework to define a universal approximator of functions over sets that has been adopted as graph aggregation [Navarin, 2019]. It has been proved that under some hypothesis, any function $f(X)$ over a set $X = \{x_1, \dots, x_M\}, x_m \in \mathfrak{X}$ can be decomposed as $f(X) = \rho\left(\sum_{x \in X} \phi(x)\right)$ for suitable transformations $\rho(\cdot)$ and $\phi(\cdot)$.

The SOM-based aggregator [Pasa, 2020] implements the $\phi(\cdot)$ function of Deep Sets exploiting a self-organizing map (SOM) [Kohonen, 1982] to map the node representations in the space defined by the activations of the SOM neurons. The resulting representation embeds information about the similarity between the various inputs. In fact, similar input structures will be mapped in similar output representations (i.e., node embeddings). The SOM is then followed by a Graph Convolution layer to partially incorporate the task supervision in the $\phi(\cdot)$ function. However, SOMs suffer from some relevant drawbacks, such as lacking an associated cost function and general proof of convergence. Because of that, it is also difficult to control the outcome of the learning process, which is driven by many heuristics requiring a careful setting of the hyperparameters, such as the shape of the function governing the width of the neighborhood used during training. While in general this may not be a practical issue, we show that in some cases the SOM-based aggregation scheme exhibits performances that are below state of the art.

In this chapter, we address these issues by developing an alternative aggregation function $\phi(\cdot)$ that is based on a principled probabilistic model, namely the Generative Topographic Mapping (GTM) [Bishop, 1998b]. Specifically, by adopting this approach, we are able to have better control of the hyperparameters defining the projection of the node representations on the 2-dimensional GTM probabilistic latent space. This should make the training procedure more effective, leading to better identification of the node representations manifold and consequently to more expressive graph-level hidden representations. In fact, contrarily to the SOM where only one winning neuron gets activated for the whole map for each input node (yielding to a global smoothing), the GTM grid of normal distributions enables a coarser transformation that preserves local structures of the representation. These transformed representations are then exploited with a dedicated training procedure, on which various pooling techniques can be applied [Bianchi, 2023; Grattarola, 2024]. An additional feature of the proposed aggregation function is the amenability to a direct inspection of the internal representations of the model *that are used to produce the output*: the GTM latent space is organized in a 2-dimensional grid that can be easily plotted and whose corresponding values have a precise probabilistic meaning. Moreover, the model uses the internal representations directly to produce the output, and they are not obtained a posteriori like other dimensionality reduction method that inevitably produces artifacts.

4.2 Generative Topographic Mapping

In more detail, the GTM algorithm [Bishop, 1998a,b] is a form of non-linear latent variable model which is based on a constrained mixture of Gaussians, whose parameters can be optimized using Expectation-Maximization (EM) [Dempster, 1977]. Formally, given a dataset $\boldsymbol{\mathcal{X}}$ of N data points $\mathbf{x}_i \in \mathbb{R}^D$, the goal of a latent variable model is to find a representation for the distribution $p(\mathbf{x})$ of data in a D-dimensional space with respect to latent variables $\mathbf{u}$ embedded in a L-dimensional latent space, where $L \ll D$. A schematic illustration of a GTM's workings is provided in Fig. 4.1. The GTM is built by first introducing a regular grid of K nodes $\mathbf{u}_i$ in the latent space, labeled by the index $i = 1, 2, \ldots, K$, and a set of M fixed non-linear radial basis functions (RBFs): $\boldsymbol{\phi}(\mathbf{u}) = \{\phi_j(\mathbf{u})\}$, with $j = 1, 2, \ldots, M$. Using the RBFs, it is possible to define a generalized linear regression model from the latent space to the data space [Park, 1991], such that each point $\mathbf{u}$ in latent space is mapped to a corresponding point $\mathbf{y}$ in the D-dimensional data space

$$\mathbf{y}(\mathbf{u}, \mathbf{W}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{u}), \tag{4.1}$$

where $\mathbf{W}$ is a $D \times M$ matrix of learnable weight parameters.

In this fashion, each node $\mathbf{u}_i$ is projected to a D-dimensional reference vector $\mathbf{m}_i = \mathbf{W}\boldsymbol{\phi}(\mathbf{u})$, and if we set a prior distribution on the latent space nodes $p(\mathbf{u})$ this mapping will also induce a corresponding distribution in the data space $p(\mathbf{y}|\mathbf{W})$ confined in a L-dimensional manifold. Since in reality the dataset $\boldsymbol{\mathcal{X}}$ will only approximately lay on a lower-dimensional manifold, it is appropriate to include a noise model for the $\mathbf{x}$ vectors. Therefore, we assume that $\mathbf{x}$, for a given $\mathbf{u}$ and $\mathbf{W}$, is distributed as a radially-symmetric Gaussian centred on $\mathbf{y}(\mathbf{u}, \mathbf{W})$ and having variance β^{-1}:

$$p(\mathbf{x}|\mathbf{u}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{ -\frac{\beta}{2}\|\mathbf{y}(\mathbf{u}, \mathbf{W}) - \mathbf{x}\|^2 \right\}. \tag{4.2}$$

By marginalizing over $p(\mathbf{u})$

$$p(\mathbf{x}|\mathbf{W}, \beta) = \int p(\mathbf{x}|\mathbf{u}, \mathbf{W}, \beta)p(\mathbf{u})d\mathbf{u}$$

and by choosing the prior distribution $p(\mathbf{u})$ to be a superposition of delta functions located at the K nodes of the regular grid in latent space (which is equivalent to the prior probabilities of each of the components are assumed to be constant and equal to $1/K$), the distribution in the data space can be expressed as $p(\mathbf{x}|\mathbf{W},\beta) = \frac{1}{K}\sum_{i=1}^{K} p(\mathbf{x}|\mathbf{u}_i,\mathbf{W},\beta)$. The posterior probabilities of the latent variables (or *responsibilities* R_i) given an input $\mathbf{x}$ can be computed by applying Bayes' theorem:

$$R_i(\mathbf{x};\mathbf{W},\beta) = p(\mathbf{u}_i|\mathbf{x},\mathbf{W},\beta) = \frac{\exp\{-\frac{\beta}{2}\|\mathbf{m}_i - \mathbf{x}\|^2\}}{\sum_{j=1}^{K}\exp\{-\frac{\beta}{2}\|\mathbf{m}_j - \mathbf{x}\|^2\}} \tag{4.3}$$

and the final response as $R(\mathbf{x};\mathbf{W},\beta) = \sum_i p(\mathbf{u}_i|\mathbf{x},\mathbf{W},\beta)$.

Since the GTM represents a parametric probability density model, it can be fitted to the dataset $\mathcal{X}$ by computing the optimal parameters $\mathbf{W}$ and β^{-1} via likelihood maximization. The log-likelihood function is given by

$$\mathcal{L}(\mathbf{W},\beta) = \sum_{n=1}^{N}\ln(p(\mathbf{x}_n|\mathbf{W},\beta)) = \sum_{n=1}^{N}\ln\left\{\frac{1}{K}\sum_{i=1}^{K}p(\mathbf{x}_n|\mathbf{u}_i,\mathbf{W},\beta)\right\}, \tag{4.4}$$

to which a regularization term can be added to reduce overfitting and improve convergence, e.g. by choosing a Gaussian prior over the weights governed by a hyperparameter $\lambda \in \mathbb{R}$. Standard optimization techniques can maximize the resulting loss function. Still, since we are dealing with a latent variable model, a viable approach is to employ the EM algorithm [Dempster, 1977]. Significant performance improvements in training can be achieved by updating the parameters incrementally using data in smaller batches [Bishop, 1998a], which is particularly suited for deep-learning applications and thus adopted in the following.

Notice that for the particular noise model given by Eq. (4.2), the distribution $p(\mathbf{x}|\mathbf{W},\beta)$ indeed corresponds to a *constrained* Gaussian mixture model since the centers of the Gaussians, i.e., $\mathbf{y}(\mathbf{u_i},\mathbf{W})$, cannot move independently but instead are adjusted indirectly through changes to the weight matrix $\mathbf{W}$. Besides, the projected points $\mathbf{m}_i$ will necessarily have a topographic ordering in the sense that any two points $\mathbf{u}_A$ and $\mathbf{u}_B$ which are close in the latent space are mapped to points $\mathbf{m_A}$ and $\mathbf{m_B}$ which are also close in the data space.

4.3 GTM-based Aggregation Function

The Generative Topographic Mapping can be employed to effectively transform data from a high-dimensional space into a low-dimensional latent space while retaining the intrinsic properties of the dataset probability distribution $p(\mathbf{x})$. Additionally, the fact that GTM preserves the topological ordering guarantees that similar node representations are mapped into similar

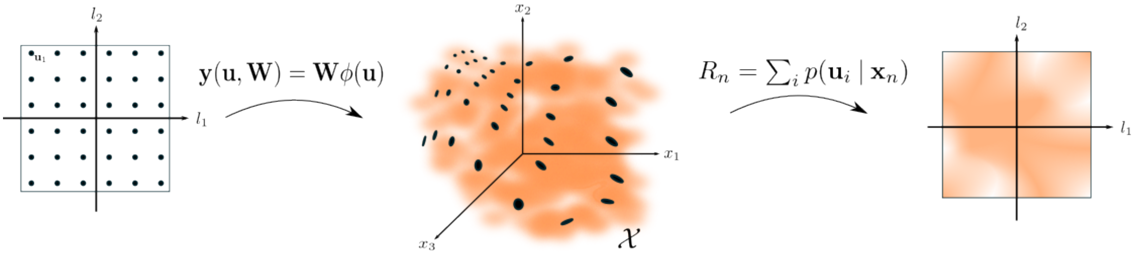

l_2
$\mathbf{u}_1$
l_1
$\mathbf{y}(\mathbf{u}, \mathbf{W}) = \mathbf{W}\phi(\mathbf{u})$
x_2
x_1
x_3
$\mathcal{X}$
$R_n = \sum_i p(\mathbf{u}_i \mid \mathbf{x}_n)$
l_2
l_1

Figure 4.1: The GTM first considers a distribution of superposition of delta functions centered at K nodes of a regular array (left). Each node $\mathbf{u}_i$ is projected into the data space, where it becomes the center of a Gaussian distribution. Then, these projections are fitted to the data manifold $\mathcal{X}$ (center), and thanks to Bayes' Theorem, the posterior distribution in the latent space is retrieved (right).

distributions in the lower-dimensional space. This method of feature extraction can be integrated into a GNN pipeline since the GTM can be well exerted as unsupervised dimensionality reduction of the graph's node representations $\mathbf{h}_v$ before being aggregated.

In this section, we describe an implementation of a GTM-based aggregation function for a GNN, or briefly GTM-GNN. Firstly we will focus on its architecture and components, and then we will describe the dedicated training procedure to learn from labeled data.

4.3.1 Architecture

The proposed architecture of the GTM-GNN is made of three main components: a *Graph Convolutional* part, the GTM-based aggregator, and a *Readout* module for the final graph classification task. A graphical rendering of the architecture is reported in Fig. 4.2.

Let us now illustrate each component of the model in detail. First, an amount d of stacked graphs convolutional layers learn stable node representations from the input dataset $\mathcal{X}$. For this implementation, we opted for *GraphConv* [Defferrard, 2016], due to its wide adoption and convincing performances, and we chose the LeakyReLU as activation function σ. All the d convolutional layers are followed by a batch normalization layer. We dubbed to output of the i^{th} graph convolutional layer as:

$$\mathbf{h}_v^{GC(i)} = \sigma\Big(\mathit{GraphConv}\Big(\mathbf{h}_v^{GC(i-1)}, \Big\{\mathbf{h}_u^{GC(i-1)} \,:\, u \in \mathcal{N}_v\Big\}\Big)\Big) \tag{4.5}$$

for $1 < i \leq d$, while the first layer directly acts on the input data. We refer to the learnable parameters of this initial *Graph Convolutional* module as $\boldsymbol{\theta}^{GC}$. The enriched node embeddings $\mathbf{h}_v^{GC(i)}$ for each layer are the one-to-one input of d independent GTMs, that constitute the

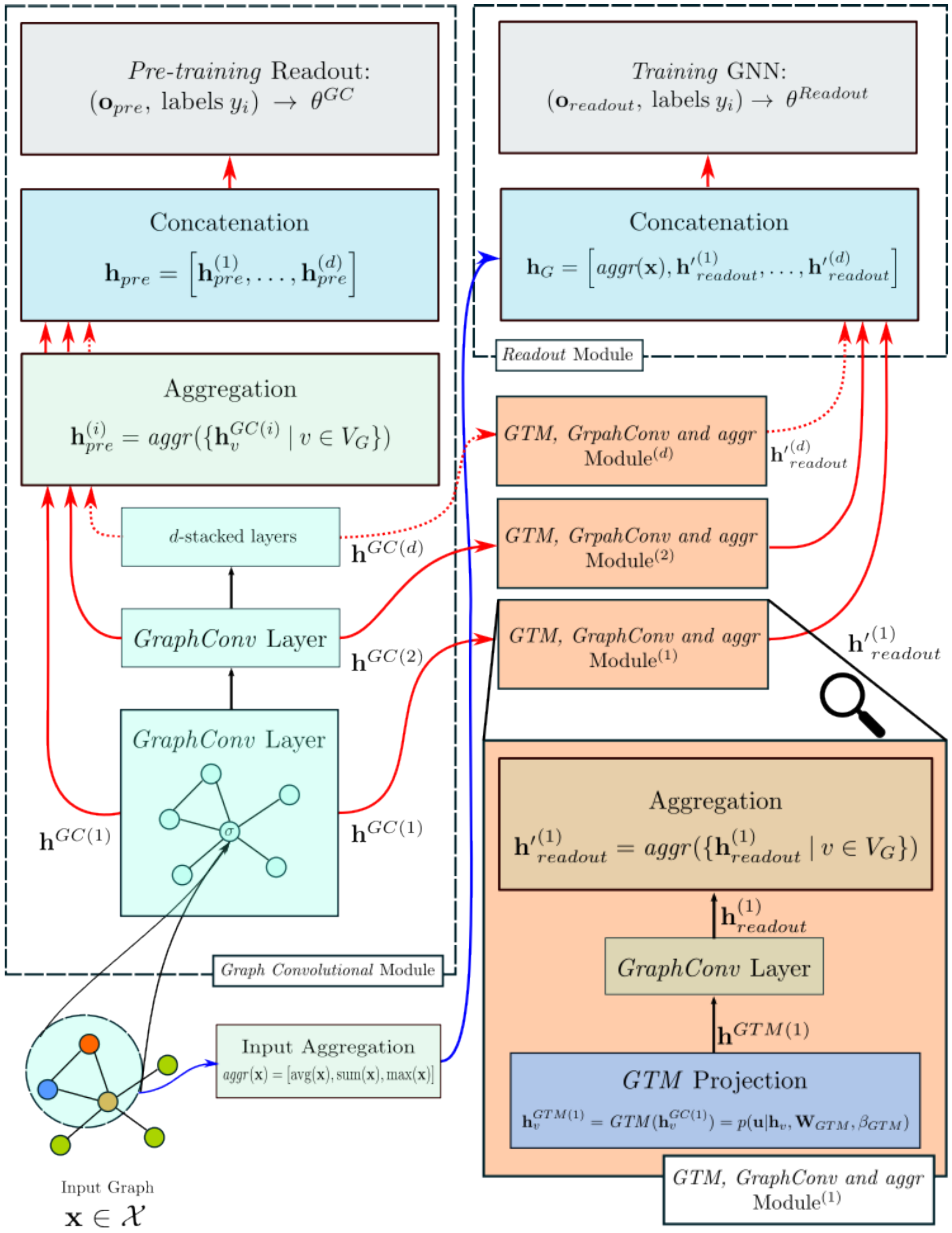

Pre-training Readout:
$(\mathbf{o}_{pre}, \text{ labels } y_i) \rightarrow \theta^{GC}$
Concatenation
$\mathbf{h}_{pre} = \left[\mathbf{h}^{(1)}_{pre}, \ldots, \mathbf{h}^{(d)}_{pre}\right]$
Aggregation
$\mathbf{h}^{(i)}_{pre} = aggr(\{\mathbf{h}^{GC(i)}_v \mid v \in V_G\})$
d-stacked layers
$\mathbf{h}^{GC(d)}$
GraphConv Layer
$\mathbf{h}^{GC(2)}$
GraphConv Layer
σ
$\mathbf{h}^{GC(1)}$
$\mathbf{h}^{GC(1)}$
Graph Convolutional Module
Input Aggregation
$aggr(\mathbf{x}) = [\text{avg}(\mathbf{x}), \text{sum}(\mathbf{x}), \text{max}(\mathbf{x})]$
Input Graph
$\mathbf{x} \in \mathcal{X}$
Training GNN:
$(\mathbf{o}_{readout}, \text{ labels } y_i) \rightarrow \theta^{Readout}$
Concatenation
$\mathbf{h}_G = \left[aggr(\mathbf{x}), \mathbf{h'}^{(1)}_{readout}, \ldots, \mathbf{h'}^{(d)}_{readout}\right]$
Readout Module
GTM, GrpahConv and aggr Module$^{(d)}$
$\mathbf{h'}^{(d)}_{readout}$
GTM, GrpahConv and aggr Module$^{(2)}$
GTM, GraphConv and aggr Module$^{(1)}$
$\mathbf{h'}^{(1)}_{readout}$
Aggregation
$\mathbf{h'}^{(1)}_{readout} = aggr(\{\mathbf{h}^{(1)}_{readout} \mid v \in V_G\})$
$\mathbf{h}^{(1)}_{readout}$
GraphConv Layer
$\mathbf{h}^{GTM(1)}$
GTM Projection
$\mathbf{h}^{GTM(1)}_v = GTM(\mathbf{h}^{GC(1)}_v) = p(\mathbf{u} \mid \mathbf{h}_v, \mathbf{W}_{GTM}, \beta_{GTM})$
GTM, GraphConv and aggr Module$^{(1)}$

Figure 4.2: Graphical representation of the GTM-based GNN architecture. On the left side, the input graph goes first to the *Graph Convolutional* module, which is trained independently on the *Pre-training* Readout layer. Then, its representations are used to train the GTMs (bottom right), and their projection is learned by a *Readout* module. Finally, all the parameters but the GTMs' are adjusted in a fine-tuning step.

aggregator module. Recall that the representations $\mathbf{h}_v^{GC(i)}$ are vectors in a high-dimensional space, whose size is governed by the number of neurons of each *GraphConv* layer. Additionally, to improve numerical stability, these representations are mapped in $[-1, 1]$ by applying the hyperbolic tangent, $\hat{\mathbf{h}}_v^{GC(i)} = \tanh\left(\mathbf{h}_v^{GC(i)}\right)$. The GTM parameters $\boldsymbol{\theta}^{GTM(i)} = \{\mathbf{W}_i, \beta_i\}$ are optimized via the EM algorithm and, once convergence has been reached, the GTMs are exploited to project the input vectors $\hat{\mathbf{h}}_v^{GC(i)}$ into the L-dimensional latent lattice, returning the posterior distribution $\forall v \in V$:

$$\mathbf{h}_v^{GTM(i)} = GTM_{(i)}\left(\hat{\mathbf{h}}_v^{GC(i)}\right) = p(\mathbf{u}|\hat{\mathbf{h}}_v^{GC(i)}, \mathbf{W}_i, \beta_i). \tag{4.6}$$

The components of $\mathbf{h}_v^{GTM(i)}$ are then normalized to reduce the variability of node representations in the same graph, $\hat{\mathbf{h}}_v^{GTM(i)} = \mathbf{h}_v^{GTM(i)} / \xi_v^{GTM(i)}$, where $\xi_v^{GTM(i)}$ is the maximum value among the components of $\mathbf{h}_v^{GTM(i)}$. In the third module, called *Readout* and defined by the parameters $\boldsymbol{\theta}^{Readout}$, each $\hat{\mathbf{h}}_v^{GTM(i)}$ is fed through another *GraphConv* layer so that the graph topology is brought back:

$$\mathbf{h}_{readout}^{(i)} = \sigma\left(GraphConv\left(\hat{\mathbf{h}}_v^{GTM(i)}, \left\{\hat{\mathbf{h}}_u^{GTM(i)} \,:\, u \in \mathcal{N}_v\right\}\right)\right). \tag{4.7}$$

Then, these transformed representations are aggregated by taking the concatenation of their average, sum, and component-wise maximum [Ying, 2018]:

$$\mathbf{h}'^{(i)}_{readout} = \left[\text{avg}\left(\mathbf{h}_{readout}^{(i)}\right), \text{sum}\left(\mathbf{h}_{readout}^{(i)}\right), \max\left(\mathbf{h}_{readout}^{(i)}\right)\right]. \tag{4.8}$$

All d feature maps are concatenated to obtain one single graph-level representation $\mathbf{h}_G$. Additionally, in $\mathbf{h}_G$ we included the aggregation via average, sum, and component-wise maximum of the features of the input nodes $\mathbf{X}$, so as to fully exploit the information associated with each node of the graph:

$$\mathbf{h}_G = \left[\text{avg}(\mathbf{X}), \text{sum}(\mathbf{X}), \max(\mathbf{X}), \mathbf{h}'^{(1)}_{readout}, \ldots, \mathbf{h}'^{(d)}_{readout}\right]. \tag{4.9}$$

Eventually, this graph-level representation is ready for the output layer and the supervised learning task, achieved by means of an MLP with an output function:

$$\mathbf{o}_{readout} = \text{LogSoftMax}(\text{MLP}(\mathbf{h}_G)). \tag{4.10}$$

4.3.2 Training Procedure

To conciliate the unsupervised framework of the GTMs with the supervised task of graph classification, the training of the GTM-based GNN takes four steps that are carried out one after the other, optimizing in each turn different sets of learnable parameters.

Pre-training: The first training step consists in optimizing the parameters $\boldsymbol{\theta}^{GC}$ by adding an ad-hoc readout layer, which we indicate as *pre-training* readout, to perform supervised learning with standard backpropagation. This pre-training readout layer further aggregates the node representations $\mathbf{h}_v^{GC(i)}$ by concatenating their average, sum, and component-wise maximum (as in Eq. (4.8)). Then, it stacks these vectors for all the d layers, applies a linear transformation, and the LogSoftmax activation function in the end. This permits the training of the graph convolutional module separately from the rest of the network, which learns stable node representations $\mathbf{h}_v^{GC(i)}$ that are later fed to the GTM module. Finally, the pre-training readout layer is discarded and thus will not make part of the final model.

GTM Training: The parameters $\boldsymbol{\theta}^{GTM}$ of the GTMs are initialized using the first two principal components of the node representations PCA [Bishop, 1998b]. Then, they are optimized via the EM algorithm with respect to the likelihood of Eq. (4.4).

Readout Training; Next, the parameters $\boldsymbol{\theta}^{Readout}$ are trained via backpropagation with reference to the negative log-likelihood loss on $\mathbf{o}_{readout}$ for the c-class graph classification.

Fine-Tuning: Finally, the last training step consists of a *fine-tuning* phase. The purpose of this step is to tune the model parameters $\boldsymbol{\theta}^{GC}$ and $\boldsymbol{\theta}^{Readout}$ while maintaining the $\boldsymbol{\theta}^{GTM}$ fixed. Additional cycles of adaptation could take place by retraining the GTMs while fixing the rest of the network, and so on; however, we did not investigate this scenario further.

The pseudo-code that summarizes the training procedure is reported in Algorithm 4.1.

4.4 Experimental Results

In this section, the model setup is presented, and the results obtained by the GNN that exploits the proposed GTM-based aggregation are discussed.

4.4.1 Setup and Hyperparameters

As already mentioned, the GTM-GNN is made of three main parts, i.e the *Graph Convolutional* section, the GTM-based aggregator, and the *Readout* module. For what concerns the

Algorithm 4.1 GTM-based GNN Training Procedure

```
Input: Graphs Dataset $\mathcal{X}$ with associated labels $y$
$\boldsymbol{\theta}^{GC}, \mathbf{o}_{pre} \leftarrow Pretraining(\mathcal{X}, y)$    ▷ Pre-training of GraphConv Module
$\forall$ stacked layer $i$ : $\mathbf{h}^{GC(i)} \leftarrow \sigma\Big(GraphConv\Big(\mathbf{h}_v^{GC(i-1)}\Big)\Big)$    ▷ Get stable representations from pre-training module
$\boldsymbol{\theta}^{GTM(i)} \leftarrow GTM\_training\big(tanh\big(\mathbf{h}^{GC(i)}\big)\big)$    ▷ Expectation Maximization
$\mathbf{h}^{GTM(i)} \leftarrow GTM\big(tanh\big(\mathbf{h}^{GC(i)}\big), \boldsymbol{\theta}^{GTM}\big)$    ▷ Projected representations
$\forall v : \hat{\mathbf{h}}_v^{GTM(i)} \leftarrow \frac{\mathbf{h}_v^{GTM(i)}}{\xi_v^{GTM(i)}}$, where $\xi_v^{GTM(i)}$ is the maximum of $\mathbf{h}_v^{GTM(i)}$ components
$\forall i : \mathbf{h}_{readout}^{(i)} \leftarrow \sigma\Big(GraphConv\Big(\hat{\mathbf{h}}_v^{GTM(i)}\Big)\Big)$    ▷ Readout Module
$\forall i : \mathbf{h'}_{readout}^{(i)} \leftarrow aggr\Big(\Big\{\mathbf{h}_{readout}^{(i)} \mid v \in V_G\Big\}\Big)$    ▷ Aggregation
$\mathbf{h}_G \leftarrow \Big[\text{avg}(\mathbf{X}), \text{sum}(\mathbf{X}), \max(\mathbf{X}), \mathbf{h'}_{readout}^{(1)}, \ldots, \mathbf{h'}_{readout}^{(d)}\Big]$    ▷ Concatenation
$\mathbf{o}_{readout} \leftarrow LogSoftMax(MLP(\mathbf{h}_G))$
$\boldsymbol{\theta}^{readout} \leftarrow Backprop(\mathcal{X}, y; \mathbf{o}_{readout})$    ▷ Readout training
$\boldsymbol{\theta}^{GC}, \boldsymbol{\theta}^{readout} \leftarrow FineTuning\big(\mathcal{X}, y; \boldsymbol{\theta}^{GC}, \boldsymbol{\theta}^{readout}\big)$    ▷ Fine-Tuning training
Output: GTM-based GNN$\big(\boldsymbol{\theta}^{GC}, \boldsymbol{\theta}^{GTM}, \boldsymbol{\theta}^{readout}\big)$
```

first *Graph Convolutional* module, we set at $d = 3$ the number of hidden layers and select *GraphConv* as the convolutional operator. In relation to the relative size of these layers, we opted for a "funnel" architecture [Navarin, 2020] in the sense that the *GraphConv* layers have an increasing number of neurons, namely $\mathbf{h}^{GC(1)} \in \mathbb{R}^{l}$, $\mathbf{h}^{GC(2)} \in \mathbb{R}^{2l}$ and $\mathbf{h}^{GC(3)} \in \mathbb{R}^{3l}$, where the size l is a hyperparameter. This architecture has been proven to improve performance, and therefore it is adopted in the GTM-GNN. Both the Graph Convolutional and the Readout module are trained via backpropagation using the AdamW optimizer [Loshchilov, 2019].

The goal of this work is to evaluate the benefit of using the GTM-based aggregation; therefore, we focused our attention on the behavior of the GTM parameters. For all GTMs, we set the latent variable dimension to $L = 2$, so that the K latent variables $\mathbf{u}_i$ lay in a bi-dimensional plane. Both their amount in the *width* and the *height* dimensions of the regular grid are hyperparameters (in this way $K = height \times width$), and the grid itself is built accordingly within a bounded $[-1, 1] \times [-1, 1]$ plane when the GTMs are initialized. Other two relevant hyperparameters concern the RBFs $\boldsymbol{\phi}(\mathbf{u})$, namely their amount M and variance σ. The former forms a $M \times M$ regular grid of RBF center points that is overlayed to the latent variables grid in the $[-1, 1]^2$ plane. On the other hand, the variance σ can be tested for any value or can be computed as the average minimum distance among the aforementioned RBF centers. Finally, as soon as the latent nodes grid and RBF function are set, the matrix $\mathbf{\Phi}_{ij} = \phi_j(\mathbf{u}_i)$ is computed, and we pad a bias column of $\mathbf{1}$ to it. Notice that this step is done only once at initialization. The parameters $\mathbf{W}$ and β can be either initially set at random from the standard normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$, or as explained beforehand, they are computed as to mimic the PCA applied to the whole training set. To do this, before the first epoch of the EM algorithm, the whole dataset is loaded into memory, and the PCA is performed. Random initialization was avoided since it can be numerically unstable and it takes longer for convergence. This step is also needed to determine the right size of the responsibility matrix R_{in} that is updated from the first epoch with incremental learning. The last hyperparameter is the regularization constant λ, which can take any fixed value or be equal to β^{-1}.

After the EM optimization, the posterior distribution of the input data is estimated. It is scaled by its maximum value $\xi_v^{GTM(i)}$ before being fed to the next *GraphConv* layer, so that the values are bounded in $[0, 1]$, restricting the learning of their relative scale on the lattice grid rather than absolute magnitude (being not-normalized probabilities). Eventually, the *Readout* module concatenates the three *GraphConv* outputs of the same fixed size l and supplies them to an MLP, whose depth q is also a hyperparameter.

The GTM-based GNN has been implemented with `Python 3.8.8` and `PyTorch 1.8.1` [Paszke, 2019]. We exploited two types of machines, respectively equipped with: 2 x Intel(R) Xeon(R) CPU E5-2630L v3, 192GB of RAM, an Nvidia Tesla V100, and 2 x Intel(R) Xeon(R) CPU E5-2650 v3, 160 GB of RAM, Nvidia T4. For all the other hyperparameters and implementation details, please check the publicly available code[2].

4.4.2 Datasets

All the considered methods were empirically validated on seven commonly adopted graph classification benchmarks. Namely, we used four datasets modeling bio-informatics problems: NCI1 [Wale, 2008], PROTEINS, [Borgwardt, 2005], D&D [Dobson, 2003] and ENZYMES [Borgwardt, 2005]. NCI1 involves chemical compounds represented by their molecular graph, where node labels represent the atom type, and bonds correspond to edges. In NCI1, the graphs represent anti-cancer screens for cell lung cancer. The remaining datasets, PROTEINS, D&D and ENZYMES involve graphs that represent proteins. Amino acids are represented by nodes and edges that connect amino acids that in the protein are less than 6Å apart. All the prediction tasks are binary classification tasks, except for the ENZYMES dataset, where a multi-

[2] `https://github.com/paolofraz/GTMBasedGraphAggregation`

Hyperparameter	**Values**
Latent variables grid	$(10 \times 15), (11 \times 16), (13 \times 18),$ $(15 \times 20), (20 \times 25)$
Amount of RBF M	$8, 12, 18$
Variance of RBF σ	$s, 2s, 1$
GTM Reg. λ	$10, 1, 0, 0.1, 0.01, \beta^{-1}$
MLP depth q	$1, 3, 5$
Hidden neurons l	$20, 30, 50$

Table 4.1: Hyperparameter grid for the random search cross-validations. Recall that s is the average spacing among RBF centers, e.g., $\sigma = 0.167$ for the (15×10) grid.

class classification of chemical compounds (six classes) is represented. We further considered two large social graph datasets: IMDB-B and IMDB-M. These are composed of graphs derived from actors/actresses who played in different movies collected on IMDB, together with the movie genre information. Each graph has a target that represents the movie genre. IMDB-B models a binary classification task, while IMDB-M contains graphs belonging to three classes. In contrast to the bio-informatics datasets, the nodes contained in the social datasets do not have any associated label, and therefore, only the graph topology is regarded. Relevant statistics about the datasets are reported in Table 4.3.

4.4.3 Model Selection

We run a 10-fold cross-validation for each dataset to select the best hyperparameter combination. Due to the long time requirements of performing an extensive grid search, we decided to limit the number of values taken into account for each hyperparameter, and we performed a random search over the grid of their combination.

Table 4.1 gives an overview of the arbitrarily chosen values of the GTM hyperparameters grid. Each one of the four training phases runs for 500 epochs. Moreover, to reduce overfitting on the training set, we adopted a validation-based early stopping regularization that chooses the epoch of the best performing model on the validation set, stopping the training if after 25 epochs no better result is achieved. For what concerns the GTMs, we use the complete-data log-likelihood to monitor the convergence and early stopping.

4.4.4 GNN Models Employed as Baselines

We compare the GTM-GNN with several GNN architectures which achieved state-of-the-art results on the used datasets. In the following, we describe the models considered for the experimental comparison. The first model we consider in our experimental comparison is the PSCN proposed by Niepert [2016]. PSCN follows a straightforward approach to define convolutions on graphs, which is conceptually closer to convolutions defined over images. First, it selects a fixed number of vertices from each graph, exploiting a canonical ordering on graph vertices. Then, for each vertex, it defines a fixed-size neighborhood (of vertices possibly at distance greater than one), exploiting the same ordering. This approach requires computing a canonical ordering over the vertices of each input graph, which is a problem as complex as graph isomorphism (no polynomial-time algorithm is known).

GraphSage [Hamilton, 2017] does modify the standard definition of graph convolution empowering the aggregation over the neighborhoods by using sum, mean or max-pooling operators, and then performs a linear projection in order to update the node representations. In addition to that, it exploits a particular neighbor's sampling scheme.

The convolution proposed in [Hamilton, 2017] has been extended by GIN [Xu, 2019], which introduces a more expressive aggregation function on multi-sets with the aim to overtake the limitation introduced by GraphSAGE using sum, mean or max-pooling operators.

