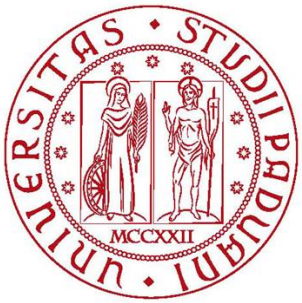


Università Degli Studi di Padova
Corso di Dottorato in Matematica
Department of Mathematics "Tullio Levi Civita"
Curriculum: Computational Mathematics


---

# Complex networks: community detection and graph semi-supervised learning on higher-order networks, with an application to the science of science.


*Candidate*
Sara Venturini

*Supervisor*
Prof. Francesco Rinaldi
Università Degli Studi di Padova
*Co-supervisor*
Prof. Francesco Tudisco
The University of Edinburgh
Gran Sasso Science Institute


---


CICLO XXXVI, 2020-2023

# Abstract

Networks represented as graphs with nodes and edges have emerged as effective tools for modeling and analyzing complex systems of interacting entities. Graphs arise naturally in many disciplines, such as social, information, infrastructure, and/or biological networks. However, advances in the study of networked systems have shown that real-world applications often require more sophisticated and diverse representations of interactions. Therefore, higher-order interactions have begun to be considered and analyzed in complex networks. Examples of higher-order models include multilayer networks, which represent different types of relationships between the nodes, and simplicial complexes or hypergraphs, which describe collective actions of groups of nodes. We deal with the community detection problem on higher-order networks, both in an unsupervised and semi-supervised context. This is a contemporary and important problem that involves identifying and analyzing communities or groups of nodes across multiple interconnected layers each representing a different type of interaction or relationship between nodes. Detecting communities in such networks allows us to uncover patterns of interaction and connectivity that might not be evident when considering each layer in isolation. The contribution of this thesis is both in the problems' modeling and in the novel approaches proposed to address these specific problems. We formulate the problems as multi-objective and bilevel optimization tasks. We solve them by applying and adapting suited and tailored modern optimization methods, like the Frank-Wolfe algorithm and block coordinate descent methods. Furthermore, the thesis offers an interesting and relevant application to the academic community related to bibliometric data, analyzing the relation between collaborations and topic switches in time-evolving collaboration networks of scholars.

# Contents

**1 Introduction** **5**

**2 A variance-aware multiobjective Louvain-like method for community detection in multiplex networks** **11**
2.1 Introduction . . . 11
2.2 Related work . . . 14
2.3 The Louvain method for single layer graphs . . . 16
2.4 Multiobjective Louvain-like method for multiplex networks . . . 18
2.4.1 Variance-aware cross-layer modularity function . . . 20
2.4.2 Positive vs negative variance regularization . . . 21
2.5 Experiments . . . 22
2.5.1 Synthetic Networks via SBM . . . 24
2.5.2 Synthetic Networks via LFR . . . 25
2.5.3 Real World Networks . . . 25
2.6 Conclusion . . . 34

**3 Learning the right layer: a data-driven layer-aggregation strategy for semi-supervised learning on multilayer graphs** **39**
3.1 Introduction . . . 39
3.2 Related work . . . 41
3.3 Bilevel optimization . . . 42
3.4 Learning the most relevant layers . . . 43
3.5 Optimization with inexact gradient computations . . . 46
3.5.1 Convergence analysis . . . 47
3.5.2 Implementation details . . . 51
3.6 Experiments . . . 51
3.6.1 Synthetic Datasets . . . 52
3.6.2 Real World Datasets . . . 54
3.7 Conclusion . . . 56

**4 Laplacian-based semi-supervised learning in multilayer hypergraphs by coordinate descent** **57**
4.1 Introduction . . . 57

4.2 Problem statement . . . 59
4.3 Block coordinate descent approaches . . . 62
4.3.1 Coordinate descent approaches . . . 64
4.3.2 Calculations . . . 65
4.4 Experiments . . . 66
4.4.1 Synthetic datasets . . . 68
4.4.2 Real datasets . . . 68
4.5 Conclusion . . . 70

**5 Collaboration and topic switches in science** **83**
5.1 Overview of the science of science . . . 83
5.2 Introduction . . . 90
5.3 Results . . . 90
5.3.1 Experiment I . . . 92
5.3.2 Experiment II . . . 96
5.4 Discussion . . . 103
5.5 Methods . . . 104
5.5.1 Data . . . 104
5.5.2 Overlap coefficient . . . 104
5.5.3 Author ranking metrics . . . 106
5.5.4 Statistical test for difference of samples . . . 106
5.5.5 Target activation probability . . . 108
5.5.6 Simple baseline for membership closure . . . 108
5.5.7 Source activation probability . . . 108
5.5.8 Chaperoning propensity . . . 109
5.6 Conclusion . . . 109

**6 Conclusion** **111**

**Bibliography** **117**

# Chapter 1

# Introduction

Complexity science is a field that investigates *complex systems*, which consist of a large number of components that interact with each other giving rise to significant phenomena that cannot be explained by analyzing each individual element in isolation [1].
Many real systems are considered to be complex systems. Examples include organisms, the human brain, living cells, social and economic organizations, transportation or communication systems, and the Earth's global climate. One of the primary challenges in complexity science lies in formally modeling and simulating these systems due to their inherent complexity.

Complex systems are difficult to model and simulate, therefore they are usually represented by *networks* in which nodes represent the components and links represent their interactions. This approach simplifies real systems while preserving essential information about the interaction structure that leads to emergent complex behaviors. Therefore, complex networks have become a powerful tool for investigating complex systems [2].
These networks are called *complex* because their structure is not *simple*. They differ from regular network models such as lattices, and random network models such as Erdös–Rényi random networks [3]. Their intrinsic structure is directly tied to the complexity observed in real-world complex systems, is typically irregular, and can evolve over time.

Some areas of applications of complex networks are:

- Technological networks, like the Internet, i.e., the computer data network in which the nodes are computers, and the edges are data connections between them; the telephone network; the transportation networks like networks of roads, rail lines, and airline routes. For instance, they are useful to better understand the flow of data traffic.
- Information networks, like the World Wide Web, where the nodes are web pages and the edges are the links between them; the citation network between academic journal articles.
- Social networks, like Facebook or Twitter, where the nodes are people and the edges between them are social connections of some kind. like friendship, communication, and collaboration. They are used to analyze the nature of social interactions, the spread of disease, and disinformation.

- Biological networks, like neural networks, concerning the connections between neurons in the brain or the macroscopic functional connectivity between large-scale regions of the brain; ecological networks, modeling the predator-prey relationships between species in an ecosystem; biochemical networks like metabolic networks, protein-protein interaction networks, genetic regulatory networks. They can help understand the complex chemical processing in the cell and perhaps even new therapies for diseases.

While models of complex systems as networks represented by a graph are popular and successful, real-world applications may require more sophisticated representations of interactions. For such cases, using a single graph is an oversimplifying assumption, which can lead to misleading models and results. Therefore, increasing attention has been devoted to *higher order interaction* models in the complex networks literature. These include multilayer networks, simplicial complexes, and hypergraphs.

- *Multilayer networks* [4, 5] represent networks that are interconnected through different layers, each of which captures a distinct type of relationship among nodes. These layers can represent different aspects or modes of interaction within a complex system. For instance, in social networks people can interact using different online platforms, each of them representing a different layer; in transportation networks, different places in a city can be connected through different types of transpositions (e.g., trains, buses, trams). In particular, multiplex networks are a subtype of widely studied multilayer graphs. Their layers share a common set of nodes and do not exhibit any inter-layer edges.

- *Simplicial complexes and hypergraphs* [6] are well-suited structures for modeling collective actions that involve groups of nodes. These types of interactions involving multiple nodes are called hyperedges. This framework has found applications in various fields, for instance in social network analysis people interact in groups; in chemical networks chemical reactions often involve multiple elements; in collaboration networks, scholars write papers in groups.

Certainly, an even more complex description of the real world may involve the combination of these two structures, considering multilayer hypergraphs.

In this thesis, we deal with contemporary and significant issues in network science extended to higher-order networks. The contributions are both in the problems' formulation and in their resolution applying and adapting suited and tailored modern optimization methods. In particular:

- We deal with the community detection problem on multiplex networks. We formulate the problem as a multi-objective optimization problem of maximizing the modularity score within each layer. In order to address it, we introduce a new method which is based on the widely employed Louvain heuristic for single-layer networks.

- We deal with the semi-supervised learning problem on multiplex networks. We formulate the problem as a bilevel optimization task. It offers the flexibility of learning a nonlinear aggregation function that adapts the weights assigned to each network based on the available labeled data. We tackle it using an inexact Frank-Wolfe algorithm in conjunction with a

parametric Label Propagation strategy. We present a comprehensive convergence analysis of the method.

- We deal with the semi-supervised learning problem on multilayer hypergraphs. We represent a hypergraph with its clique expansion, and we conduct a comparison between the application of various coordinate descent methods and the gradient descent algorithm.

- We propose an application to the science of science. We investigate the interplay between collaboration and topic switches. We found that the likelihood of a researcher engaging in a new research topic rises in conjunction with the number of prior collaborators and that the more productive/impactful an active author is, the more likely their coauthors will start working with them on a new topic.

We now present the thesis outline and the contributions per chapter in more detail.

In Chapter 2, we deal with the *community detection* problem on multiplex networks. This problem consists of finding communities in graphs, i.e., groups of nodes that are densely connected internally and loosely connected to the nodes in the other communities [7, 8]. This is one of the most relevant tasks in the analysis of graphs representing real systems as it has been shown that many real-world networks show a community structure. For instance, friendship networks have communities made out of close friends; in the World Wide Web, communities are represented by pages dealing with the same topic; in metabolic networks, they can be metabolites performing the same biochemical tasks. While many community detection algorithms have been developed over the recent years, most of these are designed for standard single-layer graphs. However, as we have discussed above, this can be an oversimplification of reality. In this chapter, we propose a method to find communities in multiplex networks. Since there are different definitions of communities for multiplex networks, we highlight that in this thesis, using the terminology introduced in [9], our aim is to find a set of communities that is total (i.e., every node belongs to at least one community), node-disjoint (i.e., no node belongs to more than one cluster on a single layer), and pillar (i.e., each node belongs to the same community across the layers).
Various community detection algorithms for multiplex networks have been proposed in the literature [9]. Many of these methods either suitably reduce the multiplex to a single-layer graph or rely on some form of aggregation of the various layers, not taking full advantage of the multi-aspect of multilayer graphs. Furthermore, they usually do not consider dealing with possibly noisy or corrupted data, because they focus on the consistency of multiple layers and do not consider possible inconsistencies. Therefore, in this chapter, we propose a method for community detection on multiplex networks that aims to simultaneously consider the information contained in the different layers and take into consideration the possible presence of noisy layers. To simultaneously consider the different layers of information, we write the problem as a multiobjective optimization problem and we solve it by extending the popular Louvain heuristic to a filter-type algorithm exploring the Pareto front. At the same time, the method takes into consideration the possible presence of noise, considering the average and the variance of modularity scores across the multiple layers. The efficacy and robustness of our method are shown through extensive experiments over both synthetic and real-world datasets.

The content of this chapter corresponds to the Journal of Complex Networks 2022 publication: "A Variance-aware Multiobjective Louvain-like Method for Community Detection in Multiplex Networks" [10].

Another issue, strictly related to community detection on networks, is the *graph semi-supervised learning problem* [11, 12, 13]. The aim is still to find a partition of the graph but take advantage of available input information on the community assignment of some nodes. This goal can also be seen as building a classifier that takes into account both labeled and unlabeled observations, by considering a suitable loss function and the underlying graph structure of the observations. Therefore, we can approach it by optimizing a continuous objective function composed of a fitting term, which takes into consideration the input labels, and a 2-Laplacian term as a regularizer. In Chapter 3 and Chapter 4, we analyze different aspects of the semi-supervised community detection problem extended to, respectively, multiplex networks and multilayer hypergraphs.

Many available algorithms for semi-supervised learning on multilayer networks deal with the available multiple information, aggregating somehow the different layers. As mentioned above, in multiplex networks, the multiple layers can carry different amounts of information. Some layers can be totally or partially corrupted. Therefore, giving all the layers the same importance can lead to incorrect results [14, 15]. In Chapter 3, the main idea is to assign to each layer a different weight in the objective function, depending on their contribution to the cluster assignment. However, this is information that we do not have apriori. We decided to learn these weights using the input available labels. To do so, we reframe the problem as a bilevel optimization problem where, at the lower level we optimize over the labels, and at the upper level, we optimize over the weights. The method allows for a non-linear combination of the layers and has as sub-cases the most popular aggregating expressions [16, 17, 18]. We solve the bilevel problem using a zeroth order version of the Frank Wolfe algorithm in conjunction with a parametric Label Propagation strategy. We also present a comprehensive convergence analysis of our method, showing its sub-linear convergence rate. We highlight that learning the layers weights gives us also a better interpretability of the multiplex network under analysis. We tested our methods over various synthetic and real multiplex networks, showing their effectiveness in dealing with diverse clustering scenarios, especially when certain layers are dominated by noise.
The content of this chapter corresponds to the ICML 2023 publication: "Learning the Right Layers: a Data-Driven Layer-Aggregation Strategy for Semi-Supervised Learning on Multilayer Graphs" [19].

In Chapter 4, we still deal with the graph semi-supervised learning problem but we analyze a different aspect of it. Considering higher-order interactions can, on the one side, lead to a more detailed modeling of a real problem but, on the other side, it can bring additional complexity and harder treatability. In this chapter, we extend the semi-supervised learning problem on multilayer hypergraphs, i.e. a set of hypergraphs each representing a different layer. Specifically, we compute a first effort in approaching this computationally demanding optimization problem, whose complexity derives from considering more sophisticated structures in the regularization term. Since this is a first attempt, albeit knowing that this modeling can be improved, we

decide to deal with the multilayer aspect summing the information over the different layers and managing the presence of hyperedges substituting them with cliques involving the corresponding group of nodes. The main idea is to solve the more complex problem by splitting it into simpler sub-problems, and we do so by applying various coordinate descent methods [20]. These methods, at each iteration, determine a coordinate or coordinate block via a particular selection rule, and then they optimize over the corresponding coordinate hyperplane while fixing all other coordinates or coordinate blocks. We compare these methods against the popular gradient descent algorithm over extensive experiments on synthetic and real-world networks. The results demonstrate the advantages of employing coordinate descent methods, especially when combined with appropriate selection rules tailored to the specific problem at hand. Furthermore, we conducted an analysis where we replaced the quadratic regularization term in the objective function with a more versatile $p$-regularizer. A choice of $p$ other than 2 changes deeply the nature of the objective function making it more difficult to handle. However, we show that it can result in improved performance across the different datasets we evaluated.
The content of this chapter corresponds to the EURO Journal on Computational Optimization 2023 publication: "Laplacian-based Semi-Supervised Learning in Multilayer Hypergraphs by Coordinate Descent" [21].

In Chapter 5, we focus on an application to the *science of science* [22, 23]. At the beginning of the chapter, we give a brief introduction to this relatively new but highly promising and rapidly developing field. The science of science uses bibliometric data (e.g., productivity, paper citations, collaboration, research funding, scholar affiliations) to analyze and model the scientific landscape. The aim is to find intricate patterns and mechanisms that characterize the structure and the evolution of science with the final goal of accelerating science. For instance, the science of science deals with the metrics used to evaluate academic careers, the gender and ethnic disparities present in scientific publications, the effect of small and big collaborations, and the evolution of different fields. The main reason why we approach the science of science is that we want to deal with a real problem that can naturally be modeled as a multilayer hypergraphs [24, 25, 26]. For instance, if we want to model the interactions between different scholars, these can be due to different factors. For example, they can collaborate on some papers, they can come from the same institution, they can cite each other works, and they can write papers about similar topics. Each of these information can represent one layer of a multilayer network. Additionally, as we have mentioned above, interactions between scholars are often group interactions, as the coauthors of a paper, and therefore they can be represented as hyperedges. In this chapter, we start an analysis of how researchers influence each other with their research topics, therefore we restrict our work to single-layer collaboration networks evolving over time for simplicity [27, 28]. Further investigations will include multiple sources of information and group interactions.
In our work, we divide the set of authors into *active*, if they wrote at least one paper with the focal topic, and *inactive* otherwise. We take into consideration the point of view of both these two sets. The first experiment concerns the inactive authors. We calculate the probability of a scholar switching on a new topic correlated to the number of interactions with active authors, discovering that these two quantities are directly proportional. We also show that the impacts of individual collaborators are interconnected and not independent. Furthermore, we find that this

probability depends on the relative prominence of the active authors, whether it is calculated according to their productivity or impact. The second experiment deals with the point of view of the active authors, comparing the more and less prominent among them. We discover that the more prominent an author is, the more inclined he is to influence others. We also find that inactive coauthors of prominent authors who collaborate with more individuals tend to have a lower probability of switching topics.
We show the robustness of our findings considering different topics in the three disciplines of physics, computer science, and biology & medicine. Our analysis is based on the bibliometric dataset OpenAlex [29].
The content of this chapter corresponds to the preprint: "Collaboration and topic switches in science" [30].

# Chapter 2

# A variance-aware multiobjective Louvain-like method for community detection in multiplex networks

In this chapter, our focus is to address the issue of identifying communities within multiplex networks. These networks consist of multiple layers, with each layer denoting various types of interactions, the layers share the same set of nodes and there are not edges between nodes of different layers. Specifically, our objective involves identifying sets of nodes that exhibit a consistent community structure across all layers. For this purpose, we introduce a novel method that extends the popular Louvain method. This extension involves two main aspects: (a) updating the average and variance of modularity scores across multiple layers simultaneously, and (b) redefining the greedy search process using a filter-based multiobjective optimization approach. In contrast to various previous strategies for maximizing modularity that involve aggregating the layers in some manner, our multiobjective approach focuses on simultaneously maximizing the modularity of each layer independently. We performed experiments using both synthetic and real-world networks to demonstrate the efficacy and robustness of our proposed strategies. Our approach proves to be effective in scenarios where all layers exhibit consistent community structures, as well as in situations where certain layers contain only noise.

the

## 2.1 Introduction

Identifying communities, which refer to clusters of nodes characterized by dense internal connections and weaker connections to nodes outside the cluster, is a critical concern in the analysis of graphs that represent real-world systems. Despite the proliferation of community detection and clustering algorithms in recent years, many of these are tailored for conventional single-layer graphs. Fortunato conducted a comprehensive survey on this subject [7]. However, the assumption of having just one graph is often overly simplistic and can yield misleading models and outcomes.

To address this limitation, researchers have acknowledged the need for methods that can handle more complex structures, such as multiplex or multilayer graphs, which capture richer interactions and dependencies among nodes.

Recent advancements in the study of networked systems have revealed that the interconnected nature of our world often consists of networks interlinked through various layers. Each layer signifies a distinct type of interaction within these systems [5]. Multilayer networks, which emerge naturally, find application across a range of fields. These applications include transportation networks [31], financial-asset markets [32], temporal dynamics [33, 34], semantic world clustering [35], multivideo face analysis [36], mobile phone networks [37], social balance [38], citation analysis [39], and many others. These examples underscore the prevalence and significance of multilayer networks in various domains.

Similar to standard (single-layer) models, community detection remains a pivotal challenge in the study of multilayer networks. However, the existence of multiple layers introduces several additional complexities. One such challenge arises from the diverse types of multilayer structures that networks can possess. Additionally, the presence of multiple layers introduces the issue of community consistency, where communities may or may not align across the various layers. This multifaceted nature of multilayer networks confirms the need for specialized methods and techniques to effectively uncover and analyze communities within these intricate systems.

In our study, we specifically concentrate on multiplex networks, which are represented as a series of graphs referred to as layers. These layers share a common set of nodes and do not exhibit any inter-layer edges. Furthermore, each layer is assumed to be undirected and simple. By the terminology introduced in the reference [9], our objective centers on identifying a collection of communities that possesses three key properties: *total* (i.e., each node is a member of at least one community), *node-disjoint* (i.e., no node is a member of more than one cluster within a single layer), *pillar* (i.e., each node retains its affiliation to the same community across all layers).

In recent years, various community detection algorithms tailored for multiplex networks have emerged. These approaches encompass a variety of successful strategies: some methods effectively simplify the multiplex structure by transforming it into a single-layer graph, specific approaches modify and extend established single-layer consensus and spectral clustering techniques to cater to the multiplex scenario, and other methods leverage information-theoretic principles and flow diffusion strategies to address community detection in multiplex networks, some approaches focus on inferring communities by fitting appropriate planted partition models to the multiplex data. These methods constitute a significant body of research aimed at solving the challenges of community detection in multiplex networks. We attempt to summarize the related literature in §2.2 and refer to the survey [9] for further details.

In our research, we introduce a novel approach that directly addresses the multiobjective optimization problem of maximizing the modularity score within each layer. To accomplish this, we adapt the well-known Louvain heuristic method, which is widely employed for single-layer networks [40]. This method involves a locally greedy procedure that incrementally enhances the modularity of node partitions. A natural extension of this method to the multiplex case, already studied in the literature, e.g. in [32, 41], is to locally maximize a weighted average $M_Q$ of the modularity of the layers, instead of the modularity of a single layer. One of the advantages of

using a linear combination of the layer modularities is that the increment of $M_Q$ can be directly computed using the increment of the modularity of each layer, whose computation is efficiently handled by the original Louvain technique.

In Section 2.3, we present the original Louvain method designed for single-layer graphs, alongside the definition of the modularity function. Likewise, in Section 2.4, we adopt a similar approach by extending the Louvain strategy to accommodate the objective of maximizing a vector-valued function representing the modularities across all layers. To tackle the ensuing multiobjective optimization problem, we introduce a mechanism for recording and dynamically refining a collection of solutions through a specially designed Pareto search technique. The size of this solution list, as well as the final community assignment, is managed using a scalar cost function. This function factors in both the average and variance of the layer modularities. By incorporating a positive or negative variance regularization term, we gain better control over the variability of modularity scores across layers. This flexibility allows us to handle cases involving informative layers and the presence of noise. Importantly, although the resulting scalar cost function involves a nonlinear combination of layer modularities, we demonstrate that by iteratively computing the modularity of each layer, we can efficiently update their variance. This efficient update process results in a multiobjective Louvain-like scheme that is variance-aware and capable of handling diverse scenarios effectively.

An essential aspect of the proposed approach is its distinctive feature of not requiring an initial assignment of the number of classes, which sets it apart from various methods outlined in the existing literature. This feature is of great importance, considering that this information about the community structure of the multiplex is often unavailable. In the absence of such information, many approaches necessitate making preliminary assumptions, which can be unwarranted, regarding the number of clusters. By obviating the need for such assumptions, our approach provides a more flexible and adaptable solution for community detection in multiplex networks.

To assess the effectiveness and robustness of our technique, we perform experiments on multiplex graphs within two distinct scenarios: the all-informative setting,i.e., all layers of the multiplex contain informative data about the community structure, and noisy layer setting,i.e., at least one layer is designated as textitnoisy layer accounting for data with corruption or noise. In Sections 2.5.1 and 2.5.2, we undertake a comparison of our method with nine baseline algorithms. We employ synthetic multilayer networks generated using the Stochastic Block Model (SBM) [42] and the Lancichinetti-Fortunato-Radicchi (LFR) Benchmark [43]. Notably, we extend these benchmarks to accommodate the multiplex framework. Subsequently, in Section 2.5.3, we extend our assessment to include real-world multiplex graphs. The outcomes of our experiments demonstrate that factoring in the variance across layers can significantly enhance performance, particularly when dealing with noisy data. Notably, our proposed filter-based algorithm often achieves either the best or second-best classification results. This confirms the value of our multiobjective approach and confirms its efficacy in community detection tasks.

## 2.2 Related work

In the subsequent section, we attempt to review and summarize some recent strategies concerning community detection within multilayer networks. As in [9], we focus on algorithms specifically designed to uncover communities in the context of multiplex networks.

Flattening methods involve the reduction of a multiplex network into a single-layer weighted network, followed by the application of conventional community detection algorithms. The most straightforward approach within this category constructs a single-layer graph in which two nodes are connected if they are neighbors in at least one of the layers [44]. Alternatively, weighted single-layer graphs can be generated. These graphs assign weights based on certain structural attributes of the multiplex [44, 45].

Layer flattening corresponds to the process of combining the adjacency matrices from individual layers into a unified aggregated adjacency matrix. This approach enables the utilization of more sophisticated community detection algorithms. For instance, merging modularity matrices or employing specialized node embeddings becomes possible through this process. The study of these extended methods is explored in [46], which introduces a novel technique that integrates the node structural features derived from the layers.

Another popular strategy involves aggregation at the level of the cost function. This involves extending single-layer community detection cost functions to the multilayer context. Notably, the Generalized Louvain (GL) method introduced by Mucha et al. [41] operates by maximizing a multilayer adaptation of Newman's modularity [47]. Additionally, Bazzi et al. [32] propose a related technique, a Louvain-based modularity maximization method tailored for community detection in multilayer temporal networks. Both of these methods utilize a cross-layer modularity function, which is essentially a weighted arithmetic mean of the modularities of individual layers. As we will discuss in the next section, our proposed approach defines a new variance-aware Louvain-like method, which leverages this idea as a starting point. Another related approach, as presented by Pramanik et al. [48], employs a weighted linear combination of modularities from each layer to define a multilayer modularity index. Their approach is focused on two-layer networks with both inter- and intra-layer connections, with the aim of simultaneously detecting inter- and intra-layer communities.

Various alternative strategies have emerged in recent times. Pizzuti and Socievole [49] introduce a genetic algorithm for community detection in multilayer networks, incorporating a multiobjective optimization framework. Their approach exploits the notion of Pareto dominance to generate new populations during each iteration. Consequently, at the end of the optimization process, the algorithm furnishes a collection of solutions representing diverse trade-offs between objectives. From this set of solutions, the optimal solution is ultimately selected through tailored strategies.

De Domenico et al. [50] extend the famous information-theoretic approach introduced in [51]. They present a method that generalizes the map equation for single graphs. This method identifies communities as sets of nodes that effectively capture flow dynamics within and across layers over an extended duration.

De Bacco et al. [52] propose a likelihood maximization-based approach. They define a mixed-membership multilayer stochastic block model and put forth a method that deduces communities by fitting this model to a given multilayer dataset through log-likelihood maximization.

Wilson et al. [53] introduce a technique for multilayer data. Their method targets the identification of densely connected sets of vertex-layer pairs. This is achieved by employing a significance-based score that measures the connectivity of these sets in comparison to a suitable fixed-degree random graph model.

Another prominent avenue of exploration involves methods inspired by data clustering techniques. Zeng et al. [54] put forth a pattern mining algorithm for identifying closed quasi-cliques that manifest on multiple layers with a frequency surpassing a given threshold. A cross-graph quasi-clique is defined as a set of vertices belonging to a maximal quasi-clique that emerges across all layers [55].

Tang et al. [39] and Dong et al. [56] introduce graph clustering algorithms tailored for multilayer graphs based on matrix factorization. The central idea is to extract common factors from multiple graphs to facilitate various clustering methods. Tang et al. [39] factorize adjacency matrices, while Dong et al. [56] factorize graph Laplacian matrices. Liu et al. [57] present a nonnegative matrix factorization-based multiview clustering algorithm. This approach involves factors that represent clustering structures across multiple views, which are simultaneously regularized toward a shared consensus.

Another popular line of research revolves around extending spectral clustering to multilayer graphs. These algorithms typically strive to formulate a graph operator that encapsulates all relevant information from the multilayer graph. The aim is to ensure that the eigenvectors corresponding to the smallest eigenvalues provide meaningful insights into the clustering structure. These methods often rely on a form of "mean operator". For instance, they might use the Laplacian of the average adjacency matrix or the average Laplacian matrix [58]. Zhou and Burges [59] have contributed to this area by developing a multiview spectral clustering approach. This method generalizes the conventional single-view normalized cut to the multi-view scenario. It aims to identify a cut that is close to optimal on each layer. Chen and Hero [60] have introduced an algorithm that employs convex aggregation of layers based on signal-plus-noise models.