DGCNN [Muhan Zhang, 2018] extends the GCN proposed by [Kipf, 2017] introducing a slightly different propagation scheme for vertices' representations based on random walks on the graph, and exploiting SortPooling as aggregation function. An extension of this model that exploits the DeepSet (DGCNN-DeepSet) was proposed by [Navarin, 2019].

DiffPool [Ying, 2018] is a pooling operator that leverages on hierarchical properties of the graph structure by learning a clustering module that makes the graph more and more coarse at every layer. In particular, it learns a new adjacency matrix for each layer where single nodes can be substituted by clusters (thus the size of the matrix becomes smaller at deeper layers).

The Funnel GCNN (FGCNN) model [Navarin, 2020] relies on the similarity of the adopted graph convolutional operator to the way the features of the Weisfeiler-Lehman (WL) Subtree Kernel [Shervashidze, 2011] is computed. Based on this observation, a novel WL-based loss term for the output of each convolutional layer is introduced to guide the network to reconstruct the corresponding explicit WL features. FGCNN also adopts a number of filters at each convolutional layer determined by a measure of the WL-kernel complexity.

Model/Dataset	PTC	NCI1	PROTEINS	D&D	ENZYMES	IMDB-B	IMDB-M
PSCN [Niepert, 2016]	60.00±4.82	76.34±1.68	75.00±2.51	76.27±2.64	-	71±2.29	45±2.84
FGCNN [Navarin, 2019]	58.82±1.80	81.50±0.39	74.57±0.80	77.47±0.86	-	-	-
DGCNN [Navarin, 2019]	57.14±2.19	72.97±0.87	73.96±0.41	78.09±0.72	-	-	-
DGCNN [Errica, 2020]	-	76.4±1.7	72.9±3.5	76.6±4.3	38.9±5.7	53.3±5.0	38.6±2.2
GIN [Errica, 2020]	-	80.0±1.4	73.3±4.0	75.3±2.9	**59.6±4.5**	66.8±3.9	42.2±4.6
DIFFPOOL [Errica, 2020]	-	76.9±1.9	73.7±3.5	75.0±3.5	59.5±5.6	69.3±6.1	45.1±3.2
GraphSAGE [Errica, 2020]	-	76.0±1.8	73.0±4.5	72.9±2.0	58.2±6.0	69.9±4.6	47.2±3.6
DGCNN-DeepSets [Navarin, 2019]	58.16±1.05	74.19±0.59	75.11±0.28	77.86±0.27	-	-	-
SOM-GCNN [Pasa, 2020]	62.24±1.7	**83.30±0.45**	**75.22±0.61**	78.10±0.60	50.01±2.92	67.65±1.99	**48.68±3.46**
GTM-GNN	**62.49±9.60**	82.48±1.33	72.88±4.82	**78.27±3.63**	59.03±5.92	**72.33±3.89**	47.69±4.44
GTM-GNN w/ Ablation	61.95±8.27	82.28±2.12	73.86±4.74	76.70±3.47	58.72±7.02	71.67±3.56	47.78±3.9
GTM-GNN Hyperparameters	(15×20) $q = 1$ $\lambda = 0.01$ $l = 30$ $M = 12$	(15×20) $q = 5$ $\lambda = 0.1$ $l = 50$ $M = 12$	(11×16) $q = 1$ $\lambda = 0.01$ $l = 20$ $M = 12$	(12×17) $q = 3$ $\lambda = 0.1$ $l = 50$ $M = 12$	(15×20) $q = 3$ $\lambda = 0.1$ $l = 50$ $M = 12$	(11×10) $q = 1$ $\lambda = 0.1$ $l = 20$ $M = 12$	(15×20) $q = 1$ $\lambda = 0.1$ $l = 30$ $M = 12$

Table 4.2: Accuracy of GTM-GNN and baselines models on the seven used datasets. Values for the selected latent variable grid size, depth of the readout MLP q, regularization parameter λ, amount of hidden neurons l, and number of RBF M are reported.

4.4.5 Discussion

In Table 4.2, we report the results achieved by the GNNs when the comparison among them is fair, i.e., the same validation strategy and the common settings for the input datasets are employed. The issue of experimental reproducibility and replicability in the field of GNN is crucial; therefore, we hold as baseline only the fair results that are reported in the literature [Errica, 2020]. The results reported in Table 4.2 were obtained by performing 5 runs of 10-fold cross-validation. The results reported in [Ying, 2018; T. Chen, 2019; Xu, 2019] are not considered in our comparison since the model selection strategy is different from the one we adopted, which makes the results not comparable.

The results reported in Table 4.2 show that the GTM-GNN achieved highly competitive performance in all considered datasets. In particular, on PTC, D&D, and IMDB-B the proposed method obtained higher results than the state-of-the-art, while in NCI1, PROTEINS, ENZYMES, and IMDB-M the accuracy results are higher than the ones achieved by most of the models considered in the comparison. On NCI1 and IMDB-M the GTM-GNN shows the second-best performance, and only the SOM-GCNN performs better than our proposed model. On PROTEINS, the accuracy reached by the GTM-GNN is lower than the ones obtained by many of the other considered models. The hyperparameter values selected in this case are very different than those selected on the other datasets. Indeed, the selected model is the simpler considered in our experimental assessment ($l = 20, q = 1$). Specifically, 20 is the smallest value for l we considered during the validation process. It is likely that by using smaller values for l, the GTM-GNN could reach better performances and avoid overfitting. Additionally, this dataset has a higher average degree (3.73, see Table 4.3) compared with NCI1 (2.16) and PTC (2.06); we argue that, compared with the other two datasets that have graphs sizes of the same magnitude, it could be more difficult to grasp the local features that differentiate between the two classes. Overall, the GTM-CGNN exhibits higher accuracy variances due to varying performances on each CV split. We also argue that being a probabilistic model, randomness plays a major role in the GTM component. Nevertheless, we recall that this probabilistic framework is theoretically well-founded, and more research can be done to exploit its characteristics.

Dataset	#Graphs	#Node	#Edge	Avg. #Nodes	Avg. #Edges	Avg. Degree	#Classes
PTC	344	4915	10108	14.29	14.69	2.06	2
NCI1	4110	122747	265506	29.87	32.30	2.16	2
PROTEINS	1113	43471	162088	39.06	72.82	3.73	2
D&D	1178	334925	1686092	284.32	715.66	5.03	2
ENZYMES	600	19580	74564	32.63	124.27	3.81	6
IMDB-B	1000	19773	193062	19.773	193.06	9.76	2
IMDB-M	600	19502	197806	13.00	131.87	10.14	3

Table 4.3: Datasets statistics.

4.4.6 Ablation Study

To investigate further the benefits of the GTM aggregation, we analyzed the case of the removal of the GraphConv layer after the aggregation that was originally inserted to restore the graph topology. The results of this ablation study are reported in Table 4.2. We can see that albeit the dismissal of a layer, the predictive performances do not show any significant difference compared to the full GTM-GNN architecture. Therefore, removing the last graph convolutional layer will be helpful in reducing the complexity of the GTM module. Moreover, it reduces the number of parameters and hyper-parameters that must be optimized.

4.5 SOM vs. GTM

The similarity between the GTM-GNN and SOM-GCNN makes the comparison between these two models interesting in evaluating the impact of the proposed GTM-based graph aggregator. From this perspective, it is worth noticing that the drop in accuracy on NCI1, PROTEINS, and IMDB-M is limited, while in ENZYMES the difference between SOM-GCNN and GTM-GNN models is considerable. Indeed, GTM-GNN improves the SOM-GCNN performance by almost nine percentage points.

We argue that the higher GTM results may be explained by: *(i)* its training being more theoretically grounded than the one exploited by the SOM—GTM optimization is based on the maximization of a likelihood function that can be carried out by standard optimization techniques; *(ii)* being able to represent more complex manifolds, its results are more suitable in managing multi-class classification tasks because it may be easier for the GTM to encode the differences in the data distributions of the various classes compared to SOM (see Section 4.5.1); *(iii)* the model is easier to set up since it does not require to define a neighborhood function with its respective hyper-parameters [Pasa, 2020].

4.5.1 Lattice representations

In order to investigate the reason for the GTM performance improvement compared to SOM on the ENZYMES dataset, in Fig. 4.3, we plot the heatmaps of the respective lattice representations. The heatmaps were computed following the same procedure proposed in Pasa [2020]. Each heatmap shows the average value of each neuron in the lattice (either SOM or GTM) computed over the set of graphs belonging to the same class. Thus, each heatmap represents, for each class, the average level of utilization of the different parts of the lattice, meaning that parts that are used by a single class represent discriminative areas for that class. The comparison shows that the GTM tends to create a more distributed pattern of specific areas.

Given the higher accuracy obtained by the GTM, it is clear that the learned node representations benefit from the greater expressiveness and local discriminative power of the GTM in comparison with the SOM. The better representations obtained by the GTM are also due to the lower sensibility of the GTM to the values of the hyperparameters compared with SOM. Indeed, as reported in Table 4.2, the selected latent space dimensions are similar regardless of the considered dataset/task complexity. These interesting features, related to the probabilistic definition of the GTMs, also help in having an effective training phase.

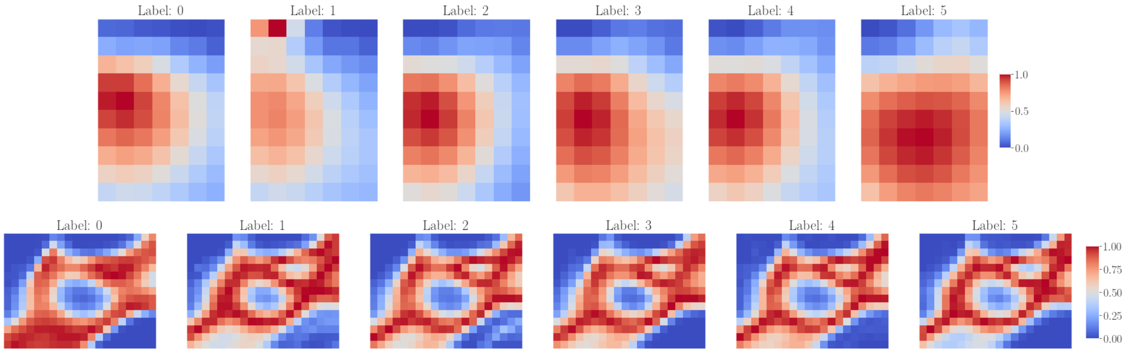


Figure 4.3: Heatmaps of first-level SOM (top row) and GTM (bottom row) projected representations for the multi-class dataset ENZYMES. Heatmaps at higher levels are similar. Each heatmap is obtained by averaging the contribution of several graphs belonging to the corresponding class.

4.5.2 End-to-end Fine Tuning

Arguably, one of the biggest downsides of having a layer trained in an unsupervised way for a supervised task, such as the SOM-GCNN, is that it is impossible to train the overall network using end-to-end backpropagation.

In this section, we show that the proposed GTM-based aggregation can also be trained end-to-end. While the aforementioned training procedure is reasonable—given the unsupervised nature of the SOM and GTM maps—thanks to the specific formulation of the GTM, it is also possible to train the map using the gradient provided by the subsequent layers. We thus propose to add a further step in the training procedure, where we fine-tune all the parameters of the network using standard backpropagation. The only modification we need to introduce for GTM is well-known when training probabilistic models using stochastic gradient descent—we have to be careful in re-normalizing the output of the GTM to ensure it remains a probability distribution. Notice that it is not possible to pursue this approach with the SOM-based formulation since it relies on an *argmax* operation that is not differentiable. Therefore, modifying the SOM formulation to exploit only differentiable operations would require defining a novel optimization algorithm since the original one is not based on the optimization of a likelihood function.

We applied this end-to-end fine-tuning on the ENZYMES dataset to see if it could improve the performances of our proposed model even more. With no additional hyper-parameter exploration, we started the fine-tuning process from the “optimal” hyper-parameters, obtaining an accuracy on the ENZYMES dataset of 60.9±6.7, where the results before end-to-end fine-tuning as reported in Table 4.2 are 59.03 ± 5.92. Even though such improvement is not statistically significant, we can see a slight increase in accuracy in all five runs thanks to the class-based supervision that is provided in this last step to the GTM representations, which are not fully unsupervised anymore.

This preliminary result suggests that a viable alternative strategy for training the GTM-based architecture can include the GTM likelihood directly in the loss function and perform a single end-to-end training phase.

4.6 Final Remarks

In this chapter, we addressed the problem of defining a more effective node aggregation function for Graph Neural Networks. Specifically, inspired by the work proposed in Pasa [2020], where the authors introduced a SOM-based graph aggregator, we developed a novel node aggregation function based on a principled probabilistic model, i.e., Generative Topographic Mapping, that owns several nice advantages over SOM: *(i)* training optimizes a well-defined cost function; *(ii)* a smaller hyperparameter space to explore for model selection; *(iii)* experimentally showed to return richer dimensionality reduction mappings, thus increasing the expressiveness of the node aggregation function that can be obtained in practice. In addition to the above advantages, the proposed approach opens the door to more interpretable GNNs since the internal 2-dimensional representations used to generate the output can be directly visualized for inspection in 2-D heatmaps. This comes without compromising the model's performance, as clearly shown by the reported state-of-the-art empirical results on seven graph-level classification tasks by a GNN exploiting the GTM-based aggregation function.

Finally, we showed how adopting the GTM aggregator enables the possibility of training the whole model in an end-to-end fashion. We exploited this option as a final fine-tuning step, but this approach will be further investigated by obtaining an aggregation layer that does not require any specific training procedure.

5 Multiplicative Integration as Graph Convolution

We have described how GNNs generate node representations using convolution operators grounded in a neighborhood aggregation methodology. Typically, a graph convolutional layer is conceptualized as additive components that merge several streams of information. Nevertheless, when integrating information in sequences, the gradient flow has demonstrated enhanced robustness through the adoption of the Multiplicative Integration (MI) technique. Consequently, it is valuable to investigate the influence of MI within GNNs[1]. We propose three distinct convolution layers designed to leverage MI with the aim of enhancing various facets of the neighborhood aggregation scheme. Finally, both theoretical and empirical analyses compare our proposed layers against the prevalent GNN operators for the graph classification task.

5.1 Motivation

GNNs define a neural architecture that follows the graph topology—from the neurons associated with a vertex and its neighbors, a hidden representation corresponds to the same ver-

[1] This chapter is based on **Frazzetto, Paolo**, Luca Pasa, Nicolò Navarin, and Alessandro Sperduti [2024a]. "Beyond the Additive Nodes' Convolutions: a Study on High-Order Multiplicative Integration". In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC 2024, Avila, Spain, April 8-12, 2024*. ACM, pp. 474–481. DOI: 10.1145/3605098.3636016

tex in another network layer. For each hidden layer within the GNN, the node representations undergo transformations that are defined by the convolution operator (Section 3.2.1). Graph Convolutions (GC) are generally predicated on a neighborhood aggregation scheme ($\oplus$ in Eq. (2.1)). This scheme takes into account, for each node, only its immediate neighbors and the various GCs proposed in literature usually exploit an additive building block.

In the structured domain, particularly in sequential learning, a different procedure of information integration has been studied: the Multiplicative Integration (MI) [Y. Wu, 2016]. The idea is that, instead of utilizing the sum operation to join the information conveyed by the various elements that compose the recurrent model equation, MI exploits the Hadamard product. Without introducing any extra parameters, the authors leverage *second-order* interactions between features, i.e., relationships or dependencies that exist *between pairs of features* within the dataset. Unlike first-order interactions, which involve individual features in isolation, second-order interactions consider how two features jointly influence the output or prediction. One of the first applications of MI on sequential domains was proposed by Goudreau [1994] that introduced the Second-Order Single-Layer Recurrent Neural Networks (Second-order SLRNN).

The most common model for sequences that adopts the MI is the LSTM [Hochreiter, 1997] (or variants of it, e.g. GRU [Cho, 2014]). This model employs the MI to implement a gating mechanism to manage long-term temporal dependencies. An enhanced version of the LSTM (and earlier, of the RNN) that exploits the MI also to define the recurrent mechanism is the multiplicative LSTM/RNN [Sutskever, 2011; Krause, 2017]. These models use the Hadamard product to combine the projection of the current time step with the hidden states that come from the previous time step. The idea of using MI to manage a gating mechanism and to combine the information flow from different temporal domains is also used in many other models, like in Highway Network [Zilly, 2017]. Another model that exploits a similar technique is the HyperNetwork [Ha, 2017]. The HyperNetwork dynamically generates the weights of a network using another (smaller) network. In particular, the recurrent version (the HyperLSTM) generates a multiplicative bias that drives the generation of dynamical weights. A similar approach that belongs to the Bayesian framework was proposed by Krueger [2017].

Graph convolution operations share some critical mechanisms with the time-based aggregation mechanism used by Recurrent Neural Networks. Indeed, in literature, the Multiplicative integration is shown to be particularly convenient to aggregate contextual information that comes from different sources [Jayakumar, 2020]. Inspired by this similarity, in this paper, we explore how MI can be applied to define novel GCs.

Some recently proposed GNN approaches do exploit some form of multiplicative mechanisms, for instance implementing gating mechanisms [Velickovic, 2018; Brockschmidt, 2020; Pasa, 2021; Tailor, 2022] or hypernetwork-like models [Ha, 2017; Pasa, 2024]. On the other hand, the recent work of Koishekenov [2023] explores how to combine features best to condition GNNs on additional information. Their "strong conditioning", is the Hadamard product between the weighted adjacency and feature matrices that replace the layers in an MLP; yet they do not exploit MI to aggregate the embeddings of the various nodes in the GNN. Furthermore, Hua, 2022 have proposed node pooling based on invariant multiplication, where they define a GNN layer that can be seen as the composition of a linear layer with a weight matrix, a multiplicative pooling layer, and another linear map. This layer is used as both the aggregation and update steps of a GNN, thus it does not explicitly leverage the graph structure, but it relies purely on higher-order feature interactions. However, these related works do not explicitly adopt the MI paradigm concerning nodes' neighborhoods.

To the best of our knowledge, this is the first paper that explores the application of MI inside a GC operator. We propose three definitions of MI-based GC operators that stem from the commonly used and very effective *GraphConv* operator [Morris, 2019]. These three operators are defined with the aim of exploring how the MI can be embedded into a graph convolution to obtain a second-order GC operator. Such second-order interactions between features can capture more intricate patterns and relationships in the data, enabling us to go beyond traditional first-order feature analysis. We empirically evaluate the proposed MI-GNNs on eight commonly adopted graph classification benchmarks. The experiments show how MI, applied in the aggregation and/or combination step, allows us to uncover hidden dependencies contributing to improved model performances. We compare the proposed methods with the most common additive graph convolutional operators. In particular, we analyze the results regarding the accuracy and computational time required for training. The results highlight how the use of MI can help obtain improved performance in terms of accuracy and speed of convergence. We apply rigorous statistical hypothesis testing to assess the statistical significance of the observed improvements. Considering that the application of MI also influences the form and the flow of the gradient of the GNN, we analyze how gradient propagation differs among multiplicative and additive GCs.

5.2 MI-GNN

The core idea of MI-GNN is to employ the Hadamard product to propagate signals in a GNN. In the following part of this section, we present three different possibilities to extend the *GraphConv* operator by Multiplicative Integration.

The first proposed version of MI-GNN (named **MI-GNN-v1**) exploits MI to integrate the information from the current node with the one from its neighborhood. This is obtained by replacing the sum of the additive version of the *GraphConv* with the Hadamard product between the current node projection and the aggregation of its neighbors:

$$\mathbf{h}_v^{(\ell)} = \phi\Big(\Big(\mathbf{W}\mathbf{h}_v^{(\ell-1)} + \mathbf{b}\Big) \odot \sum_{u \in \mathcal{N}_v} \Big(\mathbf{W}_\Sigma \mathbf{h}_u^{(\ell-1)} + \mathbf{b}_\Sigma\Big)\Big). \tag{5.1}$$

The two bias terms $\mathbf{b}$ and $\mathbf{b}_\Sigma$ are inserted to obtain a more expressive formulation. In fact, by distributing the product over the sum and rearranging the terms, we get the equivalent equation

$$\begin{aligned} \mathbf{h}_v^{(\ell)} = \phi\Big(|\mathcal{N}_v|\Big(\mathbf{W}\mathbf{h}_v^{(\ell-1)} + \mathbf{b}\Big) \odot \mathbf{b}_\Sigma + \mathbf{b} \odot \mathbf{W}_\Sigma \sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(\ell-1)} + \\ + \mathbf{W}\mathbf{h}_v^{(\ell-1)} \odot \Big(\mathbf{W}_\Sigma \sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(\ell-1)}\Big)\Big). \end{aligned} \tag{5.2}$$

Note that the first two terms of the above equation constitute a scaled version of the regular additive interaction between $\mathbf{h}_v^{(\ell-1)}$ and $\sum_u \mathbf{h}_u^{(\ell-1)}$. The scaling is dynamic since, in addition to the constant bias terms, the scaling factor also depends on the cardinality of the neighborhood. Moreover, the weights regulating the multiplicative integration between these two components, defined by the third term, are obtained by combining the two weight matrices $\mathbf{W}$ and $\mathbf{W}_\Sigma$ of the convolution, with no increase in the number of parameters with respect to the additive *GraphConv*, except for the additional bias.

To further enhance the expressiveness of the MI-GNN, at the cost of increasing the number of convolution parameters, the additive and multiplicative building blocks can be combined explicitly, each with its own weights. We named this variation **MI-GNN-v2**:

$$
\begin{aligned}
\mathbf{h}_v^{(\ell)} = \phi\Big(\Big(\mathbf{W}\mathbf{h}_v^{(\ell-1)} + \mathbf{b}\Big) + \mathbf{W}_{\Sigma} \sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(\ell-1)} + \\
+ \mathbf{W}_{\odot}\Big(\mathbf{h}_v^{(\ell-1)} \odot \sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(\ell-1)}\Big)\Big)
\end{aligned}
\tag{5.3}
$$

The third MI variation of the *GraphConv* that we propose, **MI-GNN-v3**, exploits the Hadamard product to define both the combination and aggregation steps of the GC operator:

$$
\mathbf{h}_v^{(\ell)} = \phi\Big(\Big(\mathbf{W}\mathbf{h}_v^{(\ell-1)} + \mathbf{b}\Big) \odot \Big(\mathbf{W}_{\Pi}\left[\prod_{u \in \mathcal{N}_v}^{\odot} \mathbf{h}_u^{(\ell-1)}\right] + \mathbf{b}_{\Pi}\Big)\Big) \tag{5.4}
$$

This formulation makes the interaction among all nodes involved in the convolution uniform, implementing a global gating mechanism while preserving the sharing scheme of the parameters used in *GraphConv*. The use of MI as a combination mechanism, however, involves multiplying several node embeddings (projected using the same shared weights) that can lead to numerical stability issues in case of extremely small (close to 0) or large (significantly higher than 1) values. This can make the training phase unstable. To solve this issue, by maintaining a multiplicative integration approach as an aggregation mechanism, we propose to transform the product among the neighbors into a sum by exploiting the logarithm function jointly with the ReLU function to ensure that the co-domain of $\mathcal{A}(\cdot)$ is limited to a set of values that ensure a more stable training phase:

$$
\begin{aligned}
\mathbf{h}_v^{(\ell)} = \mathrm{ReLU}\Big(\mathbf{W}\mathbf{h}_v^{(\ell-1)} + \mathbf{b}\Big) \odot \\
\odot \Big(\mathbf{W}_{log}\left[\sum_{u \in \mathcal{N}_v} \log\Big[\mathrm{ReLU}(\mathbf{h}_u^{(\ell-1)}) + \epsilon\Big]\right] + \mathbf{b}_{log}\Big),
\end{aligned}
\tag{5.5}
$$

where ϵ is a small positive constant which prevents the input to log to be 0. Notice that when $\epsilon = 1$, we get as output only positive values, avoiding negative values with high modules. We applied the ReLU function also on the current node embedding $(\mathbf{W}\mathbf{h}_v^{(\ell-1)} + \mathbf{b})$ projection because we want to ensure that the new embedding will be a positive tensor. This is critical since it will be used as input to a logarithm in the subsequent GC layer (if it exists, i.e., if $\ell < L$).

5.3 Experimental Setup and Results

Our experimental assessment aimed to empirically verify whether multiplication can be used to define graph convolutions with competitive performance. Specifically, our experimental results show that representative methods in literature exploiting multiplicative operators, i.e., GAT and GNN-FiLM, do not show a performance advantage over addition-based GCNs. We then explore the use of MI in graph neural networks, evaluating the proposals presented in Section 5.2, along with four widely adopted baselines, on seven graph classification benchmark datasets (see Section 4.4.2).

5.3.1 GNN Architecture and Model Selection

We considered as baselines two commonly adopted GC operators, GCN [Kipf, 2017] and *GraphConv* [Morris, 2019]. Moreover, we also experiment with two powerful convolutions for graphs that exploit the multiplicative operation differently than the MI-GNN: GAT [Velickovic, 2018] and GNN-FiLM [Brockschmidt, 2020]. For more details, please check the publicly available code[2].

We handled the experiments and validated all the models' hyper-parameters adopting the GraphGym [You, 2020] framework. Specifically, we started off from their findings on the GNN architecture design space by setting a common baseline configuration for all the experiments based on the work of [You, 2020]. We set the PReLU as activation function [He, 2015], batch normalization [Ioffe, 2015] for each layer, and among layers, we adopted the SKIP-CAT scheme [G. Huang, 2017]. The training is carried out with the ADAM optimizer [Kingma, 2015], cosine learning rate schedule (starting from 0.01 and annealed to 0, no restarting), 5×10^{-4} L2 weight decay for regularization. The batch size is set to 32 for all the datasets and we let every experiment run for 400 epochs. We used the libraries `PyTorch=2.0.0`, `PyTorch Geometric=2.3.0`, and our experiments have been carried out in a computing cluster equipped with GPUs Nvidia RTX A5000. Each tested network consists of MLP layers before and after the GC operator layers. This particular architectural setting is the one suggested in GraphGym. The amount of these layers and their hidden units are hyperparameters. In our evaluation, we consider graph classification tasks; therefore, all the considered models have a global pooling layer to compute a graph-level representation given the node embeddings. The pooling layer is

[2] `https://github.com/paolofraz/MI-GNN`

defined by concatenating the global mean, max, and sum aggregations (Section 3.2.2):

$$\mathbf{h}^{\mathcal{G}} = [\mathrm{mean}_{v \in V}(\mathbf{h}_v^{(L)}), \max_{v \in V}(\mathbf{h}_v^{(L)}), \mathrm{sum}_{v \in V}(\mathbf{h}_v^{(L)})], \tag{5.6}$$

where L is the number of GC layers.

Each dataset is split in train/validation/test sets according to a $[80\%, 10\%, 10\%]$ random split. Every configuration is run 3 times, and we take the average of all the evaluation metrics (accuracy, time, etc.) taken on the test set at the best epoch in validation. The random generator seed is set likewise at the beginning of each run, thus ensuring that the dataset splits are equal for each model and making their comparison more robust and fair. We performed a full grid search over all the hyper-parameters combinations reported in Table 5.1, resulting in 96 configurations tested for each of the 7 datasets and of the 7 GNN layer types. The layers for *GraphConv*, GCNConv, GATConv, and FiLMConv are taken from the PyTorch library.

Hyper-parameters	**Values**
Pre-MLP layers	$1, 2$
GC layers	$2, 4, 6, 8$
Post-MLP layers	$2, 3$
Hidden units	$64, 128, 256$
Activation ϕ	PReLU, Tanh

Table 5.1: Hyper-parameters Grid for the MI-GNNs

5.3.2 Results and Discussion

Dataset \ GNN	GNN-FiLM	GAT	GCN	GraphConv	MI-GNN-v1	MI-GNN-v2	MI-GNN-v3
COLLAB	68,9±9,2	55,7±1,4	**79,4±1,6**	75,9±1,7	76,9±2,7	75,9±0,7	74,5±0,9
DD	77,9±2,1	77,6±2,0	77,6±3,4	77,6±2,4	**79,6±2,4**	77,3±0,7	77,0±5,3
ENZYMES	58,9±0,8	58,3±3,6	62,2±4,2	62,2±0,8	62,8±0,8	**63,3±2,4**	60,6±3,9
IMDB-B	53,7±4,8	55,7±2,5	71,7±5,3	73,0±3,3	**74,0±2,2**	73,7±1,7	**74,0±4,9**
IMDB-M	41,1±7,5	40,0±2,2	50,5±2,7	50,9±2,5	49,6±3,8	49,8±3,8	**51,3±3,3**
NCI1	79,0±2,1	80,2±2,0	80,6±1,4	**81,7±1,7**	81,6±1,0	81,6±2,0	78,3±1,7
PROTEINS	73,8±3,0	74,1±2,6	73,8±2,3	75,0±4,1	75,3±3,4	**75,9±2,6**	74,4±2,8

Table 5.2: Accuracy and standard deviation, in percentages, on the test set for the best-validated models on all the datasets. The best performances are highlighted in boldface.

The results of our experiments are presented in Table 5.2. We can start noticing that the two existing multiplication-based methods (GNN-FiLM and GAT) do not result in the better-performing methods on any of the considered datasets. This observation enforces our intuition that research is still required in multiplication-based graph convolutions to achieve competitive performance. As for the proposed methods, at least one version of the proposed MI-GNNs is the best-performing method in six of the eight datasets. The GCN performs well only in the COLLAB dataset, which is the dataset that has the highest edges/graph ratio. We argue that the normalized adjacency matrix of the GCN is more robust for such cases, whereas the other multiplicative or additive operators are penalized by the higher average degree. The *GraphConv* has the highest accuracy only for the NCI1 dataset; however, MI-GNN-v1 and -v2 perform similarly. MI-GNN-v1 and MI-GNN-v2 are the best-performing methods on two datasets each. MI-GNN-v3 performs on par with MI-GNN-v1 on IMDB-B (but with a higher variance) and is the best-performing method on IMDB-M.

Additionally, we analyzed whether the improved performances of our implementation of the MI-GNNs also translate into briefer training times. In Fig. 5.1, we display each dataset's accuracy and total training time until the best validation epoch is reached. In the plot, we also report the Pareto frontier—the set of all Pareto-efficient points. In our case, those points refer to the methods for which no improvement in one dimension (either accuracy or training efficiency) is possible without losing performance on the other dimension. Such points can be informally interpreted as the best time/accuracy trade-off methods. We can see that the MI-GNNs are often present in the Pareto-front, meaning that not only do they achieve the best performances most of the time, but they also require less or comparable time than the baselines. Notice that on many datasets, some methods are extremely efficient (e.g., GCN on COLLAB, IMDB-M, and PROTEINS) but perform very poorly. Even though those points are part of the Pareto-front because of the low training times, they are not interesting solutions for their degraded predictive performance. We want to mention that GCN results are the slowest method because, by default, it does not store the normalized adjacency matrix. For this reason, its training could be tweaked and sped up; however, its predictive performances would not be altered.

5.3.3 Statistical Significance of the Results

Inspired by the analysis of Demsar [2006], we investigated the performances of our proposed models beyond a simple but naive maximum-accuracy benchmark. Indeed, when comparing

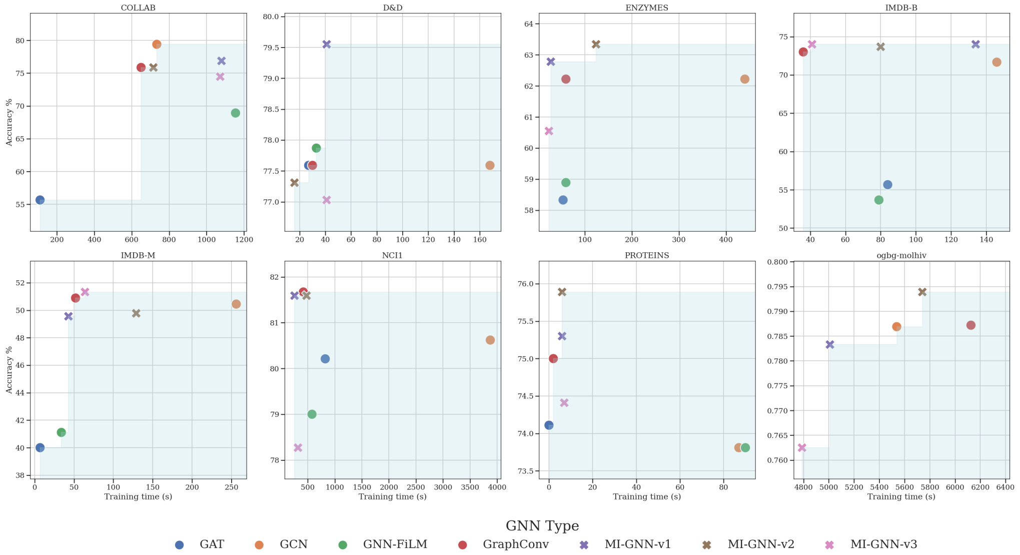


Figure 5.1: Distribution of the training times of the best performing models w.r.t their accuracy. The MI-GNNs are marked with a cross, and the other baselines are marked with a rounded point. The light-blue area (left side) helps to identify the Pareto-front. For ogbg-molhiv, the AUC is reported on the vertical axis.

multiple classifiers on multiple datasets, one should apply rigorous statistical hypothesis testing before assessing whether the improvement is statistically significant. Additionally, when testing multiple hypotheses simultaneously, multiplicity issues arise and one should adopt the proper corrections.

The Friedman test [Pereira, 2015] is a non-parametric test that does not assume the distribution and variances of the samples. For each dataset and for each configuration, it ranks the accuracies of all the models. Then, it computes a statistic χ_F^2 under the null hypothesis, which states that all the models are equivalent and their ranks should be random. In our case, this test gives a p-value< 0.01, so the null hypothesis is rejected, and we can proceed with a post-hoc test to tell which algorithms perform the best. To calculate the statistical significance of the pairwise comparisons between the models, we used the Conover post-hoc test for unreplicated blocked data [Conover, 1999] where the p-values are adjusted with the step-down method using the Sidak corrections [Šidák, 1967]. Other common p-values adjustments yielded equivalent outcomes. The outcome of this analysis is neatly presented with the critical difference (CD) diagrams in Fig. 5.2. This plot displays the averages of the normalized ranks of the models among all the configurations. On the x-axis, 1 would stand for a model that always scores better; on the contrary, a model at 0 would always be the last one in the rankings. The groups that could not be statistically deemed different by the Conover test are linked by a horizontal crossbar. We can see that *GraphConv*, MI-GNN-v1, and -v2 are significantly better ranked than all the other models. While the MI-GNN-v1 has the highest average rank of 0.71, indicating that it tends to be the best model more frequently for a given configuration and dataset, there is insufficient statistical evidence to confirm this conclusion. Therefore, conducting experiments on additional datasets would be necessary to endorse this assessment. Nevertheless, this shows that MI-GNN-v1 is a valid alternative to GraphConv, and all the parameters and settings being equal, merely replacing the additive term with the Hadamard product can lead to improved performances. Moreover, the CD diagram shows us that MI-GNN-v3 is, on average, in the middle of the rankings despite achieving the best accuracy in PROTEINS and ENZYMES. This advises us that when evaluating new machine learning models, when it is possible, it is crucial to go beyond the maximum accuracy rationale—only by testing whether the new model performs statistically better than the baselines for multiple configurations, one can ensure that such improvements are significant and applicable across different scenarios.

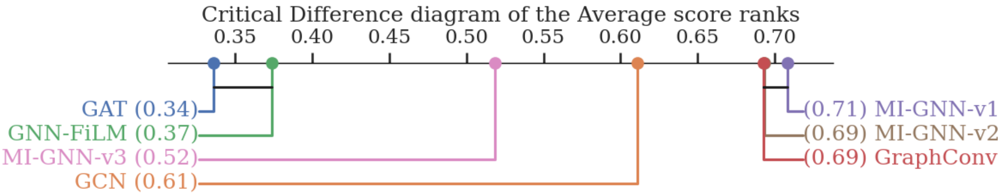


Figure 5.2: Critical Difference diagram of the Average score ranks.

5.3.4 Open Graph Benchmark

To prove the validity of our approach, we additionally evaluated the performances of the MI-GNNs on the *molhiv* dataset belonging to the Open Graph Benchmark (OGB) [Hu, 2020]. This dataset is made of 41 127 graphs, an order of magnitude more than the ones previously considered, and it follows its pre-defined pre-processing and evaluation pipeline. Due to its complexity and time constraints, we restricted our experimental setup to a smaller hyper-parameters grid and fewer models. For these reasons, it cannot be analyzed along with the other datasets and the procedure described in the previous section. Table 5.3 reports the best AUC on the test set, training time, and architecture for the model with the highest AUC score on the validation set. MI-GNN-v2 has the best performance and it requires fewer parameters, thus, this supports our intuition that MI is able to grasp relevant complex interactions among nodes. It is worth noticing that such an AUC score would place our model in the top 20 leaderboard for this dataset.

GNN Type	Pre-MLP layers	GC-layers	Post-MLP layers	Hidden units	# Param.	AUC	Training Time (s)
GCN	1	2	3	384	24 517 826	78,69±0,86	5538
GraphConv	1	4	3	384	69 506 498	78,72±2,15	6126
MI-GNN-v1	2	6	2	256	31 872 322	78,33±0,90	5009
MI-GNN-v2	2	4	3	128	7 983 554	**79,39±0,15**	5741
MI-GNN-v3	2	2	3	384	25 109 954	76,25±0,03	4788

Table 5.3: Experimental results and architecture for the best-validated model on the ogbg-molhiv dataset.

5.4 Gradient Analysis

Pursuing the goal of characterizing the strengths and the weakness of the application of the multiplicative integration in GC operators, we theoretically analyze the gradient of the various proposed versions of MI-GNN and we compare them with the gradient of the most similar additive model, the *GraphConv*. We denote as $\mathbf{H} \in \mathbb{R}^{m_\ell \times n}$ the matrix of all the nodes' embeddings at layer ℓ, $\bar{\mathbf{H}} \in \mathbb{R}^{m_{\ell+1} \times n}$ is the same matrix at the following layer $\ell + 1$, and δ_{ij} the Kronecker delta function. In the following, we use Einstein's notation of summation over repeated indexes: $a_{ij} = {\mathbf{A}^i}_j = \sum_k b_{ik} c_{kj} = {\mathbf{B}^i}_k {\mathbf{C}^k}_j$. For the *GraphConv*, the derivatives w.r.t. the weights are (omitting the Jacobian of ϕ and the bias term):

$$\frac{\partial {\bar{\mathbf{H}}^x}_v}{\partial {\mathbf{W}^a}_b} = \delta_{ax} {\mathbf{H}^b}_v \;\text{ and, }\; \frac{\partial {\bar{\mathbf{H}}^x}_v}{\partial {\mathbf{W}_{\mathbf{\Sigma}}{}^c}_d} = \delta_{cx} {\mathbf{H}^d}_u {\mathbf{A}^u}_v, \tag{5.7}$$

while the derivative of the MI-GNN-v1 w.r.t the weights are the following:

$$\frac{\partial {\bar{\mathbf{H}}^x}_v}{\partial {\mathbf{W}^a}_b} = \delta_{ax} {\mathbf{H}^b}_v \cdot \left({\mathbf{W}_{\mathbf{\Sigma}}{}^x}_\rho {\mathbf{H}^\rho}_u {\mathbf{A}^u}_v \right), \tag{5.8}$$

$$\frac{\partial {\bar{\mathbf{H}}^x}_v}{\partial {\mathbf{W}_{\mathbf{\Sigma}}{}^c}_d} = \delta_{cx} {\mathbf{H}^d}_u {\mathbf{A}^u}_v \cdot \left({\mathbf{W}^x}_\sigma {\mathbf{H}^\sigma}_v \right). \tag{5.9}$$

It is worth noticing how the use of MI instead of an additive block leads the gradient with respect to $\mathbf{W}$ to be influenced by $\mathbf{W}_{\mathbf{\Sigma}}$ and vice-versa. Moreover, in the MI-GNN, the gradient of the weights that multiply the current node v also depends on the adjacency matrix $\mathbf{A}$, making it possible to carry information about the neighbors when learning $\mathbf{W}$. This enriched gradient for the first weights matrix of Eq. (5.1) implies that the neighborhood will directly influence the projection of the node v. We can notice a similar effect comparing the gradient of the *Graph-Conv* and the MI-GNN-v1 w.r.t. the node embeddings of the previous layer. Indeed for the *GraphConv*, the gradient is the following:

$$\frac{\partial \bar{\mathbf{H}}^{x}{}_{v}}{\partial \mathbf{H}^{z}{}_{t}} = \delta_{tv} \mathbf{W}^{x}{}_{z} + \mathbf{W_{\Sigma}}^{x}{}_{z} \mathbf{A}^{t}{}_{v}, \tag{5.10}$$

while the one of the MI-GNN-v1 is:

$$\frac{\partial \bar{\mathbf{H}}^{x}{}_{v}}{\partial \mathbf{H}^{z}{}_{t}} = \delta_{tv} \mathbf{W}^{x}{}_{z} \cdot \Big(\mathbf{W_{\Sigma}}^{x}{}_{\rho} \mathbf{H}^{\rho}{}_{u} \mathbf{A}^{u}{}_{v}\Big) + \Big(\mathbf{W}^{x}{}_{\sigma} \mathbf{H}^{\sigma}{}_{v}\Big) \cdot \Big(\mathbf{W_{\Sigma}}^{x}{}_{z} \mathbf{A}^{t}{}_{v}\Big). \tag{5.11}$$

For what concerns the MI-GNN-v2, we have three weight matrices (see Eq. (5.3)). The gradient of the hidden representations w.r.t $\mathbf{W}$ and $\mathbf{W_{\Sigma}}$ are the same as reported in Eq. (5.9), so the consideration made for the MI-GNN-v1 holds also for these second version. Interestingly, the gradient w.r.t. $\mathbf{W}_{\odot}$ keeps the same capability of conveying information about the current node and the neighborhood as in the case of the other two weights matrices:

$$\frac{\partial \bar{\mathbf{H}}^{x}{}_{v}}{\partial \mathbf{W}_{\odot}{}^{e}{}_{f}} = \delta_{ex} \Big(\mathbf{H}^{f}{}_{v} \cdot \mathbf{A}^{f}{}_{l} \mathbf{H}^{l}{}_{v}\Big). \tag{5.12}$$

Differently from the gradient of the hidden representation with respect to the previously considered $\mathbf{W}$ and $\mathbf{W_{\Sigma}}$ parameter matrices, in this case, the gradient is not influenced by the other weights of the model.

Considering the MI-GNN-v3, the derivative w.r.t. the previous layer is highly influenced by the ReLU function applied to the projection of the node v and the projection of the neighborhood. Recall that the ReLU function is used to avoid instability during the training. Let us start considering the gradient of the hidden representation of a layer w.r.t. the representation of the layer before:

$$\begin{aligned} \frac{\partial \bar{\mathbf{H}}^{x}{}_{v}}{\partial \mathbf{H}^{z}{}_{t}} = {} & \delta_{tv} \Theta\Big(\mathbf{W}^{x}{}_{\sigma} \mathbf{H}^{\sigma}{}_{v}\Big) \mathbf{W}^{x}{}_{z} \cdot \Big(\mathbf{W_{log}}^{x}{}_{\rho} \log\Big[\mathrm{ReLU}\Big(\mathbf{H}^{\rho}{}_{u}\Big)\Big] \mathbf{A}^{u}{}_{v}\Big) + \\ & + \mathrm{ReLU}\Big(\mathbf{W}^{x}{}_{\sigma} \mathbf{H}^{\sigma}{}_{v}\Big) \cdot \Big(\mathbf{W_{log}}^{x}{}_{z} \mathbf{A}^{t}{}_{v}\Big), \end{aligned} \tag{5.13}$$

where $\Theta(\cdot)$ is the derivative of the ReLU. If we consider the gradient w.r.t. the weights $\mathbf{W}$, $\mathbf{W_{log}}$ we can notice that the gradient takes into account the interaction between them, as well as the adjacency matrix:

$$\frac{\partial \bar{\mathbf{H}}^{x}{}_{v}}{\partial \mathbf{W}^{a}{}_{b}} = \delta_{ax}\Theta\Big(\mathbf{W}^{x}{}_{\sigma}\mathbf{H}^{\sigma}{}_{v}\Big)\mathbf{H}^{b}{}_{v} \cdot \Big(\mathbf{W}_{\mathbf{log}}{}^{x}{}_{\rho}\log\Big[\mathrm{ReLU}\Big(\mathbf{H}^{\rho}{}_{u}\Big)\Big]\mathbf{A}^{u}{}_{v}\Big) \tag{5.14}$$

$$\frac{\partial \bar{\mathbf{H}}^{x}{}_{v}}{\partial \mathbf{W}_{\mathbf{log}}{}^{c}{}_{d}} = \mathrm{ReLU}\Big(\mathbf{W}^{x}{}_{\sigma}\mathbf{H}^{\sigma}{}_{v}\Big) \cdot \delta_{cx}\log\Big[\mathrm{ReLU}\Big(\mathbf{H}^{d}{}_{u}\Big)\Big]\mathbf{A}^{u}{}_{v}. \tag{5.15}$$

Unlike the other version of MI-GNN, this one also considers the log of the ReLU activation of the neighbors' embeddings and the ReLU derivative applied to the embedding of the current node v. In particular, for $\mathbf{W}_{\mathbf{log}}$, this behavior ensures to have always a positive value for the gradient.

5.5 Final Remarks

In this chapter, we conducted a thorough analysis of how Multiplicative Integration (MI) can be utilized to define graph convolution operators. We introduced three distinct variants, each examining different methodologies for applying MI to graph-structured data, which we have collectively named MI-GNN. Our empirical evaluation involved testing the three MI-GNN models across eight benchmark classification datasets. We ensured a fair experimental setup and adopted a robust statistical framework to analyze the results comprehensively. The experimental findings demonstrated that MI-GNNs achieve competitive accuracy levels while also exhibiting shorter training times when compared to other state-of-the-art models.

Additionally, we conducted an investigation into how the incorporation of MI impacts training by performing a theoretical analysis of the gradients. This analysis indicated that MI-GNN models are proficient at transmitting structural information during gradient computation for model weight updates. Furthermore, it was observed that the gradient flow within MI-GNN models is affected by the interplay between the various weights of the GNN. This effect demonstrates a superior capacity to convey information compared to the well-established *GraphConv* layer.

6

Precise Determination of Shapley Interactions of Arbitrary Order in GNNs

Albeit the ubiquitous use of GNNs in machine learning prediction tasks involving graph data, their interpretability remains challenging. In explainable artificial intelligence (XAI), the Shapley Value (SV) is the predominant method to quantify the contributions of individual features to an ML model's output. Addressing the limitations of SVs in complex prediction models, Shapley Interactions (SIs) extend the SV to groups of features. In this appendix[1], we explain single graph predictions of GNNs with SIs that quantify node contributions and interactions among multiple nodes. By exploiting the GNN architecture, we show that the structure of interactions in node embeddings are preserved for graph prediction. As a result, the exponential complexity of SIs depends only on the receptive fields, i.e. the message-passing ranges determined by the connectivity of the graph and the number of convolutional layers. Based on our theoretical results, we introduce GraphSHAP-IQ, an efficient approach to compute any-order SIs exactly. GraphSHAP-IQ is applicable to popular message-passing techniques in conjunction with a linear global pooling and output layer. We showcase that GraphSHAP-IQ substantially reduces the exponential complexity of computing exact SIs on multiple benchmark

[1] This Appendix is the result of the fruitful collaboration with the German universities of Bielefeld and LMU, that lead to the paper: Fumagalli, Fabian, Maximilian Muschalik, **Paolo Frazzetto**, Janine Strotherm, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara Hammer [2025]. "Exact Computation of Any-Order Shapley Interactions for Graph Neural Networks". In: *The International Conference on Learning Representations (ICLR)*. Under review.

datasets. Beyond exact computation, we evaluate GraphSHAP-IQ's approximation of SIs on popular GNN architectures and compare with existing baselines. Lastly, we visualize SIs of real-world water distribution networks (WDN) and molecule structures using a SI-Graph.

6.1 Motivation

A major drawback of GNNs is the opacity of their predictive mechanism, which they share with most deep-learning based architectures [Amara, 2022]. Reliable explanations for their predictions are crucial when model decisions have significant consequences [H. Zhang, 2024] or lead to new discoveries [McCloskey, 2019]. In XAIs, the SV [Shapley, 1953] is a prominent concept to assign contributions to entities of black box ML models [Lundberg, 2017; Covert, 2021b; H. Chen, 2023]. Entities typically represent features, data points [Ghorbani, 2019] or graph structures [Yuan, 2021; Ye, 2023]. Although SVs yield an axiomatic attribution scheme, they do not give any insights into joint contributions of entities, known as *interactions*. Yet, interactions are crucial to understanding decisions of complex black box ML models [Wright, 2016; I. E. Kumar, 2020; Sundararajan, 2020a; I. Kumar, 2021; **Frazzetto**, 2024a]. SIs [Grabisch, 1999; Bordt, 2023] extend the SV to include joint contributions of multiple entities. SIs satisfy similar axioms [Grabisch, 1999] while providing interactions up to a maximum number of entities, referred to as the *explanation order*. In this context, SVs are the least complex SIs, whereas Möbius Interactions (MI) (or Möbius transform) [Harsanyi, 1963; Rota, 1964] are the most complex SIs by assigning contributions to every group of entities. Thus, SIs convey an adjustable explanation with an *accuracy-complexity trade-off* for interpretability [Bordt, 2023]. SVs, SIs and MIs are limited by exponential complexity, e.g. with 20 features already $2^{20} \approx 10^6$ model calls per explained instance are required. Consequently, practitioners rely on model-agnostic approximation methods [Lundberg, 2017; Fumagalli, 2023] or model-specific methods [Lundberg, 2020; Muschalik, 2024] that exploit knowledge about the model's structure to reduce complexity. As a remedy for GNNs, the SV was applied as a heuristic on subgraphs [Ying, 2019; Ye, 2023], or approximated [Duval, 2021; Bui, 2024].

In this work, we address limitations of the SV for GNN explainability by computing the SIs visualized as the SI-Graph in Fig. 6.1. Our method yields exact SIs by including GNN-specific knowledge and exploiting properties of the SIs. In contrast to existing methods [Yuan, 2021; Ye, 2023], we evaluate the GNN on node level without the need to cluster nodes into subgraphs. Instead of model-agnostic approximation [Duval, 2021; Bui, 2024], we provide structure-aware approximation for graph prediction and prove that MIs of node embeddings indeed transfer

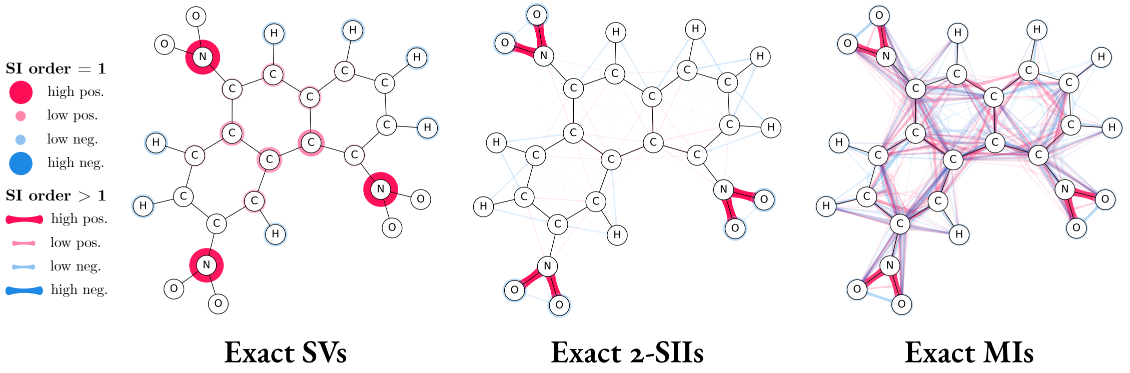


Figure 6.1: SI-Graphs overlayed on a molecule graph showing exact SIs for a molecule with 30 atoms from *MTG*. A GNN correctly identifies it as mutagenic. The SIs, in line with ground-truth knowledge, highlight the NO_2 groups. Computing exact SIs requires $2^{30} \approx 10^9$ model calls. GraphSHAP-IQ needs 7 693.

to graph prediction for linear readouts. In summary, our approach is a model-specific computation of SIs for GNNs, akin to TreeSHAP [Lundberg, 2020].

The main contributions include:

1. We introduce SIs among nodes and the SI-Graph (Definition 6.2.1) for graph predictions of GNNs that address limitations of the SV and provide theoretical results that exploit the GNN architecture.
2. We provide theoretical results that exploit the GNN architecture and formally prove that interactions learned in node embeddings are preserved for graph predictions.
3. We present GraphSHAP Interaction Quantification (GraphSHAP-IQ), an efficient mode to compute exact any-order SIs in GNNs or approximate them in restricted settings.
4. We show substantially reduced complexity when applying GraphSHAP-IQ on real-world benchmark datasets, and analyze SI-Graphs of a WDN and molecule structures.

6.1.1 Explanation Complexity: From Shapley Values to Möbius Interactions

Concepts from cooperative game theory, such as the SV [Shapley, 1953], are prominent in XAI to interpret predictions of a black box ML model via feature attributions [Strumbelj, 2014; Lundberg, 2017]. Formally, a cooperative game $\nu : \mathcal{P}(N) \to \mathbb{R}$ is defined, where individual features $N = \{1, \dots, n\}$ act as players and achieve a payout for every group of players in the power set $\mathcal{P}(N)$. To obtain feature attributions for the prediction of a single instance, ν typically refers to the model's prediction given only a subset of feature values. Since classical ML models cannot handle missing feature values, different methods have been proposed, such as model retraining [Strumbelj, 2009], conditional expectations [Lundberg, 2017; Aas, 2021; Frye, 2021], marginal expectations [Janzing, 2020] and baseline imputations [Lundberg, 2017; Sundararajan, 2020a]. In high-dimensional feature spaces, retraining models or approximating feature distributions is infeasible, imputing absent features with a baseline, known as BShap [Sundararajan, 2020b], is the prevalent method [Lundberg, 2017; Sundararajan, 2017, 2020a,b; Covert, 2021b; Jethani, 2022]. We now first introduce the MIs as a backbone of additive contribution measures. Later in Section 6.2, we exploit sparsity of MIs for GNNs to compute the SV and any-order SIs.

Möbius Interactions (MIs) $m : \mathcal{P}(N) \to \mathbb{R}$, alternatively Möbius transform [Rota, 1964], Harsanyi dividend [Harsanyi, 1963], or internal interaction index [Fujimoto, 2006], are a fundamental concept of cooperative game theory. The MI is

$$m(S) := \sum_{T \subseteq S} (-1)^{|S|-|T|} \nu(T) \text{ and they recover } \nu(T) = \sum_{S \subseteq T} m(S) \text{ for all } S, T \subseteq N. \tag{6.1}$$

From the MIs, every game value can be additively recovered, and MIs are the unique measure with this property [Harsanyi, 1963; Rota, 1964]. The MI of a subset $S \subseteq N$ can thus be interpreted as the *pure additive contribution* that is exclusively achieved by a coalition of all players in S, and cannot be attributed to any subgroup of S. The MIs are further a basis of the vector space of games [Grabisch, 2016], and therefore every measure of contribution, such as the SV or the SIs, can be directly recovered from the MIs.

Shapley Values (SVs) for players $i \in N$ of a cooperative game ν are the weighted average

$$\phi^{\text{SV}}(i) := \sum_{T \subseteq N \setminus i} \frac{1}{n \cdot \binom{n-1}{|T|}} \Delta_i(T) \text{ with } \Delta_i(T) := \nu(T \cup i) - \nu(T) \tag{6.2}$$

over marginal contributions $\Delta_i(T)$. It was shown [Shapley, 1953] that the SV is the unique attribution method that satisfies desirable axioms: linearity (the SV of linear combinations of games, e.g., model ensembles, coincides with the linear combinations of the individual SVs), dummy (features that do not change the model's prediction receive zero SV), symmetry (if a model does not change its prediction when switching two features, then both receive the same

SV), and lastly efficiency (the sum of all SVs equals the difference between the model's prediction $\nu(N)$ and the featureless prediction $\nu(\emptyset)$). A cooperative game is typically normalized, such that $\nu(\emptyset) = 0$, which does not affect the SVs. The SV assigns attributions to individual features, but it does not provide any insights about *feature interactions*, i.e. the joint contribution of multiple features to the prediction. In practice, however, understanding complex black box models requires investigating interactions [I. E. Kumar, 2020; Slack, 2020; Sundararajan, 2020a; I. Kumar, 2021]. On the contrary, the MIs provide the full additive decomposition involving all possible joint contributions. While the SVs are limited in their expressivity, the MIs are difficult to interpret due to the exponential number of components. The SIs provide a framework to bridge both concepts.

Shapley Interactions (SIs) explore model predictions beyond individual feature attributions, and provide additive contribution for all subsets up to *explanation order* $k = 1, \dots, n$. More formally, the SIs $\Phi_k : \mathcal{P}_k(N) \to \mathbb{R}$ assign interactions to subsets of N up to size k, summarized in $\mathcal{P}_k(N)$. The SIs decompose the model's prediction with $\nu(N) = \sum_{S \subseteq N, |S| \leq k} \Phi_k(S)$. The least complex SIs are the SVs, which are obtained with $k = 1$. For $k = n$, the SIs are the MIs with 2^n components, which provide the most faithful explanation of the game but entail the highest complexity. SIs are constructed based on extensions of the marginal contributions $\Delta_i(T)$, known as discrete derivatives [Grabisch, 1999]. For two players $i, j \in N$, the discrete derivative $\Delta_{ij}(T)$ for a subset $T \subseteq N \setminus ij$ is defined as $\Delta_{ij}(T) := \nu(T \cup ij) - \nu(T) - \Delta_i(T) - \Delta_j(T)$, i.e., the joint contribution of adding both players together minus their individual contributions in the presence of T. This recursion is extended to any subset $S \subseteq N$ and $T \subseteq N \setminus S$. A positive value of the discrete derivative $\Delta_S(T)$ indicates synergistic effects, a negative value indicates redundancy, and a value close to zero indicates no joint information of all players in S given T. The Shapley Iinteraction Index (SII) [Grabisch, 1999] provides an axiomatic extension of the SV and summarizes the discrete derivatives in the presence of all possible subsets T as

$$\phi^{\text{SII}}(S) = \sum_{T \subseteq N \setminus S} \frac{1}{(n - |S| + 1) \cdot \binom{n-|S|}{|T|}} \Delta_S(T) \quad \text{with} \quad \Delta_S(T) := \sum_{L \subseteq S} (-1)^{|S|-|L|} \nu(T \cup L). \tag{6.3}$$

Given an explanation order k, the k-Shapley Values (k-SIIs) [Lundberg, 2020; Bordt, 2023] construct SIs recursively based on the SII, such that the interactions of SII and k-SII for the highest order coincide. Alternatively, the Shapley Taylor Interaction Index (STII) [Sundararajan, 2020b] and the Faithful Shapley Interaction Index (FSII) [Tsai, 2023] have been proposed. In summary, SIs provide a flexible framework of increasingly complex and faithful contributions ranging from the SV ($k = 1$) to the MIs ($k = n$). Given the MIs, it is possible to reconstruct SIs of arbitrary order. In Section 6.2, we will exploit the sparse structure of MIs of GNNs to efficiently compute any-order SIs.

6.2 Any-Order Shapley Interactions for Graph Neural Networks

In the following, we are interested in explaining the prediction of a GNN f_g for a graph g with respect to nodes. We aim to decompose a model's prediction into SIs Φ_k visualized by a SI-Graph.

Definition 6.2.1 (SI-Graph) *The SI-Graph is an undirected hypergraph $g_k^{SI} := (N, \mathcal{P}_k(N), \Phi_k)$ with node attributes $\Phi_k(i)$ for $i \in N$ and hyperedge attributes $\Phi_k(S)$ for $2 \leq |S| \leq k$.*

The simplest SI-Graph displays the SVs ($k = 1$) as node attributes, whereas the most complex SI-Graph displays the MIs ($k = n$) as node and hyperedge attributes, illustrated in Fig. 6.1. The complexity of the SI-Graph is adjustable by the explanation order k, which determines the maximum hyperedge order. The sum of all contributions in the SI-Graph yields the model's prediction (for regression) or the model's logits for the predicted class (for classification). This choice is natural for an additive contribution measure due to additivity in the logit-space. To compute SIs, we introduce the GNN-induced graph game ν_g with a node masking strategy in Section 6.2.1. The graph game is defined on all nodes and describes the output given a subset of nodes, where the remaining are masked. Computing SIs on the graph game defines a perturbation-based and a decomposition-based GNN explanation [Yuan, 2023], which is an extension of node attributions [C. Agarwal, 2023]. In Section 6.2.2, we show that GNNs with linear global pooling and output layer satisfy an invariance property for the node game associated with the node embeddings (Theorem 6.2.3). This invariance implies sparse MIs for the graph game (Proposition 6.2.6), which determines the complexity of MIs by the corresponding receptive fields (Theorem 6.2.7), which substantially reduces the complexity of SIs in our experiments. In Section 6.2.3, we introduce GraphSHAP-IQ, an efficient algorithm to exactly compute and estimate SIs on GNNs.

6.2.1 A Cooperative Game for Shapley Interactions on Graph Neural Networks

Given a GNN f_g, we propose the graph game for which we compute *axiomatic and fair* SIs.

Definition 6.2.2 (GNN-induced Graph and Node Game) *For a graph* $g = (V, E, \mathbf{X})$ *and a GNN* f_g*, we let* $N := \{i : v_i \in V\}$ *be the node indices and define the graph game* $\nu_g : \mathcal{P}(N) \to \mathbb{R}$ *as*

$$\nu_g(T) := f_{g,\hat{y}}(\mathbf{X}^{(T)}) \text{ with } \mathbf{X}^{(T)} := (\mathbf{x}_1^{(T)}, ..., \mathbf{x}_n^{(T)})^t \in \mathbb{R}^{n \times d_0} \text{ and } \mathbf{x}_i^{(T)} := \begin{cases} \mathbf{x}_i & \text{if } i \in T, \\ \mathbf{b} & \text{if } i \notin T, \end{cases}$$

with $i \in N$ *and baseline* $\in \mathbb{R}^{d_0}$*. In graph regression* $f_{g,\hat{y}} \equiv f_g$ *and for graph classification* $f_{g,\hat{y}}$ *is the component of the predicted class* $\hat{y}$ *of* f_g*. We further introduce the (multi-dimensional) node game* $\nu_i : \mathcal{P}(N) \to \mathbb{R}^{d_\ell}$ *as* $\nu_i(T) := f_i(\mathbf{X}^{(T)})$ *for* $i \in N$ *and each node* $v_i \in V$*.*

The graph game outputs the prediction of the GNN for a subset of nodes $T \subseteq N$ by masking all node features of nodes v_i with $i \in N \setminus T$ using a suitable baseline , illustrated in Fig. 6.2, left. Computing such SVs is known as BShap [Sundararajan, 2020b] and a prominent approach for feature attributions [Lundberg, 2017; Covert, 2021b; H. Chen, 2023]. As a baseline $\mathbf{b}$, we propose the average of each node feature over the whole graph. By definition, the prediction of the GNN is given by $\nu_g(N) = f_g(\mathbf{X})$, and due to the efficiency axiom, the sum of contributions in the SI-Graph yields the model's prediction, and thus a decomposition-based GNN explanation [Yuan, 2023]. The graph and the node game are directly linked as

$$\nu_g(T) = f_{g,\hat{y}}(\mathbf{X}^{(T)}) = \sigma_{\hat{y}}(\Psi(\{\{f_i(\mathbf{X}^{(T)})\} \mid v \in V\})) = \sigma_{\hat{y}}(\Psi(\{\{\nu_i(T)\} \mid i \in N\})), \tag{6.4}$$

where $\sigma_{\hat{y}}$ outputs the component of σ for the predicted class $\hat{y}$. The number of convolutional layers ℓ determines the *receptive field*, i.e. the message-passing range defined by its ℓ-hop neighborhood

$$\mathcal{N}_i^{(\ell)} := \{j \in N \mid d_g(i,j) \leq \ell\} \text{ with } d_g(i,j) := \text{length of shortest path from } v_j \text{ to } v_i \text{ in } g.$$

Consequently, the node game ν_i is unaffected by maskings outside its ℓ-hop neighborhood.

Theorem 6.2.3 (Node Game Invariance) *For a graph g and an ℓ-Layer GNN f_g, let ν_i be the GNN-induced node game with $i \in N$. Then, ν_i satisfies the invariance $\nu_i(T) = \nu_i(T \cap \mathcal{N}_i^{(\ell)})$ for $T \subseteq N$.*

Node Masking: Computing SIs on the graph game is a perturbation-based explanation [Yuan, 2023], where also other masking strategies were proposed [C. Agarwal, 2023]; for example, node masks [Ying, 2019; Yuan, 2021], edge masks [Luo, 2020; Schlichtkrull, 2021] or node feature masks [Ying, 2019; Yuan, 2021; C. Agarwal, 2023]. Our method is not limited to a specific masking strategy as long as it defines a game with the invariance property (Theorem 6.2.3). We implement our method with the well-established and theoretically understood BShap [Sundararajan, 2020b]. Alternatively, the T-induced subgraph could be used, but GNNs are fit to specific graph topologies, such as molecules, and perform poorly on isolated subgraphs [Alsentzer, 2020]. However, other maskings could emphasize different aspects of GNNs, which we leave to future work. Due to the invariance, we show that MIs and SIs of the graph game are sparse. To obtain our theoretical results, we require a structural assumption.

Assumption 6.2.4 (GNN Architecture) *We require the global pooling Ψ and the output layer σ to be linear functions, e.g. Ψ is a mean or sum pooling operation and σ is a dense layer.*

Linearity Assumption: In our experiments, we show that popular GNN architectures yield competitive performances under Assumption 6.2.4 on multiple benchmark datasets. In fact, such an assumption should not be seen as a hindrance, as it is the norm in GNN benchmark evaluations [Errica, 2020]. Furthermore, simple global pooling functions, such as *sum* or *mean*, are adopted in many GNN architectures [Xu, 2019; L. Wu, 2022], while more sophisticated pooling layers do not always translate into empirical benefits [Mesquita, 2020; Grattarola, 2024]. Likewise, a linear output layer is a common design choice, and the advantage of deeper output layers must be validated for each task [You, 2020].