Various alternative approaches have also been proposed in this field. Dong et al. [61] extend spectral clustering by merging informative Laplacian eigenspaces from different layers through a subspace optimization analysis on Grassmann manifolds. Zhan et al. [62, 63, 64] introduce multiview graph learning approaches that consolidate multiple graphs into a unified graph with the desired number of connected components. Other methods exploit the concept of maximizing clustering agreement. Zong et al. [64] propose Weighted Multi-View Spectral Clustering, using the largest canonical angle to quantify differences between spectral clustering results from different views. Nie et al. [65] put forth a self-weighted scheme for fusing multiple graphs, accounting for the importance of each view, termed the Procrustes Analysis technique.

A common limitation among the proposed multiview clustering methods is their lack of consideration for dealing with potentially noisy or corrupted data. These methods focus primarily on enhancing the consistency across multiple layers and often neglect to account for potential inconsistencies. To tackle this concern, Xia et al. [66] introduced the Robust Multi-view Spectral Clustering approach. This method utilizes a Markov chain framework to learn an intrinsic transition matrix from multiple views, by restricting the transition matrix to be low-rank. Similarly, Mercado et al. [42, 14] have addressed this limitation. They propose a Laplacian operator derived by combining Laplacians from various layers using a one-parameter family of nonlinear matrix

power means. More recently, Liang et al. [67] have presented a novel multiview graph learning framework. This framework simultaneously addresses both multi-view consistency and multiview inconsistency within a unified objective function. These advancements highlight an emerging interest in addressing the issue of noisy or inconsistent data in the context of multiview community detection.
An additional approach adopted in the field involves Bayesian inference [68]. This method entails making hypotheses regarding node connections to identify the best model fit to a graph. The optimization of a suitable likelihood is then performed [69].
Bickel and Scheffer [70] extended the semi-supervised co-training approach [71] to multi-view clustering. Co-training involves iterating over all views and optimizing the objective function in each view based on results from previous views. Kumar and Daum'e [72] introduce a co-training approach aiming to identify a consistent clustering that aligns across multiple views. This method is based on the underlying assumption that each layer can be independently used for clustering. In a similar vein, Kumar et al. [73] propose Co-regularized Spectral Clustering under the same assumption. This approach emphasizes co-regularization, striving to maximize agreement between different views for enhanced clustering results.

## 2.3 The Louvain method for single layer graphs

In this work, we rely on the idea that a graph has a community structure if it is "different" enough from a random graph. A random graph should indeed not have a community structure, since any two nodes have the same probability to be adjacent. To this end, a *null model* is used as a term of comparison when checking whether a graph shows a community structure or not. This is a core concept in the definition of the *modularity*, a quality function that identifies a subgraph as a community if the number of edges inside it exceeds the expected number of internal edges that the same subgraph would have in the null model. The modularity function can be written as follows

$$Q = \frac{1}{2m} \sum_{i,j} (A_{ij} - P_{ij}) \delta_{ij} \tag{2.1}$$

where the sum runs over all pairs of vertices, $A$ is the adjacency matrix, $m$ the total number of edges of the graph (or the sum of all their weights in the case of weighted networks), $P_{ij}$ represents the expected number of edges between vertices $i$ and $j$ in the null model, and the function $\delta$ yields one if vertices $i$ and $j$ are in the same community, zero otherwise.
Even though several variations of the modularity and the null model have been proposed in the literature, the arguably most popular null model is the configuration model originally considered by Newman and Girvan in [47], where edges are linked at random, under the constraint that the expected degree of each vertex of the null model coincides with the actual degree of the corresponding node in the original graph [74]. With this choice of $P$, the modularity function reads

$$Q = \frac{1}{2m} \sum_{i,j} \Big( A_{ij} - \frac{k_i k_j}{2m} \Big) \delta_{ij} \tag{2.2}$$

where $k_i$ is the degree of the node $i$ (the sum of all the edges incident to $i$), and the function $\delta$ yields one if vertices $i$ and $j$ are in the same community, zero otherwise.
. In this model, communities are found via the maximization of $Q$. As this is known to be an NP-hard combinatorial optimization task [75], several algorithms have been proposed to approximately compute a modularity-based community assignment.
The *Louvain method*, introduced by Blondel et al. in [40], is one of the most popular and most effective methods for modularity maximization. This method is based on a greedy strategy that consists of two phases: initially, each vertex is a community and, in the first phase, the algorithm computes the gain, in terms of weighted modularity, obtained by including a node $i$ in the community of every neighbor node $j$, then selecting the one with the largest increase in modularity, as long as it is positive. Once the first level partition is obtained, in the second phase of the method, communities are replaced by super vertices. Two super vertices are connected if there is at least an edge between the vertices of the corresponding communities, with the edge weight between the two super vertices being the sum of the edge weights between the corresponding communities. The whole procedure is repeated iteratively until the modularity cannot increase and the algorithm stops.
The Louvain heuristic is very popular for its simplicity and efficiency. Part of the algorithm's efficiency results from the fact that the modularity can be calculated iteratively during the procedure. Only at the very beginning of phase 1, the method calculates the modularity from scratch.
To calculate the modularity, the algorithm uses the following formula, which is equivalent to the equation

$$Q = \sum_{r=1}^{|C|} \frac{|E(C_r)|}{m} - \left(\frac{\sum_i k_i}{2m}\right)^2 \tag{2.3}$$

where $C$=($C_1$,. . . ,$C_{|C|}$) is the clustering assignment, $m$ is the sum of weights of all the links in the network, $|E(C_r)|$ is the sum of the weights of all the links between nodes in the community $C_r$ and $k_i$ is the degree of node $i$.
During the loop, the algorithm calculates the modularity gain cheaply. The gain $\Delta Q1_i$, obtained by moving a node $i$ from its community $C_r$, can easily be computed via the formula

$$\Delta Q1_i = \frac{\sum_{tot} \cdot k_i}{2m^2} - \frac{{k_i}^2}{2m^2} - \frac{k_{i,in}}{m} \tag{2.4}$$

where $\sum_{tot}$ is the sum of weights of the links incident to the nodes in $C_r$, $k_i$ is the degree of node $i$, $k_{i,\ in}$ is the sum of weights of the links from $i$ to nodes in $C_r$.
Similarly, the following expression is used to evaluate the change of modularity $\Delta Q2_{i\to j}$ when an

$$\Delta Q2_{i\to j} = \frac{\sum_{i,in}}{m} - \frac{\sum_{tot} \cdot k_i}{2m^2} \tag{2.5}$$

where $\sum_{i,in}$ is the sum of weights of the links inside community $C_r$.
Thus, if a node $i$ changes community, the algorithm calculates the modularity of the new partition just by adding the gains obtained above to the initial modularity value, rather than calculating the function $Q$ from scratch. In particular, note that these formulas to calculate the modularity

gain can be easily extended to the weighted graph created in the second phase of the algorithm. Due to this fact, the Louvain heuristic is extremely fast.
One of the main drawbacks of modularity is its resolution limit which may prevent it from detecting clusters that are comparatively small concerning the graph as a whole [76]. Therefore, many different versions of modularity have been proposed [77]. Furthermore, it has been shown that modularity admits an exponential number of distinct high-scoring solutions and typically lacks a clear global maximum [78]. This implies that the output of any modularity maximization procedure should be interpreted cautiously in scientific contexts.

## 2.4 Multiobjective Louvain-like method for multiplex networks

In this section, we introduce our novel method designed for community detection in multiplex networks. Building upon the widely known Louvain heuristic method for single-layer networks [40], our approach is tailored to maximize the modularity across all layers simultaneously. Diverging from various alternative strategies that aggregate either the multiplex or the cost function into a single-layer representation of the original multiple layers, our method directly addresses the multiobjective nature inherent to the problem at hand, i.e., the existence of more than one objective to optimize. To achieve this, our algorithm maintains and updates a list of community assignments deemed suitable throughout its execution. Each of these assignments is favored over the others based on a particular criterion. This approach allows us to effectively address the complexity of optimizing multiple objectives in the context of multiplex community detection.
More formally, consider a multiplex with $k$ layers $G_1, \dots, G_k$, where $G_s = (V, E_s)$ is the graph forming the $s$-th layer. Thus, consider the vector of layer modularities $Q = (Q_1, \dots, Q_k)$. Here and everywhere in the text, we shall always assume vectors are column vectors unless otherwise specified. The modularity score of the $s$-th layer is defined as

$$Q_s = \frac{1}{2m_s} \sum_{ij} \Big( A_{ij}^{(s)} - \frac{d_i^{(s)} d_j^{(s)}}{2m_s} \Big) \delta_{ij} \,, \tag{2.6}$$

where the sum runs over all pairs of vertices, $A^{(s)}$ is the adjacency matrix of $G_s$, $m_s$ the total number of edges in $E_s$ (or the sum of all their weights, in the case of weighted graphs), $d_i^{(s)}$ is the degree (or weighted degree) of the node $i$ in $G_s$ and the function $\delta$ yields one if vertices $i$ and $j$ are in the same community and zero otherwise.
We aim at maximizing all entries of $Q$ simultaneously, i.e., we consider the following multiobjective optimization problem:

$$\max_{\{\text{partitions of } V\}} (Q_1, \dots, Q_k) \tag{2.7}$$

In multiobjective optimization, there is no unique way to define optimality, since there is no a-priori total order for $\mathbb{R}^k$ and each partial order leads to different strategies. Here, we consider the well-established definition of optimality according to Pareto [79]:

**Definition 2.4.1.** Given two vectors $z^1, z^2 \in \mathbb{R}^k$, we write $z^1 \succeq_P z^2$ if $z^1$ dominates $z^2$ according

to Pareto, that is:

$$\begin{aligned} z_i^1 &\geq z_i^2 \quad \text{for each index } i = 1, .., k \text{ and} \\ z_j^1 &> z_j^2 \quad \text{for at least one index } j = 1, .., k\,. \end{aligned}$$

A vector $z^* \in \mathbb{R}^k$ is Pareto optimal if there is no other vector $z \in \mathbb{R}^k$ such that $z \succeq_P z^*$. Moreover, the Pareto front is the set of all Pareto optimal points.

To tackle the optimization problem expressed in equation (2.7), we adapt the well-known Louvain method to approach a feasible solution along the Pareto front of the modularity vector. The procedure commences with an initial partition in which each node represents an individual community. This initial partition leads to an initial modularity vector, denoted as $Q$. The subsequent process involves a two-phase approach, leading to the generation of a list $L$ that contains community assignments and their associated modularity vectors. The fundamental criterion guiding this generation is that no entry in this list should be Pareto-dominated by any other entry. In simpler terms, each community assignment should not be consistently outperformed by any other assignment in terms of all objectives. The final approximate solution is determined through the use of a scalar function $F$ designed to evaluate the "quality" of a partition across the multiple layers. The choice of $F$ is explored in detail in §2.4.1. To start the process, the list $L$ is initialized with the initial modularity vector $Q = (Q_s)_s$ as well as the corresponding initial community partition and the value of $F$ associated with it. This initialization sets the foundation for the subsequent optimization steps.
During the first phase of the algorithm, a sequential node selection process is undertaken based on a specified initial node ordering. The process involves examining each node individually, and for each node, its neighboring nodes $j$ (excluding those already considered) are assessed. For each node $i$ and every layer $s$, the change in modularity $\Delta Q_s^{(i\to j)}$ is calculated as the sum of the change in modularity on layer $s$ obtained by removing $i$ from its community $C^{(i)}$ and the variation of modularity on layer $s$ obtained by including $i$ in the community $C^{(j)}$.
In the first phase, the algorithm picks one node at a time, following a given initial node ordering. For each node $i$, for every layer $s$, and for every neighbor $j$ of node $i$ (among the $j$s that have not been considered yet), we compute the change of modularity $\Delta Q_s^{(i\to j)}$ as the sum of the change in modularity on layer $s$ due to the removal of node $i$ from its community $C_i$ and the variation of modularity on layer $s$ caused by incorporating node $i$ into the community $C_j$ of its neighboring node $j$. For each alteration in community assignment, the resulting new modularity vector $Q^{(i\to j)} = (Q_1 + \Delta Q_1^{(i\to j)}, \ldots, Q_k + \Delta Q_k^{(i\to j)})$ is evaluated by efficiently updating the previous modularity scores, following the methodology utilized in the original Louvain approach. If the modularity vector $Q^{(i\to j)}$ is not Pareto-dominated by any of the modularity vectors already present in the list $L$, then it is a promising candidate for inclusion in $L$. However, akin to the original Louvain method, we seek to incorporate only new modularity vectors that signify a "strict improvement". To address this, we utilize the modularity updates $\Delta Q_s^{(i\to j)}$ to efficiently compute the change $\Delta F^{(i\to j)}$ in the scalar function $F$. If this change results in a positive increment in the quality function $F$, we add $Q^{(i\to j)}$, the updated value of the quality function $F + \Delta F^{(i\to j)}$, and the associated partition to the list $L$. Consequently, partitions in $L$ whose modularity vectors are

outperformed by the newly added vector are removed. Furthermore, to prevent an exponential growth in the size of the list $L$, a final control measure is implemented. Specifically, elements in $L$ with low values of $F$ are filtered out, ensuring that only the $h$ partitions yielding the highest values of $F$ are retained. This step effectively maintains the list $L$ at a manageable size while still considering partitions with the most promising performance metrics.
This process continues iteratively until the list $L$ ceases to change and exclusively includes non-dominating vectors. At this juncture, the method identifies the most optimal partition according to the criterion of $F$, selecting it as the new starting point. Subsequently, the algorithm advances to the second phase, involving an aggregation step where the communities within the chosen partition are merged to create condensed vertices, resulting in a smaller graph representation. The entire procedure is reiterated iteratively until no further enhancement in the Pareto sense is achievable. This follows the same approach as the original Louvain method for single-layer graphs. The complete algorithmic outline is summarized in Algorithm 1.
The selection of the function $F$ plays a pivotal role in determining the effectiveness of the proposed approach. This decision affects not only the ultimate community assignment but also the computational efficiency of the strategy, as the evaluation of $F$ can be a resource-intensive process. In the following, we contend that an appropriate choice of $F$ should consider both the average and variance of the layer modularities. We demonstrate that these two metrics can be evaluated through a cost-effective iterative update procedure.

### 2.4.1 Variance-aware cross-layer modularity function

One possible approach to quantify the quality of a partition in a multiplex network involves calculating the average of the corresponding modularity functions across all layers. In essence, this choice corresponds to setting $F$ equal to $M_Q$ , with

$$M_Q = \frac{1}{k} \sum_{s=1}^{k} Q_s \,. \tag{2.8}$$

The concept of evaluating a linear combination, possibly with weights, of the modularity functions from individual layers is a conceptually natural approach. This idea has been explored in previous studies, such as in [32, 41, 48]. One notable advantage of this approach is that the change $\Delta F^{(i \to j)}$, which quantifies how the selected function $F$ is affected when node $i$ is transferred from community $C_i$ to $C_j$ during the first phase of the algorithm, can be easily calculated due to the linear relationship between $F$ and $Q_s$:

$$\Delta M_Q^{(i \to j)} = \frac{1}{k} \sum_{s=1}^{k} \Delta Q_s^{(i \to j)} \tag{2.9}$$

This observation is at the basis of the GL method, proposed in [41], see also [9, 32].
While calculating the average modularity offers a straightforward perspective on the community structure across layers, this approach can occasionally oversimplify the analysis [42, 14]. Particularly, in scenarios involving noisy layers, linear averages over the multiple layers perform poorly

[42]. To address this limitation, we explore two functions that incorporate the sampled variance of the layer modularities:

$$F_- = (1-\gamma)M_Q - \gamma V_Q \qquad \text{and} \qquad F_+ = (1-\gamma)M_Q + \gamma V_Q \tag{2.10}$$

where $\gamma \in (0,1)$ is a parameter and $V_Q$ is the sampled variance of the modularity of the layers, which we compute as

$$V_Q = \frac{1}{k-1}\sum_{s=1}^{k}(Q_s - M_Q)^2\,. \tag{2.11}$$

While $F_\pm$ is now quadratic in $Q_s$, we observe below that, as for the linear choice $F = M_Q$, the increment $\Delta F_\pm^{(i\to j)}$ of both $F_-$ and $F_+$ can be computed efficiently during the algorithm. A direct computation shows that the following formula holds:

$$\Delta F_\pm^{(i\to j)} = (1-\gamma)\Delta M_Q^{(i\to j)} \pm \gamma\, R_Q^{(i\to j)},$$

where $R_Q^{(i\to j)}$ is the coefficient

$$R_Q^{(i\to j)} = V_{\Delta Q}^{(i\to j)} + \frac{2}{k-1}(Q - M_Q\mathbf{1})^\top(\Delta Q^{(i\to j)} - \Delta M_Q^{(i\to j)}\mathbf{1})\,,$$

$\mathbf{1} = (1,\ldots,1)$ is the vector of all ones, $Q = (Q_1,\ldots,Q_k)$ is the vector of the layer modularities, $\Delta Q^{(i\to j)}$ is the column vector whose $s$-th component is the gain $\Delta Q_s^{(i\to j)}$, and $V_{\Delta Q}^{(i\to j)}$ is the sampled variance of $\Delta Q_s^{(i\to j)}$, which we compute as follows:

$$V_{\Delta Q}^{(i\to j)} = \frac{1}{k-1}\sum_{s=1}^{k}(\Delta Q_s^{(i\to j)} - \Delta M_Q^{(i\to j)})^2\,. \tag{2.12}$$

The presented formulas highlight that, similar to the iterative updates used for $\Delta Q_s^{(i\to j)}$ within $M_Q$, the nonlinear quality functions $F_\pm$ can also be iteratively updated during the execution of Algorithm 1. Importantly, this process remains computationally efficient by utilizing the incremental changes $\Delta F_\pm^{(i\to j)}$. Consequently, this approach ensures the efficient calculation of the quality metrics associated with the novel community assignments. This simultaneous tracking of the mean and variance of the layer modularities aids in providing a comprehensive evaluation of the community structures' consistency and variability, enhancing the overall quality assessment.

### 2.4.2 Positive vs negative variance regularization

The two quality functions, $F_+$ and $F_-$, offer the flexibility to address distinct forms of modularity variability across layers. Specifically, in scenarios where all layers demonstrate consistent community structure, the use of $F_-$ is appropriate. This choice stems from the desire to achieve a balance between achieving high modularity values while keeping layer variability within acceptable limits. A higher value of the parameter $\gamma$ results in a smaller variance across the layers in the

final solution, aligning with the notion of an optimal solution when community structures across all layers coincide. Conversely, when dealing with layers that contain noise or exhibit limited community structure, the preference shifts to utilizing $F_+$. In these instances, the objective is to favor solutions that not only possess a significant modularity but also display a significant level of variability across layers. An elevated value of $\gamma$ here allows for greater permissible variability across layers, potentially facilitating the accommodation of noise or diverse layer characteristics. In summary, the selection between $F_+$ and $F_-$ hinges on the nature of the layers, and the choice of the parameter $\gamma$ offers a way to strike a balance between modularity and variability to cater to different underlying structures and potential noise in the multiplex network.
Overall, we study three variants of the proposed Louvain Multiobjective Method in Algorithm 1, which correspond to the following three different choices of the function $F$: the modularity average $M_Q$, defined in (2.8), and the functions $F_-$ and $F_+$ defined in (2.10). We refer to the corresponding algorithms respectively as *Louvain Multiobjective Average* (**MA**), *Louvain Multiobjective Variance Minus* (**MVM**) and *Louvain Multiobjective Variance Plus* (**MVP**).
The determination of the list length $h$, as mentioned earlier, involves a trade-off between exploration of the Pareto front and computational efficiency. A longer list ($h$) facilitates a more thorough approach to the Pareto front and exploration of layer modularity space. However, it also comes at the cost of increased computational complexity, which grows exponentially with $h$. Our experimental results indicate that even with a relatively small value of $h$ such as 2 or 3, significant performance improvements in terms of accuracy and normalized mutual information (NMI) can be achieved compared to $h = 1$. It's noteworthy that when $h$ is set to 1, the method essentially simplifies into an aggregation strategy where the multiobjective aspect is neglected. Instead, the focus shifts towards maximizing the selected scalar function $F$ using a Louvain-like greedy approach. In this context, methods such as MVP and MVM with $h = 1$ provide extensions to the GL method by incorporating variance considerations. On the other hand, MA with $h > 1$ introduces a multiobjective variant of the GL approach. For further clarity, when $h = 1$, we refer to the method using the $F_-$ function as *Louvain Expansion Variance Minus* (**EVM**) and the method using the $F_+$ function as *Louvain Expansion Variance Plus* (**EVP**). This distinction highlights their focus on either minimizing variance ($F_-$) or accommodating variability and noise ($F_+$) in the layers.

## 2.5 Experiments

We implemented the methods described in §2.4 using `Matlab`. Our codes are all available on the GitHub page: `https://github.com/saraventurini/A-Variance-aware-Multiobjective-Louvain-like-Method-for-Community-Detection-in-Multiplex-Networks`.
We considered both synthetic and real-world networks, performing extensive experiments to compare the proposed methods against nine multilayer community detection baselines (see §2.2 for details), namely:

- **GL**: Generalized Louvain [32, 41, 48];
- **CoReg**: Co-Regularized spectral clustering, with parameter $\lambda = 0.01$ [73];

- **AWP**: Multi-view clustering via Adaptively Weighted Procrustes [65];
- **MCGC**: Multi-view Consensus Graph Clustering, with parameter $\beta = 0.6$ [64];
- **PM**: Power mean Laplacian multilayer clustering, with parameter $p = -10$ [42];
- **MT**: Multitensor expectation maximization [52];
- **SCML**: Subspace Analysis on Grassmann Manifolds, with parameter $\alpha = 0.5$ [61];
- **PMM**: Principal Modularity Maximization, with parameters $l = 10$ and maxKmeans = 5 [46, 80];
- **IM**: Information-theoretic generalized map equation [50].

Notably, all of these methods, except GL and IM, necessitate the user to predetermine the number of communities being sought. This could pose a potential limitation in real-world applications, given that we often lack prior knowledge about the inherent community structure of the graph, which would lead us to possibly make unfounded assumptions regarding the number of clusters. Methods' performance is evaluated using two metrics: the accuracy, measured as the percentage of nodes assigned to the correct community [64], and the Normalized Mutual Information (NMI) [81]. We recall that the output of any modularity maximization procedure should be interpreted cautiously in practical contexts since there could be different existing valid partitions for the same network and different methods can find different local optima [78]. An interesting future possibility would be to integrate the comparison with another performance measure that does neither rely on the attributes nor on the learned partition, like the modularity values of the output partition. In all performed experiments, we explored two distinct scenarios: the informative scenario, where every layer contains meaningful information about the underlying clusters, and the noisy scenario, wherein some layers consist solely of randomly generated noise. As elaborated in Section 2.4.1, the inclusion of a negative variance component as seen in $F_-$ proves suitable for the informative context, whereas the presence of a positive variance term as depicted in $F_+$ becomes beneficial in the presence of noisy layers. Consequently, we assessed the performance of EVM and MVM for the informative setting, while we employed EVP and MVP for the noisy scenario. Additionally, to validate the advantages associated with the variance term in $F_\pm$, we also present the results of MA, which exclusively accounts for the sum of modularities across layers, effectively offering a form of a multiobjective variant of GL. For the Louvain Multiobjective model, we evaluated two different list lengths: $h = 2$ and $h = 3$, denoted by adding the corresponding number to the method's abbreviation (e.g., MA2 represents the Louvain Multiobjective Average method with a list length of $h = 2$).
To comprehensively assess the method's performance across various variance regularizing parameters, we considered a range of values for $\gamma$, specifically $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, which were utilized in the definitions of both $F_-$ and $F_+$. In Figures 2.1-2.4 and Tables 2.2-2.4, we present the scores achieved with the parameter that led to the highest Normalized Mutual Information (NMI) on each dataset. Notably, it is worth mentioning that a substantial level of comparability in performance was achieved for a wide array of parameter values. As such, the evaluation of the methods across varying values of $\gamma$ offers valuable empirical insight into selecting the most

suitable $\gamma$ value. This empirical guidance suggests that a well-balanced contribution, such as $\gamma \approx 0.5$, is preferable in informative scenarios, while a higher value such as $\gamma \approx 0.9$ appears to yield superior results in the presence of noise. The model with a negative parameter $\gamma$, frequently exhibits improved performance, suggesting that encouraging higher variance, i.e., promoting different partitions across layers, is beneficial for achieving better results. This perspective is further supported by considering the increased number of parameters. When all communities across layers match, there are only $N$ non-unique parameters, whereas if they are all different, the model has $NK$ parameters. Given that, a better result (in this case obtained with higher variance, hence more parameters) can be due to overfitting. However, our aim is exactly to overfit the informative layers, discarding the noisy ones. Some other existing methods allow nodes to belong to more than one community, however, this is outside the scope and setting of this analysis.

As locally greedy algorithms, the initial ordering of nodes during phase one of all our methods, much like the standard Louvain method, can influence the final performance. Our computational experience suggests that the choice of a specific ordering has a relatively minor impact on the cost function itself, but it may affect computational time. Determining the most suitable initial ordering is a nontrivial challenge and a well-known concern when employing these types of greedy strategies. In our experiments, we adopt a tailored approach to selecting the initial node ordering based on the specific characteristics of the network scenario at hand. More precisely, we arrange the nodes according to their community size in the informative setting. In contrast, for the noisy setting, we assign nodes in a random order. This distinction is primarily motivated by computational considerations: sorting nodes by their community size incurs a higher computational cost, which remains reasonable for the informative case but becomes impractical for the noisy scenario.

### 2.5.1 Synthetic Networks via SBM

Here, we analyze networks generated using the multilayer Stochastic Block Model (SBM), a generative model for graphs with planted communities defined by parameters $p$ and $q$. These parameters determine edge probabilities: between nodes $i$ and $j$, the edge probability is $p$ (resp. $q$) if nodes $i$ and $j$ belong to the same (resp. different) community. For our experiments, we set $p > q$ to generate informative layers and $p = q$ for noisy ones. Specifically, we construct networks with 4 communities, each containing 125 nodes, and either 2 or 3 layers. We keep $p = 0.1$ fixed and vary the $p/q$ ratio to control the informativeness of the layers. In the case of noisy layers, we maintain $p = q = 0.1$. Each pair of $(p, q)$ values is tested on 10 random instances, and the results are averaged. The outcomes are presented in Figures 2.1 and 2.2, categorizing the settings as follows: 2.1(a)(b) two informative layers; 2.1(c)(d) three informative layers; 2.2(a)(b) two informative layers and one noisy layer; 2.2(c)(d) two informative layers and two noisy layers. Overall, the proposed approaches demonstrate strong performance across various parameter configurations compared to the baseline methods. The variance-based multiobjective approaches (MVM and MVP) tend to perform the best, even in scenarios with multiple noisy layers (Fig. 2.2(b)). Notably, while the community detection task becomes easier with higher $p/q$ ratios, the proposed approaches consistently outperform other methods. Furthermore, we investigated the potential influence of community size disparities on the methods' performance.

We generated networks using the SBM with communities of different sizes: 100, 150, and 200 nodes. We maintained the same values for $p$ and $p/q$ as before and assessed the same cases. Results are presented in Figures 2.3 and 2.4: 2.3(a)(b) two informative layers; 2.3(c)(d) three informative layers; 2.4(a)(b) two informative layers and one noisy layer; 2.4(c)(d) two informative layers and two noisy layers. Notably, the results show that the methods' performance is minimally impacted by community size variations.

### 2.5.2 Synthetic Networks via LFR

Our next test setting involves synthetic networks generated using the Lancichinetti-Fortunato-Radicchi (LFR) benchmark [43]. This benchmark offers the advantage of modeling networks with varying node degrees and community sizes, providing a more heterogeneous structure compared to the SBM. We extended the LFR benchmark to the multilayer scenario, generating separate networks for each layer while maintaining consistent parameter values. In line with [50], we considered graphs with 128 nodes distributed across 4 communities, each containing 32 nodes. The average degree was set to 16, and the maximum degree was capped at 32. The parameter $\mu$, representing the fraction of inter-community links, was varied. For the noisy layers, we ensured a single community structure and set $\mu = 0$ as suggested in the literature. Our experiments encompass different combinations of informative and noisy layers, and the results are depicted in Figures 2.5 and 2.6. In these figures, each pair of panels corresponds to accuracy and Normalized Mutual Information (NMI) scores for the following scenarios: 2.5(a)(b) two informative layers; 2.5(c)(d) three informative layers; 2.6(a)(b) two informative layers and one noisy layer; 2.6(c)(d) two informative layers and two noisy layers. The outcomes demonstrate that the proposed methods consistently achieve high accuracy and NMI scores, making them highly competitive when compared to baseline approaches. Notably, in cases where $\mu = 0.6$, the number of inter-community links surpasses that of intra-community links, leading to less distinct community structures. As a result, all methods face challenges in achieving satisfactory solutions in this particular scenario.

### 2.5.3 Real World Networks

We consider five real-world datasets frequently used for evaluation in multilayer graph clustering, [42]:

- *3sources* comprises articles from three distinct online news sources (BBC, Reuters, and The Guardian). Each news source corresponds to a layer, and the articles are manually categorized into six topical labels, including business, entertainment, health, politics, sport, and technology [57, 82].
- *BBCSport* contains sports articles labeled with five topic categories. The dataset's two layers are created by splitting each document into segments and then randomly assigning these segments to layers [82].
- *Cora* is a citation dataset involving research papers labeled with seven classes. One layer corresponds to the citation network, while the second layer is constructed based on document features, specifically the k-nearest neighbor graph [83].

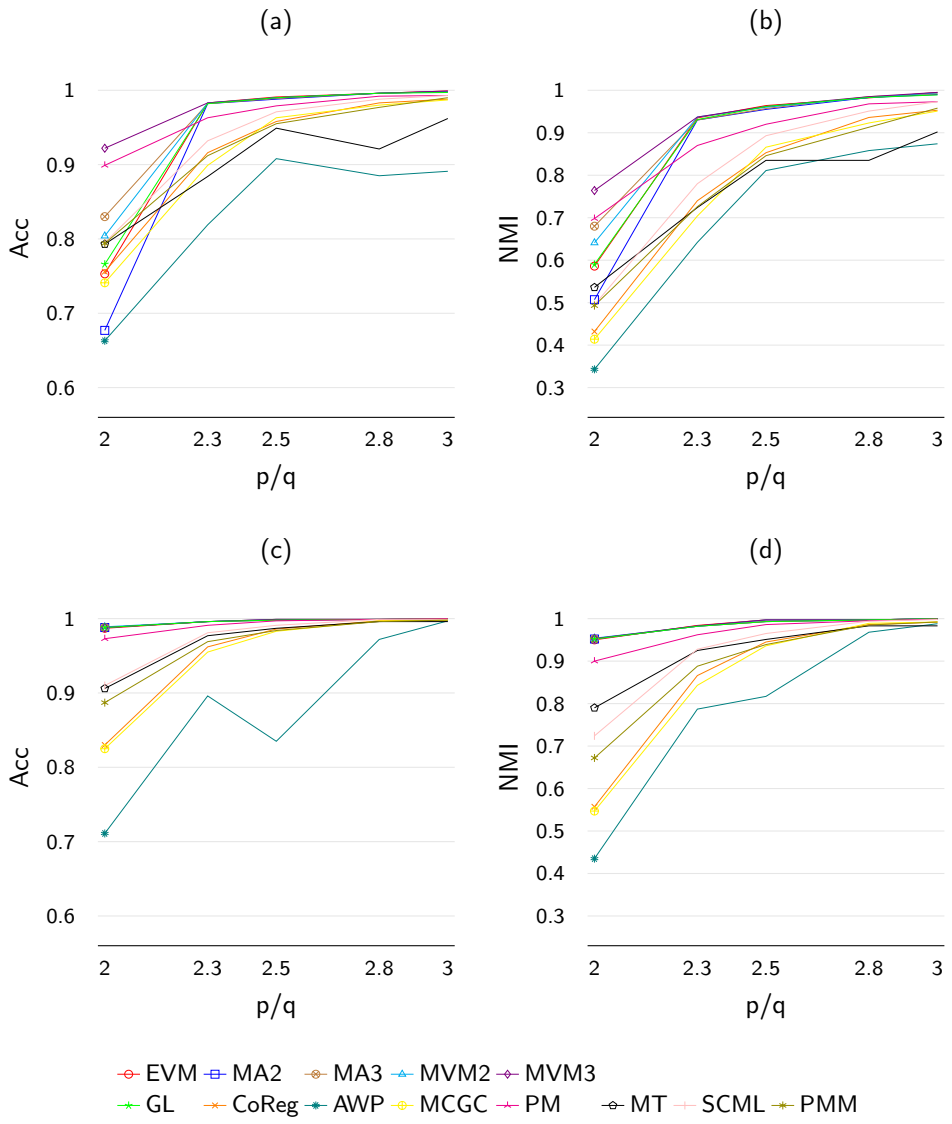


Figure 2.1: Average values of accuracy and NMI over 10 random networks sampled from SBM with equally distributed informative layers (2 layers (a)(b) and 3 layers (c)(d)) with four clusters of equal size, for $p = 0.1$ and $p/q \in \{2, 2.3, 2.5, 2.8, 3\}$.

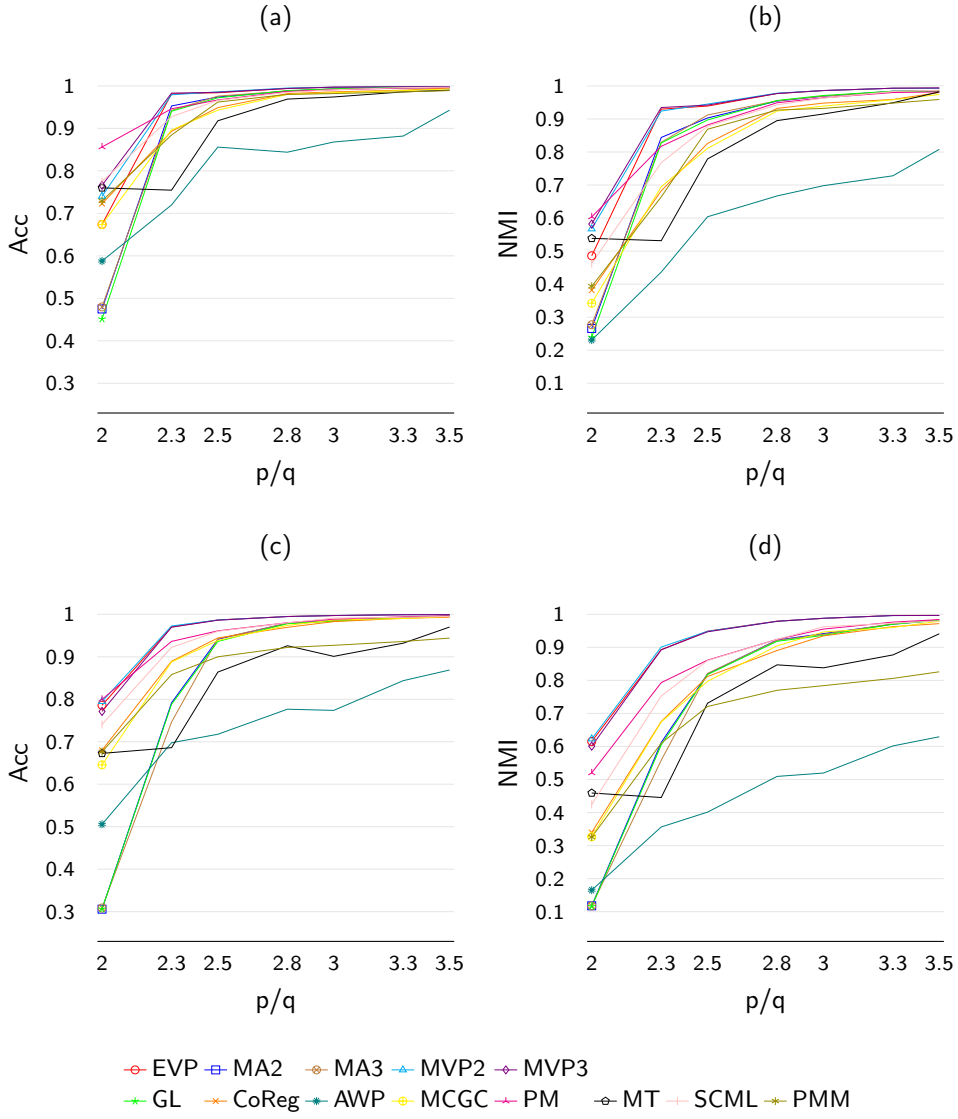


Figure 2.2: Average values of accuracy and NMI over 10 random networks sampled from SBM with both informative and noisy layers (two informative and one noisy in (a)(b); two informative and two noisy in (c)(d)). The informative layers are equally distributed SBM graphs with four clusters of equal size, $p = 0.1$ and $p/q \in \{2, 2.3, 2.5, 2.8, 3, 3.3, 3.5\}$. The noisy layers are SBM graphs with $p = q = 0.1$.

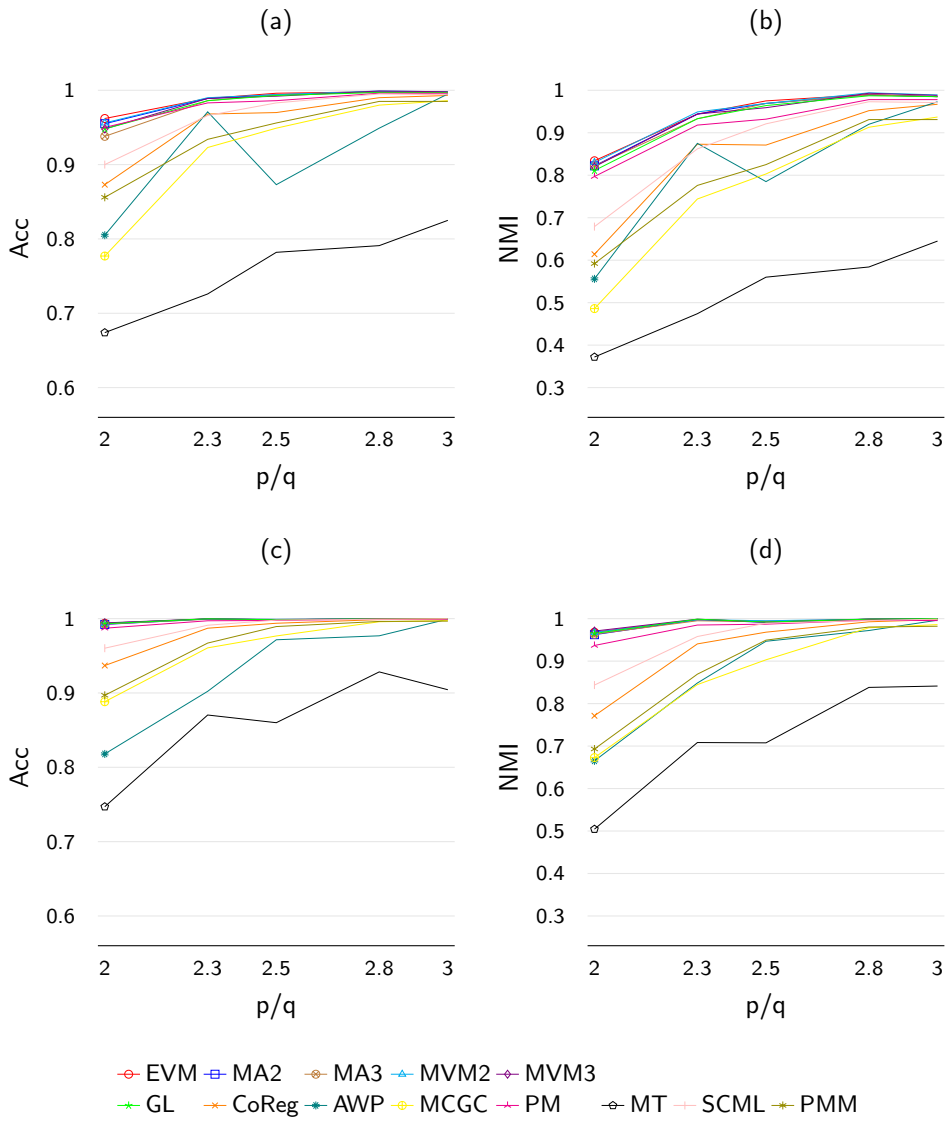


Figure 2.3: Average values of accuracy and NMI over 10 random networks sampled from SBM with equally distributed informative layers (2 layers (a)(b) and 3 layers (c)(d)) with three clusters of 100, 150 and 200 nodes, respectively, for $p = 0.1$ and $p/q \in \{2, 2.3, 2.5, 2.8, 3\}$.

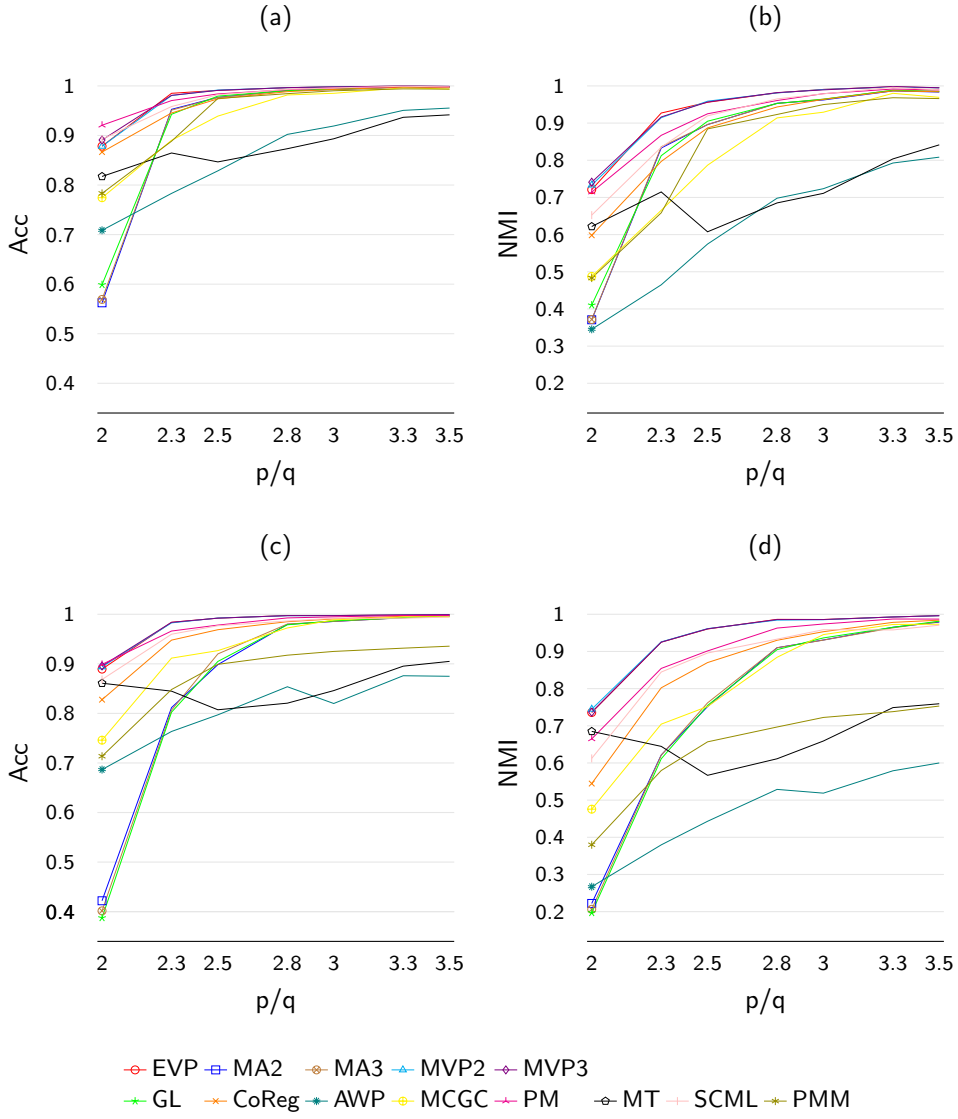


Figure 2.4: Average values of accuracy and NMI over 10 random networks sampled from SBM with both informative and noisy layers (two informative and one noisy in (a)(b); two informative and two noisy in (c)(d)). The informative layers are equally distributed SBM graphs with three clusters of 100, 150, and 200 nodes, respectively, for $p = 0.1$ and $p/q \in \{2, 2.3, 2.5, 2.8, 3, 3.3, 3.5\}$. The noisy layers are SBM graphs with $p = q = 0.1$.

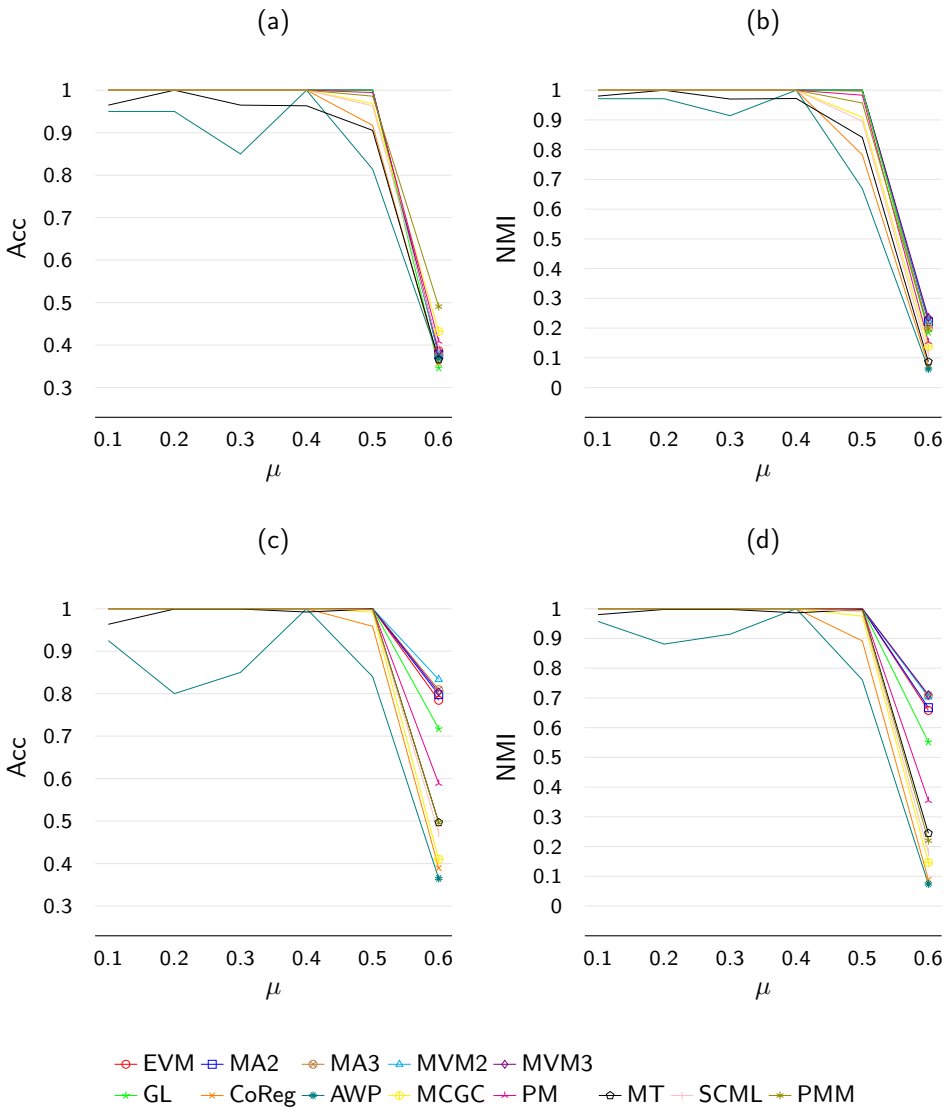


Figure 2.5: Average values of accuracy and NMI over 10 random networks sampled from LFR with equally distributed informative layers (2 layers (a)(b) and 3 layers (c)(d)), with four clusters and $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$.

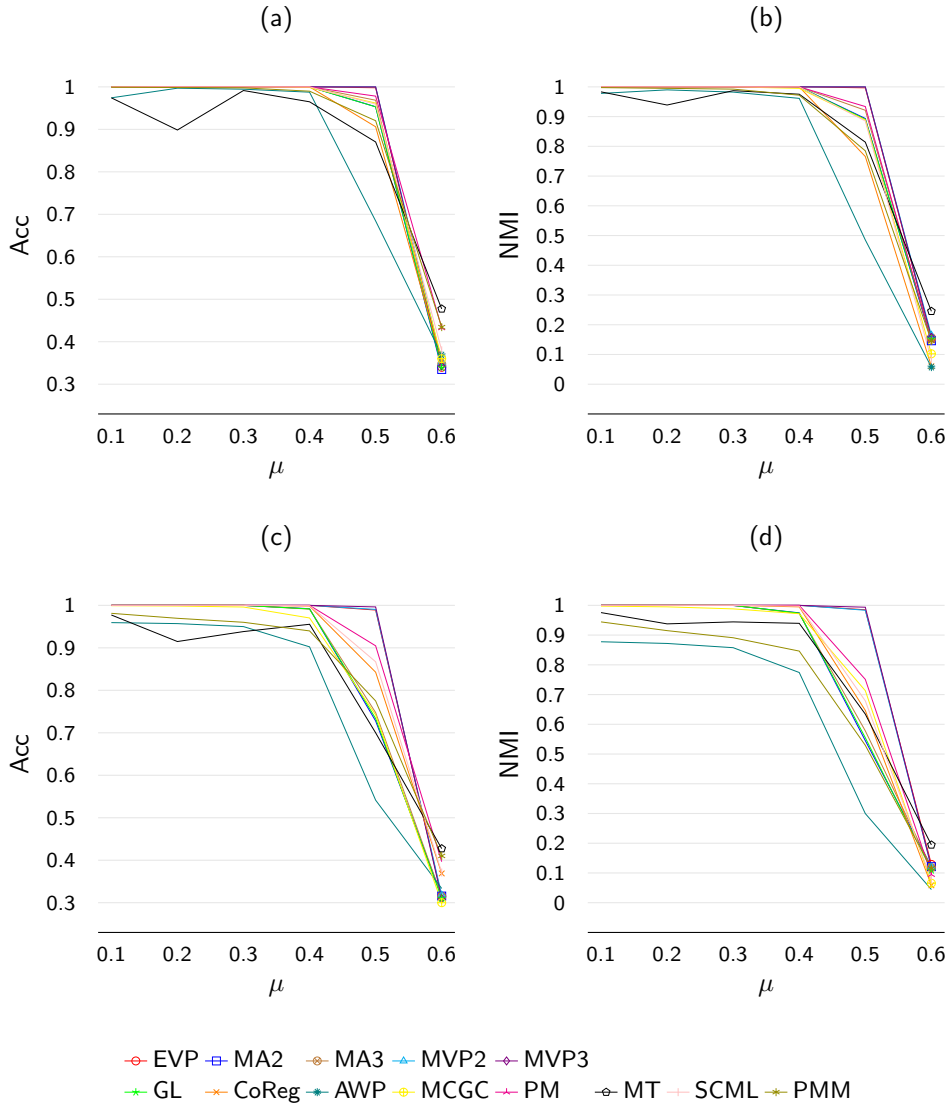


Figure 2.6: Average values of accuracy and NMI over 10 random networks sampled from LFR with both informative and noisy layers (two informative and one noisy in (a)(b); two informative and two noisy in (c)(d)). The informative layers are equally distributed LFR graphs with four clusters and $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. The noisy layers are LFR graphs with one community and $\mu = 0$.

Table 2.1: Basic statistics for the real-world datasets. For each dataset, it shows the number of nodes $N$, the number of layers $k$, the number of communities $c$, the size of each community, and, for each layer, the number of edges $|E|$, the edge density $\delta$ and the average and standard deviation of the nodes' degrees, $\langle \text{deg} \rangle$ and $\sigma$, respectively.

| | 3sources | | | | BBCSport | | | | cora | | | | UCI | | | | Wikipedia | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N$ | 169 | | | | 544 | | | | 2708 | | | | 2000 | | | | 693 | | | |
| $k$ | 3 | | | | 2 | | | | 2 | | | | 6 | | | | 2 | | | |
| $c$ | 6 | | | | 5 | | | | 7 | | | | 10 | | | | 10 | | | |
| $\lvert C_i \rvert$ | 56, 21, 11, 18, 51, 12 | | | | 62, 104, 193, 124, 61 | | | | 298, 418, 818, 426, 217, 180, 351 | | | | 200 each | | | | 34, 88, 96, 85, 65, 58, 51, 41, 71, 104 | | | |
| | $\lvert E \rvert$ | $\delta$ | $\langle \text{deg} \rangle$ | $\sigma$ | $\lvert E \rvert$ | $\delta$ | $\langle \text{deg} \rangle$ | $\sigma$ | $\lvert E \rvert$ | $\delta$ | $\langle \text{deg} \rangle$ | $\sigma$ | $\lvert E \rvert$ | $\delta$ | $\langle \text{deg} \rangle$ | $\sigma$ | $\lvert E \rvert$ | $\delta$ | $\langle \text{deg} \rangle$ | $\sigma$ |
| L1 | 1168 | 0.04 | 12.82 | 3.37 | 4075 | 0.01 | 13.98 | 5.22 | 5278 | 7e-4 | 3.90 | 5.23 | 14447 | 3e-3 | 13.45 | 3.54 | 5606 | 0.01 | 15.18 | 5.37 |
| L2 | 1223 | 0.04 | 13.47 | 4.55 | 4127 | 0.01 | 14.17 | 6.42 | 21273 | 3e-3 | 14.71 | 9.52 | 14600 | 3e-3 | 13.60 | 3.7 | 5385 | 0.01 | 28.83 | 5.37 |
| L3 | 1272 | 0.04 | 14.05 | 5.13 | - | - | - | - | - | - | - | - | 14498 | 3e-3 | 13.50 | 3.48 | - | - | - | - |
| L4 | - | - | - | - | - | - | - | - | - | - | - | - | 12729 | 3e-3 | 11.73 | 1.67 | - | - | - | - |
| L5 | - | - | - | - | - | - | - | - | - | - | - | - | 14561 | 3e-3 | 13.56 | 3.73 | - | - | - | - |
| L6 | - | - | - | - | - | - | - | - | - | - | - | - | 14421 | 3e-3 | 13.42 | 3.43 | - | - | - | - |

- *UCI* contains features of handwritten digits (0-9). The digits are represented using six distinct feature sets, forming six layers: Fourier coefficients of character shapes, profile correlations, Karhunen-Love coefficients, pixel averages, Zernike moments, and morphological features [57, 84].
- *Wikipedia* consists of Wikipedia articles categorized into ten different categories, including art & architecture, biology, geography, history, literature & theatre, media, music, royalty & nobility, sport & recreation, and warfare. Each article is assigned to one of these categories in both the text and image components, forming the two layers [85].

The layers constructed from feature sets in these datasets utilize a symmetrized $k$-nearest neighbor graph with a value of $k = 10$. This construction is based on the Pearson linear correlation between nodes, where nodes with higher correlations have smaller distances. Specifically, if $N_k(u)$ denotes the set of $k$ nodes that have highest correlation with node $u$, to each node $u$ we connect all nodes in the set

$$N_k(u) \cup \{v : u \in N_k(v)\}.$$

The main properties of the various multilayer networks are reported in Table 2.1. In general, we point out that with real data the ground truth that we take into consideration is one of the possible different existing valid partitions for the network.
Given the known community structure of the graphs, we examined both the scenarios involving informative layers and those featuring noisy layers. For the noisy case, we explored two setups. In the first, we introduced a noisy layer to accompany the informative ones. In the second setup, we considered networks with two layers: the first layer was formed by aggregating all the available layers, while the second layer consisted solely of noise. The noisy layers were created using uniform (Erdős–Rényi) random graphs, with an edge probability of $p$ taking values from the set $0.01, 0.03, 0.05$. For each specific $p$ value and each dataset, we generated 10 random instances. Within each instance, our methods were executed 10 times, each with different random initial

community orderings.