6.2.2 Computing Exact Shapley and Möbius Interactions for the Graph Game

Given a GNN-induced graph game ν_g from Definition 6.2.2 with Assumption 6.2.4, i.e. Ψ and σ are linear, then the MIs of each node game are restricted to the ℓ-hop neighborhood. Intuitively, maskings outside the receptive field do not affect the node embedding. Consequently, due to the linearity of MIs in the vector space of games, the MIs of the graph game are restricted

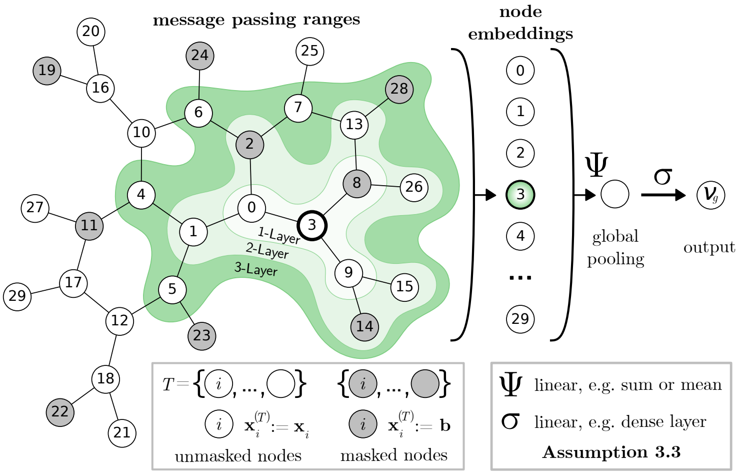


Figure 6.2: Illustration of the graph game ν_g. Masked nodes (grey) are imputed by baseline , and embeddings are determined by the receptive field (left). Subsequently, a linear pooling (Ψ) and output layer (σ) yield the GNN-induced graph game output.

Algorithm 1 GraphSHAP-IQ (simplified for exact SIs)

```
Require: Graph $g = (V, E, \mathbf{X})$, $\ell$-Layer GNN $f_g$, SI order $k$.
$\mathcal{I} \leftarrow \bigcup_{i \in N} \mathcal{P}(\mathcal{N}_i^{(\ell)})$
$\nu_g \leftarrow [f_g(\mathbf{X}^{(T)})]_{T \in \mathcal{I}}$
$m_g \leftarrow [\texttt{MI}(\nu_g, S)]_{S \in \mathcal{I}}$
$\Phi_k \leftarrow \texttt{MIToSI}(m_g, k)$
return MIs $m_g$, SIs $\Phi_k$
```

by all existing ℓ-hop neighborhoods. More formally, due to the invariance of the node games (Theorem 6.2.3), the MIs that are not fully contained in the ℓ-hop neighborhood $\mathcal{N}_i^{(\ell)}$ are zero.

Lemma 6.2.5 (Trivial Node Game Interactions) *Let $m_i : \mathcal{P}(N) \to \mathbb{R}^{d_\ell}$ be the MIs of the GNN-induced node game ν_i for $i \in N$ under Assumption 6.2.4. Then, $m_i(S) = \mathbf{0}$ for all $S \not\subseteq \mathcal{N}_i^{(\ell)}$.*

Lemma 6.2.5 yields that the interactions outside of the ℓ-hop neighborhood do not have to be computed. Due to the linearity of the MIs and Assumption 6.2.4, the interactions of the GNN-induced graph game are equally zero if they are not fully contained in any ℓ-hop neighborhood.

Proposition 6.2.6 (Trivial Graph Game Interactions) *Let $m_g : \mathcal{P}(N) \to \mathbb{R}$ be the MIs of the GNN-induced graph game ν_g under Assumption 6.2.4 and let $\mathcal{I} := \bigcup_{i \in N} \mathcal{P}(\mathcal{N}_i^{(\ell)})$ be the set of non-trivial interactions. Then, $m_g(S) = 0$ for all $S \subseteq N$ with $S \notin \mathcal{I}$.*

$\mathcal{I}$ is the set of non-trivial MIs, whose size depends on the receptive field of the GNN. The size of $\mathcal{I}$ also directly determines the required model calls to compute SIs.

Theorem 6.2.7 (Complexity) *For a graph g and an ℓ-Layer GNN f_g, computing MIs and SIs on the GNN-induced graph game ν_g requires $|\mathcal{I}|$ model calls. The complexity is thus bounded by*

$$|\mathcal{I}| \leq \sum_{i \in N} 2^{|\mathcal{N}_i^{(\ell)}|} \leq n \cdot 2^{n_{\max}^{(\ell)}} \leq n \cdot 2^{\frac{d_{\max}^{\ell+1}-1}{d_{\max}-1}},$$

where $n_{\max}^{(\ell)} := \max_{i \in N} |\mathcal{N}_i^{(\ell)}|$ is the size of the largest ℓ-hop neighborhood and $d_{\max}$ is the maximum degree of the graph instance.

In other words, Theorem 6.2.7 shows that the complexity of MIs (originally 2^n) for GNNs depends at most linearly on the size of the graph n. Moreover, the complexity depends exponentially on the connectivity $d_{\max}$ of the graph instance and the number of convolutional layers ℓ of the GNN. Note that this is very rough theoretical bound. In our experiments, we empirically demonstrate that in practice for many instances exact SIs can be computed, even for large graphs ($n > 100$). Besides this upper bound, we empirically show that the *graph density*, which is the ratio of edges compared to the number of edges in a fully connected graph, is an efficient proxy for the complexity.

6.2.3 GraphSHAP-IQ: An Efficient Algorithm for Shapley Interactions

Building on Theorem 6.2.7, we propose GraphSHAP-IQ, an efficient algorithm to compute SIs for GNNs. We first present the simplified method to compute exact SIs, which we then extend to an approximation in restricted settings. The computation of exact SIs with GraphSHAP-IQ is outlined in Section 6.2.1. First, GraphSHAP-IQ identifies the set of non-zero MIs $\mathcal{I}$ based on the given graph instance. The GNN is then evaluated for all maskings contained in $\mathcal{I}$. Given these GNN predictions, the MIs for all interactions in $\mathcal{I}$ are computed. Based on the computed MIs, the SIs are computed using the conversion formulas. Lastly, GraphSHAP-IQ outputs the exact MIs and SIs. For graphs with high connectivity and GNNs with many convolutional layers, computing exact SIs with Section 6.2.1 could remain infeasible. We thus present an extension of GraphSHAP-IQ for efficient *approximation* of SIs. The core idea is to introduce a hyperparameter λ, which limits the highest order of MIs that are computed. Hence, GraphSHAP-IQ outputs exact values, if $\lambda = n_{\max}^{(\ell)}$.

Dataset Description					*Model Accuracy by Layer (%)*								
					GCN			GAT			GIN		
Dataset	**Graphs**	d_{out}	**Nodes (avg)**	**Density (avg)**	1	2	3	1	2	3	1	2	3
Benzene (BNZ)	12000	2	20.6	22.8	84.2	88.6	90.4	83.1	85.1	85.7	84.9	90.5	90.8
FluorideCarbonil (FLC)	8671	2	21.4	21.6	82.4	83.9	83.9	82.4	82.2	82.4	84.6	87.2	87.1
Mutagenicity (MTG)	337	2	30.3	18.3	77.8	80.7	80.3	72.6	73.6	74.8	77.8	77.4	77.5
AlkaneCarbonyl (ALC)	1125	2	21.4	21.5	98.7	97.8	99.1	98.2	96.3	97.3	96.9	97.3	97.8
PROTEINS (PRT)	1113	2	39.1	42.4	75.2	71.1	74.0	75.3	60.5	79.8	79.3	74.9	67.7
ENZYMES (ENZ)	600	6	32.6	32.0	34.2	37.5	35.8	32.5	35.0	35.8	36.7	35.0	39.2
COX2 (CX2)	467	2	41.2	10.6	87.2	86.1	87.2	81.9	87.2	85.1	84.0	85.1	85.1
BZR (BZR)	405	2	35.8	13.0	90.1	87.7	90.2	88.9	86.4	87.7	88.9	88.9	88.9

Table 6.1: Summary of datasets, models, and GraphSHAP-IQ median speed-up.

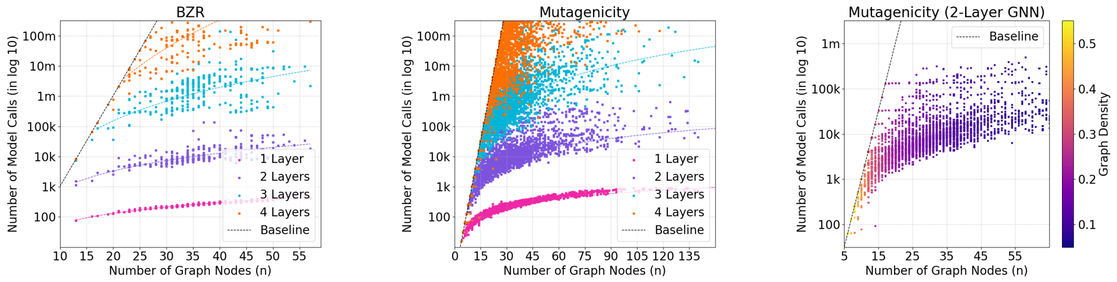


Figure 6.3: Complexity of GraphSHAP-IQ against model-agnostic baseline (dashed) in model calls (in $\log 10$) by number of nodes (n) for all instances of BZR (left) and MTG (middle, right) visualized by number of convolutional layers (left, middle) and graph density for a 2-Layer GNN (right).

6.3 Experiments

In this section, we empirically evaluate GraphSHAP-IQ for GNN explainability, and showcase a substantial reduction in complexity for exact SIs (Section 6.3.1), benefits of approximation (Section 6.3.2), and explore the SI-Graph for Water Distibution Networks (WDN) and molecule structures (Section 6.3.3). Following Amara [2022], we trained a GCN [Kipf, 2017], GIN [Xu, 2019], and GAT [Velickovic, 2018] on eight real-world chemical datasets for graph classification and a WDN for graph regression, cf. Table 6.1. We trained all models under Assumption 6.2.4 and reported test accuracies, that are comparable to existing benchmarks [Errica, 2020; You, 2020].

6.3.1 Complexity Analysis of GraphSHAP-IQ for Exact Shapley Interactions

In this experiment, we empirically validate the benefit of exploiting graph and GNN structures with GraphSHAP-IQ. The complexity is measured by the number of evaluations of the GNN-induced graph game, i.e. the number of model calls of the GNN, which is the limiting factor of SIs [Fumagalli, 2023; Kolpaczki, 2024b; Muschalik, 2024]. For every graph in the benchmark datasets, described in Table 6.1, we compute the complexity of GraphSHAP-IQ, where the first upper bound from Theorem 6.2.7 is used if $\max_{i \in N} |\mathcal{N}_i^{(\ell)}| > 23$, i.e. the complexity exceeds $2^{23} \approx 8.3 \times 10^6$. Fig. 6.3 displays the log-scale complexity (y-axis) by the number of nodes n (x-axis) for BZR (left) and MTG (middle, right) for varying number of convolutional layers ℓ (left, middle) and by graph density for a 2-Layer GNN (right). The model-agnostic baseline is represented by a dashed line. Fig. 6.3 shows that the computation of SIs is substantially reduced by GraphSHAP-IQ. Even for large graphs with more than 100 nodes, where the baseline requires over $2^{100} \approx 10^{30}$ model calls, many instances can be exactly computed for 1-Layer and 2-Layer GNNs with fewer than 10^5 evaluations. In this case, the complexity of GraphSHAP-IQ is almost constant. Figure 6.3 (right) shows that the graph density is an efficient proxy of complexity, with higher values for instances near the baseline.

6.3.2 Approximation Quality of GraphSHAP-IQ

For densely connected graphs and GNNs with many layers, exact computation of SIs might still be infeasible. We thus evaluate the approximation of SIs with GraphSHAP-IQ, which computes all MIs up to order λ. We compare GraphSHAP-IQ with current state-of-the-art model-agnostic baselines to approximate SIs for SVs and 2-SIIs and large graphs. For the SV, we apply *KernelSHAP* [Lundberg, 2017], *Unbiased KernelSHAP* [Covert, 2021a], *k-additive SHAP* [Pelegrina, 2023], *Permutation Sampling* [Castro, 2009], *SVARM* [Kolpaczki, 2024a], and *L-Shapley* [J. Chen, 2019]. We estimate 2-SII values with *KernelSHAP-IQ* [Fumagalli, 2024], *Inconsistent KernelSHAP-IQ* [Fumagalli, 2024], *Permutation Sampling* [Tsai, 2023], *SHAP-IQ* [Fumagalli, 2023], and *SVARM-IQ* [Kolpaczki, 2024b]. We select 10 graphs containing $30 \leq n \leq 40$ nodes for the *MTG*, *PRT*, and BZR benchmark datasets. For each graph, we compute MIs for each λ with GraphSHAP-IQ and estimate SVs and 2-SIIs with the same number of model calls as the baselines. Ground truth (GT) SVs and 2-SIIs are computed via GraphSHAP-IQ and compared in terms of MSE (lower is better). In Fig. 6.4, we observe

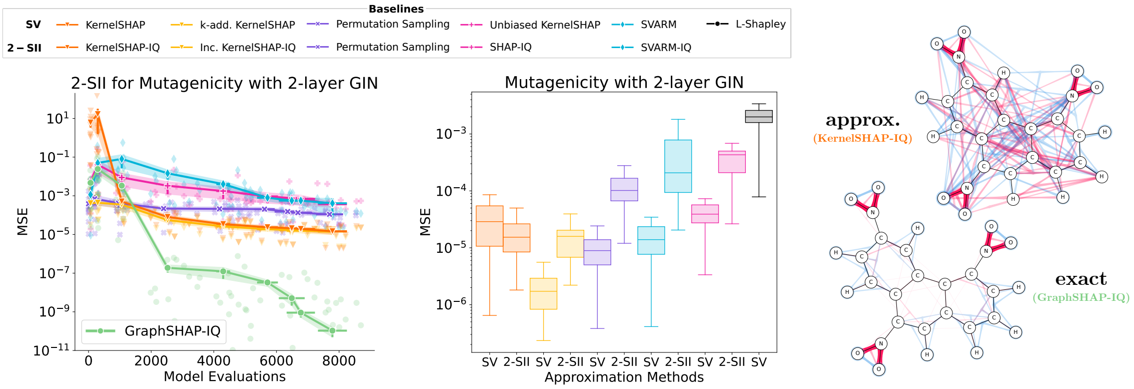


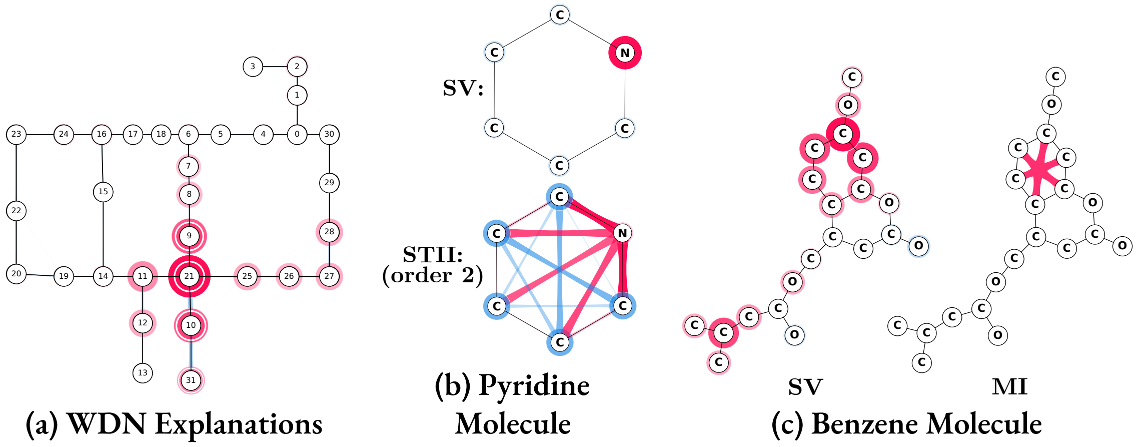
Figure 6.4: Approximation of SIs with GraphSHAP-IQ (green) and model-agnostic baselines for *MTG* (left). At budgets, where GraphSHAP-IQ reaches exact SIs, the baselines achieve varying estimation qualities (middle) which prohibits drawing reliable conclusions from the explanations (right).

that GraphSHAP-IQ outperforms the sampling-based baselines in settings with a majority of lower-order MIs (*MTG* and 2-layer GIN). Model-agnostic baselines exhibit a comparable estimation performance in settings with higher-order interactions (*PRT* and 2-layer GAT). Yet, approximated SIs exhibit an approximation error, cf. Fig. 6.4 (right), and should be interpreted with care.

6.3.3 Real-World Applications of Shapley Interactions and the SI-Graph

We now apply GraphSHAP-IQ in real-world applications. **Monitoring water quality** in WDNs requires insights into a dynamic system governed by local partial differential equations. Here, we investigate the spread of chlorine as a graph-level regression of a WDN, where a GNN predicts the fraction of nodes chlorinated after some time. Based on the Hanoi WDS [Vrachimis, 2018], we create a temporal WaterQualirt (WAQ) dataset containing 1 000 graphs consisting of 30 time steps. We train and explain a simple GNN, which processes node and edge features like chlorination level at each node and water flow between nodes. Fig. 6.5 show that 2-SIIs spread over the WDS aligned with the water flow. Therein, mostly first-order interactions influence the time-varying chlorination levels.

Benzene rings in molecules are structures consisting of six carbon (C) atoms connected in a ring with alternating single and double bonds. We expect a well-trained GNN to identify ben-



(a) WDN Explanations

(b) Pyridine Molecule

(c) Benzene Molecule

Figure 6.5: Exact SIs values for three example graph structures. SVs illustrate the trajectory of chlorination levels in a WDN (a). STII (order 2) values showcase that a Pyridine molecule is not classified as benzene (b), and the largest positive MI for a benzene molecule is the benzene ring (c).

zene rings to incorporate higher-order MIs (order ≥ 6). Fig. 6.5 shows two molecules and their SI-Graphs computed by GraphSHAP-IQ. The Pyridine molecule in Fig. 6.5 (b) is correctly predicted to be non-benzene as the hexagonal configuration features a nitrogen (N) instead of a carbon, which is confirmed by the SVs highlighting the nitrogen. STIIs of order 2 reveal that the MI of nitrogen is zero and interactions with neighboring carbons are non-zero, presumably due to higher-order MIs, since STII distributes all higher-order MIs to the pairwise STIIs. In addition, STIIs among the five carbon atoms impede the prediction towards the benzene class. Interestingly, opposite carbons coincide with the highest negative interaction. The MIs for a benzene molecule with 21 atoms in Fig. 6.5 (c) reveal that the largest positive MI coincides with the 6-way MI of the benzene ring.

Mutagenicity of molecules is influenced by special compounds like nitrogen dioxide (NO_2) [Kazius, 2005]. Fig. 6.1 shows SIs for a MTG molecule, which GNN accurately identifies as mutagenic. 2-SIIs and MIs both show that not the nitrogen atom but the interactions of the NO_2 bonds contributed the most.

6.4 Final Remarks

We presented GraphSHAP-IQ, an efficient method to compute SIs that applies to all popular message passing techniques in conjunction with a linear global pooling and output layer. Assumption 6.2.4 is a common choice for GNNs [Xu, 2019; Errica, 2020; L. Wu, 2022] and does not necessarily yield lower performance [Mesquita, 2020; You, 2020; Grattarola, 2024], which is confirmed by our experiments. However, exploring non-linear choices that preserve trivial MIs is important for future research. Masking node features with a fixed baseline, known as BShap, preserves the topology of the graph structure and is a well-established approach [Sundararajan, 2020a]. Nevertheless, alternatives such as induced subgraphs, edge-removal, or learnable masks, could emphasize other properties of the GNN and should be explored. Lastly, the deterministic approximation of GraphSHAP-IQ shows mixed results when higher-order MIs dominate, where GNN-informed sampling methods using Proposition 6.2.6 is promising future work.

Part III

Applied Contributions

7

A Graph-Based Approach for Candidate Data in HR Analytics

In the world of HR, the integration of advanced data analysis techniques has become a key factor in gaining competitive advantages [Wirtky, 2016]. This chapter embarks on an in-depth exploration of the application of ML techniques to HR data, specifically concentrating on developing and analyzing candidate networks via graph-based methodologies.

The domain of HR analytics has persistently dealt with a notable challenge: the limited availability of publicly accessible datasets. To address this issue, our primary research focus has been to generate a new dataset derived from real-world recruitment cases. Although this dataset has not been publicly released due to privacy matters, it forms a crucial underpinning for our research initiatives. The dataset includes both structured data obtained from questionnaires and unstructured information extracted from candidates' CVs, collected with the close collaboration of Amajor's team.

The study in this chapter brings two main contributions[1]. Initially, we explore methods to convert standard HR data into graph structures, where candidates are depicted as nodes, and their significant interrelationships constitute the edges. This methodology enables us to repre-

[1] This chapter is based on **Frazzetto**, **Paolo**, Muhammad Uzair-Ul-Haq, and Alessandro Sperduti [2023b]. "Enhancing Human Resources through Data Science: a Case in Recruiting". In: *Proceedings of the 2nd Italian Conference on Big Data and Data Science (ITADATA 2023), Naples, Italy, September 11-13, 2023*. Vol. 3606. CEUR Workshop Proceedings. CEUR-WS.org. URL: https://ceur-ws.org/Vol-3606/paper71.pdf

sent the complex network of links within the talent pool, potentially uncovering patterns and insights that might be missed in conventional tabular formats. Secondly, we leverage Large Language Models (LLMs) to draw out rich, contextual information from candidates' CVs. This approach allows us to capture detailed nuances about candidates' skills, experiences, and potential, which might be overlooked by traditional keyword-based systems.

The primary objective of this chapter is to investigate whether these graph-based representations of candidate data, enhanced by LLM-extracted features, can outperform traditional tabular data approaches in various HR analytics tasks. We conduct a comparative analysis, examining the efficacy of GNNs against conventional NN and other baseline models. This exploration serves as a crucial first step in understanding the potential of graph-based methods in HR analytics. The approaches presented here lay the groundwork for more sophisticated methods that will be explored in subsequent chapters.

7.1 Challenges in HR Data Analytics

The automated extraction of information from CVs, job postings (JP), and related HR documents has long caught the attention of many researchers and companies [Bizer, 2005; Yu, 2005; Yi, 2007]. However, the lack of publicly available datasets bottlenecks most of the progress. Despite the scarcity of datasets, there have been some efforts to extract information from resumes and job postings with the aim of boosting HR performance.

For example, Fernández-Reyes [2019] proposed the use of an average word embedding model for CV retrieval based on the job description (JD). The embeddings of CVs are trained from scratch and combined with the pre-trained word2vec embeddings using a hybrid embedding model. The JPs are also embedded using a pre-trained word2vec model, and cosine similarity is used as a measure to find relevance between the CV and job description. However, to the best of our knowledge, the dataset used in the research is not publicly available.

Jiang [2020] proposed the use of machine learning to match candidates with job postings on online platforms (CJM). The authors used real-case data collected by the recruitment team between January 2019 and October 2019. They collected about 13 thousand jobs with 580 thousand candidates and 1.3 million resumes. However, the authors claim the dataset to be sensitive and do not release it publicly. Similarly, Yao [2022] proposed a method to quantify the CJM. They model the task with distantly supervised skill extraction to identify the skill entities from job postings and resumes using skill entity dictionaries. The relevance between a resume and a job description is measured according to the matching score of the skill entities. The dataset used in the research contains 21 thousand job postings and 86 thousand CVs provided by a high-tech company. So far, the authors have not released the publicly available dataset.

On the other hand, Mike Zhang [2022] released a novel dataset for skill extraction on English job postings called SKILLSPAN. The authors also outlined the annotation guidelines created by domain experts to annotate hard and soft skills in job postings. The dataset consists of 14.5 thousand sentences, of which 12.5 thousand are annotated. The dataset is divided into three categories: BIG, HOUSE, and TECH. The authors only released the HOUSE and TECH categories of the dataset to the public. However, the dataset is limited to job postings, and only skill entities are annotated.

Given the limited availability of publicly available datasets, we address this problem at hand and build our HR dataset.

7.2 Data Collection

The gathering of reliable data in HR recruitment is a complex task that goes beyond the realm of academia, as it requires external support from the industry. While academic research provides valuable insights and theoretical frameworks, it often falls short of capturing the intricacies and practicalities of the recruitment process in real-world settings. Additionally, HR recruitment involves gathering a wide range of data, including resumes, application forms, psychometric assessments, interviews, and performance evaluations. This data collection process requires collaboration with organizations willing to share their recruitment data and provide access to their internal information systems.

This whole research has been made possible by the partnership and support of Amajor SB (Section 2.2). This partnership enabled us to obtain authentic and diverse datasets, allowing us to analyze and develop models that closely mirror the challenges and complexities faced by HR practitioners. This collaborative approach ensures that our research findings are relevant, applicable, and aligned with the practical needs of the industry, ultimately leading to more effective and impactful HR recruitment strategies.

7.2.1 Dataset Description

The data were collected from real-case candidates' applications for 195 different job postings and vacancies, mostly in northeast Italy. The positions vary in terms of seniority, role, business area, and corporate size. The time frame spans from January 1st, 2021 to May 31st, 2023. From a starting pool of more than $13,000$ applications, we filtered out those candidates who did not fill in the privacy consent for this research, those who did not complete the assessment questionnaire, and those who did not submit their CVs.

7.2.2 Classes Identification

The job selection process typically involves several steps that candidates must navigate. In our scenario (part of Section 2.2.4), candidates must first submit their application, including their CV and other basic data, and fill in the assessment questionnaire. After the initial screening by HR recruiters, candidates are invited to participate in one or more interviews, which can be conducted in various formats, such as video interviews or in-person meetings. From the outcome of the interviews, only a handful of candidates are shortlisted and presented to the future employer. In the final stage, the employer selects one candidate, negotiates with them the job offers, reviews the employment contracts, and completes the necessary paperwork before officially joining the organization.

In an ML framework, we can consider each of the previous states as a label, thus modeling the process as a multi-class classification problem. Alternatively, since the process is consequential, one could model it as a regression task over an interval. Nonetheless, for this first analysis and release of the dataset, we opted to consider *binary labels*. We realized that in real-case applications, there are many deviations from this standard process; besides, the data recorded in the information systems may be miss-classified or missing for some candidates and stages. Therefore, we focus on those candidates who completed all stages up to the first interview—an HR specialist has checked their CV, questionnaire, and interview and made the decision to bring them further in the selection (positive labels) or deemed that they were not the best-suited candidates for that role (negative labels). These steps left out $N = 2,647$ valid candidates, of which $1,674$ (63%) with positive labels and 973 (37%) with negative ones. More numerous and complex data structures will be presented in the following chapters, resulting from additional collection, cleaning, and pre-processing of the data.

7.2.3 CV Embeddings

State-of-the-art Large Language Models (LLMs) have succeeded in various natural language processing tasks [Zhao, 2024]. These models can be pre-trained on large corpora to capture contextual information of words in a text. CVs are unstructured documents that consist of long textual information. In this case, we use an `XLM-RoBERTa-Longformer` [Conneau, 2020], a multilingual model with an input size of $4,096$ tokens. The multilingual characteristic allows us to capture information in different languages, whereas the larger input size enables processing long documents.

The documents are pre-processed by removing stop-words, extra spaces, and special characters. These documents are tokenized and then passed through the pre-trained XLM-RoBERTa-Longformer to extract the word embeddings of all the tokens. Each token is represented by a 768-dimensional feature vector. Therefore, a document consisting of N tokens returns a $N \times 768$ dimensional feature matrix. The average document length is of 546 tokens, ranging from a minimum of 53 to a maximum of $8,526$ tokens. Processing such large matrices is computationally expensive; therefore, we average the embedding vector of all the tokens in the document, resulting in a 768-dimensional feature vector representing the document in an embedding space.

7.2.4 Questionnaire Data

Personality and behavior assessments through questionnaires are one valuable tool in HR selection processes and organizational psychology [Bailey, 2017]. These assessments aim to gain insight into candidates' traits, characteristics, and behavioral tendencies, providing a deeper understanding of their fit within the organization and job role. Some questionnaires are designed specifically for personality assessment and others are designed to measure some abilities or behaviors. There exists a plethora of different kinds of tests, with different validity and usage among HR practitioners [Furnham, 2008]. Nevertheless, the quantitative nature of and its ease of collection allow collection, allows for rigorous statistical analysis and enables standardized evaluation across candidates, promoting fairness and consistency in the selection process.

We collected questionnaire data following Amajor's business model (Section 2.2.2). The tool used for the candidates' assessment is the so-called A+ Questionnaire: a set of 242 questions with 3-scale Likert-type answers (yes/maybe/no or similar, see Fig. 7.1) covering various aspects of one's behavior, habits, and personality, developed by the company team after working alongside more than 120 clients over a period of five years [Peronato, 2022]. The answers are grouped and processed following a proprietary factor model that gives an estimate of one's hidden traits; however, its analysis and discussion go beyond the scope of this work. In the following section, we describe a novel and general approach to exploit Likert-type questionnaire data to find patterns among respondents.

Is it necessary to give up something today to be better off tomorrow?

No Maybe Yes

Figure 7.1: Example of a Likert-scale question used in the candidates' assessment A+ Questionnaire.

7.2.5 From Questionnaires to Graph

Likert-type scales are widely employed in academic and industrial settings to capture human facets due to their user-friendly nature, simplicity of development, and ease of administration [Joshi, 2015]. It enables respondents to answer questions in a closed-form way, picking only one value on an ordered scale according to some sort of preference or agreement. Due to the fact that the perceived distance between two consecutive items cannot be defined or presumed equal [Munshi, 2014], such a scale cannot be analyzed by classical statistical methods defined on a metric space or parametric tests but requires specific modeling and assumptions [Disegna, 2022]. In order to link candidates that give similar answers, we resort to graphs and Network Science. Such a more expressive data structure occurs in many fields of science and engineering [Barabási, 2013]. However, translating tabular data to graphs is not trivial [K. Zhou, 2022; J. Liu, 2023] as it requires domain knowledge and heuristics to define the nodes and their relationships.

Our approach to tackling these issues is straightforward and takes advantage of the specific structure of Likert-type data. Given any Likert scale Questionnaire $\mathcal{Q} = \{q_1, q_2, \dots, q_n\}$ made up of n questions, each possible answer a_i takes value in an ordered set that w.l.o.g. we can define as $\mathcal{A} = \{1, 2, \dots, L\} \subset \mathbb{N}^n$, so $a_i \in \mathcal{A}$, $\forall i = 1, \dots, n$. The order relation depends on each q_i, and we assume that it is universal, i.e., the questionnaire is well-written, and all respondents understand each question. In this way, each completed questionnaire can be formulated as a specific collection of all the possible answers $\mathbf{a} = \{a_1, \dots, a_n\}$, $\forall a_i \in \mathcal{A}$ and a respondent can be described as a function $r : \mathcal{Q} \to \mathcal{A} \times \ldots \times \mathcal{A} = \mathcal{A}^n$, $r(\mathcal{Q}) = \mathbf{a}$. We desire to link candidates/respondents that provide similar answers, thus having similar behaviors and personalities, without resorting to the hidden variables given by factor analysis. Next, we have to define a distance function $d(\mathbf{u}, \mathbf{v}) : \mathcal{A}^n \times \mathcal{A}^n \to \mathbb{R}$ between two responses $\mathbf{u}, \mathbf{v}$. Our desiderata are that respondents who give the exact same answers will be closer, whereas when the answers are on the opposite side of the scale, the distances should be greater. Additionally, we want to avoid the Euclidean metric since it scales quadratically with L, but Likert scales are perceived as linear [Munshi, 2014]. Therefore, the ideal candidate is the Manhattan distance or ℓ_1 norm:

$$d_M(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} |u_i - v_i| \quad u_i, v_i \in \mathcal{A}. \tag{7.1}$$

We brought this idea further by considering that Likert-type answers are usually contrasting, i.e., one end of the scale is the opposite of the other, with all the ranges in between. Therefore, in the case of an odd number of choices L, a middle value is perceived as a neutral or indefinite answer [Boateng, 2018]. We want to emphasize this contrast and penalize the neutral answers, as they provide little insight into the analysis. For these reasons, we center our answer set in zero $\widetilde{\mathcal{A}} = \{-\lfloor L/2 \rfloor, \ldots, 0, \ldots, \lfloor L/2 \rfloor\}$ and we exploit this symmetry with a redefined Bray-Curtis similarity [Bray, 1957]:

$$d_{BC}(\mathbf{u}, \mathbf{v}) = 1 - \frac{\sum_i |u_i - v_i|}{\sum_i |u_i + v_i|}. \tag{7.2}$$

Notice that this measure of similarity has the desired properties, being normalized to 1 when the answers of two questionnaires are exactly the same. On the contrary, it diverges to $-\infty$ when they are always at the opposite. In practice, $d_{BC}(\mathbf{u}, \mathbf{v}) \leq 0 \iff \sum_i |u_i - v_i| \geq \sum_i |u_i + v_i|$ and the latter holds when the majority of answers have opposite signs, hence meaning.

In our specific business case, we have $L = 3$ and thus $\widetilde{\mathcal{A}} = \{-1, 0, 1\}$, where 1 stands for the positive/affirmative answer, 0 is maybe/neutral, and -1 is no/negative. We then computed the pairwise similarity of Eq. (7.2) for all pairs. These values directly translate into a graph in which each node is a candidate, who is connected to all other candidates by means of a weighted adjacency matrix $\mathbf{A}$, whose entries are consequently defined as $\mathbf{A}_{uv} = d_{BC}(u, v) = \mathbf{A}_{vu}$. In this way, we obtain a fully connected graph with an order of $N^2 \simeq 3.5 \times 10^6$ links. We also apply two heuristics to reduce its complexity and keep only the most significant connections. First, we set the negative values to zero, enforcing no similarity between such different questionnaires. This results in the removal of 1.47×10^5 links. The corresponding boxenplot distribution is shown in Fig. 7.2, noticing that we retain a homogeneous distribution of similarities along a right-tail of candidates with almost identical answers. Secondly, our aim is to drop the links with a weight close to zero, starting from the lowest values. Therefore, we apply edge percolation [M. E. J. Newman, 2001] and heuristically stop when the largest component has 95% of the total nodes. As shown in Fig. 7.3, an additional 1.1×10^5 edges can be removed.