In Tables 2.2 - 2.4, we present the average accuracy and NMI scores across the various samples and random initializations. The optimal and second-best values are highlighted using gray shading, with the optimal values presented in bold font. Additionally, we calculate the average performance ratio scores $\rho_{\text{Acc}}$ and $\rho_{\text{NMI}}$ as follows: for a given metric $M_{a,d}$ (where $M_{a,d}$ represents either accuracy or NMI achieved by algorithm $a$ on dataset $d$), the *performance ratio* is defined as $r_{a,d} = M_{a,d}/\max M_{a,d}$ over all algorithms $a$. The average performance ratios $\rho_{\text{Acc}}$ and $\rho_{\text{NMI}}$ (for accuracy and NMI, respectively) are then computed by averaging the values of $r_{a,d}$ over all datasets $d$. For any algorithm, the closer the average performance ratio to 1, the better the overall performance.

The results show that in many instances, the proposed methods outperform the baseline approaches. Notably, methods that take into account both the variance and average of modularity across the layers tend to perform better. The multiobjective methods, namely MVM and MVP, consistently achieve superior results in almost all cases. This observation aligns with the more sophisticated Pareto-based approach and is in line with the findings from the synthetic data experiments. Furthermore, the last two tables underscore the robustness of the proposed methods to noise. In scenarios with noisy layers, the proposed methods, especially the multiobjective variants, exhibit very high-performance ratios. This is accompanied by consistently high accuracy and NMI scores across various datasets, highlighting the robustness of the methods in the presence of noise.

Lastly, we turn our attention to analyzing how the methods perform when faced with the addition of a larger number of noisy layers. To investigate this, we evaluate the different methods on the *3sources* dataset by incrementally adding up to 5 noisy layers. In Table 2.5, we present the average accuracy and NMI scores across samples and random initializations for the noisy scenarios. For comparison, we include values for the informative case studied in Table 2.2, which corresponds to having 0 noisy layers added. The results demonstrate that across all cases, the proposed methods consistently outperform the baseline approaches. Particularly, the multiobjective approaches, MVM and MVP, achieve the best performance in nearly all scenarios. It's important to note that we use MVM for the informative setting (as in Table 2.2) and MVP for settings with varying numbers of noisy layers. This decision is made to ensure higher variance across the layers in each of those cases. Remarkably, the multiobjective MVP approaches exhibit remarkable insensitivity to the number of noisy layers. The accuracy for these methods only marginally diminishes as the number of noisy layers increases from 1 to 5. In some instances, more noise even leads to better accuracy. This behavior can be attributed to the methods' capacity to leverage higher modularity variance, potentially considering it as a positive factor instead of a hindrance. Figure 2.7 provides insight into the computational time in seconds. In 2.7(a), the computational time for all methods is shown, while 2.7(b) omits the values for the most time-consuming method (MT) for clarity. The figures indicate that the proposed methods exhibit comparable time efficiency to the baseline methods. It is known that the computational complexity of the Louvain heuristic is $\mathcal{O}(|V|\log(|V|))$. However, it has been empirically observed in practical instances and lacks a theoretical demonstration to our knowledge [86, 87]. Studying the computational complexity of the proposed method, not only from an empirical but also from a theoretical point of view, would be an interesting future direction.

## 2.6 Conclusion

In summary, we introduced a novel method for detecting communities in multiplex graphs. This method extends the Louvain heuristic by incorporating a quality function that considers variance and employs a vector-valued modularity ascending approach based on a specialized Pareto search. We examined different versions of the method to analyze scenarios involving informative and noisy cases. In the informative case, each layer reflects the same community structure, while the noisy case involves some layers with communities and others consisting solely of noise. We performed extensive experiments comparing our method with nine baseline methods drawn from both network science and machine learning fields. Our evaluation covered synthetic networks created using the LFR and stochastic block models, as well as five real-world multilayer datasets (3sources, BBCSport, cora, UCI, Wikipedia), with informative and noisy settings considered in each case. The experimental outcomes underscore the competitive performance of our proposed method against the baseline approaches. Specifically, the multiobjective approach incorporating modularity variance demonstrated superior performance across a wide range of scenarios.

Table 2.2: Real-world dataset setting one: no noisy layers. Average accuracy, NMI, and performance ratio score over 10 random initializations. All layers are informative. The best and second-best values are highlighted with gray boxes.

| | **3sources** | | **BBCSport** | | **cora** | | **UCI** | | **Wikipedia** | | **Perf. Ratios** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | $\rho_{\text{Acc}}$ | $\rho_{\text{NMI}}$ |
| EVM | 0.876 | 0.789 | 0.833 | 0.798 | **0.617** | **0.537** | 0.882 | 0.921 | 0.548 | 0.520 | 0.951 | **0.966** |
| MA2 | 0.858 | 0.749 | 0.899 | 0.825 | 0.407 | 0.434 | 0.753 | 0.862 | 0.525 | 0.521 | 0.863 | 0.912 |
| MA3 | 0.876 | 0.789 | 0.596 | 0.731 | 0.425 | 0.428 | 0.876 | 0.910 | 0.558 | 0.546 | 0.838 | 0.920 |
| MVM2 | **0.888** | **0.812** | 0.844 | 0.784 | 0.597 | 0.514 | 0.883 | **0.925** | 0.544 | 0.508 | 0.952 | 0.959 |
| MVM3 | **0.888** | **0.812** | 0.915 | **0.851** | 0.603 | 0.502 | 0.883 | **0.925** | 0.530 | 0.504 | **0.966** | 0.965 |
| GL | 0.858 | 0.749 | 0.748 | 0.753 | 0.523 | 0.520 | 0.877 | 0.913 | 0.556 | 0.544 | 0.904 | 0.947 |
| CoReg | 0.651 | 0.658 | 0.858 | 0.617 | 0.530 | 0.380 | **0.958** | 0.911 | 0.522 | 0.445 | 0.905 | 0.840 |
| AWP | 0.686 | 0.662 | 0.616 | 0.722 | 0.534 | 0.293 | 0.869 | 0.891 | 0.462 | 0.332 | 0.843 | 0.758 |
| MCGC | 0.544 | 0.595 | **0.919** | 0.795 | 0.273 | 0.034 | 0.898 | 0.855 | 0.221 | 0.135 | 0.676 | 0.580 |
| PM | 0.734 | 0.707 | 0.778 | 0.690 | 0.551 | 0.456 | 0.876 | 0.879 | **0.569** | **0.560** | 0.896 | 0.892 |
| MT | 0.651 | 0.610 | 0.748 | 0.656 | 0.453 | 0.289 | 0.553 | 0.666 | 0.342 | 0.229 | 0.692 | 0.638 |
| SCML | 0.686 | 0.661 | 0.864 | 0.767 | 0.616 | 0.447 | 0.862 | 0.872 | 0.560 | 0.535 | 0.919 | 0.889 |
| PMM | 0.692 | 0.666 | 0.518 | 0.514 | 0.336 | 0.238 | 0.638 | 0.662 | 0.417 | 0.302 | 0.658 | 0.625 |
| IM | 0.539 | 0.624 | 0.531 | 0.401 | 0.431 | 0.477 | 0.721 | 0.761 | 0.123 | 0.11 | 0.570 | 0.630 |

Table 2.3: Real-world dataset setting two: informative layers plus one noisy layer. Average accuracy, NMI, and performance ratio scores over 10 random initializations and 10 random edge probabilities $p \in \{0.01, 0.03, 0.05\}$ for the noisy layer. The best and second-best values are highlighted with gray boxes.

| | 3sources | | BBCSport | | cora | | UCI | | Wikipedia | | Perf. Ratios | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | $\rho_{\text{Acc}}$ | $\rho_{\text{NMI}}$ |
| EVP | 0.703 | 0.649 | 0.825 | **0.797** | 0.541 | 0.517 | 0.880 | 0.916 | 0.577 | **0.556** | 0.964 | 0.988 |
| MA2 | 0.692 | 0.609 | 0.790 | 0.761 | 0.551 | 0.519 | 0.881 | 0.920 | 0.558 | 0.518 | 0.950 | 0.955 |
| MA3 | 0.683 | 0.612 | 0.789 | 0.758 | 0.549 | 0.519 | 0.881 | 0.921 | 0.559 | 0.518 | 0.947 | 0.955 |
| MVP2 | 0.717 | 0.668 | 0.828 | **0.797** | 0.543 | 0.518 | 0.881 | 0.920 | 0.578 | 0.555 | 0.970 | 0.994 |
| MVP3 | **0.730** | **0.677** | 0.817 | 0.792 | 0.543 | 0.520 | 0.881 | 0.919 | **0.579** | **0.556** | **0.971** | **0.996** |
| GL | 0.678 | 0.607 | 0.777 | 0.754 | **0.555** | **0.521** | 0.881 | 0.920 | 0.561 | 0.520 | 0.945 | 0.953 |
| CoReg | 0.652 | 0.650 | **0.849** | 0.753 | 0.407 | 0.191 | 0.957 | 0.912 | 0.435 | 0.324 | 0.874 | 0.767 |
| AWP | 0.658 | 0.602 | 0.737 | 0.593 | 0.411 | 0.123 | 0.924 | 0.897 | 0.410 | 0.266 | 0.835 | 0.662 |
| MCGC | 0.546 | 0.585 | 0.812 | 0.694 | 0.303 | 0.005 | 0.804 | 0.816 | 0.201 | 0.107 | 0.686 | 0.563 |
| PM | 0.714 | 0.658 | 0.730 | 0.645 | 0.548 | 0.444 | 0.876 | 0.880 | 0.568 | **0.556** | 0.943 | 0.916 |
| MT | 0.624 | 0.627 | 0.519 | 0.355 | 0.187 | 0.011 | 0.660 | 0.723 | 0.170 | 0.051 | 0.556 | 0.453 |
| SCML | 0.639 | 0.593 | 0.772 | 0.604 | 0.222 | 0.028 | **0.964** | **0.930** | 0.185 | 0.057 | 0.701 | 0.558 |
| PMM | 0.538 | 0.508 | 0.387 | 0.167 | 0.255 | 0.060 | 0.667 | 0.677 | 0.172 | 0.047 | 0.528 | 0.377 |
| IM | 0.538 | 0.624 | 0.531 | 0.401 | 0.431 | 0.477 | 0.721 | 0.761 | 0.123 | 0.110 | 0.620 | 0.671 |

Table 2.4: Real-world dataset setting three: one aggregated informative layer plus one noisy layer. Average accuracy, NMI, and performance ratio scores over 10 random initializations and 10 random edge probabilities $p \in \{0.01, 0.03, 0.05\}$ for the noisy layer. The best and second-best values are highlighted with gray boxes.

| | 3sources | | BBCSport | | cora | | UCI | | Wikipedia | | Perf. Ratios | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | $\rho_{\text{Acc}}$ | $\rho_{\text{NMI}}$ |
| EVP | 0.655 | 0.575 | 0.886 | 0.799 | 0.550 | **0.432** | 0.869 | 0.903 | 0.556 | 0.526 | 0.938 | 0.943 |
| MA2 | 0.348 | 0.217 | 0.664 | 0.555 | 0.541 | 0.418 | 0.853 | 0.890 | 0.431 | 0.361 | 0.761 | 0.717 |
| MA3 | 0.332 | 0.207 | 0.659 | 0.550 | 0.537 | 0.416 | 0.858 | 0.893 | 0.427 | 0.359 | 0.754 | 0.711 |
| MVP2 | 0.744 | 0.675 | **0.914** | 0.826 | 0.546 | 0.430 | 0.872 | 0.905 | 0.564 | 0.538 | **0.969** | 0.980 |
| MVP3 | 0.754 | 0.689 | **0.914** | **0.828** | 0.544 | 0.431 | **0.873** | **0.906** | **0.566** | **0.540** | **0.969** | **0.984** |
| GL | 0.327 | 0.203 | 0.664 | 0.549 | 0.537 | 0.418 | 0.866 | 0.898 | 0.427 | 0.360 | 0.756 | 0.713 |
| CoReg | 0.566 | 0.436 | 0.608 | 0.338 | 0.435 | 0.190 | 0.761 | 0.642 | 0.446 | 0.305 | 0.753 | 0.542 |
| AWP | 0.549 | 0.416 | 0.644 | 0.376 | 0.444 | 0.179 | 0.750 | 0.622 | 0.439 | 0.292 | 0.754 | 0.531 |
| MCGC | 0.512 | 0.479 | 0.682 | 0.480 | 0.276 | 0.020 | 0.702 | 0.691 | 0.347 | 0.291 | 0.654 | 0.514 |
| PM | 0.512 | 0.363 | 0.729 | 0.658 | 0.569 | 0.414 | 0.834 | 0.828 | 0.457 | 0.360 | 0.831 | 0.764 |
| MT | 0.693 | 0.614 | 0.729 | 0.614 | 0.272 | 0.113 | 0.634 | 0.699 | 0.406 | 0.306 | 0.714 | 0.534 |
| SCML | 0.514 | 0.410 | 0.797 | 0.661 | **0.600** | 0.416 | 0.846 | 0.835 | 0.479 | 0.365 | 0.867 | 0.783 |
| PMM | 0.440 | 0.304 | 0.424 | 0.206 | 0.311 | 0.108 | 0.641 | 0.535 | 0.386 | 0.249 | 0.591 | 0.392 |
| IM | **0.793** | **0.742** | 0.730 | 0.751 | 0.323 | 0.376 | 0.199 | 0.338 | 0.496 | 0.532 | 0.687 | 0.827 |

**Algorithm 1** Louvain Multiobjective Method

**Input** $G$ multiplex graph, $F$ scalar quality function
**Output** final partition

$L$ initialized with node-based partition, the corresponding modularity vector $Q$ and the value of $F$
Set `terminate` = false
**repeat**
  Set `updateL` = true
  **repeat**
    **for all** node $i$ of $G$ **do**
      **for all** partition $C$ in list $L$ **do**
        places $i$ in every neighboring community which yields a positive increment of $F$. If the corresponding modularity vector $Q$ is not Pareto-dominated by any of the modularity vectors in $L$, insert in $L$ the vector $Q$, the corresponding partition, and the $F$ value. Delete from $L$ all terms corresponding to modularity vectors that are dominated by $Q$.
      **end for**
    **end for**
    If $L$ is longer than $h$, cut it to length $h$ using $F$
    **if** $L$ does not change **then**
      `updateL` = false
    **end if**
  **until** ( `updateL` == false)
  Consider the partition of the list $L$ which maximizes the function $F$ gain
  **if** $L$ has changed **then**
    $G$ = reduced graph where each community of the selected partition is a node
  **else**
    Set `terminate` = true
  **end if**
**until** (`terminate` == true)

Table 2.5: Real-world dataset setting four: 3-sources dataset with an increasing number of noisy layers (from 0 to 5). Average accuracy and NMI over 10 random initializations (for the noisy cases) and 10 random edge probabilities $p \in \{0.01, 0.03, 0.05\}$ for the noisy layers. The best and second-best values are highlighted with gray boxes.

| | **0** | | **1** | | **2** | | **3** | | **4** | | **5** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI | Acc | NMI |
| EVM/P | 0.876 | 0.789 | 0.703 | 0.649 | 0.729 | 0.677 | 0.735 | 0.688 | 0.767 | 0.708 | 0.777 | 0.717 |
| MA2 | 0.858 | 0.749 | 0.692 | 0.609 | 0.615 | 0.538 | 0.577 | 0.495 | 0.526 | 0.442 | 0.503 | 0.426 |
| MA3 | 0.876 | 0.789 | 0.683 | 0.612 | 0.602 | 0.529 | 0.576 | 0.494 | 0.535 | 0.449 | 0.503 | 0.420 |
| MVM/P2 | **0.888** | **0.812** | 0.717 | 0.668 | 0.748 | **0.693** | **0.752** | **0.700** | 0.778 | 0.718 | 0.787 | **0.721** |
| MVM/P3 | **0.888** | **0.812** | **0.730** | **0.677** | **0.751** | 0.692 | 0.738 | 0.699 | **0.779** | **0.722** | **0.790** | 0.720 |
| GL | 0.858 | 0.749 | 0.678 | 0.607 | 0.615 | 0.541 | 0.571 | 0.492 | 0.508 | 0.443 | 0.506 | 0.424 |
| CoReg | 0.651 | 0.658 | 0.652 | 0.650 | 0.650 | 0.645 | 0.654 | 0.645 | 0.650 | 0.637 | 0.645 | 0.631 |
| AWP | 0.686 | 0.662 | 0.658 | 0.602 | 0.664 | 0.580 | 0.632 | 0.539 | 0.622 | 0.521 | 0.599 | 0.462 |
| MCGC | 0.544 | 0.595 | 0.546 | 0.585 | 0.541 | 0.577 | 0.550 | 0.575 | 0.538 | 0.551 | 0.486 | 0.503 |
| PM | 0.734 | 0.707 | 0.714 | 0.658 | 0.671 | 0.609 | 0.675 | 0.592 | 0.671 | 0.583 | 0.640 | 0.543 |
| MT | 0.651 | 0.610 | 0.624 | 0.627 | 0.629 | 0.619 | 0.650 | 0.654 | 0.642 | 0.632 | 0.665 | 0.633 |
| SCML | 0.686 | 0.661 | 0.639 | 0.593 | 0.658 | 0.641 | 0.666 | 0.632 | 0.652 | 0.620 | 0.649 | 0.596 |
| PMM | 0.692 | 0.666 | 0.538 | 0.508 | 0.649 | 0.608 | 0.637 | 0.603 | 0.581 | 0.555 | 0.570 | 0.563 |
| IM | 0.539 | 0.624 | 0.538 | 0.624 | 0.538 | 0.650 | 0.574 | 0.635 | 0.580 | 0.617 | 0.527 | 0.610 |

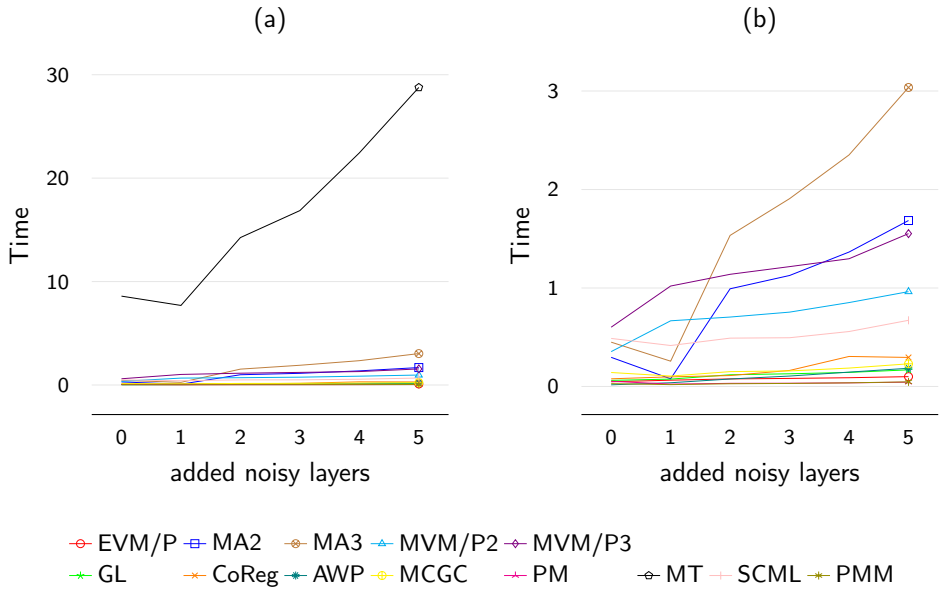


Figure 2.7: Computational time of different methods on the 3-sources dataset, with an increasing number of noisy layers (from 0 to 5). In (a) all methods' runtimes are shown, in (b) excluding MT.

# Chapter 3

# Learning the right layer: a data-driven layer-aggregation strategy for semi-supervised learning on multilayer graphs

As pointed out in the previous chapter, clustering (or community detection) on multilayer graphs poses several additional complications concerning standard graphs as different layers may be characterized by different structures and types of information. A significant challenge lies in determining the degree to which each layer contributes to the assignment of clusters. This issue is essential to fully harness the potential of the multilayer structure and enhance classification outcomes beyond what can be achieved using individual layers or their union. However, making an informed a-priori assessment about the clustering information content of the layers can be very complicated. In this study, we operate within a semi-supervised learning context, where the classes of a limited portion of nodes are initially supplied. We introduce a Laplacian-regularized model designed to acquire an optimal nonlinear combination of the diverse layers based on the provided input labels. The learning algorithm relies on a Frank-Wolfe optimization approach that incorporates an inexact gradient, coupled with a modified Label Propagation iteration. We offer a comprehensive convergence analysis of the algorithm and perform experiments using both synthetic and real-world datasets. These experiments demonstrate that our proposed method exhibits favorable outcomes when compared to a range of baseline approaches. Moreover, our method surpasses the performance of each layer when utilized in isolation. In the following chapters, we rely on the assumption that the graph correlates with the given labels, therefore giving in input a graph should always improve the classification.

## 3.1 Introduction

Graph-based Semi-Supervised Learning (GSSL) has emerged as a powerful approach for inferring labels of unlabeled nodes in various applications with limited labeled data [12]. In GSSL, the

underlying graph structure plays a crucial role, especially when the labeled data is scarce and node features are unavailable. The Laplacian regularization formulation, based on the smoothness assumption, has proven successful. It minimizes a loss function that enforces consistency with both initial labels and the graph structure. The solution can be interpreted as a new node embedding used for classification. Initially proposed by Zhou et al. [88], Belkin et al. [89], and Yang et al., [90] this approach has seen extensive exploration in the machine learning field [14, 91, 92, 93, 94, 95, 96]. While traditional graphs have proven to be a valuable tool for representing data interactions, many real-world scenarios involve intricate systems characterized by multiple types of interactions or relationships occurring concurrently. In these cases, multilayer graphs offer a more proper representation [97, 98, 99]. For instance, consider transportation systems that encompass various modes like trains, buses, and more [100]. In scientific domains, data may involve co-authorship, co-citation, and other connections, as well as affiliations based on topics or institutions [101]. In social environments, individuals engage in different interactions like friendships, acquaintanceships, or business interactions [102]. Similarly, biological systems exhibit diverse relationships among their components [103, 104]. Multilayer graphs offer an established framework for representing these complex data scenarios, allowing for the direct modeling of these multifaceted interactions. This has yielded improvements in various domains, spanning both network science and machine learning applications [9, 105].

Despite their potential usefulness and power, multilayer graph models introduce a fundamental challenge. A multilayer graph can consist of numerous layers, each describing different properties. However, it is not immediately clear whether all these layers contribute to effectively classifying the nodes. Some layers might carry the same or complementary clustering information, while others could be more informative than others. Additionally, certain layers might be mere noise, carrying no meaningful information about node clusters. Determining the situation that best characterizes a given dataset and identifying the most and least informative layers presents a significant and complex challenge. Constructing these networks in various applications isn't straightforward, and making an informed assessment about the presence of noise, the distinct types of layer structures, and the general information content related to clustering can be highly intricate [103, 15].

In recent years, numerous Graph-based Semi-Supervised Learning (GSSL) algorithms have been developed specifically for multilayer networks. Many of these methods propose aggregating the information from different layers into a single-layer graph by utilizing various aggregation functions such as sum, min, max, and so on. The objective is typically to give more weight to the most informative layers, aiming to enhance the overall classification performance [59, 14, 16, 17, 106, 107, 108, 109, 18, 110, 111]. However, a notable limitation of many existing methods is that their proposed aggregation strategies are often tailored to specific scenarios. Consequently, they usually require prior knowledge about the type of clustering information carried by the layers and the entire dataset. Furthermore, although certain methods perform well across multiple settings, such as those presented in [14, 112], they fail to provide insights into whether specific layers hold more informative value than others, whether the information is complementary, or whether certain layers are essentially uninformative noise.

In this study, we introduce a novel approach based on a Laplacian-regularized model that effectively learns the optimal combination of various layers using the available labeled data.

Our model employs nonlinear generalized mean functions to aggregate the layers, encompassing various aggregation functions that have been employed in previous literature as special cases. To determine the optimal aggregation parameters, we employ a specially designed bi-level inexact-gradient optimization strategy. We provide an exhaustive analysis of the convergence properties of this optimization method and extensively validate its performance through numerical experiments. Our experimental evaluation encompasses both synthetic and real-world datasets. The results demonstrate that our proposed Graph-based Semi-Supervised Learning (GSSL) method for multilayer networks showcases a remarkable competitive advantage in terms of classification accuracy when compared to existing alternatives. Furthermore, our approach is effective in identifying the layers that contribute the most relevant information, as well as those that are less informative, thus allowing us to highlight complementary information across the layers.

## 3.2 Related work

In this section, we will offer a brief overview of existing semi-supervised learning algorithms tailored for multilayer graphs. Our focus will primarily be on methods specifically designed to operate on featureless multilayer networks.

Similar to our model, various approaches are centered around learning an optimal set of parameters within the aggregation function of the multilayer graph. These methods make use of a multilayer adaptation of the Laplacian-regularization framework often seen in graph-based semi-supervised learning. For instance, Tsuda et al. [16] introduced a technique for protein classification that operates on multiple protein networks. They aggregate the multilayer graph through a weighted linear combination, with the weights determined through a variational min-max procedure that aims to minimize the worst-case graph consistency function. The outcome of their method demonstrates robustness in the presence of noisy or irrelevant layers. Likewise, Argyriou et al. [17] present an approach where they determine an optimal linear combination of Laplacian kernels. This is achieved by solving an expanded regularization problem on the multilayer graph, which involves minimizing both the dataset and the collection of graph kernels. Furthermore, Zhou and Burges [59] demonstrate that the resulting convex combination of graph Laplacians serves as an extension of the normalized cut function to the context of multilayer networks. An alternative perspective is introduced by Nie et al. [108], where they offer a different formulation. In their method, the optimal weights for the weighted linear combination of layers' Laplacians are implicitly determined through a dual Lagrangian formulation. The outcome is a parameter-free technique for finding optimal layer weights. Lastly, Karasuyama and Mamitsuka [107] present an efficient strategy for the linear aggregation of multiple graphs under the Laplacian regularization framework. They achieve this by employing an approach involving alternate optimization through label propagation, combined with sparse integration. In contrast to the approach we present here, all of the aforementioned methods rely on linear aggregation functions (such as convex combinations), where the optimal weights are determined based on model-driven considerations, often aimed at addressing worst-case scenarios. Incorporating nonlinear layer aggregation functions, such as max, min, and their generalizations, introduces additional modeling flexibility. Mercado et al. [14] employ a Log-Euclidean matrix function formulation to define a generalized power mean of graph Laplacians, which includes arithmetic, geometric, and harmonic means as special

cases. They propose a regularizer based on a one-parameter family of matrix means. This approach is further refined and enhanced in Bergermann et al. [111], using diffuse interface methods and efficient matrix-vector products. Although this approach achieves competitive performance, it necessitates extensive parameter exploration for the chosen mean, which can become computationally intensive. In other works such as [105, 109, 113], entrywise minimum and maximum aggregation functions are employed in the multilink model.

Diverging from the traditional Laplacian regularization framework, there are some alternative approaches in the literature. Eswaran et al. [114] introduce a method that utilizes fast belief propagation on heterogeneous graphs, where nodes belong to distinct types. This approach is tailored to capture the intricate relationships present in such heterogeneous graphs. Gujral and Papalexakis [112] present a parameter-free algorithm that leverages tensor factorization techniques. This algorithm is designed to uncover both overlapping and non-overlapping communities within multilayer networks.

While geometric deep learning and graph neural networks are widely used in the single-layer setting, their extension to the multilayer context is still relatively limited and often focuses on scenarios where multilayer graphs contain connections between different layers (inter-layer edges). Here are a couple of examples. Among the available ones, Ghorbani et al. [115] is based on an extension of the graph convolutional filter by Welling & Kipf [92], while Grassia et al. [116] proposes a graph neural network whose graph filter is a parametric aggregated Laplacian, parametrized in terms of an MLP.

## 3.3 Bilevel optimization

In this section, we provide a basic overview of bilevel optimization. Bilevel programming involves solving an upper optimization problem (UOP) while taking into account the optimality conditions of a lower optimization problem (LOP). This framework is used when optimizing one problem depends on the solution of another. Bilevel optimization has found applications in various fields, such as engineering, economics, and machine learning [117, 118, 119, 120]. In the context of hyperparameter optimization, bilevel programming arises when tuning both model parameters (usually learned from the training data) and hyperparameters (which control the learning process). The available data is typically split into a training set for parameter tuning and a validation set for hyperparameter tuning. The goal is to find the best combination of model parameters and hyperparameters that minimize the validation error or some other performance metric. This creates a hierarchical optimization structure where the inner optimization involves selecting the optimal model parameters given fixed hyperparameters, and the outer optimization involves selecting the best hyperparameters.

Bilevel optimization in hyperparameter tuning is challenging due to the nested nature of the optimization problems and the potential non-convexity of the search space. Various strategies, including grid search, random search, Bayesian optimization, and gradient-based methods, have been proposed to address this challenge and efficiently search for the optimal combination of model parameters and hyperparameters.

A bilevel optimization problem has the following general form:

$$\min_{\xi,\psi} \zeta(\xi,\psi) \quad s.t. \quad \psi \in \operatorname*{argmin}_{\psi'} \upsilon(\xi,\psi') \tag{3.1}$$

Here, the goal is to minimize an outer objective function $\zeta$ by considering both inner and outer variables $\xi$ and $\psi$. The inner variables $\psi$ are subject to a constraint that requires them to minimize an inner objective function $\upsilon$ concerning $\psi'$. The nested structure of this problem introduces complexity and makes bilevel optimization challenging.

In our context, the available data are the input labels which are divided into a training set and a test set. In the LOP we deal with the standard GSSL, while in the UOP we aim to tune the weights to give to the different layers. To solve the bilevel formulation, we will calculate the explicit expression of the solution of the LOP. We will substitute it in the UOP and we will solve it using a zeroth order Frank Wolfe algorithm since the calculation of the hypergradient can be computationally demanding.

## 3.4 Learning the most relevant layers

### Problem set-up

We consider a multiplex (alternatively known as a multicolor, or multiview graph) with $k$ layers $G_1,\dots,G_k$, each being a weighted undirected graph $G_s = (V, E_s, w_s)$ with $V = \{1,\dots,|V|\}$, $E_s \subseteq V \times V$, and $w_s : E_s \to \mathbb{R}_+$. To each layer correspond a weighted adjacency matrix $A^{(s)}$, whose entries $A^{(s)}_{ij} = w_s(ij) > 0$ represent the strength of the tie between $i$ and $j$, if $ij \in E$, and $A^{(s)}_{ij} = 0$ if $ij \notin E$.
Using the terminology proposed in [9], we assume $G$ consists of a set $C = \{C_1,\dots,C_{|C|}\}$ of communities (or labels) that is total (i.e., every node belongs to at least one $C_j \in C$), node-disjoint (i.e., no node belongs to more than one cluster), and pillar (i.e., each node belongs to the same community across the layers). Further, we assume that for each $C_r \in C$ we are given a set of input known labels $O_r \subseteq V$ which are one-hot encoded into the matrix $Y \in \mathbb{R}^{n\times m}$, with $Y_{ir} = 1$ if $i \in C_r$, and $Y_{ir} = 0$ otherwise.
The goal is to learn the unknown labels. In our setting, we assume no node feature is available. In other words, we focus on the setting in which one has access only to topological information about the graph structure and has some input knowledge about the community assignment of some nodes. This is a common setting in e.g. network and social science applications [9].

### Generalized mean adjacency model

To learn a classifier that effectively takes into account the multilayer graph structure, we design a nonlinear aggregation strategy that optimally learns the aggregation parameters and computes a classifier based on a multilayer Laplacian-regularization model. To this end, we first briefly review the standard Laplacian-regularization model for single-layer graphs.

When dealing with a single-layer graph, a common strategy to ensure both local and global consistency with the provided input labels and the graph structure is to minimize the following GSSL loss function, which incorporates Laplacian regularization:

$$\varphi(X) := \|X - Y\|_F^2 + \tfrac{\lambda}{2}\mathrm{Tr}(X^\top \mathcal{L} X) \tag{3.2}$$

over all $X \in \mathbb{R}^{|V|\times|C|}$. Here $\mathcal{L} = D - A$ is the Laplacian matrix of the single-layer graph at hand, with $D = \mathrm{diag}(A\mathbf{1})$ the diagonal matrix of the (weighted) degrees. Simple linear algebra passages show that the obtained solution $X^* = \operatorname{argmin} \varphi(X)$ is entrywise positive, thus the entries $X^*_{ir}$ can be interpreted as a classifier that provides a score quantifying the likelihood that node $i$ belongs to the community $C_r$. Hence, we assign to each node $i$ the label $C_{r^*}$, with $r^* = \operatorname{argmax}_j X^*_{ij}$. Note that, if $y^{(r)}$ is the $r$-th column of $Y$, with one-hot information about the input labels in $O_r$, then one can equivalently write $\varphi(X) = \sum_{r=1}^{|C|} \varphi_r(x^{(r)})$, where $x^{(r)}$ are the columns of $X$ and

$$\varphi_r(x) = \sum_{i=1}^{N} |x_i - y_i^{(r)}|^2 + \frac{\lambda}{2}\sum_{i,j=1}^{N} A_{ij}\,(x_i - x_j)^2 \ . \tag{3.3}$$

As the $\varphi_r$ are independent of each other, in this single-graph setting minimizing $\varphi$ is equivalent to minimizing each $\varphi_r$ individually.

When we are given $k$ layers, imposing smoothness concerning the edge structure is more challenging. As the communities are assumed to be consistent across the layers, a standard approach is to tackle the problem after layer aggregation. If the aggregating function is linear, this boils down to choosing a set of weights $\beta_s > 0$ with $\sum_s \beta_s = 1$ and replace $A$ in (3.2) or (3.3) with $A^{lin} = \sum_s \beta_s A^{(s)}$. This approach has been widely explored and is considered for example in [16, 17, 18]. As the cost function in (3.2) is quadratic, this is equivalent to considering the classifier $X^* = \sum_s \beta_s X^*_s$, with $X^*_s$ solution to (3.2) for $A = A^{(s)}$. Another possibility is to consider "nonlinear aggregations" [14, 109, 111, 113].

For example, using the concept of multilink [105], Mondragon et al. [109] replace the multilayer network with a single-layer graph with adjacency matrix with entries $A^{max}_{ij} = \max_s A^{(s)}_{ij}$ or $A^{min}_{ij} = \min_s A^{(s)}_{ij}$. The maximum-based model assumes that an edge between nodes $i$ and $j$ exists in the aggregated graph if at least one edge connecting them is present in any of the layers. On the other hand, the minimum-based model keeps edges in the aggregated graph only if they are present in all layers. These approaches work well when either all the edges in every layer are reliable and can be trusted, as in the case of the maximum-based model, or when no individual layer can be fully trusted, leading to the minimum-based approach. However, in real-world scenarios, the layers might contain distinct and complementary community information, while some layers could be considered "noisy," indicating that they might provide limited or no meaningful information about the underlying communities [14, 15].

The introduction of parameters $\beta_s$ in the linear model enables us to assign varying weights to the layers, which can be particularly useful if information about their community-related content is known. However, determining these weights based on prior knowledge can be challenging, especially when assessing the presence of noise or different types of community structures across the layers [15]. To address this challenge, we propose a nonlinear aggregation approach

Table 3.1: Entries of the generalized mean adjacency matrix $A(\theta)$, for particular choices of the parameters $\theta = (\alpha, \beta) \in \mathbb{R}^{K+1}$.

| $\alpha \to -\infty$ | $\alpha = -1,\ \beta_k = 1/K$ | $\alpha \to 0,\ \beta_k = 1/K$ | $\alpha = 1,\ \beta_k = 1/K$ | $\alpha \to +\infty$ |
|---|---|---|---|---|
| Minimum (MIN) | Harmonic (HARM) | Geometric (GEO) | Arithmetic (ARIT) | Maximum (MAX) |
| $\min_{k=1,\dots,K} A_{ij}^{(k)}$ | $\left(\frac{1}{K}\sum_{k=1}^{K}\frac{1}{A_{ij}^{(k)}}\right)^{-1}$ | $\left(\prod_{k=1}^{K} A_{ij}^{(k)}\right)^{1/K}$ | $\frac{1}{K}\sum_{k=1}^{K} A_{ij}^{(k)}$ | $\max_{k=1,\dots,K} A_{ij}^{(k)}$ |

that encompasses linear, maximum, and minimum aggregation strategies as specific instances. Moreover, this approach dynamically learns the optimal aggregation parameters directly from the provided input labels. This way, the method can adapt to the specific characteristics of the data and the layer relationships, without requiring explicit prior knowledge about the layers' noise levels or community structures.

Both the linear combination $A^{lin}$ and the minimum, maximum matrices $A^{min}$, $A^{max}$ can be seen as particular cases of more general nonlinear aggregations based on the *generalized mean adjacency matrix* $A(\theta)$, entrywise defined as

$$A(\theta)_{ij} = \left(\sum_{s=1}^{k} \beta_s \big(A_{ij}^{(s)}\big)^{\alpha}\right)^{1/\alpha}, \tag{3.4}$$

where the parameters $\theta = (\alpha, \beta)$ are such that $\sum_s \beta_s = 1$, $\beta_s > 0$ as above, and $\alpha \in \mathbb{R}$. In fact, $A^{lin} = A((1, \beta))$, and $A^{min} = \lim_{\alpha\to-\infty} A((\alpha, \beta))$, $A^{max} = \lim_{\alpha\to+\infty} A((\alpha, \beta))$. As illustrated in Table 3.1, a variety of well-known means is modeled by (3.4), including the minimum, the harmonic, and the geometric means. Note that, despite a similar terminology, $A(\theta)$ is an elementwise function and thus is very different from the matrix function generalized mean considered in [14].

Letting $D(\theta) = \mathrm{diag}(A(\theta)\mathbf{1})$ be the degree matrix of the generalized mean adjacency matrix, and $\mathcal{L}(\theta) = D(\theta) - A(\theta)$ its Laplacian matrix, we extend (3.2) to the multilayer setting by considering the following

$$\varphi(X, Y; \theta) = \|X - Y\|_F^2 + \tfrac{\lambda}{2}\mathrm{Tr}(X^\top \mathcal{L}(\theta) X)$$

and the corresponding class-wise function $\varphi_k(x, y; \theta)$, obtained by replacing $A$ with $A(\theta)$ in (3.3). In order to learn the parameters $\theta$, we split the available input labels into training and test sets, with corresponding one-hot matrices $Y^{tr}$ and $Y^{te}$, and consider the bilevel optimization model

$$\begin{aligned} \min_{\theta} \quad & H(Y^{te}, X_{Y^{tr};\theta}) \\ \text{s.t.} \quad & X_{Y^{tr};\theta} = \mathrm{argmin}_X\, \varphi(X, Y^{tr}; \theta) \\ & \theta = (\alpha, \beta),\ \alpha \in \mathbb{R},\ \beta \geq 0,\ \textstyle\sum_s \beta_s = 1 \end{aligned} \tag{3.5}$$

where $H$ is the multiclass cross-entropy loss

$$H(Y, X) = -\frac{1}{|V|}\sum_{i=1}^{|V|}\sum_{j=1}^{|V|} Y_{ij} \log\left(\frac{X_{ij}}{\sum_{j=1}^{|V|} X_{ij}}\right).$$

The resulting embedding $X^*$ for the learned parameters is then used to classify the unlabeled nodes in the usual way.
Note that, unlike the single-layer case, using (3.3) rather than (3.2) in this setting may yield different results. In particular, if different layers carry information about different communities, using a one-vs-all cross-entropy model may be more effective. Thus, as an alternative to (3.5), we consider

$$\begin{aligned} \min_{\theta} \quad & h(y^{te}, x_{y^{tr};\theta}) \\ \text{s.t.} \quad & x_{y^{tr};\theta} = \operatorname{argmin}_x \varphi_k(x, y^{tr}; \theta) \\ & \theta = (\alpha, \beta),\ \alpha \in \mathbb{R},\ \beta \geq 0,\ \textstyle\sum_s \beta_s = 1 \end{aligned} \tag{3.6}$$

which we solve for each community $c$, individually, using the binomial cross-entropy loss

$$h(y, x) = -\frac{1}{N} \sum_{i=1}^{N} \big( y_i \log(x_i) + (1 - y_i) \log(1 - x_i) \big) .$$

## 3.5 Optimization with inexact gradient computations

To compute the classifier based on the generalized mean aggregation, we employ a gradient-free optimization algorithm in conjunction with a parametric Label Propagation technique. The details of this approach are outlined in Section §3.5.2. In particular, we utilize a Frank-Wolfe algorithm [121, 122, 123] with inexact gradient computation and a customized line search strategy. This algorithm is specifically designed for constrained optimization problems, and in our context, we focus on the formulation presented in (3.5) for simplicity. The same principles and procedures can be directly extended to the case of (3.6).
Note that, fixing the parameters $\theta = (\alpha, \beta)$, the inner problem $\min_X \varphi(X, Y; \theta)$ can be solved explicitly. A direct computation shows that $\nabla_X \varphi(X, Y; \theta) = 2\{(X - Y) + \lambda L(\theta)\}X$. Thus,

$$\operatorname{argmin}_X \varphi(X, Y; \theta) = \big(I + \lambda L(\theta)\big)^{-1} Y . \tag{3.7}$$

Using (3.7), we can rewrite (3.5) by replacing the optimality constraint at the inner level with its explicit solution. Moreover, as the generalized mean converges fast to maximum and minimum for $\alpha \to \pm\infty$, we limit $\alpha$ within an interval $\alpha \in [-a, a]$, for a large enough $a > 0$. Altogether, we reformulate (3.5) as

$$\min_{\theta \in S} \ f(\theta) , \tag{3.8}$$

where $f(\theta) := H(Y^{te}, (I + \lambda L(\theta))^{-1} Y^{tr})$ and, for $a > 0$, $S = \{(\alpha, \beta) \in \mathbb{R}^{k+1} : \alpha \in [-a, a], \theta_s > 0, \sum_s \theta_s = 1\}$.
As previously mentioned, we employ a Frank-Wolfe-based technique to address the optimization problem (3.8). The underlying concept of this algorithm is to determine, at each iteration $n$, a direction $d_n$ that minimizes a linear approximation of the objective function $f$ around the current point $\theta_n$. Subsequently, we update the next point $\theta_{n+1}$ by advancing along the direction $d_n$ with a step size $\eta_n$, which is chosen using an appropriate line search strategy. Although the function $f$ is smooth and real-valued, computing its gradient $\nabla f$ can be computationally intensive in

**Algorithm 2** Frank-Wolfe algorithm with inexact gradient

1: **Given** $\theta_0 \in S$
2: **For** $n = 0, 1, \ldots$
3: Compute $\widetilde{\nabla} f(\theta_n)$ as an estimate of $\nabla f(\theta_n)$
4: Compute $\hat{\theta}_n \in \operatorname{argmin}_{\theta \in S} \widetilde{\nabla} f(\theta_n)^\top (\theta - \theta_n)$ and set $d_n = \hat{\theta}_n - \theta_n$
5: Compute a stepsize $\eta_n \in (0, 1]$ by a line search
6: Set $\theta_{n+1} = \theta_n + \eta_n d_n$
7: **End for**

practice. To circumvent this challenge, we use an estimated gradient $\widetilde{\nabla} f$ within the algorithm. The resulting approach is presented in Algorithm 2.
Note that the linear problem at line 3 of Algorithm 2 is particularly simple due to the box-plus-simplex form of the constraint set $S$, and it can be solved separately in the variables $\alpha$ and $\beta$. In fact, for the variable $\alpha$, we aim at minimizing a linear function over the box $[-a, a]$, which implies $\hat{\alpha}_n = -a$ if $\widetilde{\nabla}_\alpha f(\theta_n) > 0$, and $\hat{\alpha}_n = a$ otherwise. Similarly, for the variables $\beta \in \mathbb{R}^K$, we have to minimize a linear function over the unit simplex, which yields $\hat{\beta}_n = e_{\hat{\jmath}}$, where $\hat{\jmath} = \operatorname{argmin}_{j=1,\ldots,K} [(\widetilde{\nabla}_\beta f(\theta_n)]_j$ and $e_{\hat{\jmath}}$ is the $\hat{\jmath}$-th vector of the canonical basis of $\mathbb{R}^K$.

### 3.5.1 Convergence analysis

To analyze the convergence of Algorithm 2, we first introduce some useful notation. Let $g_n = -\nabla f(\theta_n)^\top d_n$, $\widetilde{g}_n = -\widetilde{\nabla} f(\theta_n)^\top d_n$ and $g_n^{FW} = -\nabla f(\theta_n)^\top d_n^{FW}$, where $d_n^{FW} \in \operatorname{argmin}_{\theta \in S} \{\nabla f(\theta_n)^\top (\theta - \theta_n)\} - \theta_n$ is the direction obtained by the Frank-Wolfe algorithm with exact gradient. Inspired by [124], we assume that the estimate $\widetilde{\nabla} f$ satisfies the following condition.

**Assumption 3.5.1.** For every $n$, there exists $\epsilon_n \geq 0$ such that

$$|(\nabla f(\theta_n) - \widetilde{\nabla} f(\theta_n))^\top (\theta - \theta_n)| \leq \epsilon_n \quad \forall\, \theta \in S. \tag{3.9}$$

Since $S$ is a convex set, a point $\theta^* \in S$ is said to be stationary for (3.8) when $\nabla f(\theta^*)^\top (\theta - \theta^*) \geq 0$ for all $\theta \in S$. Then, $g_n^{FW}$ is an optimality measure, i.e. $g_n^{FW} = 0$ if and only if $\theta_n \in S$ is a stationary point. Now we show that when Assumption 3.5.1 is satisfied with a sufficiently small $\epsilon_n$ and the stepsize $\eta_n$ is generated with a suitable line search, Algorithm 2 obtains a stationary point at a sublinear rate on non-convex objectives with a Lipschitz continuous gradient. The constant in the convergence rate depends on the quality of the gradient estimate (the more precise the estimate, the smaller the constant).

**Theorem 3.5.2.** *Let $\nabla f$ be Lipschitz continuous with constant $M$, and let $S$ be compact with finite diameter $\Delta$. Let $\{\theta_n\}$ be a sequence generated by Algorithm 2, where $\widetilde{\nabla} f$ satisfies Assumption 3.5.1 with*

$$\epsilon_n \leq \frac{\sigma}{1 + \sigma}\, \widetilde{g}_n, \quad 0 \leq \sigma < \frac{1}{3}, \tag{3.10}$$

*and the step size $\eta_n$ satisfies*

$$\eta_n \geq \bar{\eta}_n = \min\left(1, \frac{\widetilde{g}_n}{M\|d_n\|^2}\right), \tag{3.11}$$

$$f(\theta_n) - f(\theta_n + \eta_n d_n) \geq \rho\bar{\eta}_n\widetilde{g}_n, \tag{3.12}$$

*with some fixed $\rho > 0$. Then,*

$$g_n^* \leq \max\left(\sqrt{\frac{\Delta^2 M(f(\theta_0) - f^*)}{n\rho(1-\sigma)^2}}, \frac{2(f(\theta_0) - f^*)}{n(1-3\sigma)}\right), \tag{3.13}$$

*where $g_n^* = \min\limits_{0\leq i\leq n-1} g_i^{FW}$ and $f^* = \min_{\theta\in S} f(\theta)$.*

*Proof.* The following chain of inequalities holds:

$$-\nabla f(\theta_n)^\top d_n^{FW} \geq -\nabla f(\theta_n)^\top d_n \geq -\widetilde{\nabla} f(\theta_n)^\top d_n - \epsilon_n \geq -\widetilde{\nabla} f(\theta_n)^\top d_n^{FW} - \epsilon_n \geq -\nabla f(\theta_n)^\top d_n^{FW} - 2\epsilon_n, \tag{3.14}$$

where we used (3.9) in the second and the last inequality, while the first and the third inequality follow from the definition of $d_n^{FW}$ and $d_n$. In particular, using the definitions of $\widetilde{g}^n$, $g_n$ and $g_n^{FW}$, from (3.14) we can write

$$g_n^{FW} \geq \widetilde{g}_n - \epsilon_n, \tag{3.15}$$

$$\widetilde{g}_n \geq g_n^{FW} - \epsilon_n, \tag{3.16}$$

$$g_n \geq \widetilde{g}_n - \epsilon_n \tag{3.17}$$

Using (3.10) and (3.15), we also have

$$\epsilon_n \leq \sigma(\widetilde{g}_n - \epsilon_n) \leq \sigma g_n^{FW}, \tag{3.18}$$

Now, let us distinguish two cases.

- If $\bar{\eta}_n < 1$, from (3.11) it follows that $\dfrac{\widetilde{g}_n}{M\|d_n\|^2} < 1$. Using (3.12) we can write

$$f(\theta_n) - f(\theta_n + \eta_n d_n) \geq \rho\bar{\eta}_n\widetilde{g}_n = \frac{\rho}{M\|d_n\|^2}\widetilde{g}_n^2 \geq \frac{\rho\widetilde{g}_n^2}{\Delta^2 M},$$

  where the last inequality follows from $\|d_n\| \leq \Delta$. Observe that, from (3.16) and (3.18), we have $\widetilde{g}_n \geq (1-\sigma)g_n^{FW}$. Therefore,

$$f(\theta_n) - f(\theta_{n+1}) \geq \frac{\rho(1-\sigma)^2}{\Delta^2 M}(g_n^{FW})^2. \tag{3.19}$$

- If $\bar{\eta}_n = 1$, from (3.11) it follows that $\dfrac{\widetilde{g}_n}{M\|d_n\|^2} \geq 1$ and, since $\eta_n \leq 1$ from the instructions of the algorithm, then $\eta_n = 1$. By the standard descent lemma, we can write

$$f(\theta_{n+1}) = f(\theta_n + d_n) \leq f(\theta_n) - g_n + \frac{M}{2}\|d_n\|^2 \leq f(\theta_n) - (\widetilde{g}_n - \epsilon_n) + \frac{M}{2}\|d_n\|^2,$$

where we used (3.17) in the last inequality. Since we are analyzing the case where $\widetilde{g}_n \geq \|d_n\|^2 M$, we obtain

$$f(\theta_n) - f(\theta_{n+1}) \geq \frac{\widetilde{g}_n}{2} - \epsilon_n.$$

Using (3.16) and (3.18), we also have

$$\frac{\widetilde{g}_n}{2} - \epsilon_n \geq \frac{g_n^{FW}}{2} - \frac{3}{2}\epsilon_n \geq \frac{g_n^{FW}}{2} - \frac{3}{2}\sigma g_n^{FW} = \frac{1-3\sigma}{2} g_n^{FW}.$$

Therefore,

$$f(\theta_n) - f(\theta_{n+1}) \geq \frac{1-3\sigma}{2} g_n^{FW}. \tag{3.20}$$

Now, based on the two cases analyzed above, we partition the iterations $\{0, 1, \ldots, T-1\}$ into two subsets $N_1$ and $N_2$ defined as follows:

$$N_1 = \{n < T \colon \bar{\eta}_n < 1\}, \quad N_2 = \{n < T \colon \bar{\eta}_n = 1\}.$$

Using (3.19) and (3.20), we can write:

$$\begin{aligned}
f(\theta_0) - f^* &\geq \sum_{n=0}^{T-1} (f(\theta_n) - f(\theta_{n+1})) \\
&= \sum_{N_1} (f(\theta_n) - f(\theta_{n+1})) + \sum_{N_2} (f(\theta_n) - f(\theta_{n+1})) \\
&\geq \sum_{N_1} \frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_n^{FW})^2 + \sum_{N_2} \frac{1-3\sigma}{2} g_n^{FW} \\
&\geq |N_1| \min_{n \in N_1} \frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_n^{FW})^2 + |N_2| \min_{n \in N_2} \frac{1-3\sigma}{2} g_n^{FW} \\
&\geq (|N_1| + |N_2|) \min\left( \frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2, \frac{1-3\sigma}{2} g_T^* \right) \\
&= T \min\left( \frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2, \frac{1-3\sigma}{2} g_T^* \right),
\end{aligned}$$

where the last inequality follows from the definition of $g_T^*$. Hence,

$$\begin{aligned}
&\frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2 \leq \frac{1-3\sigma}{2} g_T^* \;\Rightarrow\; g_T^* \leq \sqrt{\frac{\Delta^2 M (f(\theta_0) - f^*)}{T\rho(1-\sigma)^2}}, \\
&\frac{\rho(1-\sigma)^2}{\Delta^2 M} (g_T^*)^2 > \frac{1-3\sigma}{2} g_T^* \;\Rightarrow\; g_T^* \leq \frac{2(f(\theta_0) - f^*)}{T(1-3\sigma)},
\end{aligned}$$

leading to the desired result. □

Note that, in our setting, $\Delta \leq 2a + \sqrt{2}$. Condition (3.10) can be easily satisfied by a proper calculation of the gradient estimate $\widetilde{\nabla} f$ (see §3.5.2). Conditions (3.11)–(3.12) can be satisfied with suitable line searches/stepsize rules (see, e.g., [123, 125, 126]). In particular, Lemma 3.5.3 below shows that this is the case for the modified Armijo line search rule which sets

$$\eta_n = \delta^j, \tag{3.21}$$

where $j$ is the smallest non-negative integer such that

$$f(\theta_n) - f(\theta_n + \eta_n d_n) \geq \gamma \eta_n \widetilde{g}_n, \tag{3.22}$$

with $\gamma \in (0, 1/2)$ and $\delta \in (0, 1)$ being two fixed parameters.

**Lemma 3.5.3.** *Let Assumption 3.5.1 hold with*

$$\epsilon_n \leq \frac{\sigma}{1+\sigma}\, \widetilde{g}_n, \quad 0 \leq \sigma < \frac{1}{2}. \tag{3.23}$$

*At iteration $n$, if $\eta_n$ is determined by the Armijo line search described in* (3.21)–(3.22)*, then*

$$\eta_n \geq \min\{1, 2\delta(1-\gamma-\sigma)\}\bar{\eta}_n, \tag{3.24}$$

*with $\bar{\eta}_n$ being defined as in* (3.11)*.*

*Proof.* Reasoning as in the proof of Theorem 3.5.2, we have that (3.17) holds. By the standard descent lemma, we have

$$f(\theta_n) - f(\theta_n + \eta d_n) \geq \eta g_n - \eta^2 \frac{M\|d_n\|^2}{2} \geq \eta(\widetilde{g}_n - \epsilon_n) - \eta^2 \frac{M\|d_n\|^2}{2}, \quad \forall \eta \in \mathbb{R}, \tag{3.25}$$

where the last inequality follows from (3.17). Then,

$$f(\theta_n) - f(\theta_n + \eta d_n) \geq \gamma \eta \widetilde{g}_n \quad \forall \eta \in \left[0, 2\frac{(1-\gamma)\widetilde{g}_n - \epsilon_n}{M\|d_n\|^2}\right].$$

Since $\eta_n$ is computed by (3.21)–(3.22), we can write

$$\begin{aligned} \eta_n &\geq \min\left(1, 2\delta\frac{(1-\gamma)\widetilde{g}_n - \epsilon_n}{M\|d_n\|^2}\right) \\ &\geq \min\left(1, 2\delta\frac{(1-\gamma-\sigma)\widetilde{g}_n}{M\|d_n\|^2}\right) \\ &\geq \min(1, 2\delta(1-\gamma-\sigma))\bar{\eta}_n, \end{aligned} \tag{3.26}$$

where the second inequality follows from (3.23). □

Note that in the proof of Theorem 3.5.3 we prove

$$\eta_n \geq \min\left(1, c\frac{\widetilde{g}_n}{M\|d_n\|^2}\right) \text{ for some } c > 0$$

for the Armijo line search. When $c \geq 1$ then $\bar{\eta}_n$ is of course a lower bound for the step size $\eta_n$, and when $c < 1$ we can still recover (3.11) by considering $\widetilde{M} = M/c$ instead of $M$ as Lipschitz constant.