The corresponding threshold of the similarity $d_{BC}(u,v)$ is 0.07, so all the remaining links connect candidates with a similarity greater than this threshold. The basic statistics of the obtained graph are reported in Table 7.1, and an illustrative example is shown in Fig. 7.4.

Such a process allowed us to find a reasonable graph of candidates based on their responses on a Likert-type questionnaire. Our research question is to test whether such an approach can improve the identification of patterns and the prediction of the class of new candidates, given that they provided similar answers to other labeled candidates.

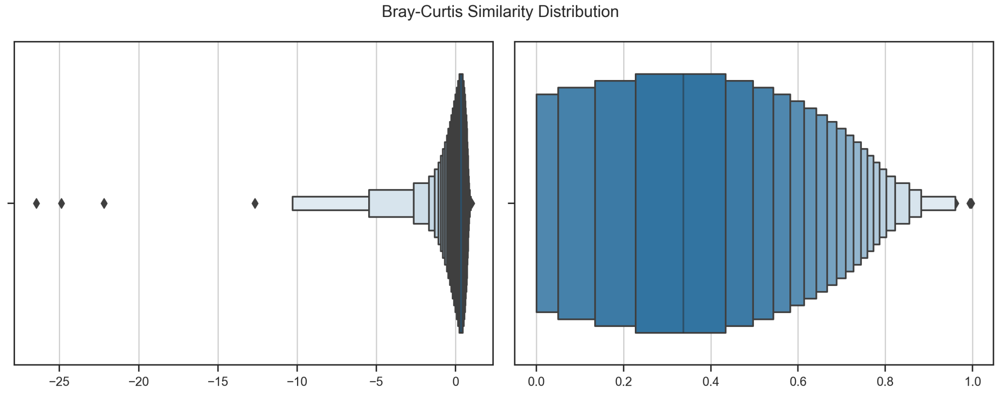
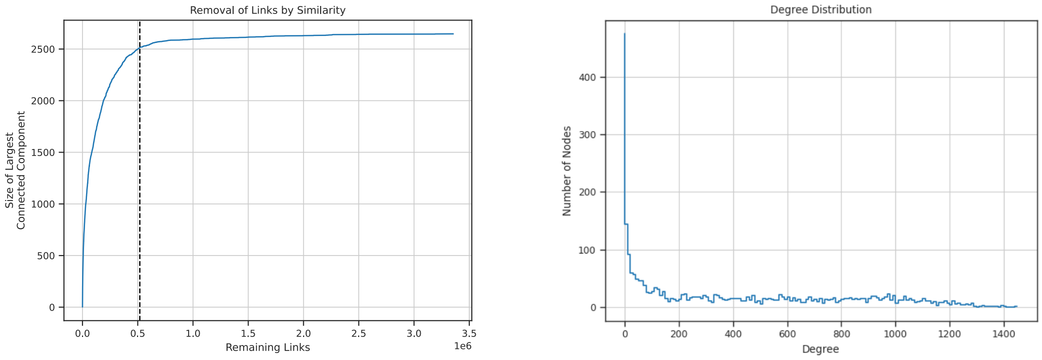


Figure 7.2: Boxenplot distribution of the Bray-Curtis similarity for all N^2 questionnaire pairs; before (left) and after (right) clipping the negative values.



(a) (b)

Figure 7.3: (a): Size of the largest connected component by removal of the links with ascending similarity. The dotted line indicates the point where the largest component has 95% of the original nodes; thus, we retain all the connections up to that point. (b): Degree distribution of the resulting graph.

#Nodes	#Links	Avg. Degree	Avg. Clustering Coeff.	Avg. Path Length	Graph Diameter	Graph Density
2647	518994	392	0.615	2.141	8	0.148

Table 7.1: Basic Properties of the Candidates Graph

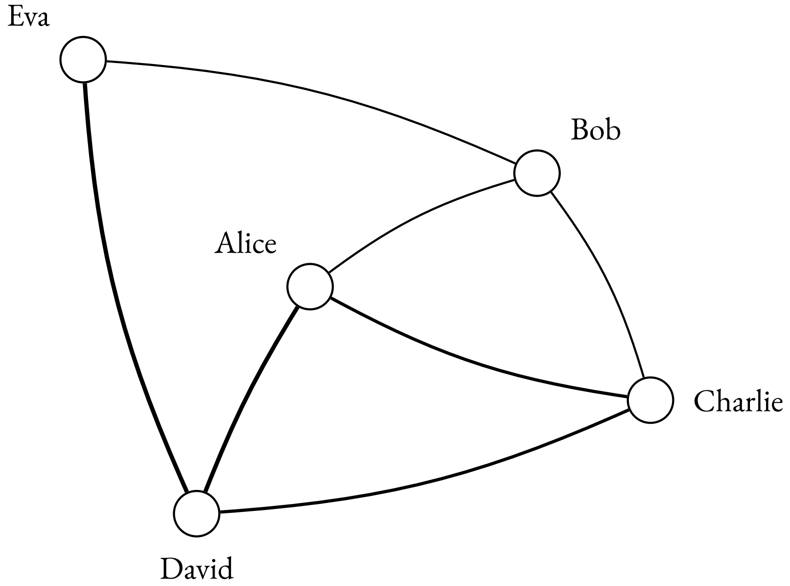


Figure 7.4: Simplified example of Graph of Candidate Profiles based on the A+ Questionnaire answers. Each candidate is a node and is connected with other nodes if their answer distance is $d_{BC}(u, v) \geq 0.07$, and the edge thickness denotes its value.

7.3 Candidates Classification

This section explores how ML can be leveraged to perform candidate classification based on this novel dataset. Two different approaches have been investigated—relying on unstructured or structured data.

The first approach focused on tabular data analysis, where traditional machine learning algorithms were applied to extract insights and patterns from unstructured data. This pipeline usually involves feature engineering, model selection, architecture search, hyper-parameter optimizations, and training on the tabular dataset to make predictions and classifications. Each of these steps may be challenging on its own; therefore, we resorted to off-the-shelf AutoML tools to automate this procedure. In particular, we exploited AutoGluon [Erickson, 2020] for its simplicity and the availability of Neural Networks among its models.

Besides employing the tabular method, we investigated the application of GNNs to analyze the previously extracted graph structures of the dataset. By representing the data as a graph, we leveraged GNNs to capture the relationships and dependencies that emerge from questionnaire data among the entities. The GNN model enables us to learn from both the nodes' attributes (i.e., the CV embeddings) and the relational information present in the graph, thereby capturing complex patterns and interactions that traditional tabular approaches might miss. Contrary to tabular or multi-modal data, AutoML tools for GNNs are still in their infancy and are under active development [K. Cao, 2023]. However, we adopted our graph for the AutoGL framework [Guan, 2021], which enables us to test some of the most common GNN layers for the node classification task.

By employing these two complementary approaches, we aimed to comprehensively understand the dataset and extract meaningful insights from different perspectives. All the experiments were conducted in the same environment and employed open-source libraries. We tested different models for each scenario, with 10-fold cross-validation on a $[80, 10, 10]$ train/validation/test split. We disabled any other feature selection/engineering techniques, as we already employ features extracted from raw HR data, and the models' comparisons would be unfair. For the same reason, we turned off bagging and multi-layer stack ensembling useful to boost predictive accuracy [Caruana, 2004].

7.3.1 Results

The experimental results are reported in Table 7.2. Our baseline model is a naive class prior probability classifier that always predicts the most common class (the "positive" candidates) with an expected accuracy of 63.24%. We considered RandomForest [Breiman, 2001] and CatBoost [Prokhorenkova, 2018] since they have been proven to be fast to train and effective on tabular data in many domains. Concerning the GNNs, we selected the modules GraphSAGE [Hamilton, 2017], GAT [Velickovic, 2018], and CGN [Kipf, 2017]. The Neural Networks and GNNs were trained with the default hyperparameters and architecture spaces; therefore, their performance could improve with more extensive model selections.

CV embeddings alone are substantially equivalent to the class prior classifier. This suggests that CV embeddings should be improved, and the baseline LLMs are unable to grasp essential information without any domain knowledge or fine-tuning [Uzair-Ul-Haq, 2024]. Considering that the Questionnaire is an essential element in the examined HR process, the fact that

Data	Model	Accuracy [%]	Std. Dev.
Baseline	Class Prior Classifier	63.24	-
Tabular (CV Embeddings)	NN	63.40	2.16
	RandomForest	61.51	1.79
	CatBoost	64.53	1.51
Tabular (Questionnaire)	NN	75.78	2.54
	RandomForest	76.99	1.42
	CatBoost	77.16	1.20
Tabular (CV Emb. + Qst.)	NN	76.45	2.44
	RandomForest	75.27	1.09
	CatBoost	76.91	1.46
Graph (CV Emb.)	GraphSAGE	**77.35**	**1.77**
	GAT	76.60	1.96
	GCN	74.33	2.77

Table 7.2: Experimental results on the test set for different models and data structures.

the answers alone are a better predictor is unsurprising. CatBoost performs the best, being designed for categorical data such as Likert-type questionnaires. Adding the embeddings along with questionnaire data does not lead to improvements but rather results in slightly degraded performances. Therefore, the CV embeddings are not as informative, and also considering their high dimensionality, they deteriorate classification. In spite of that, the graph topology we proposed turns out to be valuable for classification, performing slightly better than the corresponding tabular dataset of embeddings and answers to the questionnaire. It seems that our rationale for linking Likert-type data is effectively linking similar candidates in a meaningful fashion.

In summary, we gathered, processed, and evaluated an initial version of an HR recruitment dataset. We utilized LLM to ensure anonymity and investigate the current limitations of these models within this field. We examined the dataset through a data-driven approach, treating it as a binary classification task. Traditional ML methods showed efficacy when paired with assessment questionnaire data, and we presented a method to transform Likert-scale data into graphs, maintaining their inherent patterns and relationships. The subsequent chapters provide a detailed discussion of the advancements made in this area.

8
LLMs for Heterogeneous Graph of Candidate Profiles

The landscape of HRM is undergoing a profound transformation driven by the rapid advancement of digital technologies and AI, particularly due to LLMs. The recruitment process involves abundant digital data, leading to increased interest in utilizing automated techniques for candidates-job matching (CJM). Nevertheless, current methods frequently fail to grasp the subtle relationships between job requirements and candidate attributes, while also overlooking the significance of diverse job vacancies. This chapter describes a novel pipeline that harnesses the power of LLMs and GNNs to gain deeper insights into the recruitment process and improve CJM, paving the path for future research[1].

8.1 Methodology

The main idea is that similar candidates—with similar CVs—are also similarly suited for a job opening. Although this is quite a strong assumption and belittles the complexity of resourcing, this research question leads us to build models that can nevertheless support HR recruiters in

[1] This chapter is based on **Frazzetto, Paolo**, Muhammad Uzair Ul Haq, Flavia Fabris, and Alessandro Sperduti [2024b]. "From Text to Talent: A Pipeline for Extracting Insights from Candidate Profiles". In: *The 3rd Italian Conference on Big Data and Data Science, (ITADATA 2024), Pisa, Italy, September 17-19, 2024*. Proceedings not yet available.

their everyday tasks. This specific pipeline is then distinguished by three key points:

1. **Utilization of Recent LLMs:** We employ state-of-the-art Large Language Models to parse and extract multifaceted information from CVs, addressing the challenges posed by unstructured data. This allows for a more sophisticated understanding of candidate profiles, including both hard and soft skills, educational background, and work experience.
2. **Multiple Job Vacancies Assessment:** Unlike previous studies that often focus on matching candidates to a single job description (JD), our pipeline is designed to handle multiple job vacancies simultaneously. This approach more accurately reflects the reality of authentic recruitment teams that deal with numerous and diverse selections.
3. **Heterogeneous Graph Construction:** Based on the entities extracted from CVs, we construct a heterogeneous graph that represents the complex relationships among candidates. This graph-based representation allows for a more nuanced understanding of how different attributes and experiences interconnect, providing a richer context for candidate evaluation.

We then train various GNN architectures on this heterogeneous graph, allowing the model to learn the patterns of such a recruitment ecosystem. In the upcoming sections, we provide the theoretical underpinnings of our method, give a detailed explanation of our procedure, and analyze the outcomes of our experimental tests.

8.2 Dataset Feature

The data is based on actual activities performed by Amajor's HR recruiters and applying candidates. As it happens, real-world data is often characterized by its heterogeneity, inconsistency, and fragmentation across multiple sources, necessitating a labor-intensive and time-consuming data integration and cleansing process. This complexity manifested in our research through the need to gather information from various databases, use purposely-build software and API calls to various cloud systems, and incorporate offline data sources, frequently requiring manual intervention to ensure data quality and coherence. Thanks to this team effort, it was possible to collect comprehensive HR data of $S = 39$ completed selection processes of different clients, totaling $C = 5,461$ candidates. This dataset comprises diverse roles and positions in multiple sectors, featuring a heterogeneous population of employees from various backgrounds and geographic locations. Some candidates have applied to multiple similar vacancies (or had been reconsidered by the recruiters for other suited positions), yielding to $6,624$ unique candidate-application combinations. The selection processes had $170_{\pm 120}$ applicants, ranging from a minimum of 31 to a maximum of 648. To protect the candidates' privacy, we removed all personally identifiable information from the dataset (such as names, emails, and phone numbers). We also eliminated potential biasing factors such as age and gender. However, biases and fairness are not the scope of this research and should be further investigated (Section 3.1).

For our proposes, the dataset has two main components for each candidate: their CV and A+ Questionnaire results. The CV provides rich, unstructured text data about the candidate's qualifications, experience, and skills, while the questionnaire's traits offer a standardized, quantitative perspective on each candidate's habits and behaviors, complementing the qualitative information in their CV. Each questionnaire trait takes numerical values in the $[-100, 100]$ range and has been standard normalized. Missing values have been imputed by their median.

Target Labels

Candidates were assigned an ordinal label y for each selection process or vacancy, reflecting their progression through the recruitment pipeline. The labels y were defined as follows: `Rejected` → 0, `Screened/Interviewed` → 1, `Shortlisted` → 2, `Hired` → 3. This labeling scheme was developed after discussions with experienced recruiters. It was determined that predicting the higher categories (1, 2, and 3) would provide the most value for the recruitment process, as these represent candidates who progressed in the application stage. Our predictive modeling efforts thus focus on these more meaningful outcomes, aiming to identify candidates likely to reach the interview, offer, or hiring stages. Due to the nature of this setting, the classes are highly unbalanced, posing substantial challenges for model training. The class distribution statistics, grouped across all 39 recruitment processes, are reported in Table 8.1.

y	Mean	Min	Max	Std
0	94.77	78.87	99.22	3.88
1	3.95	0.39	18.31	3.20
2	1.47	0.27	4.17	1.09
3	0.95	0.27	3.23	0.71

Table 8.1: Candidate label distribution for selected 39 job openings (%).

Feature Extraction

Given that CVs are typically unstructured data, traditional rule-based approaches often struggle to accurately parse and interpret the various formats. Moreover, ML-based approaches rely on annotated datasets for information extraction [Mike Zhang, 2022; Uzair-Ul-Haq, 2024]. Therefore, to transform the unstructured text of CVs into structured data, we leveraged LLMs (namely, GPT-4 [Achiam, 2024]) since their advanced natural language processing capabilities proved to be effective for information extraction [Tang, 2024]. Also, the GPT-4 model can efficiently parse through diverse document formats, identifying relevant entities from the text [Wei, 2024].

In this setting, we harnessed the capabilities of GPT-4 to extract five entities from each CV: in mathematical terms $\mathcal{E} = \{$*Soft Skills*, *Hard Skills*, *Industry Sector*, *Education*, *Language Skills*$\}$, thereby creating a structured representation of each candidate's profile containing the keywords related to these entities. We denote one of these entities with $\epsilon \in \mathcal{E}$. To achieve this, we first normalize the text by removing extra spaces, special characters, and sensitive private data. Then, we use GPT-4 to extract entities using the prompting approach and store them in a structured data format for all CVs. The prompts and outcomes were iteratively refined with the support of HR recruiters, who evaluated these extractions.

8.2.1 Embedding Generation

To create vector representations of all features, we employed OpenAI's `text-embedding-3-large` model [OpenAI, 2024], representing the state-of-the-art embedding technology. This model generates a vector $\mathbf{v}_{\epsilon,i} \in \mathbb{R}^{768}$ for each textual feature belonging to category ϵ for candidate i, allowing us to capture rich semantic information. In this real-world scenario, the number of entities varies across candidates. For instance, Candidate A might possess five soft skills, while Candidate B has only three. To accommodate this variability, we represent each CV for candidate i (CV_i) as a *set of sets* of the five entity categories $\mathcal{E}$, where each entity category is encoded as the sets of its vector embeddings $E_{\epsilon,i} = \{\mathbf{v}_{\epsilon,i}\}$. Formally, $\mathrm{CV}_i = \{E_{\epsilon,i} \mid \epsilon \in \mathcal{E}\}$. It is worth mentioning that the vector embeddings allow us to overcome the issues of synonyms and CVs written in different languages since such words are closer in the embedding space. In total, this yields $39,752$ unique embeddings.

8.2.2 Similarity Measure and Heterogeneous Graph

Our approach is predicated on assuming that candidates with comparable CVs are likely similarly suited for the same vacancies. To actualize this concept, we aimed to design a robust similarity measure, ultimately constructing a graph of candidates as in the previous chapter. Given that each CV is represented by a set of embeddings for all entities with varying cardinalities, we employed an approximate nearest neighbor (ANN) algorithm [Douze, 2024] to identify the k-nearest neighbors (with $k = 10$) for each embedding $\mathbf{v}_{\epsilon,i}$ across all CVs. See Table 8.2 for an example of retrieved neighbors, showcasing the efficacy of this method. Next, we define the kNN vectors retrieved for one embedding $\mathbf{v}_{\epsilon,i}$ as $\text{kNN}(\mathbf{v}_{\epsilon,i})$. Inspired by the Jaccard Index J for sets [Z. Wang, 2023], we compute the overlap between two candidates i and j for a given entity ϵ as

Entity	**Nearest Neighbors**
Soft Skills: *communication*	communications, comunication, communications and relations, communicating, communication and writing
Hard Skills: *python*	python programming, python), coding (python, programming languages (python, python (numpy
Industry Sector: *startups*	start ups, start up companies, technology startups, innovative start ups, technology startup
Education: *management diploma*	master in management, marketing management diploma, master degree in management, master in general management, professional diploma in management
Languages: *english*	english (good), english (medium), english (school), english (native), english language

Table 8.2: Example of Top 5-NN based on a given query for all entities.

$$J_\epsilon(i,j) = \frac{\sum_{\mathbf{v}_{\epsilon,i},\mathbf{v}_{\epsilon,j}} [\![\text{kNN}(\mathbf{v}_{\epsilon,i}) \subseteq \text{kNN}(\mathbf{v}_{\epsilon,j})]\!]}{|E_{\epsilon,i}| + |E_{\epsilon,j}|}, \tag{8.1}$$

where $[\![\cdot]\!]$ is the indicator function. In other words, $J_\epsilon(i,j)$ is the count of all embeddings that share at least one neighbor, normalized by the total amount of the embeddings. The similarity is then computed as:

$$\text{sim}_\epsilon(i,j) = \max(1 - e^{-\lambda J_\epsilon(i,j)} - \theta, 0) \tag{8.2}$$

so that $\text{sim}_\epsilon(i,j) \in [0, 1-\theta]$, λ is a scaling factor and θ is a tunable threshold to discard smaller values. Running this algorithm for all entities, we obtained a *weighted heterogeneous graph* $\mathcal{H}$ of candidates connected by different types of weighted edges according to their similarity. We set $\lambda = 2$ and $\theta = 0.2$, such that we obtained a graph with $\approx 10^8$ edges grouped as in Table 8.3. Computing all pairwise similarities requires $\mathcal{O}(C^2) \approx 3 \times 10^8$ evaluations, but then the computation of J has linear complexity and can be computed efficiently—given the pre-computed

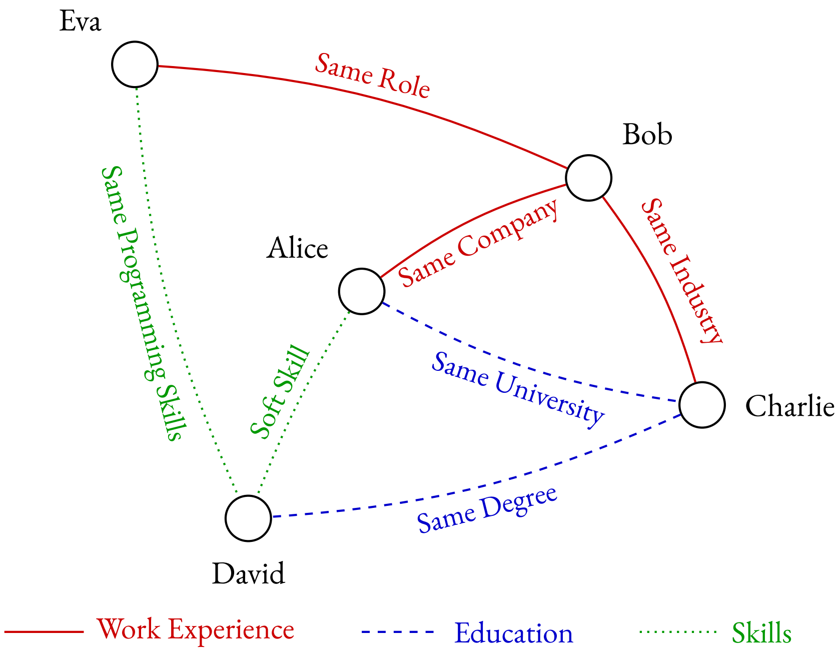


Figure 8.1: Simplified example of Heterogeneous Graph of Candidate Profiles. Each candidate is a node and is connected with other nodes if in their CV there is an overlapping embedding (its kNN to be precise) which encodes the feature labels (e.g., "Same Degree" could be "Ph.D.", that belongs to the Education entity type). In this picture, only 3 types of entities are represented with vague feature labels for illustration purposes.

embeddings, all similarities can be calculated in a matter of minutes on consumer hardware. An explanatory illustration of such a simplified graph is given in Fig. 8.1.

8.2.3 Model Architecture

In our graph-based approach, we employed Heterogeneous Graph Convolution as the core component of our model architecture [Xiao Wang, 2023]. Within this framework, we implemented two popular graph convolution methods: Graph Convolutional Networks (GCN) [Kipf, 2017] and Relational Graph Convolutional Networks (RGCN) [Schlichtkrull, 2018]. GCN provides a powerful mechanism for aggregating information from local graph neighborhoods. In this heterogeneous formulation, each edge type is treated individually, so the message passing happens on all five induced subgraphs. Finally, each hidden representation for each type is summed. RGCN extends this capability by explicitly modeling different types of edges in the graph but, contrary to GCN, does not consider edge weights.

Category	#Edges
Soft Skills	1,247,845
Hard Skills	162,605
Industry Sector	478,838
Education	287,724
Languages	7,723,929
Total	9,900,941

Table 8.3: Amount of heterogeneous edges with $\lambda = 2$ and $\theta = 0.2$. Language edges are the most prevalent, being the majority of candidates fluent in Italian.

Learning Framework

The final goal would be to predict the HR labels y so that new vacancies and candidates can be added to the graph in a future stage. We treated these selections in a *multi-task* learning approach to exploit their shared structure and learn relationships among candidates and vacancies. Another viable solution would be to train a classifier by first merging all selections (as in the previous chapter, where job opening attributes were neglected), train a model on the subgraphs induced by each selection, or treat the selection as a distinct node type linked to pertinent candidates. These options are left open to future works. Additionally, given the nature of our recruitment outcome data, which consists of four ordered classes, we explored two distinct problem formulations. First, we approached the task as *ordinal regression*, recognizing the inherent order and progressive nature of the recruitment stages. Alternatively, we framed the problem as a *multi-label classification* task, where each candidate could be assigned one or more labels corresponding to the stages they reached, neglecting its ordinal nature. This approach offers flexibility when a candidate might skip certain stages, or the recruitment process does not strictly follow a linear progression.

8.3 Experimental Setup

The graph $\mathcal{H}$ entails all $C = 5,461$ candidates, whose features are the 18 numerical questionnaire's traits; its edges are given by their CVs similarity according to the five entities embeddings, each one weighted by its similarity score. Every candidate has a categorical label belonging to at least one of the $S = 39$ job selections. Dealing with a single graph with possibly new unlabeled nodes, we fall into the *transductive* learning case. For training and validation, we split the nodes evenly for all tasks with $[80\%, 20\%]$ splits as train/test sets. Due to class imbalances, we made sure the splits were properly stratified. Rare labels $y = 4$ were also randomly split with the same ratio. The models have been implemented with the `PyTorch Geometric` library [Fey, 2019]. We ran each trial for 300 epochs with the Adam optimizer. The hyperparameters were chosen over the values in Table 8.4 with the `Optuna` library [Akiba, 2019] for 100 trials.

Hyperparameters	Range/Values
Hidden Units	$\{16, 32, 48, 64\}$
#Deep Layers	$\{1, 2, 3, 4, 5\}$
Learning Rate	10^{-4} to 10^{-1}
Activation Function	LeakyRelu, Elu, Tanh, Sigmoid

Table 8.4: Hyperparameters grid chosen to validate the models on the heterogeneous graph.

8.4 Results

Model	Task	Acc.	MAE	RMSE	F1	AUC
RGCN	Ordinal	25.2	0.532	0.924	0.729	0.566
	Multi-label	20.1	1.07	1.58	0.615	0.503
GCN	Ordinal	27.4	0.565	0.900	0.684	0.539
	Multi-label	30.1	0.662	1.29	0.796	0.606

Table 8.5: Experiment Results for RGCN and GCN with Ordinal Regression and Multi-label Learning

The results in Table 8.5 demonstrate the performance of the RGCN and GCN models under two different learning scenarios: ordinal regression and multi-label learning. Due to classes' distribution, we report the *balanced* accuracy and *weighted* F1-score. We also grouped classes $(0, 1)$ and $(2, 3)$ to compute the AUC, as for a binary classification setting.

The GCN model generally outperforms the RGCN model in both learning types. Specifically, for the ordinal regression task, the GCN model achieved a higher accuracy compared to the RGCN. The GCN model also demonstrated a lower RMSE and a competitive F1 score. Despite these gains, the RGCN model exhibited a slightly better binary AUC, indicating a marginally better binary classification capability. In the multi-label learning scenario, the GCN model again showed superior performance in terms of accuracy, which is significantly higher than the RGCN. The GCN's F1 weighted score and AUC highlight its robustness in handling multiple labels, which is crucial for such tasks requiring simultaneous prediction of several categories.

While these results are not overwhelming in absolute terms, they are nonetheless promising, and the balanced accuracy is above random guess (25%). Thus, such a model could be used to give hints to recruiters. Additionally, when considering the challenges posed by real-world, scarce, noisy, and imbalanced data in this complex learning setting. These findings not only demonstrate the feasibility of the proposed approach but also open up numerous avenues for future research directions, suggesting that further refinements could yield significant improvements for recruitment decision support systems.

8.5 Final Remarks

This study explored the integration of GNNs and LLMs to support recruiters in personnel selections. By leveraging the strengths of GNNs in capturing relational data and the advanced text-understanding capabilities of LLMs, we developed a pipeline to process this type of HR data. The experimental results demonstrated that our approach can be applied in this real-world scenario. Finally, it showcases the potential of combining graph-based and language-based models for complex classification tasks, offering a primal solution for CV analysis.

The primary objective of this study was to establish a robust pipeline capable of handling the complexities of real-world HR data. While this chapter explores and evaluates several architectural choices, learning strategies, and models, it is acknowledged that the vast landscape of possible approaches leaves ample room for future research and improvements.

In the next chapter, we leave the homogeneous graph in favor of *inductive* graph classification for the CJM.

9

Inductive Graph Classification: A New Paradigm for Candidate-Job Matching

The landscape of talent acquisition is undergoing a profound transformation, driven by the need for more efficient and accurate methods of matching candidates to job vacancies (CJM). While previous chapters have explored the application of GNNs to a global graph representing the entire candidate-job ecosystem, this approach, though promising, has revealed certain limitations in capturing the nuanced interplay between individual candidates and specific job roles, particularly with limited and real-world data.

The transition from traditional CJM methods to our graph approach required a thoughtful data translation and connection process. Our goal was to create a flexible graph structure that could capture the nuanced relationships between candidates and Job Descriptions (JD) while allowing for new, unseen data generalization. This chapter marks a significant departure from the previous global graph paradigm, introducing a novel perspective that treats each candidate-job pair as a distinct, purpose-built graph. This additionally frames the CJM problem into a graph classification task in an inductive learning setting, enabling our system to make informed predictions about the suitability of candidates for job positions that were not present in the training data.

By treating each candidate-job pair as an individual graph, we gain the ability to model the matching process with remarkable granularity. This approach allows us to capture subtle compatibilities that might be overlooked in a more generalized model. For instance, it can account for how a candidate's unique combination of skills might align with a job's multifaceted requirements or how a candidate's past experiences might translate into potential in a new role. Moreover, this graph classification approach offers significant advantages in terms of scalability and adaptability—our model can seamlessly accommodate new candidates and job descriptions without the need for retraining the entire system.

In the following sections, we present the methodology of constructing these purpose-built CJM graphs, exploring how various candidate and job attributes can be effectively encoded in this format. We will then train GNN architectures on these graphs, aiming to learn and infer complex matching patterns. Through a series of experiments and case studies, we will demonstrate how this approach can benefit the CJM.

9.1 Dataset Feature

The evaluated dataset expands upon the previous work described in Section 8.2. It now includes $S = 62$ job selections along with corresponding JDs, which are structured and homogeneus documents. On the other hand, there are $C = 8,360$ distinct candidates, all with their own CV. These are normally stored as PDFs or docx, and while text extraction from standard document formats is relatively straightforward, PDFs pose particular difficulties due to their complex structure—such CVs often contain a variety of elements such as images, graphs, and diverse fonts and styles, which can confound Optical Character Recognition (OCR) systems. Moreover, the inherently unstructured nature of CVs further complicates the extraction process, as there is no standardized format across applicants (with the exception of the Europass) and conversely, CVs tends to be quite creative. This is also the reason why candidates are often required to manually input their information into online application systems, despite they had already submitted a CV. The heterogeneity of CV formats and content thus necessitates sophisticated text extraction techniques to effectively parse and analyze these documents, and therefore there are commercial solutions available. We resorted to an open-source library, `PyPDF`, that was capable to extract text for all but 362 of them that were therefore discarded. Among these, $3,895$ have also completed the A+ Questionnaire, and their 18 resultant traits have been normalized and used as feature. For the $4,465$ candidates lacking the questionnaire, their traits have been set to zero. Furthermore, each JD specifies minimal requirements for some of the A+ Questionnaire traits, as determined by the HR recruiters based on the job role. Several candidates have applied to various similar positions, or were reevaluated by the recruiters for other fitting roles, resulting in a total of $N = 9,532$ unique candidate-application pairings to be translated into graphs. The selection processes attracted $159_{\pm 103}$ applicants, with figures varying from a low of 26 to a high of 648. These serve as the foundation for constructing CJM graphs, each of which is assigned an ordinal label y reflecting the selection stages: `Rejected` $\rightarrow$ 0, `Screened/Interviewed` $\rightarrow$ 1, `Shortlisted` $\rightarrow$ 2, `Hired` $\rightarrow$ 3. Typically, a dozen candidates get interviewed, only a few get shortlisted (in some cases, this step may be skipped altogether), and at least one candidate is ultimately hired. All this renders the dataset significantly imbalanced, with most of the applicants being rejected. Its statistics are presented in Table 9.1.

Label y	**Total**	**Grouped by S**			
		Mean	Min	Max	Std
0	95.10	94.50	69.23	99.22	4.77
1	3.40	4.32	0.39	30.77	4.75
2	0.84	1.65	0.30	4.17	0.97
3	0.66	0.94	0.27	3.70	0.74

Table 9.1: Overall and grouped candidate label distribution and statistics (in %) for $S = 62$ selected job openings. The 'Total' column represents the distribution across all job openings, while the 'Grouped by S' columns show statistics for the distribution within each selected job opening.

Feature Selection

Feature selection is a cornerstone of candidate assessment, as it provides a structured framework to evaluate attributes that directly impact their potential to succeed in a role. From an HRM perspective, the selected features encapsulate both tangible and intangible qualities critical to workplace performance. These features, identified and validated by HR experts, reflect years of professional experience and research into effective recruitment practices.

Soft Skills, encompassing interpersonal and behavioral competencies, are indispensable to fostering effective collaboration, adaptability to change, leadership potential, and conflict resolution in dynamic team environments. These skills often serve as the differentiating factor in roles requiring high levels of emotional intelligence and cultural fit. *Hard Skills*, on the other hand, represent the technical abilities and language proficiency necessary for candidates to perform specific job tasks with efficiency and precision. These skills are particularly critical in roles involving specialized tools, programming, or multilingual communication, as they ensure immediate job readiness.

Education credentials, including academic qualifications and certifications, provide a measure of a candidate's foundational knowledge and commitment to professional growth. They serve as benchmarks for assessing whether a candidate meets the baseline requirements of a role. Similarly, the *Field of Education* offers insight into the candidate's specific area of study or specialization, helping recruiters align their expertise with the technical and strategic needs of the organization.