### 3.5.2 Implementation details

In Algorithm 2, we approximate the gradient with the finite difference method:

$$\widetilde{\nabla} f(\theta_n) = \sum_{i=1}^{K+1} \frac{f(\theta_n + h_n e_i) - f(\theta_n)}{h_n} e_i, \tag{3.27}$$

where $h_n$ is a suitably chosen positive parameter. As shown in [127], this approach gives good gradient approximations in practice.
From [128], we have that Eq. (3.9) is satisfied when using the finite difference approach with $\epsilon_n = \frac{M\Delta d}{2} h_n$. We notice that condition (3.10) can in turn be satisfied at each iteration $k$ by suitably choosing $h_n$ in the finite difference approximation, e.g., $h_n \leq \xi\tau$, with $\xi = \frac{2\theta}{(1+\theta)M\Delta d}$ and $\tau$ stopping condition tolerance.
In our experiments, we start with $h_0 = 10^{-4}$ and set $h_n = \frac{h_{n-1}}{2}$ for $n = 1, 2, \ldots$. Furthermore, we stop the algorithm when $\widetilde{g}_n = -\widetilde{\nabla} f(\theta_n)^\top d_n \leq \tau$, with $\tau = 10^{-4}$.
To compute $f(\theta)$ for a given set of parameters $\theta = (\alpha, \beta)$, we run a form of modified parametric Label Propagation algorithm:

$$X^{(r+1)} = \lambda A\theta (I + \lambda D\theta)^{-1} X^{(r)} + (I + \lambda D\theta)^{-1} Y\,,$$

which propagates the input labels in $Y$ and converges to the solution of the linear system (3.7). In fact, as $\sum_j A(\theta)_{ij} = D(\theta)_{ii}$ for all $i$, a direct application of the first Gershgorin circle theorem [129] to the matrix $A(\theta)(I + \lambda D(\theta))^{-1}$ implies that the spectral radius of $A(\theta)(I + \lambda D(\theta))^{-1}$ is smaller than one and thus $X^{(r)} \to \operatorname{argmin}_X \varphi(X, Y; \theta)$, as $r \to \infty$.
We apply the multistart version of Frank Wolfe [130], where the algorithm is applied with different initial points, and we choose the best solution according to the value of the optimized function $f$. In particular, we start from 10 random points $\theta_0$, among which we include the particular choices $\theta_0 = (1, 1/K, \ldots, 1/K)$ and $\theta_0 = (1, 1/K, \ldots, 1/K)$, which correspond to the arithmetic and the harmonic means. In all the experiments, we fixed $\lambda = 1$ in the objective function (3.4), and we restricted the study of the parameter in $\alpha \in [-20, 20]$.

## 3.6 Experiments

We conduct experiments on various synthetic and real-world multilayer networks. For each dataset, we are provided with a set of known labels denoted as $Y$. This set is partitioned into a training subset $Y^{tr}$ containing 80% of the available labels, and a test subset $Y^{te}$ containing the remaining 20%. To mitigate the potential influence of different training and test set selections on the optimal parameters, we initially split the input-labeled nodes into five equal-sized subsets. Subsequently, one of these subsets is assigned cyclically to $Y^{te}$ while the others are assigned to $Y^{tr}$. We apply Algorithm 2 with 10 different starting points for each of these five choices. Ultimately, we select the parameters that result in the lowest value of the loss function $f(\theta)$ over the test set among the five runs. Once the optimal aggregation weights are determined, we employ standard Label Propagation on the aggregated graph that emerges. Subsequently, we assess the accuracy performance on the held-out test set, consisting of all initially unlabeled points.

We implement both the multiclass (MULTI) and the binomial (BINOM) versions of our method, corresponding to the bilevel optimization problems in (3.2) and (3.3), respectively. Our `Python` implementation is available at the GitHub page: `https://github.com/saraventurini/Learning-the-right-layers-semi-supervised-learning-on-multilayer-graphs-`

We conducted a comprehensive series of experiments that encompassed both synthetic and real-world networks. The primary objective was to compare the efficacy of our proposed approach, which involves learning the parameters of the generalized mean from the available input data. This was contrasted with standard Label Propagation applied to each individual layer. Additionally, we evaluated the performance of methods that utilize the proposed generalized mean aggregation function with specific parameter selections (as detailed in Table 3.1). Furthermore, we included four multilayer graph semi-supervised learning baseline methods in our comparison:

- **SGMI**: Sparse Multiple Graph Integration [107], based on label propagation by sparse integration, with parameters $\lambda_1 = 1, \lambda_2 = 10^{-3}$;
- **AGML**: Auto-weighted Multiple Graph Learning [108], which is a parameter-free method for optimal graph layer weights;
- **SMACD**: Semi-supervised Multi-Aspect Community Detection [112], which is a tensor factorization method for semi-supervised learning;
- **GMM**: Generalized Matrix Means, which is a Laplacian-regularization approach based on the Log-Euclidean matrix function formulation of the power mean Laplacian, with parameter $p = -1$ [14];

Notice that SGMI and GMM need a parameter choice, which we have made following the indications in the corresponding papers, while AGML and SMACD, as well as the proposed methods MULTI and BINOM, are parameter-free. We also tested against the two multilayer graph neural networks discussed in §3.2, which however performed poorly, probably due to the absence of features in our test settings.

### 3.6.1 Synthetic Datasets

We created synthetic datasets with 3 communities of 400 nodes each, and 3 layers. In particular, for each layer, we generated 3 isotropic Gaussian blobs of points $p_i \in \mathbb{R}^5$, with a variable standard deviation. The adjacency matrix of the network is then formed using a symmetrized $k$-NN graph with $k = 5$, weighted with the Euclidean kernel $\exp(-\|p_i - p_j\| + \min_{ij} \|p_i - p_j\|)$. We considered three settings (illustrated in Figure 3.6):

- *Informative case*: layers are constructed using three isotropic Gaussian clusters, and all layers exhibit the same community structure.;
- *Noisy case*: involves one informative layer and two noisy layers. The noisy layers are created by randomly permuting the informative layers.
- *Complementary case*: each layer contains information about only one cluster while being noisy for the other clusters. The noisy layers in this case are sparser compared to the previous scenario, achieved by shuffling $k$-nearest neighbor ($k - NN$) layers with $k = 1$.

The informative isotropic Gaussian blobs have standard deviation $std \in \{5, 6, 7, 8\}$ for the informative case, as this is the easiest setting, while we test for $std \in \{2, 3, 4, 5\}$ in the other two settings, as these are more challenging. The percentage of input labels is 20% of the overall

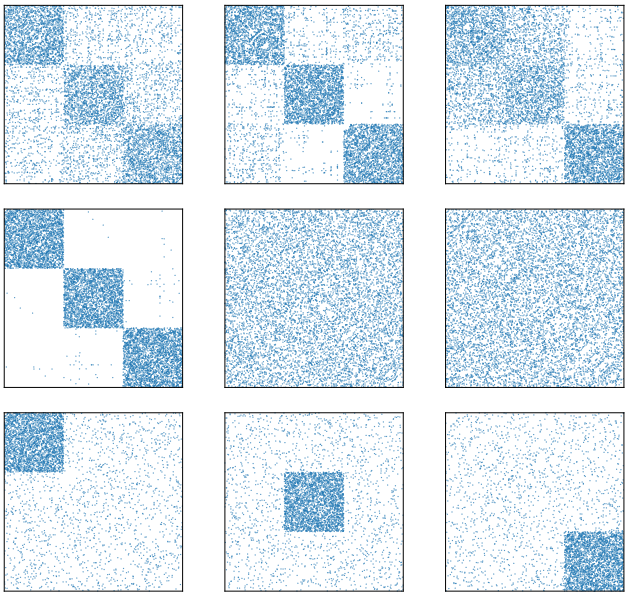

Figure 3.1: Synthetic datasets settings. (top) informative case, (middle) noisy case, (bottom) complementary case.

number of nodes in each of the communities.

Table 3.3 reports the average accuracy and standard deviation score across 5 network samples, as compared to the accuracy of the individual layers (computed ignoring the other layers), reported in the first three columns, and those achieved with fixed a-priori choices of the parameters (as in Table 3.1).

The BINOM and MULTI methods we propose consistently demonstrate strong performance across all experimental scenarios, often surpassing the individual layers and the baselines we considered. Specifically, the performance of MIN, GEO, and SMACD is generally poor across all settings. AGML exhibits favorable results primarily in the informative setting. ARIT, HARM, and MAX showcase good performance in the informative and complementary cases, but not in the noisy scenario. SGMI achieves high accuracy only in the noisy case. GMM's performance is noteworthy in informative and noisy settings.

While the best performance is sometimes achieved by some particular aggregation function (such as MAX or HARM), all the baselines are setting-specific and have poor performance in certain settings, e.g. in the presence of noise. When measured across all settings, BINOM and MULTI perform best. This is highlighted by the Average Performance Ratio (APR) score values, reported in the last line of Table 3.3, which are quantified as follows: denoting the accuracy of algorithm $a$ on dataset $d$ as $\mathcal{A}_{a,d}$, let the performance ratio be $r_{a,d} = \mathcal{A}_{a,d} / \max\{\mathcal{A}_{a,d} \text{ over all } a\}$. The APR of each algorithm is then obtained by averaging $r_{a,d}$ over all the datasets $d$. For any algorithm, the closer the average performance ratio is to 1, the better the overall performance.

Additionally, upon analyzing the learned weights, the proposed methods enable us to evaluate the structure of the multilayer network and identify the presence of noisy or less informative layers. This is exemplified in Table 3.2, which displays an instance of learned weights obtained from

Table 3.2: Example of learned parameters by BINOM (B) and MULTI (M) on the synthetic datasets of Table 3.3.

| | | INFO | | | | NOISY | | | | COMPL | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $k$ | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\alpha$ | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\alpha$ | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\alpha$ |
| B | 1 | 0.2 | 0.6 | 0.3 | 7.2 | 0.7 | 0.2 | 0.1 | 3.3 | 0.6 | 0.2 | 0.3 | 15.7 |
| | 2 | 0.7 | 0.3 | 0.0 | 0.1 | 1 | 0 | 0 | 20 | 0.2 | 0.8 | 0.0 | 2.9 |
| | 3 | 0.2 | 0.8 | 0.0 | 0.1 | 0.3 | 0.1 | 0.6 | 11.7 | 0.1 | 0.3 | 0.6 | 12.7 |
| M | - | 0.3 | 0.4 | 0.3 | 0.6 | 1 | 0 | 0 | 20 | 0.3 | 0.4 | 0.3 | 1.7 |

Table 3.3: Accuracy (mean ± standard deviation) over five random samples of synthetically generated multilayer graphs, for different levels of *std* in the isotropic Gaussian blobs forming the clusters.

| | *std* | I | II | III | MIN | GEOM | ARIT | HARM | MAX | BINOM | MULTI | SGMI | AGML | SMACD | GMM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INFO | 5 | 0.83±0.10 | 0.86±0.05 | 0.84±0.10 | 0.33±0.00 | 0.33±0.00 | 0.95±0.03 | 0.96±0.03 | 0.96±0.03 | 0.96±0.03 | 0.93±0.03 | 0.87±0.04 | 0.93±0.05 | 0.48±0.18 | 0.93±0.04 |
| | 6 | 0.77±0.11 | 0.80±0.05 | 0.79±0.11 | 0.33±0.00 | 0.33±0.00 | 0.90±0.05 | 0.92±0.04 | 0.92±0.05 | 0.92±0.05 | 0.88±0.06 | 0.80±0.06 | 0.86±0.09 | 0.47±0.16 | 0.89±0.05 |
| | 7 | 0.71±0.10 | 0.74±0.05 | 0.74±0.11 | 0.33±0.00 | 0.33±0.00 | 0.86±0.07 | 0.88±0.06 | 0.87±0.07 | 0.88±0.06 | 0.82±0.06 | 0.74±0.07 | 0.80±0.10 | 0.46±0.14 | 0.85±0.07 |
| | 8 | 0.66±0.10 | 0.69±0.04 | 0.70±0.11 | 0.33±0.00 | 0.33±0.00 | 0.81±0.08 | 0.83±0.08 | 0.82±0.09 | 0.82±0.08 | 0.77±0.06 | 0.69±0.07 | 0.74±0.10 | 0.44±0.14 | 0.80±0.08 |
| NOISY | 2 | 0.99±0.02 | 0.35±0.01 | 0.36±0.01 | 0.33±0.00 | 0.33±0.00 | 0.67±0.02 | 0.68±0.03 | 0.68±0.03 | 0.97±0.02 | 0.99±0.02 | 0.98±0.03 | 0.63±0.03 | 0.35±0.03 | 0.86±0.04 |
| | 3 | 0.95±0.05 | 0.35±0.00 | 0.36±0.01 | 0.33±0.00 | 0.33±0.00 | 0.66±0.04 | 0.66±0.04 | 0.66±0.04 | 0.86±0.13 | 0.95±0.05 | 0.94±0.07 | 0.60±0.06 | 0.36±0.04 | 0.82±0.07 |
| | 4 | 0.89±0.08 | 0.36±0.00 | 0.35±0.01 | 0.33±0.00 | 0.33±0.00 | 0.63±0.05 | 0.63±0.05 | 0.63±0.05 | 0.78±0.13 | 0.89±0.08 | 0.88±0.10 | 0.57±0.07 | 0.36±0.03 | 0.77±0.10 |
| | 5 | 0.83±0.10 | 0.35±0.01 | 0.36±0.01 | 0.33±0.00 | 0.33±0.00 | 0.58±0.06 | 0.59±0.06 | 0.58±0.06 | 0.72±0.15 | 0.83±0.10 | 0.80±0.12 | 0.53±0.07 | 0.35±0.04 | 0.70±0.11 |
| COMPL | 2 | 0.41±0.00 | 0.47±0.00 | 0.48±0.01 | 0.33±0.00 | 0.33±0.00 | 0.89±0.02 | 0.93±0.02 | 0.92±0.02 | 0.93±0.02 | 0.89±0.02 | 0.43±0.04 | 0.30±0.01 | 0.52±0.05 | 0.69±0.03 |
| | 3 | 0.41±0.01 | 0.47±0.01 | 0.47±0.01 | 0.33±0.00 | 0.33±0.00 | 0.86±0.04 | 0.90±0.05 | 0.89±0.05 | 0.90±0.05 | 0.87±0.04 | 0.44±0.04 | 0.30±0.02 | 0.44±0.08 | 0.67±0.04 |
| | 4 | 0.40±0.01 | 0.46±0.01 | 0.47±0.01 | 0.33±0.00 | 0.33±0.00 | 0.82±0.06 | 0.86±0.06 | 0.85±0.06 | 0.86±0.06 | 0.83±0.05 | 0.43±0.04 | 0.18±0.16 | 0.71±0.17 | 0.65±0.05 |
| | 5 | 0.40±0.01 | 0.46±0.02 | 0.47±0.01 | 0.33±0.00 | 0.33±0.00 | 0.79±0.07 | 0.82±0.08 | 0.81±0.07 | 0.82±0.08 | 0.80±0.07 | 0.43±0.04 | 0.18±0.17 | 0.46±0.13 | 0.63±0.06 |
| APR | | - | - | - | 0.37 | 0.37 | 0.88 | 0.90 | 0.89 | **0.97** | **0.97** | 0.78 | 0.61 | 0.50 | 0.86 |

Algorithm 2. Furthermore, it's worth noting that while BINOM excels in the informative and complementary cases, as anticipated, MULTI appears to be the most robust approach across all scenarios.

### 3.6.2 Real World Datasets

We consider nine real-world datasets frequently used to assess the performance of multilayer graph clustering (some of which have already been mentioned in §2.5.3) [9, 14, 131]:

- *3sources*: news articles covered by news sources BBC, Reuters, and Guardian (169 nodes, 6 communities, 3 layers) [57, 132];
- *BBC*: BBC news articles (685 nodes, 5 communities, 4 layers) [133];
- *BBCSport*: BBC Sport articles (544 nodes, 5 communities, 2 layers) [132];
- *Wikipedia*: Wikipedia articles (693 nodes, 10 communities, 2 layers) [85];
- *UCI*: hand-written UCI digits dataset with six different sets of features (2000 nodes, 10 communities, 6 layers) [57, 134];
- *cora*: citations dataset (2708 nodes, 7 communities, 2 layers) [83];
- *citeseer*: citations dataset (3312 nodes, 6 communities, 2 layers) [135];
- *dkpol*: five types of relationships between employees of a university department (490 nodes, 10 communities, 3 layers) [136];

Table 3.4: Accuracy (mean ± standard deviation) over three random samples of the input labels, on real-world datasets (+ one layer of noise).

| | I | II | III | IV | V | VI | | MIN | GEOM | ARIT | HARM | MAX | BINOM | MULTI | SGMI | SMACD | GMM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3sources | 0.77 | 0.75 | 0.78 | - | - | - | | 0.75±0.06 | 0.75±0.06 | 0.83±0.05 | 0.76±0.04 | 0.80±0.03 | 0.79±0.05 | 0.80±0.06 | 0.74±0.04 | 0.66±0.09 | 0.80±0.03 |
| | | | | | | | (+noise) | 0.35±0.00 | 0.35±0.00 | 0.78±0.07 | 0.59±0.04 | 0.69±0.04 | 0.77±0.07 | 0.74±0.08 | 0.75±0.05 | 0.61±0.09 | 0.77±0.02 |
| BBC | 0.85 | 0.84 | 0.8 | 0.84 | - | - | | 0.39±0.02 | 0.39±0.02 | 0.91±0.01 | 0.89±0.01 | 0.90±0.01 | 0.91±0.01 | 0.89±0.01 | 0.77±0.02 | 0.64±0.12 | 0.89±0.01 |
| | | | | | | | (+noise) | 0.33±0.00 | 0.33±0.00 | 0.91±0.01 | 0.88±0.01 | 0.90±0.01 | 0.88±0.01 | 0.87±0.02 | 0.79±0.01 | 0.62±0.05 | 0.87±0.01 |
| BBCSport | 0.91 | 0.9 | - | - | - | - | | 0.81±0.01 | 0.81±0.01 | 0.94±0.01 | 0.93±0.00 | 0.93±0.00 | 0.94±0.01 | 0.92±0.01 | 0.84±0.02 | 0.77±0.09 | 0.91±0.02 |
| | | | | | | | (+noise) | 0.36±0.00 | 0.36±0.00 | 0.90±0.02 | 0.87±0.01 | 0.88±0.02 | 0.92±0.01 | 0.89±0.02 | 0.84±0.03 | 0.85±0.02 | 0.86±0.02 |
| Wikipedia | 0.17 | 0.65 | - | - | - | - | | 0.21±0.01 | 0.21±0.01 | 0.56±0.02 | 0.56±0.02 | 0.56±0.02 | 0.64±0.02 | 0.66±0.02 | 0.62±0.02 | 0.31±0.05 | 0.62±0.01 |
| | | | | | | | (+noise) | 0.15±0.00 | 0.15±0.00 | 0.50±0.01 | 0.50±0.01 | 0.50±0.01 | 0.64±0.01 | 0.66±0.01 | 0.61±0.01 | 0.32±0.06 | 0.55±0.01 |
| UCI | 0.92 | 0.82 | 0.96 | 0.59 | 0.97 | 0.83 | | 0.11±0.00 | 0.11±0.00 | 0.96±0.01 | 0.88±0.00 | 0.93±0.00 | 0.97±0.01 | 0.97±0.00 | 0.95±0.01 | 0.35±0.11 | 0.96±0.01 |
| | | | | | | | (+noise) | 0.10±0.00 | 0.10±0.00 | 0.97±0.00 | 0.90±0.01 | 0.94±0.00 | 0.97±0.00 | 0.97±0.01 | 0.95±0.01 | 0.26±0.06 | 0.96±0.01 |
| cora | 0.75 | 0.64 | - | - | - | - | | 0.35±0.01 | 0.35±0.01 | 0.71±0.00 | 0.70±0.00 | 0.71±0.00 | 0.75±0.03 | 0.70±0.01 | 0.75±0.01 | 0.41±0.03 | 0.73±0.01 |
| | | | | | | | (+noise) | 0.30±0.00 | 0.30±0.00 | 0.61±0.00 | 0.59±0.01 | 0.60±0.01 | 0.68±0.02 | 0.76±0.02 | 0.75±0.01 | 0.35±0.07 | 0.65±0.00 |
| citeseer | 0.56 | 0.65 | - | - | - | - | | 0.31±0.01 | 0.31±0.01 | 0.67±0.00 | 0.67±0.01 | 0.67±0.01 | 0.69±0.01 | 0.68±0.02 | 0.55±0.02 | 0.39±0.08 | 0.61±0.01 |
| | | | | | | | (+noise) | 0.21±0.00 | 0.21±0.00 | 0.57±0.01 | 0.55±0.00 | 0.56±0.01 | 0.66±0.00 | 0.62±0.03 | 0.55±0.03 | 0.35±0.07 | 0.54±0.01 |
| dkpol | 0.35 | 0.16 | 0.69 | - | - | - | | 0.16±0.01 | 0.16±0.01 | 0.74±0.05 | 0.65±0.07 | 0.69±0.06 | 0.69±0.05 | 0.78±0.03 | 0.34±0.04 | 0.25±0.06 | 0.69±0.05 |
| | | | | | | | (+noise) | 0.14±0.00 | 0.14±0.00 | 0.67±0.05 | 0.64±0.05 | 0.64±0.05 | 0.64±0.05 | 0.75±0.04 | 0.36±0.05 | 0.26±0.05 | 0.39±0.04 |
| aucs | 0.34 | 0.36 | 0.61 | 0.8 | 0.72 | - | | 0.30±0.01 | 0.30±0.01 | 0.85±0.04 | 0.79±0.06 | 0.85±0.05 | 0.81±0.06 | 0.81±0.06 | 0.75±0.07 | 0.55±0.04 | 0.81±0.05 |
| | | | | | | | (+noise) | 0.27±0.00 | 0.27±0.00 | 0.83±0.04 | 0.47±0.04 | 0.67±0.04 | 0.87±0.01 | 0.87±0.01 | 0.76±0.09 | 0.48±0.06 | 0.78±0.05 |
| APR | | | | | | | | 0.45 | 0.39 | 0.87 | 0.87 | 0.89 | 0.89 | **0.97** | 0.86 | 0.57 | 0.91 |

Table 3.5: Accuracy (mean ± standard deviation) over three random samples of the input labels, on real-world datasets with two additional noisy layers.

| | MIN | GEOM | ARIT | HARM | MAX | BINOM | MULTI | SGMI | SMACD | GMM |
|---|---|---|---|---|---|---|---|---|---|---|
| 3sources | 0.34±0.00 | 0.34±0.00 | 0.74±0.09 | 0.48±0.02 | 0.64±0.04 | 0.77±0.08 | 0.77±0.07 | 0.75±0.05 | 0.63±0.15 | 0.76±0.07 |
| BBC | 0.33±0.0 | 0.33±0.00 | 0.89±0.00 | 0.84±0.00 | 0.88±0.00 | 0.85±0.02 | 0.88±0.01 | 0.79±0.01 | 0.64±0.01 | 0.84±0.01 |
| BBCSport | 0.35±0.00 | 0.35±0.00 | 0.86±0.01 | 0.81±0.01 | 0.83±0.0 | 0.91±0.01 | 0.87±0.02 | 0.84±0.03 | 0.85±0.07 | 0.8±0.03 |
| Wikipedia | 0.15±0.00 | 0.15±0.00 | 0.47±0.02 | 0.47±0.02 | 0.47±0.02 | 0.63±0.00 | 0.66±0.01 | 0.61±0.01 | 0.23±0.02 | 0.51±0.01 |
| UCI | 0.10±0.00 | 0.10±0.00 | 0.97±0.00 | 0.89±0.01 | 0.94±0.00 | 0.97±0.00 | 0.96±0.01 | 0.95±0.01 | 0.29±0.01 | 0.96±0.01 |
| cora | 0.30±0.00 | 0.30±0.00 | 0.54±0.00 | 0.52±0.00 | 0.53±0.00 | 0.63±0.02 | 0.64±0.11 | 0.75±0.01 | 0.36±0.05 | 0.62±0.01 |
| citeseer | 0.21±0.00 | 0.21±0.00 | 0.51±0.01 | 0.47±0.01 | 0.49±0.01 | 0.61±0.03 | 0.62±0.05 | 0.55±0.03 | 0.36±0.13 | 0.52±0.01 |
| dkpol | 0.14±0.00 | 0.14±0.00 | 0.62±0.05 | 0.58±0.05 | 0.59±0.05 | 0.57±0.04 | 0.65±0.03 | 0.36±0.05 | 0.19±0.04 | 0.32±0.05 |
| aucs | 0.27±0.00 | 0.27±0.00 | 0.77±0.04 | 0.42±0.04 | 0.64±0.05 | 0.83±0.04 | 0.83±0.04 | 0.76±0.09 | 0.54±0.05 | 0.78±0.07 |

- *aucs*: three types of online relations between Danish Members of the Parliament on Twitter (61 nodes, 9 communities, 5 layers) [137].

For each dataset, we assume that initially, only 15% of the labels are known for each class. The average accuracy along with the standard deviation across 3 samples of known labels is presented in Table 3.4. Additionally, we provide the performance after introducing one additional noisy layer. The results after adding two layers of noise are included in Table 3.5. Notably, we exclude the AGML baseline from the comparison since it is tailored for graphs with communities of the same size. The findings reaffirm the trends observed in the synthetic case, demonstrating that BINOM and MULTI consistently match or surpass the baselines in most scenarios. These two methods exhibit the most favorable performance overall, spanning all settings.

It's worth highlighting that among all the techniques considered, BINOM and MULTI are the only ones that consistently achieve results on par with or better than using the individual layers

in isolation. This indicates that our proposed approach effectively harnesses the benefits of the multilayer structure across various scenarios. This attribute is particularly significant and valuable, as it addresses a recent data challenge [15].

## 3.7 Conclusion

In summary, our approach introduces a novel method for semi-supervised community detection in multiplex networks. It offers the flexibility of learning a nonlinear aggregation function that adapts the weights assigned to each network based on the available labeled data. The problem is formulated as a bilevel optimization task, which we tackle using an inexact Frank-Wolfe algorithm in conjunction with a parametric Label Propagation strategy. We present a comprehensive convergence analysis of our method. Through extensive experimentation, we compare our approach to single-layer methods and various baseline techniques across synthetic and real-world datasets. The results consistently showcase the method's ability to identify informative layers, resulting in reliable and robust performance across diverse clustering scenarios, especially when certain layers are dominated by noise.

# Chapter 4

# Laplacian-based semi-supervised learning in multilayer hypergraphs by coordinate descent

Similar to the preceding chapter, we deal with the semi-supervised learning problem considering an optimization-based formulation. In this case, we do not extend it just to multilayer graphs but also hypergraphs and multilayer hypergraphs (i.e. a set of hypergraphs each representing a different layer). We focus on the additional complexity and harder treatability that usually come from considering more sophisticated structures than simple graphs. Specifically, we conduct a comparison between the application of various coordinate descent methods and the gradient descent algorithm. The performed experiments on both synthetic and real-world datasets demonstrate the advantage of employing coordinate descent methods, especially when combined with appropriate selection rules tailored to the specific problems at hand. In addition, we carried out an analysis replacing the standard quadratic regularization term in the objective function with a more general $p-$regularizer. The reported results clearly show that this modification can lead to better performance.
The aim is to present this problem to the optimization community.

## 4.1 Introduction

Consider a finite, weighted and undirected graph $G = (V, E)$, with node set $V$ and edge-weight function $w$ such that $w(e) = w(uv) > 0$ if $e = (u, v) \in E$, edge set $E \subseteq V \times V$ and 0 otherwise. Suppose each node $u \in V$ can be assigned to one of $|C|$ classes, or labels, $C_1, \ldots, C_{|C|}$. In graph-based Semi-Supervised Learning (SSL), given a graph $G$ and an observation set of labeled nodes $O \subset V$ whose vertices $i \in O$ are pre-assigned to some label $y_i \in \{C_1, \ldots, C_{|C|}\}$, the aim is to infer the labels of the remaining unlabeled nodes in $V \setminus O$, using the information encoded by the graph [11, 12, 13].
Extending labels in a meaningful way is inherently problematic due to the infinite possible solutions. Therefore, a common strategy is to address this challenge by adopting the semi-

supervised smoothness assumption. This assumption posits that effective labeling functions $x_r : V \to \mathbb{R}_+$ for the $r$-th class, where $x_{i,r}$ denotes the likelihood that node $i \in V \setminus O$ belongs to class $C_r$, should exhibit smoothness in densely connected regions of the graph. In the context of this assumption, smoothness refers to the idea that labeling functions tend to vary gradually across nodes that are closely connected in the graph. This aligns with the notion that nodes with similar characteristics or attributes are expected to possess similar labels. This assumption becomes particularly relevant when the edges of the graph represent some form of similarity or relationship between pairs of nodes.
Consider the following $\ell_2$-based Laplacian regularizer [138]

$$\bar{r}_2(x) = \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(x_i - x_i)^2 \,. \tag{4.1}$$

Minimizing $\bar{r}_2(x)$ while adhering to either hard label constraints, such as $z_i = y_i$ for $i \in O$, or incorporating a soft penalty constraint like the mean squared error $\sum_i (y_i - x_i)^2$ concerning the provided labels $y$, has proven to be an effective strategy for promoting smoothness concerning the graph edges. In both scenarios, the resulting objective function becomes strictly convex, leading to a unique optimal solution for the associated minimization problem.
Despite the widespread popularity and effectiveness of the $\ell_2$-based Laplacian regularizer in various scenarios, it has been shown that this approach can lead to degenerate solutions when the number of input labels in $O$ is very small. In such cases, the learned function $z$ tends to become nearly constant across the entire graph, with abrupt spikes localized near the labeled data points in $O$ [139, 140]. As a response to this limitation, researchers have put forward alternative formulations [141, 142], which encompass strategies based on total variation [143, 91]. Moreover, a class of $p$-Laplacian based regularizers has been introduced to address this issue more comprehensively [139]. This class of regularizers is defined as follows:

$$\bar{r}_p(x) = \frac{1}{p} \sum_{(i,j)\in E} w_{ij}|x_i - x_j|^p \,. \tag{4.2}$$

Note that this modified objective function remains strictly convex. Additionally, the introduction of the $p$-Laplacian based regularizer $\bar{r}_p$ has the effect of discouraging the solution from developing sharp spikes, especially for values of $p$ greater than 2. In this context, higher values of $p$ impose a more substantial penalty on large gradients $|x_i - x_j|$. Conversely, when $1 \leq p < 2$, the objective function encourages sparsity in the gradients. Remarkably, as the value of $p$ approaches 1, the resulting objective function becomes directly linked to graph cuts and modular clustering [144, 145, 146]. A considerable amount of research has been dedicated to studying the behavior of $\bar{r}_p$ across varying values of $p$, particularly in the context of graphs generated using the geometric random graph model [139, 147, 148, 149, 150].
We aim to explore the efficacy of these types of Laplacian regularizers within the context of graph semi-supervised learning while also considering interactions beyond pairwise connections. Numerous complex systems have been effectively modeled as networks, where pairs of interacting nodes are linked. However, real-world applications often require a more nuanced and diverse representation of interactions [6, 4]. On one hand, we have simplicial complexes or hypergraphs,

which provide a suitable framework for describing collective actions involving groups of nodes [151, 152, 153, 154, 155]. On the other hand, we have multilayer networks, i.e., networks that are coupled to each other through different layers, all of them representing different types of relationships between the nodes [59, 14, 16, 112, 17, 107, 108, 114]. Empirical evidence indicates that these tools can significantly enhance modeling capabilities compared to standard single-layer graphs. Multilayer hypergraphs find their natural occurrence in a wide array of applications. Examples include the field of science of science, where nodes represent authors, and in one layer, a hyperedge connects a group of authors who collaborated on a paper, while in another layer, pairs of nodes are linked if they cite each other. In protein networks, nodes correspond to proteins, and connections between them, whether in pairs or groups, are established using various complementary genomic datasets, each forming a distinct layer. Similarly, in social networks, users are represented as nodes, and they can engage in interactions within groups across different platforms, each platform forming a unique layer in the multilayer hypergraph. In our study, we concentrate on multiplex hypergraphs, which are represented by a sequence of hypergraphs (referred to as layers) that share a common set of nodes. Importantly, there are no hyperedges connecting nodes from different layers. Moreover, with the terminology introduced in [9] in the context of multilayer networks, we aim to find a set of communities that is *total* (i.e., every node belongs to at least one community), *node-disjoint* (i.e., no node belongs to more than one cluster on a single layer), and *pillar* (i.e., each node belongs to the same community across the layers). Indeed, the exploration of both multilayer and higher-order structures in complex networks has been relatively limited in the existing literature [156]. This lack is primarily attributed to the increased computational cost associated with obtaining accurate solutions for models involving such intricate structures. In this study, we embark on a preliminary investigation into semi-supervised learning within the context of multilayer hypergraphs, aiming to tackle the inherent complexity that arises from combining these two aspects. This endeavor represents an initial step towards understanding and addressing the challenges posed by these advanced network structures. We approach the problem by employing various coordinate descent strategies and then juxtapose the outcomes with those derived from conventional first-order methodologies, such as gradient descent/label spreading. Even though coordinate descent approaches were used in the literature to deal with other semi-supervised learning problems [157, 158], the analysis reported here represents, to the best of our knowledge, the first attempt to give a thorough analysis of those methods for semi-supervised learning in multilayer hypergraphs. The rest of the chapter is organized as follows. In Section 4.2, we introduce the graph semi-supervised problem and the formulation for multilayer hypergraphs. In Section 4.3, we briefly review the block coordinate descent approaches, and we report some computations needed to apply the methods to our specific problem, pointing out the differences in the special case $p = 2$. In Section 4.4, we report the results of experiments on synthetic and real-world datasets. In Section 4.5, we draw some conclusions.

## 4.2 Problem statement

In this section, we formalize the notation and formulate the problem under analysis. Consider first an undirected and weighted graph $G = (V, E, w)$ with node set $V$ and edge set $E$. Let

$A = (A_{ij})_{i,j \in V}$ be the adjacency matrix of $G$, with weights $A_{ij} = w(e) > 0$ for $e = (i, j) \in E$, measuring the strength of the tie between nodes $i$ and $j$, and $A_{ij} = 0$ if $(i, j) \notin E$. We assume that $V$ can be partitioned into $|C|$ classes $C_1, \dots, C_{|C|}$ and that, only for a few nodes in $O \subset V$, it is known the class $C_r$ to which they belong. The problem consists of assigning the remaining nodes to a class.
Here, we review the approach based on the $p$-Laplacian regularization and the corresponding optimization problem. Define the $(|V| \times |C|)$-dimensional matrix of the input labels $Y$, such that

$$Y_{i,r} = \begin{cases} \dfrac{1}{|C_r \cap O|} & \text{if node } i \in O \text{ belongs to the class } C_r, \\ 0 & \text{otherwise,} \end{cases}$$

where $|C_r \cap O|$ is the cardinality of the known class $C_r$, i.e., the number of nodes that are initially known to belong to $C_r$. Now, let $y^r$ be the $r$-th column of $Y$ and, for all $i \in V$, let $\delta_i$ be the weighted degree of $i$, that is, $\delta_i = \sum_{j \in V} A_{ij}$.
The Laplacian regularized SSL problem boils down to the following minimization problem for all classes $r \in \{1, \dots, |C|\}$:

$$\min_{x \in \mathbb{R}^{|V|}} \|x - y^r\|^2 + \lambda \sum_{i,j=1}^{|V|} A_{ij} \left| \frac{x_i}{\sqrt{\delta_i}} - \frac{x_j}{\sqrt{\delta_j}} \right|^p, \tag{4.3}$$

with given $p \geq 1$ and regularization parameter $\lambda \geq 0$. Equivalently, as the minimization problems above are independent for $r \in \{1, \dots, |C|\}$, we can simultaneously optimize their sum, which can be written in compact matrix notation as

$$\min_{X \in \mathbb{R}^{|V| \times |C|}} \|X - Y\|_{(2)}^2 + \lambda \|W^{1/p} B D^{-1/2} X\|_{(p)}^p, \tag{4.4}$$

where $\|M\|_{(p)}$ denotes the entry-wise $\ell^p$ norm of the matrix $M$, $D$ is the $|V| \times |V|$ diagonal matrix of the graph degrees

$$D = \begin{bmatrix} \delta_1 & 0 & \cdots & 0 \\ 0 & \delta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_{|V|} \end{bmatrix},$$

$B$ is the $|E| \times |V|$ (signed) incidence matrix of the graph, which for any chosen orientation of the edges is entrywise defined as

$$B_{e,i} = \begin{cases} 1 & \text{if node } i \text{ is the source of edge } e, \\ -1 & \text{if node } i \text{ is the tip of edge } e, \\ 0 & \text{otherwise,} \end{cases}$$

and $W$ is the diagonal $|E| \times |E|$ matrix of the edge weights $W_{e,e} = w(e)$. Note that, even though we are dealing with undirected graphs, $B$ requires fixing an orientation for the edges of $G$.

However, all the arguments presented here are independent of the chosen orientation. For $p = 2$, a direct computation shows that the the optimal solution $X^*$ of the above problem is entrywise nonnegative. The same property carries over to any $p \geq 1$, as one can interpret the minimizer of (4.4) as the smallest solution of a $p$-Laplacian eigenvalue equation on $G$ with boundary conditions, see e.g. [159]. Thus, we can interpret the entry $X^*_{i,r} \geq 0$ as a score that quantifies how likely it is for the node $i \in V$ to belong to the class $C_r$ and we then assign each node $i \in V$ to the class $r^* \in \operatorname{argmax}_{r=1,\ldots,|C|} X^*_{i,r}$.

Now, we want to extend the formulation (4.4) to the case where rather than a graph $G$, we have a multilayer hypergraph $H$. Specifically, assume that we have $k$ layers $H_1, \ldots, H_k$, where $H_s(V, E_s)$ is the hypergraph forming the $s$th layer and $E_s$ is a hyperedge set, that is, $E_s$ contains interactions of order greater than 2. In other words, each $e \in E_s$ is a set of arbitrarily many nodes, weighted by $w_s l(e) > 0$. The topological information of a hypergraph $H_s$ can be all included in the (signless) incidence matrix $K^{(s)} \in \mathbb{R}^{|E_s| \times |V|}$, defined as $K^{(s)}_{e,i} = 1$ if $i \in e$, and $K^{(s)}_{e,i} = 0$ if $i \notin e$, for all $i \in V$ and $e \in E_s$, see e.g. [96, 6, 160]. Using $K^{(s)}$, we can represent each $H_s(V, E_s)$ via a clique-expanded graph $G(H_s)$, which corresponds to the adjacency matrix

$$A^{(s)} = K^{(s)T} W^{(s)} K^{(s)} - D^{(s)},$$

with $W^{(s)}$ being the $|E_s| \times |E_s|$ diagonal matrix of the relative hyperedge weights, defined as

$$W^{(s)}_{e,e} = \frac{w_s(e)}{|e|} > 0,$$

and $D^{(s)}$ being the diagonal matrix of the node degrees of the hypergraph $H_s$, defined as

$$D^{(s)}_{i,i} = (\delta_s)_u = \sum_{e \in E} w_s(e) |e|^{-1} K^{(s)}_{e,i} = (K^{(s)T} W^{(s)} K^{(s)})_{i,i}\,.$$

Note that the edge $(i, j)$ is in the resulting clique-expanded graph $G(H_s)$ if and only if $i \neq j$ and there exists at least one hyperedge in $E_s$ such that both $i \in E_s$ and $j \in E_s$. In that case, the weight of the edge $(i, j)$ in $G(H_s)$ is

$$A^{(s)}_{i,j} = \sum_{e : i,j \in e} \frac{w_s(e)}{|e|}$$

Using the clique-expanded representation of $H_s$ has the advantage that one can directly transfer established techniques from the graph literature to the hypergraph case. However, note that the clique expansion is not always a good representation of a hypergraph, as a given clique expansion can represent more than one hypergraph (while one hypergraph implies one particular clique expansion, i.e. the mapping $H_s \mapsto G(H_s)$ is not bijective). Moreover, the choice of the weights in the clique-expanded adjacency matrix $A^{(s)}$ is somewhat ambiguous as one can equivalently choose any positive function of the original hyperedge weights. At the same time, we notice that if the average size of the hyperedges is not much larger than two, thus interactions are mostly pairwise, using the clique-expansion should be a good approximation of the original hyper-edge structure. Other possible drawbacks are due to the fact that the clique-expansion projection results in

a graph that is considerably denser than the original input data. This increased density may not accurately reflect the correlation with the actual underlying hyperedges, potentially causing a distortion of the observations provided in the initial input [161, 162]. Despite its potential drawbacks, here we decided to use the clique expansion of the hypergraph and to consider this definition of assigning weights to edges since it is by far the most popular and used technique in this context and often leads to very good performance in homophilic node classification tasks [163, 164].

Proceeding as before, we can define $B_s$ as the signed incidence matrix of $G(H_s)$ and we can sum the corresponding regularization terms across all the layers, obtaining the following formulation:

$$\min_{X\in\mathbb{R}^{|V|\times|C|}} \vartheta(X) := f(X) + \bar{r}_p(X) \tag{4.5}$$
$$\text{where } f(X) = \|X-Y\|_{(2)}^2, \quad \bar{r}_p(X) = \sum_{s=1}^{k} \lambda_s \|W^{(s)1/2} B_s D^{(s)-1/2} X\|_{(p)}^p,$$

where $\lambda_1, \ldots, \lambda_k \geq 0$ are regularization parameters. Note that, if $H$ is a standard graph, i.e., if $|e| = 2$ for all edges and $k = 1$, then (4.5) boils down to (4.4), up to the constant term $1/|e| = 1/2$. Note moreover that, as in the graph case, we can equivalently write the objective function $\vartheta(X)$ as $\sum_c \vartheta_c(x^r)$, where $x^r$ is the $r$-th column of $X$, and

$$\vartheta_r(x) = \|x - y^r\|_2^2 + \sum_{s=1}^{k} \lambda_s \sum_{e\in E_s} \frac{w_s(e)}{|e|} \sum_{i,j\in e} \left| \frac{x_i}{\sqrt{(\delta_s)_i}} - \frac{x_j}{\sqrt{(\delta_s)_j}} \right|^p.$$

The above expression shows that the regularizers $\vartheta_r$ enforce a form of higher-order smoothness assumption in the solution across all the nodes of each layer's hyperedge by imposing the minimizer $X^*$ to have similar values on pairs of nodes in the same hyperedge. This immediately justifies the choice of the objective function (4.5) for SSL on multilayer hypergraphs. Also note that, as in the graph setting, the optimal solution $X^*$ to (4.5) has to be entrywise nonnegative and thus, once $X^*$ is computed, we can assign each node $i \in V$ to the class $r^* \in \operatorname{argmax}_{r=1,\ldots,|C|} X^*_{i,r}$.

## 4.3 Block coordinate descent approaches

When confronted with extensive optimization challenges, as encountered in the domain of semi-supervised learning within real-world multilayer hypergraphs, traditional optimization approaches may become unwieldy. In such contexts, block coordinate descent methods present themselves as valuable tools for attaining enhanced computational efficiency. In each iteration of a block coordinate descent method, a subset of variables, known as a working set, is meticulously chosen and systematically updated, while the remaining variables are held constant. The overarching framework for a block coordinate descent method, aimed at minimizing an objective function $f(x)$, is outlined in Algorithm 3.

In the existing literature, numerous block coordinate descent methods have been introduced, catering to both unconstrained and constrained optimization problems. These methods vary in their strategies for calculating $W^n$ and $t^n$ (refer to [20] and its references for more details). When

**Algorithm 3** Generic block coordinate descent method

1: **Given** $x^0 \in \mathbb{R}^{|V|}$
2: **For** $n = 0, 1, \ldots$
3: Choose a working set $W^n \subseteq \{1, \ldots, |V|\}$
4: Compute $t^n \in \mathbb{R}^n$ such that $t_i^n = 0$ for all $i \notin W^n$
5: Set $x^{n+1} = x^n + t^n$
6: **End for**

determining the working set $W^n$, a viable option is the adoption of a *cyclic rule*, alternatively referred to as the *Gauss-Seidel rule*[165]. This technique entails partitioning the variables into distinct blocks and selecting each block in a sequential rotation. A more extensive form of this approach is the *essentially cyclic rule* or the *almost cyclic rule*[166], which mandates that each variable block must be selected at least once within a predetermined number of iterations.
In unconstrained optimization, blocks can be as minimal as containing only one variable. In such cases, each update (i.e., the computation of $h^n$) can be executed through either an exact or an inexact minimization approach [165, 166, 167, 168, 169]. These methods have been extended to constrained optimization scenarios as well, and the nature of the constraints can influence the composition of the variable blocks. For instance, non-separable constraints might necessitate variable blocks that encompass more than one variable [165, 170, 171, 172, 173, 174, 175]. One notable advantage of cyclic-based rules lies in the fact that, during each iteration, only a small fraction of the components of $\nabla f$ need to be computed. This can lead to exceptional efficiency, particularly when calculating an individual component of $\nabla f$ is substantially less resource-intensive than evaluating the entire gradient vector.
Another strategy for selecting the working set is to employ a *random rule*, which entails the computation of $W^n$ based on a random distribution. These methods are often referred to as *random coordinate descent methods* and exhibit favorable convergence properties in expectation, both for unconstrained optimization [176, 177] and constrained problems [178, 179, 180, 181, 182]. Notably, random rules, along with cyclic rules, do not rely on first-order information to determine the working set. This aspect contributes to their high efficiency, particularly in scenarios where computing a single component of $\nabla f$ is significantly less resource-intensive than calculating the entire gradient vector.
An alternative approach for selecting the working set $W^n$ is to employ a *greedy rule*, often referred to as the *Gauss-Southwell rule*. This strategy involves choosing, at each iteration, a block containing the variable(s) that most significantly violate a specified optimality condition. In the context of unconstrained optimization, one potential choice for the working set is the block associated with the largest absolute gradient component. It's worth noting that this rule can facilitate more substantial progress in the objective function, as it leverages first (or higher) order information to determine the working set. However, it may be computationally more expensive compared to cyclic or random selection methods. Exact or inexact minimizations can be employed to update the variables using this rule [166, 168, 183, 184], and extensions to constrained settings have been explored as well [185, 186, 187, 188, 189]. Recent research has demonstrated that certain problem structures allow for efficient calculation of this class of rules

in practice, mitigating some of the potential computational challenges (refer to [187] and its references for further insights).

**Algorithm 4** Block coordinate descent method for problem (4.5) - matrix form

1: **Given** $X^0 \in \mathbb{R}^{|V|\times|C|}$
2: **For** $n = 0, 1, \ldots$
3: Choose a working set $W^n = W_1^n \times \ldots \times W_{|C|}^n \subseteq \{1, \ldots, |V|\}^{|C|}$
4: Compute $T^n \in \mathbb{R}^{|V|\times|C|}$ such that $T_{ir}^n = 0$ for all $i \notin W_r^n$
5: Set $X^{n+1} = X^n + T^n$
6: **End for**

In this work, we adapt block coordinate descent methods to solve problem (4.5), leading to the method reported in Algorithm 4. In particular, we start with a matrix $X^0 \in \mathbb{R}^{|V|\times|C|}$ and, at each iteration $n$, we choose a working set $W_r^n$ for each class $r \in \{1, \ldots, |C|\}$. We highlight that problem (4.5) solves the same problem for the different classes $C_r$ with $r = 1, \ldots, |C|$ in a matrix form, but each of them is independent and can eventually be solved in parallel.

### 4.3.1 Coordinate descent approaches

Here, we focus on block coordinate descent approaches that use blocks $W_r^n$ of dimension 1, i.e., $W_r^n = \{i_r^n\}$, with $i_r^n$ being a variable index for class $r$ at iteration $n$. Then, $X^{n+1}$ is obtained by moving the variables $X_{i_r^n r}^n$ along $-\nabla_{i_r^n r}\vartheta(X^n)$ with a proper stepsize $\alpha_r^n$. Namely, for any class $r \in \{1, \ldots, |C|\}$,

$$X_{hr}^{n+1} = \begin{cases} X_{hr}^n - \alpha_r^n \nabla_{hr}\vartheta(X^n) & \text{if } h = i_r^n, \\ X_{hr}^n & \text{otherwise.} \end{cases} \tag{4.6}$$

Taking into account the possible choices described in Section 4.3, we consider the following algorithms:

- **Cyclic Coordinate Descent (CCD).** At every iteration $n$, a variable index $i^n \in \{1, \ldots, |V|\}$ is chosen in a cyclic fashion (i.e., by a Gauss-Seidel rule), and then $X^{n+1}$ is obtained as in (4.6) by setting $i_r^n = i^n$ for all $r \in \{1, \ldots, |C|\}$. A random permutation of the variables every $n$ iteration is also used since it is known that this might lead to better practical performances in several cases (see, e.g., [20, 190]).

- **Random Coordinate Descent (RCD).** At every iteration $n$, a variable index $i^n \in \{1, \ldots, |V|\}$ is randomly chosen from a uniform distribution, and then $X^{n+1}$ is obtained as in (4.6) by setting $i_r^n = i^n$ for all $r \in \{1, \ldots, |C|\}$.

- **Greedy Coordinate Descent (GCD).** At every iteration $n$, a variable index $i_r^n \in \{1, \ldots, |V|\}$ is chosen for every class $r \in \{1, \ldots, |C|\}$ as

$$i_r^n \in \underset{i=1,\ldots,|V|}{\operatorname{argmax}} |\nabla_{ir}\vartheta(X^n)|$$

  (i.e., by a Gauss-Southwell rule), and then $X^{n+1}$ is obtained as in (4.6).

The GCD method can provide favorable convergence rates under appropriate conditions [187]. However, it's worth noting that this method can become computationally expensive in practice due to the need to evaluate the entire gradient and search for the optimal index to select the block for updating at each iteration.
This becomes particularly relevant when dealing with large-scale problems encountered in semi-supervised learning scenarios. To practically implement these methods, specific strategies need to be developed. To practically implement those methods, specific strategies hence need to be implemented. Importantly, in our context where semi-supervised learning problems often exhibit sparsity, it's possible to efficiently implement the basic GCD rule. Techniques such as tracking the gradient element using a max-heap structure and employing caching strategies can help mitigate computational challenges, as discussed in works like [187].
Given that the practical efficiency of coordinate descent methods hinges on the implementation details, we provide comprehensive calculations outlining the steps required to update the gradient of the objective functions at a specific iteration in the next section. This information serves as a guide to ensure the proper execution of the algorithm and achieve the desired computational efficiency.

### 4.3.2 Calculations

To compare the gradient descent method to block coordinate descent approaches, we need to calculate the gradient of the function $\vartheta(X)$ that we want to minimize in (4.5). The gradient of $\vartheta(X)$ can be expressed as:

$$\nabla\vartheta(X) = 2(X - Y) + p\sum_{s=1}^{L}\lambda_s\mathcal{L}_s^p(X), \tag{4.7}$$

where $\mathcal{L}_s^p(X)$ is the normalized $p-$laplacian and it is applied on each column of $X$ in this way:

$$\mathcal{L}_s^p(X) = (B_s D_s^{-\frac{1}{2}})^T \phi_p(B_s D_s^{-\frac{1}{2}} X),$$

with $\phi_p(y) = |y|^{p-1}\text{sgn}(y)$ component-wise.

At the beginning, we calculate $B_s D_s^{-\frac{1}{2}}\ \forall s \in \{1, .., k\}$ layer. Then, at each iteration $n$, the gradient is calculated iteratively. Break the formula of the gradient in (4.7) into two parts:

$$\nabla\vartheta(X) = \nabla f(X) + \nabla\bar{r}_p(X),$$

with

$$\nabla f(X) = 2(X - Y),$$
$$\nabla\bar{r}_p(X) = p\sum_{s=1}^{k}\lambda^s\mathcal{L}_s^p(X).$$

Then,

$$\nabla f(X^{n+1}) = \nabla f(X^n) + 2(X^{n+1}_{W^n} - X^n_{W^n}),$$
$$\nabla \bar{r}_p(X^{n+1}) = p \sum_{s=1}^{n} \lambda^s \mathcal{L}^p_s(X^{n+1}),$$

where $\mathcal{L}^p_s(X^{n+1})$ can be iteratively calculated using

$$B_s D_s^{-\frac{1}{2}} X^{n+1} = B_s D_s^{-\frac{1}{2}} X^n + (B_s D_s^{-\frac{1}{2}})_{W^n}(X^{n+1}_{W^n} - X^n_{W^n})$$

with the appropriate subscript $W^n$ to take just the $W^n_c$ coordinates of column $c$, for all $c \in \{1, \ldots, |C|\}$.

**Special case** $p = 2$

In this section, we discuss the special case of problem (4.5) with $p = 2$. The optimization problem (4.5) is equivalent to:

$$\min_{X \in \mathbb{R}^{|V| \times |C|}} \|X - Y\|^2_{(2)} + \sum_{s=1}^{k} \lambda_s X^T \bar{\mathcal{L}}_s X,$$

where $\bar{\mathcal{L}}_s = I - \bar{A}^{(s)}$ is the normalized laplacian matrix of layer $s = 1, \ldots, k$ and $\bar{A}^{(s)}$ is the normalized adjacency matrix of layer $s = 1, \ldots, k$ with entries

$$(\bar{A}^{(s)})_{ij} = \frac{(A^{(s)})_{ij}}{\sqrt{(\delta_s)_i}\sqrt{(\delta_s)_j}}.$$

In this case, the gradient of the function to minimize can be expressed as

$$\nabla_X \vartheta(X) = 2(X - Y) + \sum_{s=1}^{k} 2\lambda_s \bar{\mathcal{L}}_s X$$

and the Hessian as $2I + \sum_{s=1}^{k} 2\lambda_s \bar{\mathcal{L}}_s$. In the experiments where $p = 2$, this last expression can be used in the calculation of the step size.

## 4.4 Experiments

In the context of semi-supervised learning, first-order methods like gradient descent and label spreading are widely utilized [13, 191, 192]. Consequently, we compare the coordinate descent approaches described in Subsection 4.3.1, namely the Cyclic Coordinate Descent method (CCD), Random Coordinate Descent method (RCD), and Greedy Coordinate Descent method (GCD), with the Gradient Descent (GD) algorithm in our experiments.
In the first setting, when $p = 2$, the objective function is quadratic and we used a step size depending on the coordinatewise Lipschitz constants (see, e.g., [193]). We highlight that while

calculating the coordinatewise Lipschitz constants or a good upper bound is pretty straightforward in the considered case for coordinate approaches, the calculation of the global Lipschitz constant might get expensive for GD (especially when dealing with large-scale instances). For the $p \neq 2$ setting, for simplicity, we used a step size depending on an upper bound of the Lipschitz constants for all the methods (see, e.g., [187, 194, 195]). The performance of the coordinate descent algorithms might, of course, be further improved by choosing a more sophisticated coordinate-dependent stepsize strategy [196, 197, 198, 199]. In the first scenario, when $p = 2$, the objective function is quadratic. We employed a stepsize strategy that relies on the coordinatewise Lipschitz constants, as described in, for instance, [193]. It's worth noting that computing these coordinatewise Lipschitz constants or finding a good upper bound for them is relatively straightforward in the context of coordinate-based approaches. However, when it comes to gradient descent (GD), particularly with large-scale instances, calculating the global Lipschitz constant can be computationally expensive. For the case when $p \neq 2$, we adopted a stepsize strategy that depends on an upper bound of the Lipschitz constants for all the methods, as outlined in, for instance, [187, 194, 195]. It's important to acknowledge that the performance of coordinate descent algorithms can potentially be further enhanced by employing more sophisticated stepsize strategies that are tailored to the specific coordinates, as discussed in references like [196, 197, 198, 199]. To highlight the benefits of using coordinate methods compared to gradient descent-like approaches and to showcase the practical efficiency of these methods, we conducted comprehensive experiments on both synthetic and real-world datasets.