The *Industry Sector* feature contextualizes a candidate's experience within specific business domains or market segments. This is critical for roles requiring an understanding of sector-specific challenges, regulatory frameworks, or operational nuances. Finally, the *Role* feature highlights the job titles and professional positions previously held by a candidate, offering a window into their career trajectory, level of responsibility, and potential for leadership. It also helps assess whether their prior experience aligns with the scope and expectations of the prospective role.

Collectively, these features, as identified by HR experts, provide a comprehensive and multidimensional view of a candidate's profile. They enable recruiters to make informed decisions by matching candidates not only to the technical requirements of a job but also to the organizational culture, strategic goals, and team dynamics. This holistic approach ensures that both the employer and the candidate benefit from an optimal fit, fostering long-term success.

Entity Extraction

Analogously to Section 8.2, GPT-4 extracted via prompting six entities from each CV and JD. Thus, $\mathcal{E} = \{$*Soft Skills*, *Hard Skills*, *Education*, *Field of Education*, *Industry Sector*, *Role*$\}$. *Languages* have been merged into *Hard Skills*, while *Field of Education* is a focus on the qualification major. Refer to Table 9.2 for an overview and examples.

We denote one of these entities with $\epsilon \in \mathcal{E}$, and we feed their texts into `text-embedding-3-large` model [OpenAI, 2024] to translate them into numerical embeddings $\mathbf{v}_{\epsilon,i} \in \mathbb{R}^{768}$. Each CV and JD includes multiple embeddings per entity, resulting in a *set of sets* across the six entity categories $\mathcal{E}$, that is $\{\{\mathbf{v}_{\epsilon,\{i,j\}} : \epsilon \in \mathcal{E}\}$ for all i indexing CVs or j for JDs. Missing entities are imputed with the null vector $\mathbf{0} \in \mathbb{R}^{768}$.

Entities $\mathcal{E}$	**#Embeddings**	**Examples**
*Soft Skill*s	6,776	Campaign development, Desire to learn, Mentoring start-ups, Problem solving mindset
Hard Skills	16,406	.NET framework, CAD, Credit analysis, Data management, English (B2)
Education	6,960	Bachelor's degree in Tourism, Diploma in Accounting, Erasmus course, SAP certificate
Field of Education	2,780	3D design, Science, Economics, Marketing, Neuroscience
Sector	7,628	Manufacturing, Law enforcement, Pharma, Restaurant management
Role	15,749	1^{st} officer, CTO, Medical representative, Researcher, Technician

Table 9.2: Overview of the entity embeddings, with their amount and some examples.

9.2 Constructing Inductive Graphs for Candidate-Job Matching

The core of our proposed method lies in the construction of purpose-built graphs that capture the intricate relationships between candidates and JD. This graph construction process is crucial as it transforms the raw data extracted from CVs and job postings into a structured representation that can be effectively processed by GNNs. Our method creates individual graphs for each CJM pair, allowing for a granular analysis of the compatibility between the two. Furthermore, this inductive approach enables our model to generalize to new, unseen candidates and job descriptions, making it highly adaptable to real-world HR scenarios. In the following subsections, we detail the step-by-step process of constructing these graphs, starting from the primary nodes and expanding to include various entity relationships and edge connections.

Adding Nodes The process of constructing each CJM graph starts with two primary nodes: a *Candidate node* connected to a *Job Description node*. These nodes serve as the anchors of our graph, representing the core elements we aim to match and use for label predictions. Initial assessments indicated that adding a self-loop to these nodes enhanced performance and optimized message passing to stacked convolutional layers. From these central points, we extend symmetrical connections to each of the entities $\mathcal{E}$, creating a star-like structure on each side of

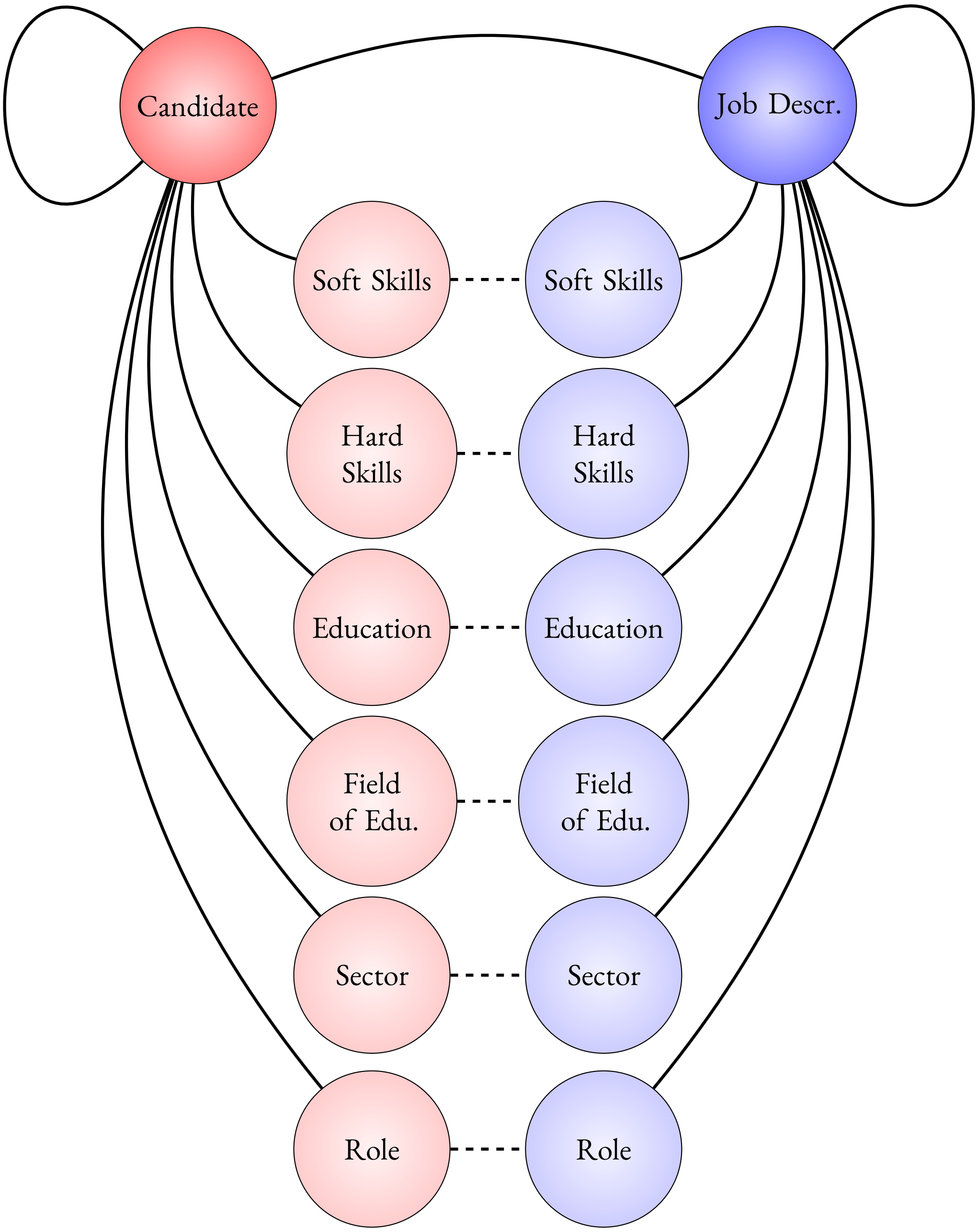


Figure 9.1: Bipartite graph representation of the Candidate-Job Matching (CJM) model. Candidate and Job Description have a self-loop and are connected with each other plus their respective entity nodes. In turn, these are connected only if there is a match based on the embeddings (dashed lines).

the graph. The resulting bipartite graph representation has thus 14 nodes and at least 15 edges, as shown in Fig. 9.1. Certainly, exploring additional design principles is a valid path, given that the process of converting data into graphs is complex [Y. Liu, 2022; K. Zhou, 2022].

Connecting Edges The true power of our approach lies in how we connect these two halves of the graph. We implemented a system of weighted edges, represented by dashed lines in our visualization, that indicate the strength of the match between the corresponding entities. Similarly to Eq. (8.1), we introduce a similarity function that computes the normalized count of common embedding within the $k = 10$ closest neighbors in the embedding space.

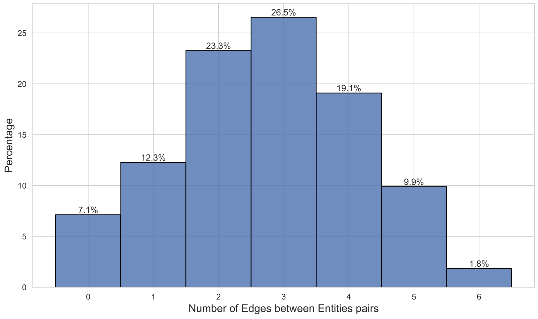
$$\mathrm{sim}_\epsilon(i,j) = \left(\frac{\sum_{\mathbf{v}_{\epsilon,i},\mathbf{v}_{\epsilon,j}} [\![\mathrm{kNN}(\mathbf{v}_{\epsilon,i}) \subseteq \mathrm{kNN}(\mathbf{v}_{\epsilon,j})]\!]}{|E_{\epsilon,i}| + |E_{\epsilon,j}|} \right)^{1/p} \in [0,1]\,, \tag{9.1}$$

where i denotes embeddings related to the Candidate node's entities, and j is the analogous for the Job Description. The exponent $1/p$ with $p = 4$ was introduced to enhance the weights of sets that overlap more. This aims to capture cases in which, for instance, a candidate's coding abilities may closely align with the technical skills needed for a position, and are thus represented by a more weighted edge.

In summary, the dataset comprises $N = 9,532$ graphs, collectively containing $|V| = 133,448$ nodes (each graph has 14 nodes by design) and $|E| = 169,213$ edges. On average, each graph contains $17.75_{\pm 1.43}$ edges, ranging from a minimum of 15 to a maximum of 21 edges per graph.

Insights from the Graph Structures Now, let us investigate the edges that connect the bipartite graph further. There are $8,853$ graphs with at least one edge between candidate and job entities, which is rich semantic information embedded as a graph. The overall distribution of the amount of edges between entities $\mathcal{E}$ is reported in Fig. 9.2. The edge count statistics presented in Table 9.2(b) reveal a noteworthy pattern: there is a statistical difference in the average number of edges across different labels. Specifically, nodes with labels 0 and 1 exhibit lower mean edge counts (2.74 and 3.03, respectively) compared to those with labels 2 and 3 (3.16 and 3.13, respectively). This variation in edge density among different label categories presents a valuable structural feature that GNNs are particularly well-suited to exploit.

It is now compelling to observe the overall distribution of entities $\mathcal{E}$ and their respective edges among the labels. To visualize these patters, we normalize their count over all graphs, and highlight their difference w.r.t. the mean count in Fig. 9.3. The heatmap of edges ratios reveals intriguing patterns across different labels y and entity categories $\mathcal{E}$. Notably, the *Hard Skill* category consistently exhibits the highest edge ratios across all labels, suggesting its central



Label y	**Mean**	**Std. Dev.**
0	2.74	1.43
1	3.03	1.44
2	3.16	1.55
3	3.13	1.36

(a) Distribution of graph edges.

(b) Edge count statistics grouped by label.

Figure 9.2: Analysis of graph edges between nodes matching entities in the CVs and JDs: (a) distribution of edge counts, and (b) statistical summary grouped by graph label.

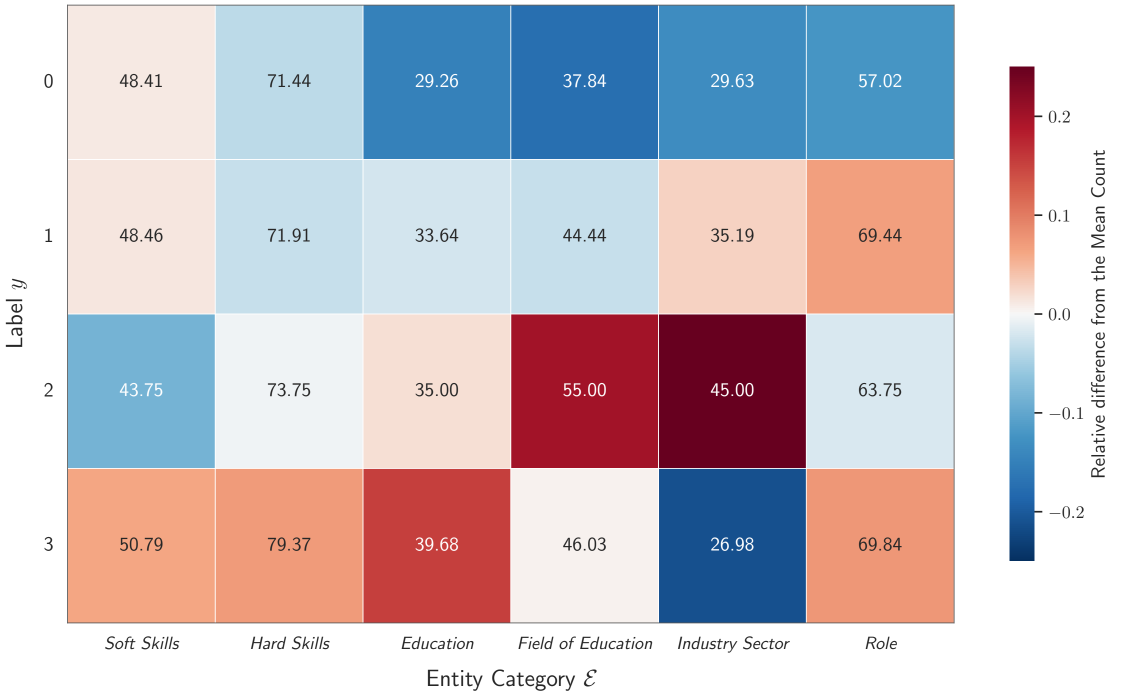


Figure 9.3: Distribution of edge ratios across labels and entity categories $\mathcal{E}$. The heatmap displays the relative difference from the mean (normalized) count for each category. Colors range from dark blue (significantly below mean) through white (at mean) to dark red (significantly above mean). Numerical values represent the original edge ratios (% of cases whether there is a link between those entities in the bipartite CJM graphs).

role in connecting candidates to job vacancies. Interestingly, the `Hired` label ($y = 3$) shows the highest ratio for this category at 79.37%, indicating that hard skills match are particularly crucial for successful hires. The *Role* category also demonstrates high edge ratios, especially for shortlisted ($y = 2$) and hired candidates, emphasizing its importance in the later stages of the hiring process. In contrast, the *Education* and `Industry Sector` categories generally show lower edge ratios, with some variation across labels. `Industry Sector` and *Field of Education* are more prevalent for $y = 2$ than the $y = 3$ as desired. This could mean that those qualities are biasing factors for being shortlisted, suggesting that these two labels could also be merged into one. Finally, the *Soft Skill* category shows relatively consistent ratios across all labels, implying its consistent importance throughout the hiring process.

Adding Features The last step of graphs' construction involves incorporating node features $\mathbf{X}$ into these CJM graphs, such that a GNN can identify patterns beyond just the graph topology. The natural choice is to use the Questionnaire results and traits minimum requirements for the Candidate and JD node, respectively. These are denoted as $\mathbf{x}_{q,\{i,j\}} \in \mathbb{R}^{18}$. Regarding the entity nodes, we opted to process their set of embeddings with set aggregation techniques similar to graph pooling:

$$\mathbf{x}_{\epsilon,\{i,j\}} = \left[\text{mean}\left(\{\{\mathbf{v}_{\epsilon,\{i,j\}}\}\}\right), \text{sum}\left(\{\{\mathbf{v}_{\epsilon,\{i,j\}}\}\}\right), \text{max}\left(\{\{\mathbf{v}_{\epsilon,\{i,j\}}\}\}\right)\right] \tag{9.2}$$

One could argue that their semantic meaning is already captured within the graph structure; however, their vector representations can be integrated with one another and with questionnaire data using GNNs. These values come from hidden LLM representations, and such formulation in a GNN can be seen as DeepSet [Zaheer, 2017] instance. An alternative strategy might involve utilizing a GTM as discussed in Chapter 4, which would focus on learning probabilistic representations of these embeddings. This method has been deferred to future work because of time limitations. Besides, we experimented with a one-hot encoding of the entities; however, this yielded inferior results.

9.3 Dataset Preparation and Splitting

For our experiments, we implemented a comprehensive data-splitting strategy to ensure robust evaluation and statistical significance. Our dataset comprises $S = 62$ distinct recruitment processes, each associated with its corresponding set of CJM graphs. To facilitate inductive learning for graph classification, we partitioned these selections S into training (70%), validation

(20%), and test (10%) sets. Unlike traditional k-fold cross-validation, we employed a random selection method that groups CJM graphs according to their related selection processes. This approach maintains the integrity of each recruitment scenario by ensuring all graphs pertaining to a specific process are consistently assigned to the same subset, emulating realistic deployment conditions. We repeated this splitting process five times, each iteration involving a new random selection while maintaining the 70-20-10 ratio of S. This repetition yields five distinct partitioned datasets, enhancing the reliability and reproducibility of our results, as well as mitigating potential biases from a single, arbitrary division.

Furthermore, to address the inherent imbalance in our dataset, we created a supplementary version that more closely aligns with Amajor's business process. We have seen that approximately half of the candidates did not complete the A+ Questionnaire and were consequently neglected in the recruitment process. This situation introduced further bias to a dataset whose labels are already highly unbalanced. To tackle this issue, we also retain only 10% of candidates without questionnaire data, randomly selected. This version was split five times using the same methodology described above. While this approach may discard potentially valuable CV information and candidate data, it is justified by the importance of ensuring a good Person-Environment fit in the recruitment process, and Amajor considers their questionnaire an essential tool for assessing this fit. Therefore, with this additional dataset version, we aim to reflect the real-world recruitment scenario more accurately, albeit at the cost of reduced data volume.

9.4 Loss Function and Learning Task

In our HR framework, the learning task is based on the outcomes of the recruitment process according to the labels $y = \{0, 1, 2, 3\}$. Given the inherent order in these labels, we explored two distinct approaches to formulate our learning task: ordinal and binary (graph) classification.

Ordinal Classification

Recognizing the progressive nature of the recruitment stages, we implemented ordinal classification using a Consistent Rank Logits (CORAL) layer [W. Cao, 2020]. The CORAL approach models the ordinal nature of our labels by learning a series of binary classifiers for the output layer, each corresponding to a threshold between consecutive ranks. The loss function for ordinal classification can be expressed as:

$$\mathcal{L}_{\text{ordinal}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K-1} \left[y_g^k \log(\sigma(f_k(\mathbf{x}_i))) + (1 - y_g^k) \log(1 - \sigma(f_k(\mathbf{x}_i))) \right] \tag{9.3}$$

where N is the number of samples, $K = 4$ is the number of classes, y_g^k is the binary indicator for the g-th graph sample and k-th threshold, $f_k(\mathbf{x}_g)$ is the model's output for the k-th threshold, and $\sigma(\cdot)$ is the sigmoid function.

Binary Classification

In addition to ordinal classification, we also considered a binary classification setting. This method groups the valued labels $y^* = \{1, 2, 3\}$ together, effectively creating a binary distinction between rejected candidates (0) and those who progressed in the recruitment process (y^*). The loss function for binary classification is given by:

$$\mathcal{L}_{\text{binary}} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_g \log(\tilde{y}_g) + (1 - y_g) \log(1 - \tilde{y}_g) \right] \tag{9.4}$$

where y_g is the binary label (0 or $1 = y^*$) and $\tilde{y}_g$ is the model's prediction for the g-th graph sample. To mitigate the imbalance in label distributions, multiplying a weight $w = 10$ to the labels y^* in both losses has led to improved performances.

From a recruitment and business standpoint, our dual approach to the learning task offers valuable insights into the hiring process. The ordinal classification aligns closely with the sequential nature of many recruitment pipelines, where candidates progress through distinct stages. Successfully learning this task would help HR professionals understand the factors that contribute to a candidate's advancement at each stage, potentially optimizing the screening and interview processes. Conversely, the binary classification provides a more streamlined view, focusing on identifying candidates likely to progress beyond the initial rejection stage. This approach can be particularly useful for large-scale recruitment purposes or when rapid initial screening is necessary, allowing recruiters to quickly identify a pool of promising candidates for further in-depth consideration.

9.5 Model Architecture

The GNN employed in this study comprises several key components. Initially, a pre-aggregation step is applied to the node features, projecting the $\mathbf{x}_{\epsilon,\{i,j\}}$ of Eq. (9.4) and $\mathbf{x}_{q,\{i,j\}}$ in a common hidden dimension. Specifically, seven Linear layers are employed: one for the primary nodes and six for the entity nodes, effectively capturing the unique characteristics of each node type.

The core of our model consists of multiple graph convolutional (GC) layers. We experiment with five different types of GC operators:

- **Graph Convolutional Network (GCN)** [Kipf, 2017]: The standard graph convolution operator that aggregates neighborhood information through normalized adjacency matrices.
- **Multiplicative Integration GNN (MIGNN)** (v1, see Section 5.2): An adaptation that employs multiplicative integration of node features, enabling the model to capture higher-order interactions between nodes' attributes.
- **Graph Isomorphism Network (GIN)** [Xu, 2019]: A theoretically powerful architecture that can distinguish different graph structures through injective neighborhood aggregation.
- **Graph Attention Network (GAT)** [Velickovic, 2018]: Incorporates attention mechanisms to weigh the importance of different neighbors dynamically.
- **GraphConv** [Defferrard, 2016]: A general formulation that combines both local and global graph properties through separate weight matrices.

These layers are followed by *Graph Normalization* [Y. Chen, 2022] and non-linear activation (*LeakyReLU*). A tunable feature is an option for *Jumping Knowledge* connections [Xu, 2018], allowing for adaptive, structure-aware representations. The model culminates in a *deep readout* phase, where global graph pooling operations (sum, mean, and max) are applied, followed by fully connected layers. Key hyperparameters include the number of layers, hidden channel dimensions, dropout rate for regularization, and the number of deep readout layers.

9.6 Experimental Setup

The experiments were conducted using `Python 3.11` and `PyTorch 2.4`. For optimization, we employed the `AdamW` optimizer [Loshchilov, 2019] with a learning rate selected through hyperparameter tuning. Hyperparameter optimization was performed using the `Optuna` library [Akiba, 2019]. We conducted 100 trials, each running for 300 epochs, with early stopping implemented using a patience of 50 epochs to prevent overfitting. The objective of the optimizer was to minimize the loss on the validation set. The hyperparameters grid is reported in Table 9.3, while all the other unspecified choices are left as their default values.

Hyperparameters	**Range/Values**
Hidden Channels	32 to 256 (step 32)
Number of GC Layers	$\{1, 2, 3, 4, 5\}$
Learning Rate	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
Dropout Rate	$\{0.0, 0.5\}$
Jumping Knowledge (JK)	{True, False}
Deep Readout Layers	$\{0, 1, 2, 3\}$

Table 9.3: Hyperparameters grid for the GNN model in candidate-job vacancy matching.

To rigorously test the efficacy of our graph-based approach, we ran the experimental evaluations against a traditional MLP. This MLP is designed to operate with the same experimental framework as our GNN model: its architecture begins with the same pre-aggregation step used in the GNN, processing the node features of the CJM graphs. Following this, instead of applying graph convolutions, the MLP utilizes the deep readout phase with a tunable number of layers ranging from 1 to 5. In this fashion, we aimed to isolate the impact of the graph-based learning approach and validate the hypothesized benefits of leveraging graph structural information in the context of CJM.

9.7 Results

In this section, we present and analyze the experimental results of our study on GNN for CJM. We compare the performance of six models, MLP, GCN, MIGNN, GAT, GIN, and GraphConv, across the two dataset versions (Filtered and Complete) and two task types (Ordinal and Binary). Table 9.4 summarizes the train, validation, and test losses for each model configuration, while additional metrics for a more comprehensive evaluation of our models' performance in the test sets are presented in Table 9.5. The latter table compares the balanced accuracy, weighted F1 score, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). We chose to use balanced accuracy and weighted F1 score due to the unbalanced nature of our dataset, as these metrics provide a more reliable assessment of model performance when class distributions are skewed. MAE and RMSE are included to provide insight into the magnitude of prediction errors, particularly relevant for the ordinal task.

Dataset	Task	Model	Train Loss	Validation Loss	Test Loss
Filtered	Ordinal	MLP	323.7$_{\pm 36.2}$	115.2$_{\pm 15.0}$	67.5$_{\pm 10.3}$
		GCN	307.3$_{\pm 25.5}$	112.3$_{\pm 14.3}$	65.0$_{\pm 10.9}$
		MIGNN	310.5$_{\pm 20.8}$	108.9$_{\pm 15.3}$	63.4$_{\pm 9.6}$
		GIN	325.3$_{\pm 26.0}$	112.0$_{\pm 13.9}$	65.0$_{\pm 11.6}$
		GAT	321.2$_{\pm 19.6}$	109.7$_{\pm 16.0}$	62.7$_{\pm 9.1}$
		GraphConv	316.6$_{\pm 34.7}$	110.8$_{\pm 14.8}$	65.0$_{\pm 10.4}$
	Binary	MLP	131.6$_{\pm 18.3}$	52.4$_{\pm 5.5}$	31.5$_{\pm 5.0}$
		GCN	109.7$_{\pm 11.0}$	50.3$_{\pm 5.4}$	30.5$_{\pm 4.1}$
		MIGNN	101.1$_{\pm 6.4}$	47.7$_{\pm 5.3}$	30.0$_{\pm 4.2}$
		GIN	111.9$_{\pm 9.3}$	50.2$_{\pm 6.7}$	29.8$_{\pm 4.3}$
		GAT	127.1$_{\pm 6.9}$	50.0$_{\pm 7.2}$	30.3$_{\pm 5.1}$
		GraphConv	102.4$_{\pm 7.4}$	49.0$_{\pm 7.5}$	29.7$_{\pm 5.5}$
Complete	Ordinal	MLP	389.3$_{\pm 49.0}$	112.2$_{\pm 14.2}$	65.9$_{\pm 8.2}$
		GCN	343.2$_{\pm 29.6}$	103.0$_{\pm 17.0}$	64.4$_{\pm 11.7}$
		MIGNN	402.4$_{\pm 19.3}$	107.5$_{\pm 16.2}$	60.3$_{\pm 11.2}$
		GIN	418.4$_{\pm 28.1}$	111.7$_{\pm 15.4}$	63.2$_{\pm 10.0}$
		GAT	430.0$_{\pm 57.2}$	109.1$_{\pm 15.6}$	62.1$_{\pm 8.8}$
		GraphConv	436.5$_{\pm 65.5}$	109.3$_{\pm 16.0}$	62.2$_{\pm 9.0}$
	Binary	MLP	180.5$_{\pm 26.0}$	50.4$_{\pm 5.8}$	30.4$_{\pm 4.5}$
		GCN	140.3$_{\pm 17.0}$	45.0$_{\pm 7.8}$	28.5$_{\pm 3.7}$
		MIGNN	146.5$_{\pm 13.2}$	48.3$_{\pm 6.8}$	29.1$_{\pm 4.0}$
		GIN	181.7$_{\pm 18.1}$	50.2$_{\pm 7.0}$	29.0$_{\pm 4.3}$
		GAT	190.2$_{\pm 9.8}$	49.7$_{\pm 6.8}$	29.7$_{\pm 3.0}$
		GraphConv	185.6$_{\pm 13.5}$	49.4$_{\pm 6.2}$	29.8$_{\pm 4.2}$

Table 9.4: Comparison of losses across different datasets and tasks for different models.

Dataset	Task	Model	B. Accuracy	W. F1	MAE	RMSE
Filtered	Ordinal	MLP	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		GCN	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		MIGNN	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		GIN	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		GAT	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		GraphConv	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
	Binary	MLP	$50.8_{\pm 2.3}$	$89.5_{\pm 6.1}$	$0.107_{\pm 0.105}$	$0.298_{\pm 0.134}$
		GCN	$61.8_{\pm 1.8}$	$80.5_{\pm 5.6}$	$0.268_{\pm 0.082}$	$0.511_{\pm 0.082}$
		MIGNN	$56.7_{\pm 3.8}$	$86.1_{\pm 2.5}$	$0.184_{\pm 0.045}$	$0.426_{\pm 0.050}$
		GIN	$58.6_{\pm 2.3}$	$85.5_{\pm 1.9}$	$0.194_{\pm 0.032}$	$0.439_{\pm 0.038}$
		GAT	$60.5_{\pm 3.9}$	$81.5_{\pm 7.3}$	$0.251_{\pm 0.103}$	$0.492_{\pm 0.096}$
		GraphConv	$59.1_{\pm 3.5}$	$83.5_{\pm 4.0}$	$0.225_{\pm 0.063}$	$0.469_{\pm 0.070}$
Complete	Ordinal	MLP	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		GCN	$25.1_{\pm 0.2}$	$92.8_{\pm 1.1}$	$0.072_{\pm 0.014}$	$0.353_{\pm 0.024}$
		MIGNN	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.347_{\pm 0.018}$
		GIN	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		GAT	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.348_{\pm 0.017}$
		GraphConv	$25.0_{\pm 0.0}$	$92.9_{\pm 0.8}$	$0.068_{\pm 0.007}$	$0.346_{\pm 0.016}$
	Binary	MLP	$55.0_{\pm 4.8}$	$85.0_{\pm 6.9}$	$0.191_{\pm 0.114}$	$0.412_{\pm 0.146}$
		GCN	$65.4_{\pm 3.6}$	$87.0_{\pm 1.8}$	$0.173_{\pm 0.029}$	$0.414_{\pm 0.036}$
		MIGNN	$62.2_{\pm 1.4}$	$86.7_{\pm 2.0}$	$0.176_{\pm 0.034}$	$0.418_{\pm 0.044}$
		GIN	$58.1_{\pm 4.2}$	$85.3_{\pm 4.9}$	$0.191_{\pm 0.088}$	$0.422_{\pm 0.116}$
		GAT	$56.9_{\pm 2.8}$	$87.3_{\pm 4.2}$	$0.159_{\pm 0.074}$	$0.387_{\pm 0.097}$
		GraphConv	$58.9_{\pm 3.7}$	$84.8_{\pm 2.2}$	$0.206_{\pm 0.035}$	$0.452_{\pm 0.038}$

Table 9.5: Comparison of models performance metrics (Balanced Accuracy (%), Weighted F1, MAE, RMSE) on the test set across different datasets and tasks. The values averaged over the test set across five runs have rounded standard deviations to the nearest decimal point.

Ordinal Task In this task, all models show nearly identical performance across all metrics, with a balanced accuracy of 25% for both Filtered and Complete datasets. This performance is equivalent to a prior classifier that always predicts the most frequent class ($y = 0$). The consistency across models suggests that the ordinal prediction task, as currently formulated, may be too challenging or the features insufficient for meaningful discrimination between ordered classes. Despite the low balanced accuracy, the weighted F1 scores are surprisingly high (around 92.9). This apparent contradiction can be explained by the class imbalance in our dataset. The high F1 score likely reflects good performance in the majority class, while the balanced accuracy reveals poor performance across all classes. The MAE of approximately 0.068 indicates that, on average, predictions are off by less than one class. Given that we have four ordinal classes, this suggests that when the models make errors, they tend to predict adjacent classes rather than making extreme misclassifications.

Binary Task This learning setting shows more promising results, with balanced accuracies consistently above 50%, indicating that all models perform better than a naive classifier. In the Filtered dataset, GCN achieves the highest balanced accuracy (61.8%), followed by GAT (60.5%) and GraphConv (59.1%), while GIN (58.6%) and MIGNN (56.7%) show less competitive performance. The MLP baseline, despite its simplicity, achieves 50.8%. For the Complete dataset, GCN maintains its lead with 65.4% balanced accuracy, followed by MIGNN (62.2%) and GIN (58.1%), while GAT and GraphConv show comparable performance around 57%. The MAE for the binary task ranges from 0.107 to 0.268, higher than the ordinal task as expected, since any misclassification results in an error of 1.

Furthermore, this scenario offers a clearer understanding of how the models address class imbalance: with a significant disparity of 945 samples in class 0 compared to just 47 in class 1, the models' effectiveness on the minority class can be compared and visualized in the normalized confusion matrices of Fig. 9.4. The MLP, while achieving a high accuracy of 92.80% of corrected predictions on the majority class in the filtered dataset, struggles significantly with the minority class, correctly classifying only 8.51% of them. In contrast, the GCN demonstrates a more balanced approach, correctly identifying 48.94% of the minority class at the cost of reduced majority class accuracy (74.39%). The MIGNN achieves a compromise, offering 84.55% accuracy on the majority class and 29.17% on the minority. Interestingly, the complete dataset improves minority class detection across all models. This suggests that additional data in training, although all belonging to class 0, benefits the learning process.

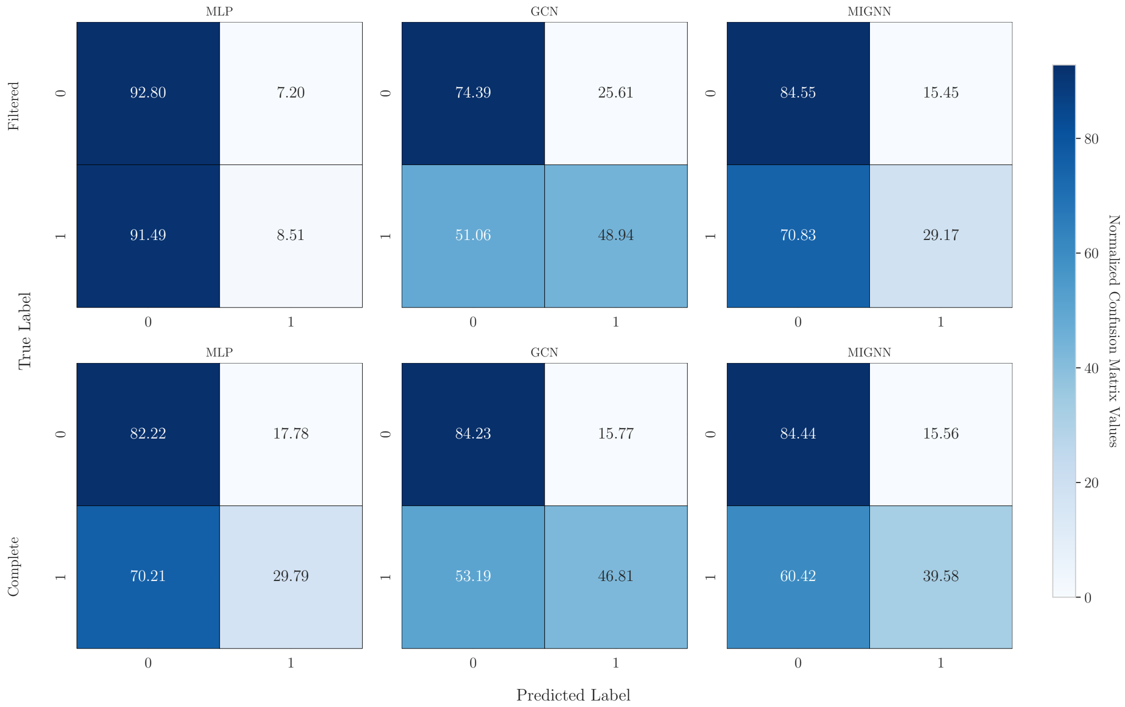


Figure 9.4: Comparison of confusion matrices for six different models in the CJM task. Each cell displays the normalized percentage (over true labels) averaged over the test splits. The class distribution is divided into 945 samples for class 0 and 47 in class 1. The color intensity indicates the proportion of samples in each true class that were predicted as each possible class, with darker blues representing higher proportions.

Graph Convolutions The consistently strong performance of graph-based models validates our hypothesis that graph structures can effectively capture meaningful relationships in HR data. While each architecture shows distinct strengths, GCN emerges as the most reliable across our experiments, particularly excelling in balanced accuracy. This suggests that its straightforward neighborhood aggregation mechanism is well-suited for our CJM graphs. The attention mechanism in GAT and the multiplicative interactions in MIGNN also prove beneficial, especially in handling class imbalance, while GIN and GraphConv offer secondary alternatives. The superior performance of these graph-based approaches over the MLP baseline highlights the value of incorporating structural information in the CJM task. This has practical implications for HR recruiters—out of hundreds of applications, they could prioritize their manual assessment on candidates classified as promising by these models, striking a reasonable balance between efficient evaluation of large applicant pools and identification of high-potential candidates.

9.7.1 Best Selected Hyperparameters

Table 9.6 presents the validated best-selected hyperparameters for the best-performing models across different datasets and tasks. It is immediately noticeable how the hyperparameters vary in all cases between the two dataset versions. The amount of GC is usually small, particularly for binary classification, suggesting that messages propagate well with just one step in these small CJM graphs. Also, the number of hidden channels varies widely across all models, from 32 to 192, reflecting the task-specific optimal model capacities. Regularization strategies and architectural choices differ across configurations, with varying dropout rates and Jumping Knowledge (JK) connections. Overall, the selected hyperparameters may hint at the fact that shallow and simpler architectures should be preferred and could prevent overfitting with such a relatively small dataset.

9.7.2 Limitations and Possible Improvements

Our research provides notable insights into using GNNs for CJM, but it is also crucial to acknowledge its limitations and seek ways to progress.

One of the most striking observations from our results is the poor performance of all models in the ordinal task. The fact that our architectures perform no better than a simple prior classifier is a clear indication that our current approach to ordinal classification in this context is suboptimal. The very nature of treating CJM as an ordinal classification problem may be flawed,

Dataset	Task	Model	Hidden Channels	#GC Layers	Dropout Rate	JK	#Deep Readout Layers	Learning Rate
Filtered	Ordinal	MLP	32	-	0.0	False	1	0.001
		GCN	128	5	0.5	False	3	0.01
		MIGNN	160	3	0.0	False	2	0.0001
	Binary	MLP	128	-	0.5	False	3	0.01
		GCN	64	3	0.0	True	3	0.0001
		MIGNN	128	1	0.0	True	0	0.0001
Complete	Ordinal	MLP	64	-	0.5	False	4	0.01
		GCN	160	2	0.5	True	2	0.0001
		MIGNN	192	2	0.0	True	2	0.01
	Binary	MLP	160	-	0.0	False	1	0.001
		GCN	128	1	0.0	False	0	0.001
		MIGNN	96	1	0.0	True	2	0.01

Table 9.6: Best selected hyperparameters for MLP, GCN, and MIGNN.

as the assumption of clear, ordered levels of suitability between candidates and job positions might be an oversimplification of a complex, multidimensional problem. To address this issue, we could consider reformulating the problem as a multi-class classification task, where each suitability level is treated as a separate class without an inherent order. This approach would allow the models to learn more nuanced relationships between features and outcomes without imposing an ordinal scale. Additionally, exploring regression-based approaches or even multi-task learning, where we predict multiple aspects of CJM or a matching preference, could equally support HR recruiters.

Another potential limitation lies in the features we are using to represent candidates and jobs. Important aspects of a candidate's suitability, such as soft skills, cultural fit, ambitions, or specific experiences that are not easily quantified, are definitely missing from our data only based on CVs and assessment questionnaires. To improve this, we could look into incorporating more detailed candidate and job information, like interview transcripts or data about the hiring company. Additional pertinent information, including age, gender, years of professional experience, salary, and geographical location, are also considered for the recruitment process. Nevertheless, these have been excluded from our analysis due to privacy and ethical concerns, thereby possibly hindering the effectiveness of the models. In general, more HR data needs to be collected to increase the pool of candidates and selections and consequently extend this analysis, yet this is a costly and time-consuming task.

The graph structure of our data presents another area for potential improvement. While the graph-based models showed benefits in the binary classification task, the graph construction relies on a heuristic method. We could explore different methods of graph construction, perhaps incorporating more domain knowledge to create edges that better represent meaningful relationships between candidates and jobs, or even let the model learn the optimal graph topology.

9.8 Final Remarks

This exploratory study shows promise in the capability of the latest and novel AI tools to support HR decisions. Our current approach has provided valuable insights into real HR data and how it can be processed. Our work makes several notable contributions, all paving the way for new advancements in this domain:

1. **Integration of LLM and GNN Technologies**: By leveraging Large Language Models for feature extraction and Graph Neural Networks for prediction, we demonstrated an innovative combination of the latest cutting-edge AI technologies in the HR domain.
2. **Novel Graph-Based Representation**: We introduced a new paradigm for representing candidate-job matching as a graph classification problem. Compared to other methods in the literature, this approach allows for a more nuanced capture of the complex relationships between candidates' attributes and job requirements.
3. **Comprehensive Experimental Framework**: Our study provides a rigorous comparison of different model architectures across various tasks and dataset versions, showcasing its versatility and documenting the strengths and limitations of each approach when applied to new data.
4. **Real-World Application**: We addressed the challenges of utilizing AI in a real-world HR setting by collaborating with HR recruiters and processing actual CVs. This resulted in a graph-based AI model that could effectively support recruiters in the screening and assessment of new candidates.
5. **Identification of Key Challenges**: Our results and discussions highlight important areas for future research and improvements in candidate-job matching, potentially guiding the direction of subsequent studies in this field.

Future investigations might delve into the utilization of the GTM approach for the aggregation of distinct entity sets as discussed in Chapter 4. Furthermore, the employment of GraphSHAP-IQ, as presented in Chapter 6, could be applied to these synthesized graphs to comprehensively study the interactions and interrelationships among the nodes.

While our current results show promise, they also reveal the complexity of HR data and the CJM problem. The performance gap between our models and ideal predictors underscores the need for continued research and development in this area, in particular for trustworthy and production-ready applications.

10
Conclusion

This dissertation has explored the application of Graph Neural Networks (GNNs) to the domain of Human Resource Management (HRM), with a particular focus on the challenging task of candidate-job matching (CJM). Through a series of theoretical contributions and applied studies, we have demonstrated the potential of graph-based approaches to enhance and innovate HR analytics and personnel selection. This work bridges the gap between cutting-edge machine learning techniques and real-world HR challenges, offering new perspectives and methodologies for addressing the complexities of modern recruitment.

Addressing the Research Questions

At the outset of this research, we posed several key questions to guide our investigation. Let us revisit these questions and summarize the insights gained through our research.

Research Question 1

How can real-world HR data, including candidate profiles, job descriptions, and assessment questionnaires, be effectively represented and processed using Deep Learning?

Our research has demonstrated that Deep Learning techniques, particularly GNNs, can effectively represent and process complex HR data. To achieve this, we developed novel methods to convert various types of HR data into graph structures:

- We transformed Likert-scale questionnaire data into graphs, preserving inherent relationships between respondents (Chapter 7).

- We created heterogeneous graphs capturing the multifaceted nature of candidate profiles, each characterized by the selection process's status (Chapter 8).

- We introduced a new paradigm for representing individual candidate-job pairs as purpose-built graphs for inductive learning (Chapter 9).

These graph-based representations proved to be more effective than traditional tabular approaches, hinting at the fact that the relational information captured by graphs provides valuable context. Moreover, throughout this investigation, we highlighted the challenges of collecting, cleaning, processing, and analyzing data from multiple sources in a business setting. Data are precious and should be fully understood to leverage Deep Learning models thoroughly—this particularly holds in the delicate, complex setting of resourcing.

Research Question 2

How can HR data be translated into graph-based structures? What additional insights and benefits would this approach provide?

Our investigation identified multiple techniques for the conversion of HR data into graph-based structures:

- For the questionnaire data, we developed a similarity measure based on response patterns to create edges between candidates (Chapter 7). This enables ML models to leverage candidates based on a more comprehensive and standardized assessment based on their estimated habits and behaviors.

- For CV data, we leveraged LLMs to extract meaningful features, which were then used to create nodes and edges in heterogeneous graphs (Chapter 8), allowing for a fine-grained comparison of CVs based on semantic entities.

- In our final approach (Chapter 9), we constructed purpose-built and interpretable graphs for each candidate-job pair, incorporating various entity types as nodes and using similarity measures to create edges, discovering patterns in the personnel selection.

The implementation of these graph-based structures facilitated the deployment of both standard and novel GNN architectures, which demonstrated superior performance compared to conventional machine learning techniques across a range of diverse HR analytics tasks.

Research Question 3

What are the most effective GNN architectures and training methodologies for learning such HR graphs?

Through extensive experimentation, we found that:

- Graph Convolutional Networks (GCNs) and our novel Multiplicative Integration GNNs (MI-GNNs) showed strong performance across various HR tasks.
- Binary graph classification yielded the best results on such complex and unbalanced HR graph data.
- The effectiveness of different architectures varied depending on the specific task and data structure. Specific training methodologies, exploring different loss functions, and careful hyperparameter tuning were crucial for optimal performance.

Research Question 4

What are the key considerations and best practices for applying GNNs to HR data, taking into account the unique characteristics and constraints of this domain?

Our research highlighted several key considerations:

- While our research focused primarily on model development and performance, it also highlighted the critical need to address privacy, ethical considerations, and potential biases in AI-driven recruitment tools.
- In many HR tasks, class labels are highly unbalanced towards few but high-value data points, requiring careful handling in model design and evaluation.
- The interpretability of model predictions is crucial in the sensitive HR domain. Graph data effectively meets this requirement by providing intelligible representations through its nodes and edges, which HR practitioners can interpret.
- Domain knowledge is essential in feature engineering and graph construction to capture relevant relationships in HR data. Therefore, close collaboration with the industry and HR teams is fundamental for such research.

Research Question 5

How accurate and efficient are the AI-based methods that have been examined? What are the limitations of these methodologies, and in what ways can they assist HR recruiters in personnel selection?

In the course of this study, we observed that:

- The combination of LLMs for feature extraction and GNNs for predictive modeling proved to be a powerful approach for handling the unstructured nature of CV and job description data, offering a promising direction for developing AI tools.

- The integration of multiple data sources (CVs, questionnaires, job descriptions) into unified graph representations promotes a more holistic evaluation of candidates, following the increasing recognition in HRM of the importance of considering various factors beyond traditional qualifications in assessing candidate suitability. Moreover, such graph-based approaches in HR open up opportunities for more personalized recruitment processes, where the unique combination of a candidate's attributes can be matched more precisely with the specific requirements and context of a job opening, such as the fit within the company and the entrepreneurial values.

- Graph-based approaches offer the potential to provide HR professionals with more sophisticated decision support tools. We demonstrated that these models—in particular for CJM in Chapter 9—could aid HR recruiters by suggesting who to asses first, prioritizing high-potential candidates from a vast pool of applicants, identifying non-obvious matches between candidates and job requirements and by providing explainable graph recommendations to support human decision-making.

- AI-based technologies are meant to exclusively assist HR recruiters, letting them dedicate their efforts to the intricate and human-centered elements of hiring, which demand empathy, intuitive understanding, and sophisticated reasoning.

Final Remarks and Future Outlook

This investigation was undertaken with an inquisitive mindset to apply the latest advancements in AI to recruitment in a real business setting. Throughout our research, we focused on keeping humans at the center of the recruitment process, recognizing that behind every data point and graph node are real people with hopes, aspirations, and potential. We have aimed to harness the power of AI to understand better and serve these individuals, matching them not just with jobs but with opportunities for growth and fulfillment. In this regard, this study wishes to represent a significant step towards AI that augments and supports, rather than replaces, human decision-making in recruitment.

We approached this challenge well aware of its intricate nature. The intersection of AI, HR, and individual careers represents a complex ecosystem with profound implications. While developing sophisticated GNN models is technically demanding, the true complexity lies in creating AI systems that can navigate the nuanced, often subjective world of human potential and organizational fit. As AI is rapidly revolutionizing the world we live in, we envision a future where AI serves as a trusted ally to both recruiters and job seekers, helping unlock human potential and guide individuals toward fulfilling career paths. However, we must remain vigilant to the ethical considerations and potential biases inherent in AI recruiting systems, and ensure that technological advancements enhance, rather than diminish, the human element in one of the most personal aspects of our lives—our careers.

We aim for future AI research in HR to foster fair, effective, and human-centric hiring methods, ultimately benefiting both organizations and individuals in their journey for meaningful professional engagement. The quest towards truly intelligent and ethical AI-driven recruitment systems is still in its infancy, and we hope that our contributions will serve as inspiration and guidance for future research endeavors in this exciting and impactful domain, that connects people with meaningful work and helps organizations build diverse, talented teams, all while prioritizing the human aspect of human resources.

References

Aas, Kjersti, Martin Jullum, and Anders Løland (2021). “Explaining individual predictions when features are dependent: More accurate approximations to Shapley values”. In: *Artificial Intelligence* 298, p. 103502. DOI: 10.1016/j.artint.2021.103502.

Achiam, Josh, Steven Adler, Sandhini Agarwal, et al. (2024). “GPT-4 technical report”. In: arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

Agarwal, Chirag, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik (2023). “Evaluating explainability for graph neural networks”. In: *Scientific Data* 10.1, p. 144.

AI Act (2024). *REGULATION (EU) 2024 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act).* https://www.artificialintelligenceact.eu.

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Akkas, Selahattin and Ariful Azad (2024). “GNNShap: Scalable and Accurate GNN Explanation using Shapley Values”. In: *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*. Ed. by Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee. ACM, pp. 827–838. DOI: 10.1145/3589334.3645599. URL: https://doi.org/10.1145/3589334.3645599.

Alsentzer, Emily, Samuel G. Finlayson, Michelle M. Li, and Marinka Zitnik (2020). “Subgraph Neural Networks”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. URL: https://proceedings.neurips.cc/paper/2020/hash/5bca8566db79f3788be9efd96c9ed70d-Abstract.html.

Amara, Kenza, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang (2022). "GraphFramEx: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks". In: *Learning on Graphs Conference, LoG 2022, 9-12 December 2022, Virtual Event*. Ed. by Bastian Rieck and Razvan Pascanu. Vol. 198. Proceedings of Machine Learning Research. PMLR, p. 44. URL: `https://proceedings.mlr.press/v198/amara22a.html`.

Armstrong, Michael and Stephen Taylor (2020). *Armstrong's Handbook of Human Resource Management Practice*. Kogan Page Publishers. ISBN: 9780749498283.

Ashraf, Inaam, Luca Hermes, André Artelt, and Barbara Hammer (2023). "Spatial Graph Convolution Neural Networks for Water Distribution Systems". In: *Proceedings of the 21th International Symposium on Intelligent Data Analysis (IDA)*. Lecture Notes in Computer Science. Springer, pp. 29–41.

Bacciu, Davide, Federico Errica, Alessio Micheli, and Marco Podda (2020). "A gentle introduction to deep learning for graphs". In: *Neural Networks* 129, pp. 203–221.

Bailey, Rob (2017). "HR applications of psychometrics". In: *Psychometric Testing: Critical Perspectives*, pp. 85–111.

Barabási, Albert-László (2013). "Network science". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1987, p. 20120375.

Becker, Brian and Barry Gerhart (1996). "The impact of human resource management on organizational performance: Progress and prospects". In: *Academy of management journal* 39.4, pp. 779–801.

Beer, Michael (1984). "Managing Human Assets". In: *Free Prees*.

Beer, Michael, Paul Boselie, and Chris Brewster (2015). "Back to the future: Implications for the field of HRM of the multistakeholder perspective proposed 30 years ago". In: *Human Resource Management* 54.3, pp. 427–438.

Bianchi, Filippo Maria and Veronica Lachi (2023). "The expressive power of pooling in Graph Neural Networks". In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine. URL: `http://papers.nips.cc/paper%5C_files/paper/2023/hash/e26f31de8b13ec569bf507e6ae2cd952-Abstract-Conference.html`.

Bishop, Christopher, Markus Svensén, and Christopher K. I. Williams (1998a). “Developments of the generative topographic mapping”. In: *Neurocomputing* 21.1-3, pp. 203–224. DOI: 10.1016/S0925-2312(98)00043-5.

– (1998b). “GTM: The Generative Topographic Mapping”. In: *Neural Comput.* 10.1, pp. 215–234. DOI: 10.1162/089976698300017953.

Bizer, Christian, Ralf Heese, Malgorzata Mochól, Radoslaw Oldakowski, Robert Tolksdorf, and Rainer Eckstein (2005). “The Impact of Semantic Web Technologies on Job Recruitment Processes”. In: *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety, 7. Internationale Tagung Wirtschaftsinformatik 2005, Bamberg, 23.2.2005 - 25.2.2005*. Ed. by Otto K. Ferstl, Elmar J. Sinz, Sven Eckert, and Tilman Isselhorst. Physica-Verlag, pp. 1367–1382. URL: http://aisel.aisnet.org/wi2005/72.

Boateng, Godfred O, Torsten B Neilands, Edward A Frongillo, Hugo R Melgar-Quiñonez, and Sera L Young (2018). “Best practices for developing and validating scales for health, social, and behavioral research: a primer”. In: *Frontiers in public health* 6, p. 149.

Bondarouk, Tanya, Emma Parry, and Elfi Furtmueller (2017). “Electronic HRM: four decades of research on adoption and consequences”. In: *The International Journal of human resource management* 28.1, pp. 98–131.

Bordt, Sebastian and Ulrike von Luxburg (2023). “From Shapley Values to Generalized Additive Models and back”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*. Vol. 206. Proceedings of Machine Learning Research. PMLR, pp. 709–745.

Borgatti, Stephen P, Ajay Mehra, Daniel J Brass, and Giuseppe Labianca (2009). “Network analysis in the social sciences”. In: *science* 323.5916, pp. 892–895.

Borgwardt, Karsten M, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel (2005). “Protein function prediction via graph kernels”. In: *Bioinformatics* 21.suppl_1, pp. i47–i56.

Bratton, John, Jeff Gold, Andrew Bratton, and Laura Steele (2021). *Human resource management*. Bloomsbury Publishing.

Bray, J Roger and John T Curtis (1957). “An ordination of the upland forest communities of southern Wisconsin”. In: *Ecological monographs* 27.4, pp. 326–349.

Breaugh, James A (2013). “Employee recruitment”. In: *Annual review of psychology* 64.1, pp. 389–416.

Breiman, Leo (2001). “Random Forests”. In: *Mach. Learn.* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324.

Brockschmidt, Marc (2020). "GNN-FiLM: Graph Neural Networks with Feature-wise Linear Modulation". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 1144–1152. URL: http://proceedings.mlr.press/v119/brockschmidt20a.html.

Bronstein, Michael M, Joan Bruna, Taco Cohen, and Petar Veličković (2021). "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges". In: *arXiv preprint arXiv:2104.13478*.

Bui, Ngoc, Hieu Trung Nguyen, Viet Anh Nguyen, and Rex Ying (2024). "Explaining Graph Neural Networks via Structure-aware Interaction Index". In: *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. URL: https://openreview.net/forum?id=2T00oYk54P.

Burrell, Jenna and Marion Fourcade (2021). "The society of algorithms". In: *Annual Review of Sociology* 47.1, pp. 213–237.

Campion, Michael C, Michael A Campion, Emily D Campion, and Matthew H Reider (2016). "Initial investigation into computer scoring of candidate essays for personnel selection." In: *Journal of Applied Psychology* 101.7, p. 958.

Cao, Kaidi, Jiaxuan You, Jiaju Liu, and Jure Leskovec (2023). *AutoTransfer: AutoML with Knowledge Transfer - An Application to Graph Neural Networks*. URL: https://openreview.net/forum?id=y81ppNf%5C_vg.

Cao, Wenzhi, Vahid Mirjalili, and Sebastian Raschka (2020). "Rank consistent ordinal regression for neural networks with application to age estimation". In: *Pattern Recognition Letters* 140, pp. 325–331. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2020.11.008.

Caruana, Rich, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes (2004). "Ensemble Selection from Libraries of Models". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, p. 18. ISBN: 1581138385. DOI: 10.1145/1015330.1015432.

Castro, Javier, Daniel Gómez, and Juan Tejada (2009). "Polynomial calculation of the Shapley value based on sampling". In: *Computers & Operations Research* 36.5, pp. 1726–1730. DOI: 10.1016/j.cor.2008.04.004.

Chen, Hugh, Ian C Covert, Scott M Lundberg, and Su-In Lee (2023). "Algorithms to estimate Shapley value feature attributions". In: *Nature Machine Intelligence* 5.6, pp. 590–601.

Chen, Jianbo, Le Song, Martin J. Wainwright, and Michael I. Jordan (2019). "L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data". In: *7th International Con-*

ference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net. URL: `https://openreview.net/forum?id=S1E3Ko09F7`.

Chen, Ting, Song Bian, and Yizhou Sun (2019). "Are powerful graph neural nets necessary? a dissection on graph classification". In: *arXiv preprint arXiv:1905.04579*.

Chen, Yihao, Xin Tang, Xianbiao Qi, Chun-Guang Li, and Rong Xiao (2022). "Learning graph normalization for graph neural networks". In: *Neurocomputing* 493, pp. 613–625. DOI: `10.1016/J.NEUCOM.2022.01.003`.

Cho, Kyunghyun, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans, pp. 1724–1734. DOI: `10.3115/V1/D14-1179`.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, et al. (2020). "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, pp. 8440–8451. DOI: `10.18653/V1/2020.ACL-MAIN.747`.

Conover, William Jay (1999). *Practical nonparametric statistics*. Vol. 350. john wiley & sons.

Corso, Gabriele, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Velickovic (2020). "Principal Neighbourhood Aggregation for Graph Nets". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. URL: `https://proceedings.neurips.cc/paper/2020/hash/99cad265a1768cc2dd013f0e740300ae-Abstract.html`.

Covert, Ian and Su-In Lee (2021a). "Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression". In: *The 24th International Conference on Artificial Intelligence and Statistics, (AISTATS 2021)*. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 3457–3465.

Covert, Ian, Scott M. Lundberg, and Su-In Lee (2021b). "Explaining by Removing: A Unified Framework for Model Explanation". In: *J. Mach. Learn. Res.* 22, 209:1–209:90. URL: `http://jmlr.org/papers/v22/20-1316.html`.

Dastin, Jeffrey (2018). "Amazon Scraps Secret AI Recruiting Tool That Showed Bias Against Women". In: *Reuters*. URL: `https://www.reuters.com/article/world/insight-amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK0AG/`.

De Neve, Jan-Emmanuel, Micah Kaats, and George Ward (2023). "Workplace wellbeing and firm performance". In: URL: https://wellbeing.hmc.ox.ac.uk/papers/2304-workplace-wellbeing-and-firm-performance/.

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). "Convolutional neural networks on graphs with fast localized spectral filtering". In: *Advances in neural information processing systems* 29.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, pp. 1–38. ISSN: 00359246.

Demsar, Janez (2006). "Statistical Comparisons of Classifiers over Multiple Data Sets". In: *J. Mach. Learn. Res.* 7, pp. 1–30. URL: https://jmlr.org/papers/v7/demsar06a.html.

Disegna, Marta, Nicolò Biasetton, Elena Barzizza, and Luigi Salmaso (2022). "A New Adaptive Membership Function with CUB Uncertainty with Application to Cluster Analysis of Likert-Type Data". In: *Available at SSRN 4115553*.

Dobson, Paul D and Andrew J Doig (2003). "Distinguishing enzyme structures from non-enzymes without alignments". In: *Journal of molecular biology* 330.4, pp. 771–783.

Doshi-Velez, Finale and Been Kim (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. arXiv: 1702.08608 [stat.ML].

Douze, Matthijs, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou (2024). *The Faiss library*. arXiv: 2401.08281 [cs.LG].

Duval, Alexandre and Fragkiskos D. Malliaros (2021). "GraphSVX: Shapley Value Explanations for Graph Neural Networks". In: *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part II*. Vol. 12976. Lecture Notes in Computer Science. Springer, pp. 302–318. DOI: 10.1007/978-3-030-86520-7_19. URL: https://doi.org/10.1007/978-3-030-86520-7%5C_19.

Easley, David, Jon Kleinberg, et al. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Vol. 1. Cambridge university press Cambridge.

Erickson, Nick, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander J. Smola (2020). "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data". In: *CoRR* abs/2003.06505. arXiv: 2003.06505.

Errica, Federico, Marco Podda, Davide Bacciu, and Alessio Micheli (2020). "A Fair Comparison of Graph Neural Networks for Graph Classification". In: *8th International Conference* *on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: `https://openreview.net/forum?id=HygDF6NFPB`.

Escalante, Hugo Jair, Heysem Kaya, Albert Ali Salah, et al. (2020). "Modeling, recognizing, and explaining apparent personality from videos". In: *IEEE Transactions on Affective Computing* 13.2, pp. 894–911.

Fabris, Alessandro, Nina Baranowska, Matthew J. Dennis, Philipp Hacker, Jorge Saldivar, Frederik J. Zuiderveen Borgesius, and Asia J. Biega (2023). "Fairness and Bias in Algorithmic Hiring". In: *CoRR* abs/2309.13933. DOI: `10.48550/ARXIV.2309.13933`. arXiv: `2309.13933`.

Faliagka, Evanthia, Lazaros Iliadis, Ioannis Karydis, Maria Rigou, Spyros Sioutas, Athanasios Tsakalidis, and Giannis Tzimas (2014). "On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV". In: *Artificial Intelligence Review* 42, pp. 515–528.

Fernández-Reyes, Francis C. and Suraj Shinde (2019). "CV Retrieval System based on job description matching using hybrid word embeddings". In: *Comput. Speech Lang.* 56, pp. 73–79. DOI: `10.1016/J.CSL.2019.01.003`. URL: `https://doi.org/10.1016/j.csl.2019.01.003`.

Fey, Matthias and Jan E. Lenssen (2019). "Fast Graph Representation Learning with PyTorch Geometric". In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Frazzetto, Paolo, Luca Pasa, Nicolò Navarin, and Alessandro Sperduti (2023a). "Topology preserving maps as aggregations for Graph Convolutional Neural Networks". In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC 2023, Tallinn, Estonia, March 27-31, 2023*. ACM, pp. 536–543. DOI: `10.1145/3555776.3577751`.

– (2024a). "Beyond the Additive Nodes' Convolutions: a Study on High-Order Multiplicative Integration". In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC 2024, Avila, Spain, April 8-12, 2024*. ACM, pp. 474–481. DOI: `10.1145/3605098.3636016`.

Frazzetto, Paolo, Muhammad Uzair Ul Haq, Flavia Fabris, and Alessandro Sperduti (2024b). "From Text to Talent: A Pipeline for Extracting Insights from Candidate Profiles". In: *The 3rd Italian Conference on Big Data and Data Science, (ITADATA 2024), Pisa, Italy, September 17-19, 2024*. Proceedings not yet available.

Frazzetto, Paolo, Muhammad Uzair-Ul-Haq, and Alessandro Sperduti (2023b). "Enhancing Human Resources through Data Science: a Case in Recruiting". In: *Proceedings of the 2nd Italian Conference on Big Data and Data Science (ITADATA 2023), Naples, Italy, September*

11-13, 2023. Vol. 3606. CEUR Workshop Proceedings. CEUR-WS.org. URL: `https://ceur-ws.org/Vol-3606/paper71.pdf`.

Frye, Christopher, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige (2021). "Shapley explainability on the data manifold". In: *International Conference on Learning Representations*.

Fujimoto, Katsushige, Ivan Kojadinovic, and Jean-Luc Marichal (2006). "Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices". In: *Games and Economic Behavior* 55.1, pp. 72–99. DOI: `10.1016/j.geb.2005.03.002`.

Fumagalli, Fabian, Maximilian Muschalik, **Paolo Frazzetto**, Janine Strotherm, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara Hammer (2025). "Exact Computation of Any-Order Shapley Interactions for Graph Neural Networks". In: *The International Conference on Learning Representations (ICLR)*. Under review.

Fumagalli, Fabian, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer (2024). *KernelSHAP-IQ: Weighted Least-Square Optimization for Shapley Interactions*. arXiv: `2405.10852 [cs.LG]`.

Fumagalli, Fabian, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Eva Hammer (2023). "SHAP-IQ: Unified Approximation of any-order Shapley Interactions". In: *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*.

Furnham, Adrian (2008). "HR professionals' beliefs about, and knowledge of, assessment techniques and psychometric tests". In: *International Journal of Selection and Assessment* 16.3, pp. 300–305.

GDPR (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. Official Journal of the European Union. GDPR consolidated version. URL: `https://eur-lex.europa.eu/eli/reg/2016/679/oj`.

Ghorbani, Amirata and James Y. Zou (2019). "Data Shapley: Equitable Valuation of Data for Machine Learning". In: *Proceedings of the 36th International Conference on Machine Learning, (ICML 2019)*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 2242–2251.

Gilmer, Justin, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl (2017). "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine

Learning Research. PMLR, pp. 1263–1272. URL: http://proceedings.mlr.press/v70/gilmer17a.html.

Goretzko, David and Laura Sophia Finja Israel (2021). “Pitfalls of Machine Learning-Based Personnel Selection”. In: *Journal of Personnel Psychology*.

Goudreau, Mark W, C Lee Giles, Srimat T Chakradhar, and Dong Chen (1994). “First-order versus second-order single-layer recurrent neural networks”. In: *IEEE Transactions on Neural Networks* 5.3, pp. 511–513.

Grabisch, Michel (2016). *Set Functions, Games and Capacities in Decision Making*. Vol. 46. Springer International Publishing Switzerland. ISBN: 978-3-319-30690-2. DOI: 10.1007/978-3-319-30690-2.

Grabisch, Michel and Marc Roubens (1999). “An axiomatic approach to the concept of interaction among players in cooperative games”. In: *International Journal of Game Theory* 28.4, pp. 547–565. DOI: 10.1007/s001820050125.

Grattarola, Daniele, Daniele Zambon, Filippo Maria Bianchi, and Cesare Alippi (2024). “Understanding Pooling in Graph Neural Networks”. In: *IEEE Trans. Neural Networks Learn. Syst.* 35.2, pp. 2708–2718. DOI: 10.1109/TNNLS.2022.3190922.

Guan, Chaoyu, Ziwei Zhang, Haoyang Li, Heng Chang, Zeyang Zhang, Yijian Qin, Jiyan Jiang, Xin Wang, and Wenwu Zhu (2021). “AutoGL: A Library for Automated Graph Learning”. In: *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*. URL: https://openreview.net/forum?id=0yHwpLeInDn.

Guest, David (2002). “Human resource management, corporate performance and employee wellbeing: Building the worker into HRM”. In: *The journal of industrial relations* 44.3, pp. 335–358.

– (2011). “Human resource management and performance: still searching for some answers”. In: *Human resource management journal* 21.1, pp. 3–13.

Guest, David, Jonathan Michie, Maura Sheehan, and Neil Conway (2000). “Getting inside the HRM-Performance Relationship.” In.

Guo, Shiqiang, Folami Alamudun, and Tracy Hammond (2016). “RésuMatcher: A personalized résumé-job matching system”. In: *Expert Systems with Applications* 60, pp. 169–182.

Ha, David, Andrew M. Dai, and Quoc V. Le (2017). “HyperNetworks”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=rkpACe1lx.

Hamilton, William L., Zhitao Ying, and Jure Leskovec (2017). “Inductive Representation Learning on Large Graphs”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1024–1034. URL: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

Hansen, Jonas Berg, Stian Normann Anfinsen, and Filippo Maria Bianchi (2023). “Power Flow Balancing With Decentralized Graph Neural Networks”. In: *IEEE Transactions on Power Systems* 38.3, pp. 2423–2433. DOI: 10.1109/TPWRS.2022.3195301.

Harsanyi, John C (1963). “A simplified bargaining model for the n-person cooperative game”. In: *International Economic Review* 4.2, pp. 194–220.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. DOI: 10.1109/ICCV.2015.123.

Hmoud, Bilal and Várallyai László (2019). “Will Artificial Intelligence Take Over Humanresources Recruitment And Selection”. In: *Network Intelligence Studies*, pp. 21–30.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780. DOI: 10.1162/NECO.1997.9.8.1735.

Hu, Weihua, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec (2020). “Open Graph Benchmark: Datasets for Machine Learning on Graphs”. In: URL: https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html.

Hua, Chenqing, Guillaume Rabusseau, and Jian Tang (2022). *High-Order Pooling for Graph Neural Networks with Tensor Decomposition*. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/282967f8abaae52a452a97ee961410f3-Abstract-Conference.html.

Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger (2017). “Densely Connected Convolutional Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

Huang, Qiang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang (2023). “GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks”. In: *IEEE Trans. Knowl. Data Eng.* 35.7, pp. 6968–6972. DOI: 10.1109/TKDE.2022.3187455.

Ioffe, Sergey and Christian Szegedy (2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd Interna-*

tional Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 448–456. URL: http://proceedings.mlr.press/v37/ioffe15.html.

Jain, Lorick, M. A. Harsha Vardhan, Ganesh Kathiresan, and Ananth Narayan (2021). "Optimizing People Sourcing Through Semantic Matching of Job Description Documents and Candidate Profile Using Improved Topic Modelling Techniques". In: *Advances in Artificial Intelligence and Data Engineering*. Ed. by Niranjan N. Chiplunkar and Takanori Fukao. Singapore: Springer Nature Singapore, pp. 899–908. ISBN: 978-981-15-3514-7.

Jansen, Karen J and Amy Kristof-Brown (2006). "Toward a multidimensional theory of person-environment fit". In: *Journal of Managerial issues*, pp. 193–212.

Janzing, Dominik, Lenon Minorics, and Patrick Blöbaum (2020). "Feature relevance quantification in explainable AI: A causal problem". In: *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*. Vol. 108. Proceedings of Machine Learning Research. PMLR, pp. 2907–2916.

Jayakumar, Siddhant M., Wojciech M. Czarnecki, Jacob Menick, Jonathan Schwarz, Jack W. Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu (2020). "Multiplicative Interactions and Where to Find Them". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=rylnK6VtDH.

Jethani, Neil, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath (2022). "FastSHAP: Real-Time Shapley Value Estimation". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. URL: https://openreview.net/forum?id=Zq2G%5C_VTV53T.

Jiang, Junshu, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo (2020). "Learning Effective Representations for Person-Job Fit by Feature Fusion". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. CIKM '20. Virtual Event, Ireland: Association for Computing Machinery, pp. 2549–2556. DOI: 10.1145/3340531.3412717.

Joshi, Ankur, Saket Kale, Satish Chandel, and D Kumar Pal (2015). "Likert scale: Explored and explained". In: *British journal of applied science & technology* 7.4, p. 396.

Kavas, Hamit, Marc Serra-Vidal, and Leo Wanner (2023). *Job Offer and Applicant CV Classification Using Rich Information from a Labour Market Taxonomy*. DOI: 10.2139/ssrn.4519766.

Kazius, Jeroen, Ross McGuire, and Roberta Bursi (2005). "Derivation and Validation of Toxicophores for Mutagenicity Prediction". In: *Journal of Medicinal Chemistry* 48.1, pp. 312–320. ISSN: 0022-2623. DOI: 10.1021/jm040835a.

Kessler, Rémy, Nicolas Béchet, Mathieu Roche, Marc El-Bèze, and Juan Manuel Torres-Moreno (2008). "Automatic profiling system for ranking candidates answers in human resources". In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, pp. 625–634.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: URL: http://arxiv.org/abs/1412.6980.

Kipf, Thomas N. and Max Welling (2017). "Semi-Supervised Classification with Graph Convolutional Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=SJU4ayYgl.

Kohonen, Teuvo (1982). "Self-organized formation of topologically correct feature maps". In: *Biological cybernetics* 43.1, pp. 59–69.

Koishekenov, Yeskendir and Erik J. Bekkers (2023). "An Exploration of Conditioning Methods in Graph Neural Networks". In: *CoRR* abs/2305.01933. DOI: 10.48550/ARXIV.2305.01933. arXiv: 2305.01933.

Kolpaczki, Patrick, Viktor Bengs, Maximilian Muschalik, and Eyke Hüllermeier (2024a). "Approximating the Shapley Value without Marginal Contributions". In: *Thirty-Eighth AAAI Conference on Artificial Intelligence, (AAAI 2024)*. AAAI Press, pp. 13246–13255. DOI: 10.1609/AAAI.V38I12.29225.

Kolpaczki, Patrick, Maximilian Muschalik, Fabian Fumagalli, Barbara Hammer, and Eyke Hüllermeier (2024b). "SVARM-IQ: Efficient Approximation of Any-order Shapley Interactions through Stratification". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics, (AISTATS 2024)*. Vol. 238. Proceedings of Machine Learning Research. PMLR, pp. 3520–3528.

König, Cornelius and Markus Langer (2022). "Machine learning in personnel selection". In: *Handbook of research on artificial intelligence in human resource management*. Edward Elgar Publishing, pp. 149–167.

König, Cornelius, Andrew M Demetriou, Philipp Glock, et al. (2020). "Some advice for psychologists who want to work with computer scientists on big data". In: *Personnel Assessment and Decisions* 6.1, pp. 17–23.

Krause, Ben, Iain Murray, Steve Renals, and Liang Lu (2017). "Multiplicative LSTM for sequence modelling". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net. URL: `https://openreview.net/forum?id=SJCS5rXFl`.

Kristof, Amy L (1996). "Person-organization fit: An integrative review of its conceptualizations, measurement, and implications". In: *Personnel psychology* 49.1, pp. 1–49.

Krueger, David, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron C. Courville (2017). "Bayesian Hypernetworks". In: *CoRR* abs/1710.04759. arXiv: `1710.04759`.

Kumar, I. Elizabeth, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle A. Friedler (2020). "Problems with Shapley-value-based explanations as feature importance measures". In: *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 5491–5500.

Kumar, Indra, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle A. Friedler (2021). "Shapley Residuals: Quantifying the limits of the Shapley value for explanations". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021 NeurIPS 2021*, pp. 26598–26608.

Kuncel, Nathan R, David M Klieger, and Deniz S Ones (2014). "In hiring, algorithms beat instinct". In: *Harvard business review* 92.5, p32–32.

Langer, Markus, Cornelius J König, and Vivien Busch (2021). "Changing the means of managerial work: effects of automated decision support systems on personnel selection tasks". In: *Journal of business and psychology* 36.5, pp. 751–769.

Langer, Markus, Cornelius J König, and Andromachi Fitili (2018). "Information as a double-edged sword: The role of computer experience and information on applicant reactions towards novel technologies for personnel selection". In: *Computers in Human Behavior* 81, pp. 19–30.

Laumer, Sven, Christian Maier, and Tim Weitzel (2022). "HR machine learning in recruiting". In: *Handbook of Research on Artificial Intelligence in Human Resource Management*. Edward Elgar Publishing, pp. 105–126.

Liu, Jixiong, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin (2023). "From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods". In: *J. Web Semant.* 76, p. 100761. DOI: `10.1016/J.WEBSEM.2022.100761`.

Liu, Yixin, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan (2022). "Towards Unsupervised Deep Graph Structure Learning". In: *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. ACM, pp. 1392–1403. DOI: 10.1145/3485447.3512186.

Loshchilov, Ilya and Frank Hutter (2019). "Decoupled Weight Decay Regularization". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Lundberg, Scott M., Gabriel G. Erion, Hugh Chen, et al. (2020). "From local explanations to global understanding with explainable AI for trees". In: *Nature Machine Intelligence* 2.1, pp. 56–67. DOI: 10.1038/s42256-019-0138-9.

Lundberg, Scott M. and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4765–4774. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Luo, Dongsheng, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang (2020). "Parameterized Explainer for Graph Neural Network". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 19620–19631. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/e37b08dd3015330dcbb5d6663667b8b8-Paper.pdf.

Marler, Janet H and John W Boudreau (2017). "An evidence-based review of HR Analytics". In: *The International Journal of Human Resource Management* 28.1, pp. 3–26.

Martinez-Gil, Jorge, Alejandra Lorena Paoletti, and Mario Pichler (2020). "A novel approach for learning how to automatically match job offers and candidate profiles". In: *Information Systems Frontiers* 22, pp. 1265–1274.

McAfee, Andrew, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton (2012). "Big data: the management revolution". In: *Harvard business review* 90.10, pp. 60–68.

McCloskey, Kevin, Ankur Taly, Federico Monti, Michael P Brenner, and Lucy J Colwell (2019). "Using attribution to decode binding mechanism in neural network models for chemistry". In: *Proceedings of the National Academy of Sciences* 116.24, pp. 11624–11629.

Mesquita, Diego P. P., Amauri H. Souza Jr., and Samuel Kaski (2020). "Rethinking pooling in graph neural networks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December*

6-12, 2020, virtual. URL: https://proceedings.neurips.cc/paper/2020/hash/1764183ef03fc7324eb58c3842bd9a57-Abstract.html.

Micheli, Alessio (2009). "Neural Network for Graphs: A Contextual Constructive Approach". In: *IEEE Trans. Neural Networks* 20.3, pp. 498–511. DOI: 10.1109/TNN.2008.2010350.

Morris, Christopher, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe (2019). *Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks*. DOI: 10.1609/AAAI.V33I01.33014602.

Munshi, Jamal (2014). "A Method for Constructing Likert Scales". In.

Muschalik, Maximilian, Fabian Fumagalli, Barbara Hammer, and Eyke Hüllermeier (2024). "Beyond TreeSHAP: Efficient Computation of Any-Order Shapley Interactions for Tree Ensembles". In: *Thirty-Eighth AAAI Conference on Artificial Intelligence, (AAAI 2024)*. AAAI Press, pp. 14388–14396. DOI: 10.1609/AAAI.V38I13.29352.

Myerson, Roger B. (1977). "Graphs and Cooperation in Games". In: *Math. Oper. Res.* 2.3, pp. 225–229. DOI: 10.1287/MOOR.2.3.225.

Naim, Iftekhar, Md Iftekhar Tanveer, Daniel Gildea, and Mohammed Ehsan Hoque (2016). "Automated analysis and prediction of job interview performance". In: *IEEE Transactions on Affective Computing* 9.2, pp. 191–204.

Navarin, Nicolò, Paolo Frazzetto, Luca Pasa, Pietro Verzelli, Filippo Visentin, Alessandro Sperduti, and Cesare Alippi (2024). "Physics-Informed Graph Neural Cellular Automata: an Application to Compartmental Modelling". In: *International Joint Conference on Neural Networks, IJCNN 2024, Yokohama, Japan, June 30 - July 5, 2024*. IEEE, pp. 1–9. DOI: 10.1109/IJCNN60899.2024.10650578.

Navarin, Nicolò, Dinh Van Tran, and Alessandro Sperduti (2019). "Universal Readout for Graph Convolutional Neural Networks". In: *International Joint Conference on Neural Networks*. Budapest, Hungary.

– (2020). "Learning Kernel-Based Embeddings in Graph Neural Networks". In: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*. Ed. by Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang. Vol. 325. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 1387–1394. DOI: 10.3233/FAIA200243.

Newman, M. E. J. and R. M. Ziff (2001). "Fast Monte Carlo algorithm for site or bond percolation". In: *Phys. Rev. E* 64 (1), p. 016706. DOI: 10.1103/PhysRevE.64.016706.

Newman, Mark (2018). *Networks*. Oxford University Press. ISBN: 9780198805090. DOI: 10.1093/oso/9780198805090.001.0001.

Niepert, Mathias, Mohamed Ahmed, and Konstantin Kutzkov (2016). "Learning Convolutional Neural Networks for Graphs". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 2014–2023.

Nikolaou, Ioannis and Janneke K Oostrom (2015). "Employee recruitment, selection, and assessment". In: *Contemporary Issues for Theory and Practice. Hove, East Sussex: Routledge*.

OpenAI (2024). *text-embedding-3-large Model*. Accessed: 2024-06-28. URL: https://www.openai.com/models/text-embedding-3-large.

Oswald, Frederick L, Tara S Behrend, Dan J Putka, and Evan Sinar (2020). "Big data in industrial-organizational psychology and human resource management: Forward progress for organizational research and practice". In: *Annual Review of Organizational Psychology and Organizational Behavior* 7.1, pp. 505–533.

Park, J. and I. W. Sandberg (1991). "Universal Approximation Using Radial-Basis-Function Networks". In: *Neural Computation* 3.2, pp. 246–257.

Pasa, Luca, Nicolò Navarin, Wolfgang Erb, and Alessandro Sperduti (2024). "Empowering Simple Graph Convolutional Networks". In: *IEEE Trans. Neural Networks Learn. Syst.* 35.4, pp. 4385–4399. DOI: 10.1109/TNNLS.2022.3232291.

Pasa, Luca, Nicolò Navarin, and Alessandro Sperduti (2020). "SOM-based aggregation for graph convolutional neural networks". In: *Neural Computing and Applications*, pp. 1–20.

– (2021). "Simple Multi-resolution Gated GNN". In: *IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021*. IEEE, pp. 1–7. DOI: 10.1109/SSCI50451.2021.9660046.

Pastor-Satorras, Romualdo and Alessandro Vespignani (2001). "Epidemic Spreading in Scale-Free Networks". In: *Phys. Rev. Lett.* 86 (14), pp. 3200–3203. DOI: 10.1103/PhysRevLett.86.3200.

Paszke, Adam, Sam Gross, Francisco Massa, et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035.

Peccei, Riccardo and Karina Van De Voorde (2019). "Human resource management–well-being–performance research revisited: Past, present, and future". In: *Human resource management journal* 29.4, pp. 539–563.

Pelegrina, Guilherme Dean, Leonardo Tomazeli Duarte, and Michel Grabisch (2023). "A *k*-additive Choquet integral-based approach to approximate the SHAP values for local interpretability in machine learning". In: *Artificial Intelligence* 325, p. 104014. DOI: 10.1016/J.ARTINT.2023.104014.

Pereira, Dulce G, Anabela Afonso, and Fátima Melo Medeiros (2015). "Overview of Friedman's test and post-hoc analysis". In: *Communications in Statistics-Simulation and Computation* 44.10, pp. 2636–2653.

Peronato, Eros, Flavia Fabris, Nicolò D'Agnolo, and Roberta D'Orazio (2022). *Entrepreneurial Values as a key for CSR in SMEs*. Presented at the XXXIII ISPIM, Copenhagen, 2022. URL: https://www.conferencesubmissions.com/ispim/copenhagen2022/index.html.

Perotti, Alan, Paolo Bajardi, Francesco Bonchi, and André Panisson (2022). "GRAPHSHAP: Motif-based Explanations for Black-box Graph Classifiers". In: *CoRR* abs/2202.08815. arXiv: 2202.08815.

Pessach, Dana, Gonen Singer, Dan Avrahami, Hila Chalutz Ben-Gal, Erez Shmueli, and Irad Ben-Gal (2020). "Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming". In: *Decision Support Systems* 134, p. 113290. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2020.113290.

Phan, Tung T., Vinh Q. Pham, Hien D. Nguyen, Anh T. Huynh, Dung A. Tran, and Vuong T. Pham (2021). "Ontology-Based Resume Searching System for Job Applicants in Information Technology". In: *Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices*. Ed. by Hamido Fujita, Ali Selamat, Jerry Chun-Wei Lin, and Moonis Ali. Cham: Springer International Publishing, pp. 261–273. ISBN: 978-3-030-79457-6.

Pope, Phillip E., Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann (2019). "Explainability Methods for Graph Convolutional Neural Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 10772–10781. DOI: 10.1109/CVPR.2019.01103.

Prokhorenkova, Liudmila Ostroumova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin (2018). "CatBoost: unbiased boosting with categorical features". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6639–6649. URL: https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

Pudasaini, Shushanta, Subarna Shakya, Sagar Lamichhane, Sajjan Adhikari, Aakash Tamang, and Sujan Adhikari (Jan. 2022). "Scoring of Resume and Job Description Using Word2vec and Matching Them Using Gale–Shapley Algorithm". In: Springer International Publishing, pp. 705–713. ISBN: 978-981-16-2125-3. DOI: 10.1007/978-981-16-2126-0_55.

Raghavan, Manish, Solon Barocas, Jon Kleinberg, and Karen Levy (2020). "Mitigating bias in algorithmic hiring: Evaluating claims and practices". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 469–481.

Rota, Gian-Carlo (1964). "On the foundations of combinatorial theory: I. Theory of Möbius functions". In: *Classic Papers in Combinatorics*. Springer, pp. 332–360.

Roy, Pradeep Kumar, Sarabjeet Singh Chowdhary, and Rocky Bhatia (2020). "A Machine Learning approach for automation of Resume Recommendation system". In: *Procedia Computer Science* 167. International Conference on Computational Intelligence and Data Science, pp. 2318–2327. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.03.284.

Russell, Stuart J and Peter Norvig (2016). *Artificial intelligence: a modern approach*. Pearson.

Sajjadiani, Sima, Aaron J Sojourner, John D Kammeyer-Mueller, and Elton Mykerezi (2019). "Using machine learning to translate applicant work history into predictors of performance and turnover." In: *Journal of Applied Psychology* 104.10, p. 1207.

Salas-Vallina, Andrés, Joaquín Alegre, and Álvaro López-Cabrales (2021). "The challenge of increasing employees' well-being and performance: How human resource management practices and engaging leadership work together toward reaching this goal". In: *Human Resource Management* 60.3, pp. 333–347.

Sanchez-Gonzalez, Alvaro, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia (2020). "Learning to Simulate Complex Physics with Graph Networks". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 8459–8468. URL: http://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.

Sánchez-Lengeling, Benjamín, Jennifer N. Wei, Brian K. Lee, Emily Reif, Peter Wang, Wesley Wei Qian, Kevin McCloskey, Lucy J. Colwell, and Alexander B. Wiltschko (2020). "Evaluating Attribution for Graph Neural Networks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. URL: https://proceedings.neurips.cc/paper/2020/hash/417fbbf2e9d5a28a855a11894b2e795a-Abstract.html.

Scarselli, Franco, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini (2008). "The graph neural network model". In: *IEEE transactions on neural networks* 20.1, pp. 61–80.

Schlichtkrull, Michael Sejr, Nicola De Cao, and Ivan Titov (2021). "Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=WznmQa42ZAx.

Schlichtkrull, Michael Sejr, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling (2018). "Modeling Relational Data with Graph Convolutional Networks". In: *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*. Vol. 10843. Lecture Notes in Computer Science. Springer, pp. 593–607. DOI: 10.1007/978-3-319-93417-4_38.

Schlippe, Tim and Koen Bothmer (2023). "Skill Scanner: An AI-Based Recommendation System for Employers, Job Seekers and Educational Institutions". In: vol. 16. 1, pp. 55–64. DOI: 10.3991/IJAC.V16I1.34779.

Schnake, Thomas, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, and Grégoire Montavon (2022). "Higher-Order Explanations of Graph Neural Networks via Relevant Walks". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.11, pp. 7581–7596. DOI: 10.1109/TPAMI.2021.3115452. URL: https://doi.org/10.1109/TPAMI.2021.3115452.

Sekiguchi, Tomoki (2004). "Person-organization fit and person-job fit in employee selection: A review of the literature". In: *Osaka keidai ronshu* 54.6, pp. 179–196.

Shapley, L. S. (1953). "A Value for n-Person Games". In: *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, pp. 307–318.

Shervashidze, Nino, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt (2011). "Weisfeiler-Lehman Graph Kernels". In: *J. Mach. Learn. Res.* 12, pp. 2539–2561. DOI: 10.5555/1953048.2078187.

Šidák, Zbyněk (1967). "Rectangular confidence regions for the means of multivariate normal distributions". In: *Journal of the American Statistical Association* 62.318, pp. 626–633.

Slack, Dylan, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju (2020). "Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods". In: *AAAI/ACM Conference on AI, Ethics, and Society (AIES 2020)*. ACM, pp. 180–186. DOI: 10.1145/3375627.3375830.

Sperduti, Alessandro and Antonina Starita (1997). “Supervised neural networks for the classification of structures”. In: *IEEE Trans. Neural Networks* 8.3, pp. 714–735. DOI: 10.1109/72.572108.

Stärk, Hannes, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Lió (2022). “3D Infomax improves GNNs for Molecular Property Prediction”. In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 20479–20502. URL: https://proceedings.mlr.press/v162/stark22a.html.

Strohmeier, Stefan (2022). *Handbook of Research on Artificial Intelligence in Human Resource Management*. Edward Elgar Publishing.

Strumbelj, Erik and Igor Kononenko (2014). “Explaining prediction models and individual predictions with feature contributions”. In: *Knowledge and Information Systems* 41.3, pp. 647–665. DOI: 10.1007/s10115-013-0679-x.

Strumbelj, Erik, Igor Kononenko, and Marko Robnik-Sikonja (2009). “Explaining instance classifications with interactions of subsets of feature values”. In: *Data Knowl. Eng.* 68.10, pp. 886–904. DOI: 10.1016/J.DATAK.2009.01.004.

Sundararajan, Mukund, Kedar Dhamdhere, and Ashish Agarwal (2020a). “The Shapley Taylor Interaction Index”. In: *Proceedings of the 37th International Conference on Machine Learning, (ICML 2020)*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 9259–9268.

Sundararajan, Mukund and Amir Najmi (2020b). “The Many Shapley Values for Model Explanation”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 9269–9278. URL: http://proceedings.mlr.press/v119/sundararajan20b.html.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). “Axiomatic Attribution for Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 3319–3328.

Sutskever, Ilya, James Martens, and Geoffrey E. Hinton (2011). “Generating Text with Recurrent Neural Networks”. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. Ed. by Lise Getoor and Tobias Scheffer. Omnipress, pp. 1017–1024. URL: https://icml.cc/2011/papers/524%5C_icmlpaper.pdf.

Tailor, Shyam A., Felix L. Opolka, Pietro Liò, and Nicholas Donald Lane (2022). "Do We Need Anisotropic Graph Neural Networks?" In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. URL: `https://openreview.net/forum?id=hl9ePdHO4%5C_s`.

Tambe, Prasanna, Peter Cappelli, and Valery Yakubovich (2019). "Artificial intelligence in human resources management: Challenges and a path forward". In: *California Management Review* 61.4, pp. 15–42.

Tang, Yiyi, Ziyan Xiao, Xue Li, Qingpeng Zhang, Esther W Chan, and Ian CK Wong (2024). "Large Language Model in Medical Information Extraction from Titles and Abstracts with Prompt Engineering Strategies: A Comparative Study of GPT-3.5 and GPT-4". In: *medRxiv*. DOI: `10.1101/2024.03.20.24304572`. eprint: `https://www.medrxiv.org/content/early/2024/03/21/2024.03.20.24304572.full.pdf`.

Tian, Xiaoguang, Robert Pavur, Henry Han, and Lili Zhang (2023). "A machine learning-based human resources recruitment system for business process management: using LSA, BERT and SVM". In: *Bus. Process. Manag. J.* 29.1, pp. 202–222. DOI: `10.1108/BPMJ-08-2022-0389`.

Tsai, Che-Ping, Chih-Kuan Yeh, and Pradeep Ravikumar (2023). "Faith-Shap: The Faithful Shapley Interaction Index". In: *Journal of Machine Learning Research* 24.94, pp. 1–42.

Uzair-Ul-Haq, Muhammad, **Paolo Frazzetto**, Alessandro Sperduti, and Giovanni Da San Martino (2024). "Improving Soft Skill Extraction via Data Augmentation and Embedding Manipulation". In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC 2024, Avila, Spain, April 8-12, 2024*. Ed. by Jiman Hong and Juw Won Park. ACM, pp. 987–996. DOI: `10.1145/3605098.3636010`.

Van Esch, Patrick, J Stewart Black, and Joseph Ferolie (2019). "Artificial intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy". In: *International Journal of Information Management* 57, p. 101994.

Vanetik, Natalia and Genady Kogan (2023). "Job Vacancy Ranking with Sentence Embeddings, Keywords, and Named Entities". In: *Information* 14.8. ISSN: 2078-2489. DOI: `10.3390/info14080468`.

Vanhala, Sinikka and Kaija Tuomi (2006). "HRM, company performance and employee well-being". In: *management revue*, pp. 241–255.

Velickovic, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio (2018). "Graph Attention Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference*

Track Proceedings. OpenReview.net. URL: https://openreview.net/forum?id=rJXMpikCZ.

Vinyals, Oriol, Samy Bengio, and Manjunath Kudlur (2016). "Order Matters: Sequence to sequence for sets". In: ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1511.06391.

Vrachimis, Stelios G, Marios S. Kyriakou, Demetrios G. Eliades, and Marios M. Polycarpou (2018). "LeakDB: A Benchmark Dataset for Leakage Diagnosis in Water Distribution Networks". In: *Proceedings of the 1st International Joint Conference on Water Distribution Systems Analysis & Computing and Control for the Water Industry (WDSA/CCWI).*

Vrontis, Demetris, Michael Christofi, Vijay Pereira, Shlomo Tarba, Anna Makrides, and Eleni Trichina (2023). "Artificial intelligence, robotics, advanced technologies and human resource management: a systematic review". In: *Artificial intelligence and international HRM*, pp. 172–201.

Vu, Minh N. and My T. Thai (2020). "PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. URL: https://proceedings.neurips.cc/paper/2020/hash/8fb134f258b1f7865a6ab2d935a897c9-Abstract.html.

Wale, Nikil, Ian A Watson, and George Karypis (2008). "Comparison of descriptor spaces for chemical compound retrieval and classification". In: *Knowledge and Information Systems* 14.3, pp. 347–375.

Wang, Xiao, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and Philip S. Yu (2023). "A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources". In: *IEEE Trans. Big Data* 9.2, pp. 415–436. DOI: 10.1109/TBDATA.2022.3177455.

Wang, Xiaowei, Zhenhong Jiang, and Lingxi Peng (2021). "A Deep-Learning-Inspired Person-Job Matching Model Based on Sentence Vectors and Subject-Term Graphs". In: *Complexity* 2021.1, p. 6206288. DOI: https://doi.org/10.1155/2021/6206288.

Wang, Yan, Yacine Allouache, and Christian Joubert (May 2021). "Analysing CV Corpus for Finding Suitable Candidates using Knowledge Graph and BERT". In: *DBKDA 2021, The Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications*. hal-03325062. Valencia, Spain.

Wang, Zifu, Xuefei Ning, and Matthew B. Blaschko (2023). "Jaccard Metric Losses: Optimizing the Jaccard Index with Soft Labels". In: *Advances in Neural Information Processing Sys-*

tems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/ee208bfc04b1bf6125a6a34baa1c28d3-Abstract-Conference.html.

Wei, Xiang, Xingyu Cui, Ning Cheng, et al. (2024). *ChatIE: Zero-Shot Information Extraction via Chatting with ChatGPT*. arXiv: 2302.10205 [cs.CL]. URL: https://arxiv.org/abs/2302.10205.

Wirtky, Thomas, Sven Laumer, Andreas Eckhardt, and Tim Weitzel (2016). "On the untapped value of e-HRM: A literature review". In: *Communications of the Association for Information Systems* 38.1, p. 2.

Wright, Marvin N., Andreas Ziegler, and Inke R. König (2016). "Do little interactions get lost in dark random forests?" In: *BMC Bioinform.* 17, p. 145. DOI: 10.1186/s12859-016-0995-8.

Wu, Lingfei, Peng Cui, Jian Pei, and Liang Zhao (2022). *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer Singapore, p. 725.

Wu, Yuhuai, Saizheng Zhang, Ying Zhang, Yoshua Bengio, and Ruslan Salakhutdinov (2016). "On Multiplicative Integration with Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2856–2864. URL: https://proceedings.neurips.cc/paper/2016/hash/f69e505b08403ad2298b9f262659929a-Abstract.html.

Xiong, Ping, Thomas Schnake, Michael Gastegger, Grégoire Montavon, Klaus-Robert Müller, and Shinichi Nakajima (2023). "Relevant Walk Search for Explaining Graph Neural Networks". In: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 38301–38324. URL: https://proceedings.mlr.press/v202/xiong23b.html.

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka (2019). "How Powerful are Graph Neural Networks?" In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=ryGs6iA5Km.

Xu, Keyulu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka (2018). "Representation Learning on Graphs with Jumping Knowledge Networks". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 5449–5458. URL: http://proceedings.mlr.press/v80/xu18c.html.

Yan, Rui, Ran Le, Yang Song, Tao Zhang, Xiangliang Zhang, and Dongyan Zhao (2019). “Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories”. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 914–922.

Yao, Kaichun, Jingshuai Zhang, Chuan Qin, Peng Wang, Hengshu Zhu, and Hui Xiong (2022). “Knowledge Enhanced Person-Job Fit for Talent Recruitment”. In: *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, pp. 3467–3480. DOI: 10.1109/ICDE53745.2022.00325.

Ye, Ziyuan, Rihan Huang, Qilin Wu, and Quanying Liu (2023). “SAME: Uncovering GNN Black Box with Structure-aware Shapley-based Multipiece Explanations”. In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/14cdc9013d80338bf81483a7736ea0
Abstract-Conference.html.

Yi, Xing, James Allan, and W. Bruce Croft (2007). “Matching resumes and jobs based on relevance models”. In: *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Ying, Zhitao, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec (2019). “GNNExplainer: Generating Explanations for Graph Neural Networks”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, pp. 9240–9251. URL: https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html.

Ying, Zhitao, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec (2018). *Hierarchical Graph Representation Learning with Differentiable Pooling*. URL: https://proceedings.neurips.cc/paper/2018/hash/e77dbaf6759253c7c6d0efc5690369c7
Abstract.html.

You, Jiaxuan, Zhitao Ying, and Jure Leskovec (2020). “Design Space for Graph Neural Networks”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

URL: https://proceedings.neurips.cc/paper/2020/hash/c5c3d4fe6b2cc463c7d7ecba17cc9de7 Abstract.html.

Yu, Kun, Gang Guan, and Ming Zhou (2005). "Resume Information Extraction with Cascaded Hybrid Model". In: *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. Ed. by Kevin Knight, Hwee Tou Ng, and Kemal Oflazer. The Association for Computer Linguistics, pp. 499–506. DOI: 10.3115/1219840.1219902.

Yuan, Hao, Haiyang Yu, Shurui Gui, and Shuiwang Ji (2023). "Explainability in Graph Neural Networks: A Taxonomic Survey". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 45.5, pp. 5782–5799. DOI: 10.1109/TPAMI.2022.3204236. URL: https://doi.org/10.1109/TPAMI.2022.3204236.

Yuan, Hao, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji (2021). "On Explainability of Graph Neural Networks via Subgraph Explorations". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 12241–12252. URL: http://proceedings.mlr.press/v139/yuan21c.html.

Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola (2017). "Deep Sets". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3391–3401. URL: https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Zhang, He, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei (2024). "Trustworthy graph neural networks: Aspects, methods, and trends". In: *Proceedings of the IEEE*.

Zhang, Mike, Kristian Nørgaard Jensen, Sif Dam Sonniks, and Barbara Plank (2022). "SkillSpan: Hard and Soft Skill Extraction from English Job Postings". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*. Association for Computational Linguistics, pp. 4962–4984. DOI: 10.18653/V1/2022.NAACL-MAIN.366.

Zhang, Muhan, Zhicheng Cui, Marion Neumann, and Yixin Chen (2018). "An end-to-end deep learning architecture for graph classification". In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhang, Shichang, Yozen Liu, Neil Shah, and Yizhou Sun (2022). "GStarX: Explaining Graph Neural Networks with Structure-Aware Cooperative Games". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. URL: `http://papers.nips.cc/paper%5C_files/paper/2022/hash/7d53575463291ea6b5a23cf6e571f59b-Abstract-Conference.html`.

Zhang, Yue, David DeFazio, and Arti Ramesh (2021). "RelEx: A Model-Agnostic Relational Model Explainer". In: *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*. ACM, pp. 1042–1049. DOI: `10.1145/3461702.3462562`. URL: `https://doi.org/10.1145/3461702.3462562`.

Zhao, Wayne Xin, Kun Zhou, Junyi Li, et al. (2024). *A Survey of Large Language Models*. arXiv: `2303.18223 [cs.CL]`. URL: `https://arxiv.org/abs/2303.18223`.

Zhou, Jie, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun (2020). "Graph neural networks: A review of methods and applications". In: *AI open* 1, pp. 57–81.

Zhou, Kaixiong, Zirui Liu, Rui Chen, Li Li, Soo-Hyun Choi, and Xia Hu (July 2022). "Table2Graph: Transforming Tabular Data to Unified Weighted Graph". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Main Track. International Joint Conferences on Artificial Intelligence Organization, pp. 2420–2426. DOI: `10.24963/ijcai.2022/336`.

Zhu, Chen, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li (2018). "Person-job fit: Adapting the right talent for the right job with joint representation learning". In: *ACM Transactions on Management Information Systems (TMIS)* 9.3, pp. 1–17.

Zilly, Julian Georg, Rupesh Kumar Srivastava, Jan Koutník, and Jürgen Schmidhuber (2017). "Recurrent Highway Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 4189–4198. URL: `http://proceedings.mlr.press/v70/zilly17a.html`.