In our evaluation, we present efficiency plots for both the objective function and the accuracy of the final partition, assessed on the subset of unlabeled nodes. For our performance comparison, we utilize the number of flops, which refers to one-dimensional moves, as our metric. To elaborate, when dealing with a graph containing $N$ nodes, the Gradient Descent (GD) algorithm uses $N$ flops per iteration, indicating that it updates all $N$ components of the iterate in a single iteration. In contrast, the coordinate methods require just one flop per iteration, as they modify only one component at a time. It's worth mentioning that, as previously highlighted in [187], this measurement isn't perfect, particularly for greedy methods, as it disregards the computational expense of each iteration. Nonetheless, it offers an implementation- and problem-independent gauge of efficiency. Additionally, in our scenario, it's relatively straightforward to estimate the cost per iteration, which is minimal when the strategy is effectively implemented. Consequently, we will observe how a faster-converging method like Greedy Coordinate Descent (GCD) leads to substantial performance improvements in the context of the considered application.

We fixed the regularization parameters at $\lambda_s = 1$ for $s = 1, \ldots, k$ and we initialized the methods with $X^0 = 0$. We implemented all the methods using Matlab.[1] We want to underscore that our decision to set the parameters $\lambda_\ell = 1$ does not impact the performance analysis conducted in this study. This choice was made to ensure a fair and equal balance across all layers for the sake of our analysis. However, it's important to note that in practical applications, determining these parameters may involve a non-trivial tuning phase. Typically, this tuning can be based on either the model's characteristics or the data itself. For more insights on parameter tuning, you can refer to references like [19, 16, 108].

[1]Our codes are available at the GitHub page: `https://github.com/saraventurini/Semi-Supervised-Learning-in-Multilayer-Hypergraphs-by-Coordinate-Descent`.

### 4.4.1 Synthetic datasets

We generated synthetic datasets using the Stochastic Block Model (SBM) [200], already explained in §2.5.1.
It's important to note that solving the problem described by Eq. (4.5) on a multilayer hypergraph can be equivalently tackled by addressing the same problem on a simple graph. In this scenario, the adjacency matrix is formed by the weighted sum of the adjacency matrices of the clique-expanded graphs from each individual layer. Consequently, for comparison, we generated single-layer datasets by maintaining a fixed $p_{in}$ value of 0.2, while varying the ratio $p_{in}/p_{out}$ across values such as 3.5, 3, 2.5, and 2. To be more specific, we crafted networks with four distinct communities, each consisting of 125 nodes. During our experimentation, we also varied the proportion of known labels per community, denoted as *perc*, which was examined at levels of $3\%, 6\%, 9\%$, and $12\%$. These percentages correspond to having 3, 7, 11, and 15 known nodes per community, respectively. We focused our investigation on optimizing the problem outlined in Eq. (4.5) while keeping the parameter $p$ constant at 2. Our experimentation procedure entailed sampling 5 random instances for each combination of $(p_{out}, perc)$, and subsequently calculating average scores. The results have been visualized in Figures 4.1 and 4.2, where the objective function values and accuracy are plotted about the number of flops. Each row in these figures corresponds to a specific ratio value of $p_{in}/p_{out}$, ranging from 2 to 3.5 (from top to bottom). In addition, each column represents distinct percentages of known labels *perc* (ranging from 3% to 12%, left to right). In Table 4.1, we have compiled summarized results related to the objective function and accuracy, using synthetic datasets (see Figures 4.1 and 4.2). For each method, we present the average and standard deviation of the number of floating-point operations (flops) normalized by the total number of nodes in the network, necessary to achieve a specific level of objective function or accuracy. This level is determined by a parameter called a "gate" which serves as a convergence tolerance, akin to the approach in [201]. Additionally, we provide information about the fraction of failures, which denotes the proportion of instances where a method fails to converge within a specified number of iterations (four times the number of nodes). The reported averages exclude instances of failure, and if all instances failed, a hyphen is displayed. The provided plots and tables clearly illustrate that the Greedy Coordinate Descent method (GCD) consistently achieves favorable results in terms of both the objective function value and accuracy with a much lower number of flops compared to the other methods under consideration. While the other coordinate methods appear to be slower in terms of attaining a favorable objective function value when compared to Gradient Descent (GD), they exhibit faster convergence in terms of accuracy. Consequently, given the appropriate selection of coordinates, a coordinate method has the potential to outperform GD in practical applications. This highlights the efficiency advantage of employing coordinate methods, especially the GCD variant, in addressing optimization problems like the one investigated in this study.

### 4.4.2 Real datasets

We further consider seven real-world datasets frequently used for assessing algorithm performance in graph clustering (information can be found in the GitHub repository) [131, 202]:

- *3sources*: 169 nodes, 6 communities, 3 layers;

Table 4.1: Aggregated results of the objective function (upper table) and the accuracy (lower table) across the synthetic datasets with $p = 2$ (see Figures 4.1 and 4.2). Using a tolerance *gate*, for each algorithm *flop* indicates the normalized number of flops (mean ± standard deviation) and *fail* indicates the fraction of failures (i.e., stopping criterion not satisfied within the maximum number of iterations, set equal to 4 times the number of nodes). The averages are calculated without considering the failures and, in case of all failures, a hyphen is reported.

| | CCD | | RCD | | GCD | | GD | |
|---|---|---|---|---|---|---|---|---|
| gate | flop | fail | flop | fail | flop | fail | flop | fail |
| 0.75 | 0.65±0.07 | 0.00 | 0.80±0.06 | 0.00 | 0.15±0.04 | 0.00 | 2.00±0.00 | 0.00 |
| 0.5 | 0.66±0.04 | 0.00 | 0.88±0.06 | 0.00 | 0.02±0.01 | 0.00 | 1.00±0.00 | 0.00 |
| 0.25 | 1.05±0.04 | 0.00 | 1.83±0.06 | 0.25 | 0.02±0.01 | 0.00 | 1.00±0.00 | 0.00 |
| 0.1 | 1.82±0.05 | 0.00 | 3.08±0.16 | 0.00 | 0.04±0.02 | 0.00 | 1.00±0.00 | 0.00 |
| 0.05 | 2.44±0.08 | 0.00 | 3.81±0.06 | 0.50 | 0.05±0.03 | 0.00 | 1.00±0.00 | 0.00 |

| | CCD | | RCD | | GCD | | GD | |
|---|---|---|---|---|---|---|---|---|
| gate | flop | fail | flop | fail | flop | fail | flop | fail |
| 0.75 | 0.65±0.07 | 0.00 | 0.80±0.06 | 0.00 | 0.15±0.04 | 0.00 | 2.00±0.00 | 0.00 |
| 0.5 | 1.06±0.15 | 0.00 | 1.71±0.28 | 0.00 | 0.24±0.03 | 0.00 | 2.00±0.00 | 0.00 |
| 0.25 | 1.75±0.13 | 0.00 | 3.28±0.29 | 0.25 | 0.54±0.27 | 0.00 | 2.44±0.51 | 0.00 |
| 0.1 | 3.07±0.53 | 0.00 | - | 1.00 | 0.82±0.28 | 0.00 | 3.00±0.00 | 0.00 |
| 0.05 | 3.62±0.32 | 1.38 | - | 1.00 | 1.02±0.32 | 0.00 | 3.50±0.52 | 0.00 |

- *BBCSport*: 544 nodes, 5 communities, 2 layers;
- *Wikipedia*: 693 nodes, 10 communities, 2 layers;
- *UCI*: 2000 nodes, 10 communities, 6 layers;
- *cora*: 2708 nodes, 7 communities, 2 layers;
- *primary-school*: 242 nodes, 11 communities, 2.4 mean hyperedge size;
- *high-school*: 327 nodes, 9 communities, 2.3 mean hyperedge size.

The first five datasets in the list are related to multilayer graphs (which have already been mentioned in §2.5.3), while the last two are related to single-layer hypergraphs.
We tested the methods considering different percentages of known labels per community, sampling them randomly 5 times and showing the average scores. In particular, we suppose to know $perc \in [3\%, 6\%, 9\%, 12\%]$ percentage of nodes per community in all the datasets except for *Wikipedia*, where we considered to know a higher percentage of nodes, $perc \in [15\%, 18\%, 21\%, 24\%]$, to have significant results.
The results corresponding to quadratic regularization in equation (4.5) (with $p = 2$) are presented and analyzed. In Figures 4.3 and 4.4, the average values of the objective function are shown, while Figures 4.5 and 4.6 display the corresponding accuracy values. In Table 4.2, we present aggregated results of the objective function and the accuracy, as explained in Section 4.4.1. The

Table 4.2: Aggregated results of the objective function (upper table) and the accuracy (lower table) across the real datasets with $p = 2$ (see Figures 4.3-4.4 and Figures 4.5-4.6). The table indices are the same as in Table 4.1.

| | CCD | | RCD | | GCD | | GD | |
|---|---|---|---|---|---|---|---|---|
| gate | flop | fail | flop | fail | flop | fail | flop | fail |
| 0.75 | 0.39± 0.06 | 0.00 | 0.40±0.02 | 0.00 | 0.01±0.00 | 0.00 | 1.00±0.00 | 0.00 |
| 0.5 | 0.74±0.08 | 0.00 | 1.06±0.09 | 0.00 | 0.01±0.00 | 0.00 | 1.00±0.00 | 0.00 |
| 0.25 | 1.30±0.24 | 0.00 | 2.06±0.16 | 0.00 | 0.02±0.01 | 0.00 | 1.00±0.00 | 0.00 |
| 0.1 | 2.16±0.40 | 0.00 | 3.37±0.36 | 0.07 | 0.03±0.01 | 0.00 | 1.32±0.48 | 0.00 |
| 0.05 | 2.90±0.49 | 0.00 | 3.49±0.00 | 0.96 | 0.04±0.02 | 0.00 | 1.61±0.50 | 0.00 |

| | CCD | | RCD | | GCD | | GD | |
|---|---|---|---|---|---|---|---|---|
| gate | flop | fail | flop | fail | flop | fail | flop | fail |
| 0.75 | 0.69±0.17 | 0.00 | 0.97±0.28 | 0.00 | 0.13±0.08 | 0.00 | 2.11±0.31 | 0.00 |
| 0.5 | 0.99±0.20 | 0.00 | 1.61±0.31 | 0.00 | 0.25±0.16 | 0.00 | 2.15±0.36 | 0.00 |
| 0.25 | 1.51±0.26 | 0.00 | 2.78±0.49 | 0.00 | 0.41±0.24 | 0.00 | 2.32±0.48 | 0.00 |
| 0.1 | 2.15±0.52 | 0.00 | 3.21±0.23 | 0.82 | 0.67±0.39 | 0.00 | 2.71±0.54 | 0.04 |
| 0.05 | 2.70±0.56 | 0.29 | 3.86±0.00 | 0.96 | 0.91±0.52 | 0.00 | 3.10±0.44 | 0.25 |

patterns observed in these results mirror the findings from the analysis of synthetic datasets. Specifically, the Greedy Coordinate Descent method (GCD) consistently reaches a favorable solution in terms of both the objective function and accuracy at a significantly lower number of flops compared to the other methods.

To investigate the influence of the regularization parameter $p$ on the methods' behavior and the accuracy of the results, we conducted experiments with different values of $p$, both larger and smaller than 2. Specifically, we considered values of $p$ in the set $1.8, 1.9, 2.25, 2.5$, with a fixed percentage of known labels ($perc = 6\%$) or, for the Wikipedia dataset, $perc = 18\%$. Figures 4.7 and 4.8 present the average values of the objective function, while Figures 4.9 and 4.10 display the corresponding accuracy values. In Table 4.3, we present aggregated results of the objective function and the accuracy, as explained in Section 4.4.1. The observed patterns indicate that the methods' behavior remains relatively consistent across varying values of $p$. The GCD method consistently outperforms the others in terms of the number of flops required for convergence. Analyzing the objective function values, the CCD method exhibits a behavior similar to that of the GD method. Meanwhile, the RCD method performs less favorably in both objective function and accuracy. It's worth noting that selecting $p \neq 2$ can lead to improvements in the final accuracy. In Table 4.4, we report the maximum value of accuracy achieved in the real datasets with fixed $perc = 6\%$ (resp. $perc = 18\%$ for Wikipedia) and varying $p \in \{1.8, 1.9, 2, 2.25, 2.5\}$.

## 4.5 Conclusion

In this study, we conducted a comprehensive comparison of various coordinate descent methods against the standard Gradient Descent approach for solving an optimization-based formulation of

Table 4.3: Aggregated results of the objective function (upper table) and the accuracy (lower table) across the real datasets with $p \neq 2$ (see Figures 4.7-4.8 and Figures 4.9-4.10). The table indices are the same as in Table 4.1.

| | CCD | | RCD | | GCD | | GD | |
|---|---|---|---|---|---|---|---|---|
| gate | flop | fail | flop | fail | flop | fail | flop | fail |
| 0.75 | 0.35±0.06 | 0.00 | 0.35±0.05 | 0.00 | 0.01±0.00 | 0.00 | 1.00±0.00 | 0.00 |
| 0.5 | 0.66±0.09 | 0.00 | 0.93±0.14 | 0.00 | 0.01±0.00 | 0.00 | 1.00±0.00 | 0.00 |
| 0.25 | 1.06±0.26 | 0.00 | 1.83±0.21 | 0.00 | 0.01±0.01 | 0.00 | 1.43±0.50 | 0.00 |
| 0.1 | 1.68±0.49 | 0.00 | 2.99±0.43 | 0.00 | 0.02±0.01 | 0.00 | 1.93±0.60 | 0.00 |
| 0.05 | 2.13±0.72 | 0.00 | 3.40±0.39 | 0.50 | 0.03±0.01 | 0.00 | 2.50±0.29 | 0.00 |

| | CCD | | RCD | | GCD | | GD | |
|---|---|---|---|---|---|---|---|---|
| gate | flop | fail | flop | fail | flop | fail | flop | fail |
| 0.75 | 0.70±0.16 | 0.00 | 0.97±0.28 | 0.00 | 0.09±0.03 | 0.00 | 2.15±0.36 | 0.00 |
| 0.5 | 1.01±0.19 | 0.00 | 1.64±0.29 | 0.00 | 0.23±0.24 | 0.00 | 2.15±0.36 | 0.00 |
| 0.25 | 1.51±0.26 | 0.00 | 2.72±0.45 | 0.00 | 0.41±0.45 | 0.00 | 2.36±0.49 | 0.00 |
| 0.1 | 1.93±0.26 | 0.00 | 3.46±0.46 | 0.68 | 0.72±0.76 | 0.00 | 2.90±0.57 | 0.00 |
| 0.05 | 2.39±0.54 | 0.04 | 3.90±0.11 | 0.89 | 0.94±0.90 | 0.00 | 3.05±0.56 | 0.18 |

the Graph Semi-Supervised Learning problem on multilayer hypergraphs. Our extensive experiments encompassed both synthetic and real-world datasets, revealing the superior convergence speed of well-chosen coordinate methods in contrast to the Gradient Descent approach. This outcome underscores the potential of developing specialized coordinate methods tailored to the resolution of semi-supervised learning problems in this context. Additionally, we explored the impact of replacing the standard quadratic regularization term in the objective function with a more generalized $p$-regularizer. The results presented in this context clearly show that this modification can lead to better performances. This study thus contributes to our understanding of effective optimization strategies for complex semi-supervised learning tasks involving multilayer hypergraphs.

Table 4.4: The maximum value of accuracy achieved in the real datasets with fixed $perc = 6\%$ (resp. $perc = 18\%$ for Wikipedia) and varying $p \in \{1.8, 1.9, 2, 2.25, 2.5\}$.

| | $p$ | | | | |
|---|---|---|---|---|---|
| dataset | 1.8 | 1.9 | 2 | 2.25 | 2.5 |
| 3sources | 0.84 | 0.82 | 0.79 | 0.76 | 0.74 |
| BBCSport | 0.91 | 0.89 | 0.87 | 0.85 | 0.83 |
| Wikipedia | 0.60 | 0.58 | 0.56 | 0.53 | 0.50 |
| UCI | 0.94 | 0.93 | 0.91 | 0.88 | 0.86 |
| cora | 0.71 | 0.69 | 0.66 | 0.62 | 0.60 |
| primary school | 0.89 | 0.85 | 0.82 | 0.77 | 0.75 |
| high school | 0.96 | 0.95 | 0.93 | 0.89 | 0.87 |

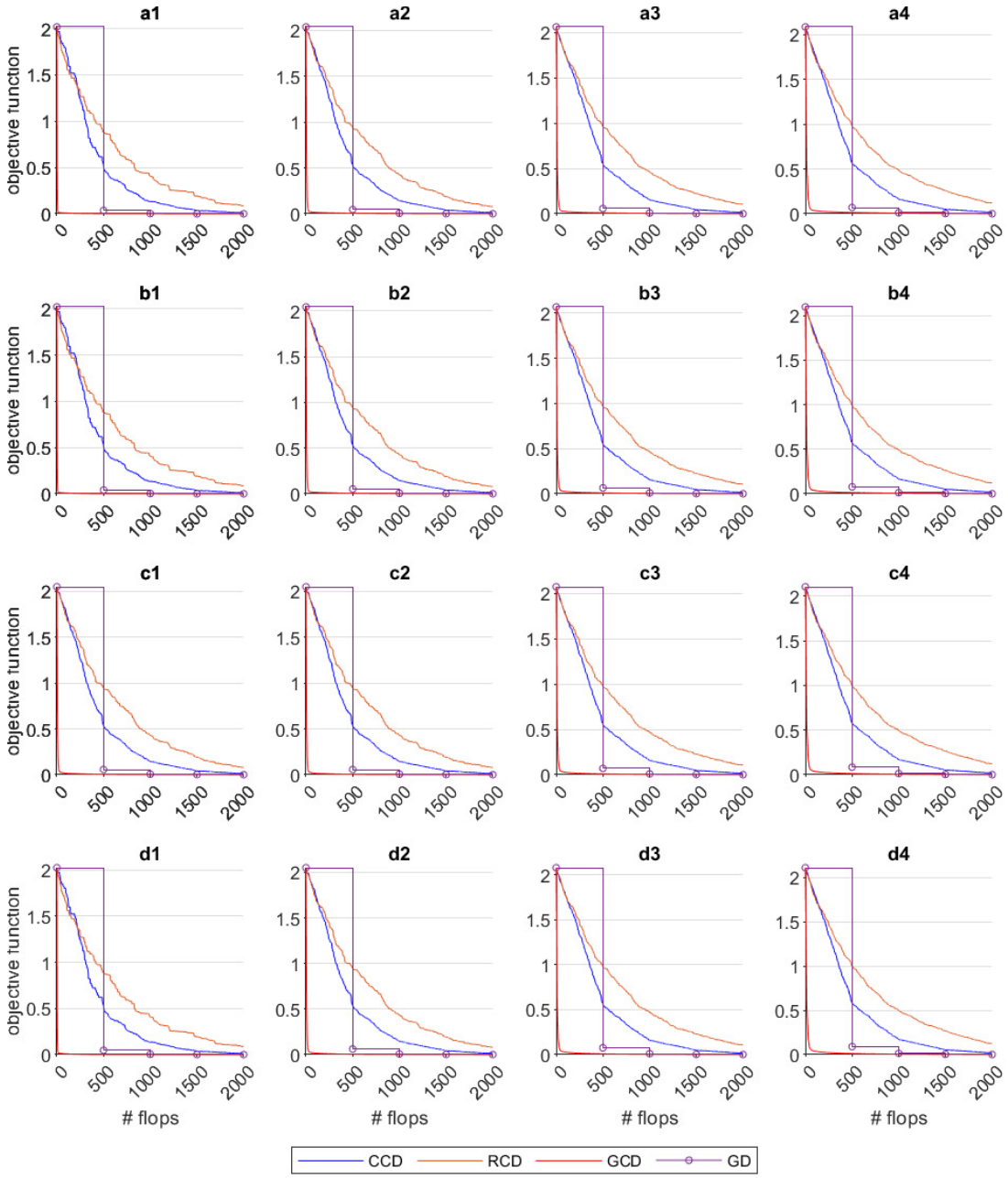


Figure 4.1: Average values of the objective function over 5 random networks sampled from SBM for $p = 2$, $p_{in} = 0.2$, $\frac{p_{in}}{p_{out}} \in \{2, 2.5, 3, 3.5\}$ varies in the rows and $perc \in [3\%, 6\%, 9\%, 12\%]$ varies in the columns.

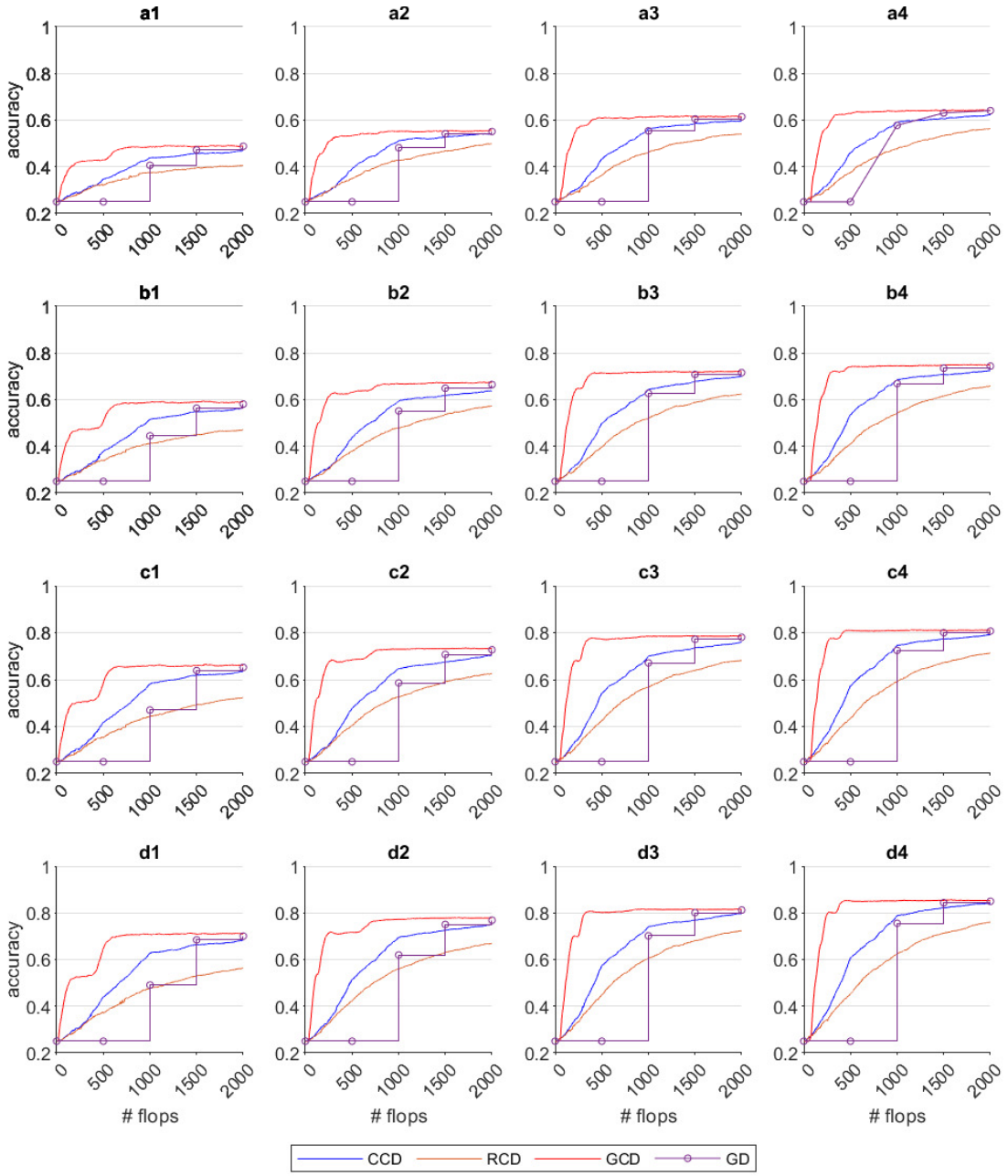


Figure 4.2: Average values of the accuracy over 5 random networks sampled from SBM for $p = 2$, $p_{in} = 0.2$, $\frac{p_{in}}{p_{out}} \in \{2, 2.5, 3, 3.5\}$ varies in the rows and $perc \in [3\%, 6\%, 9\%, 12\%]$ varies in the columns.

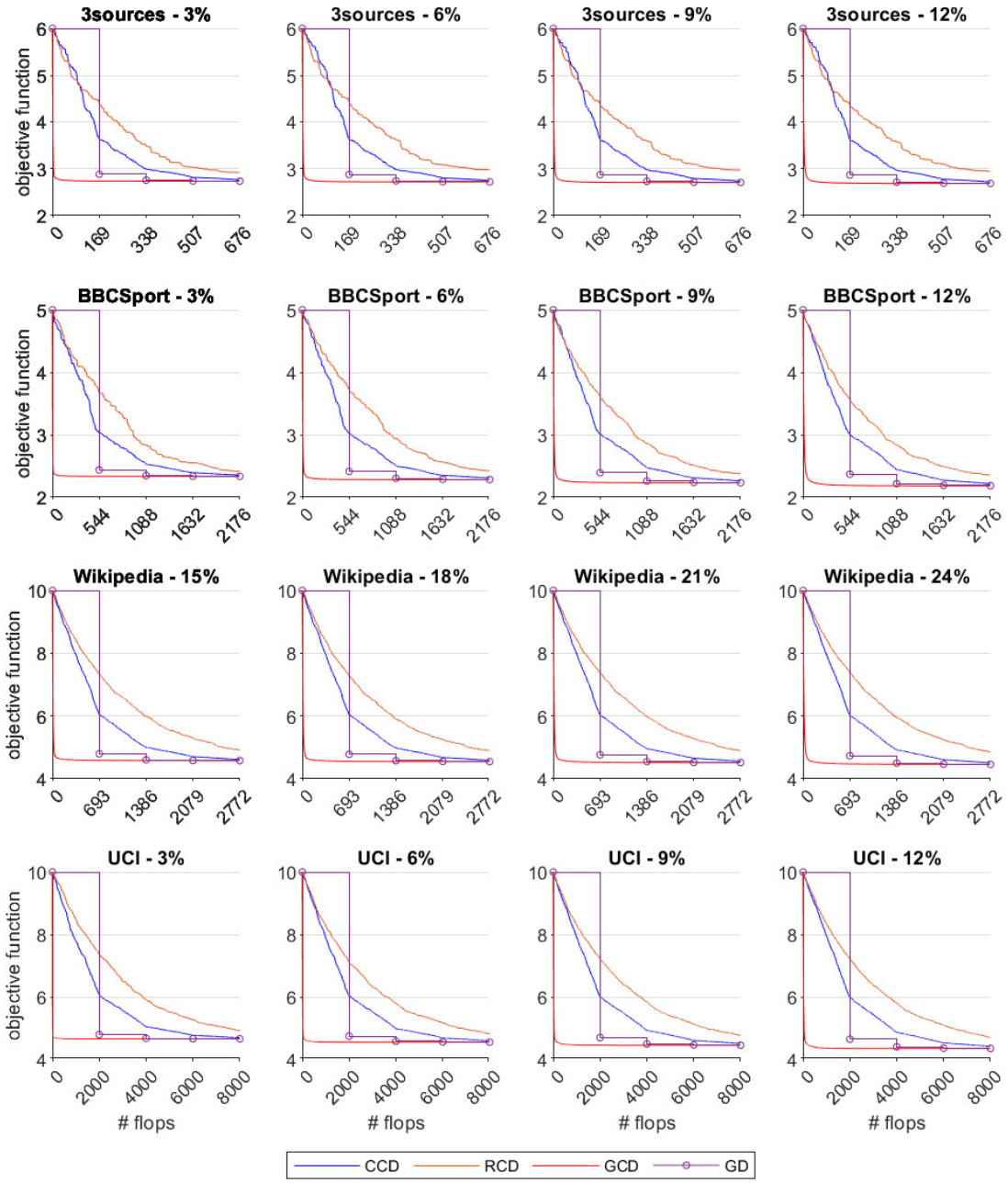


Figure 4.3: Average values of the objective function over 5 sampling of know labels, referring to 4 multilayer real-world datasets (3sources, BBCSport, Wikipedia, UCI) with quadratic regularizer. $perc \in [3\%, 6\%, 9\%, 12\%]$ (resp. $perc \in [15\%, 18\%, 21\%, 24\%]$ for Wikipedia) varies in the columns.

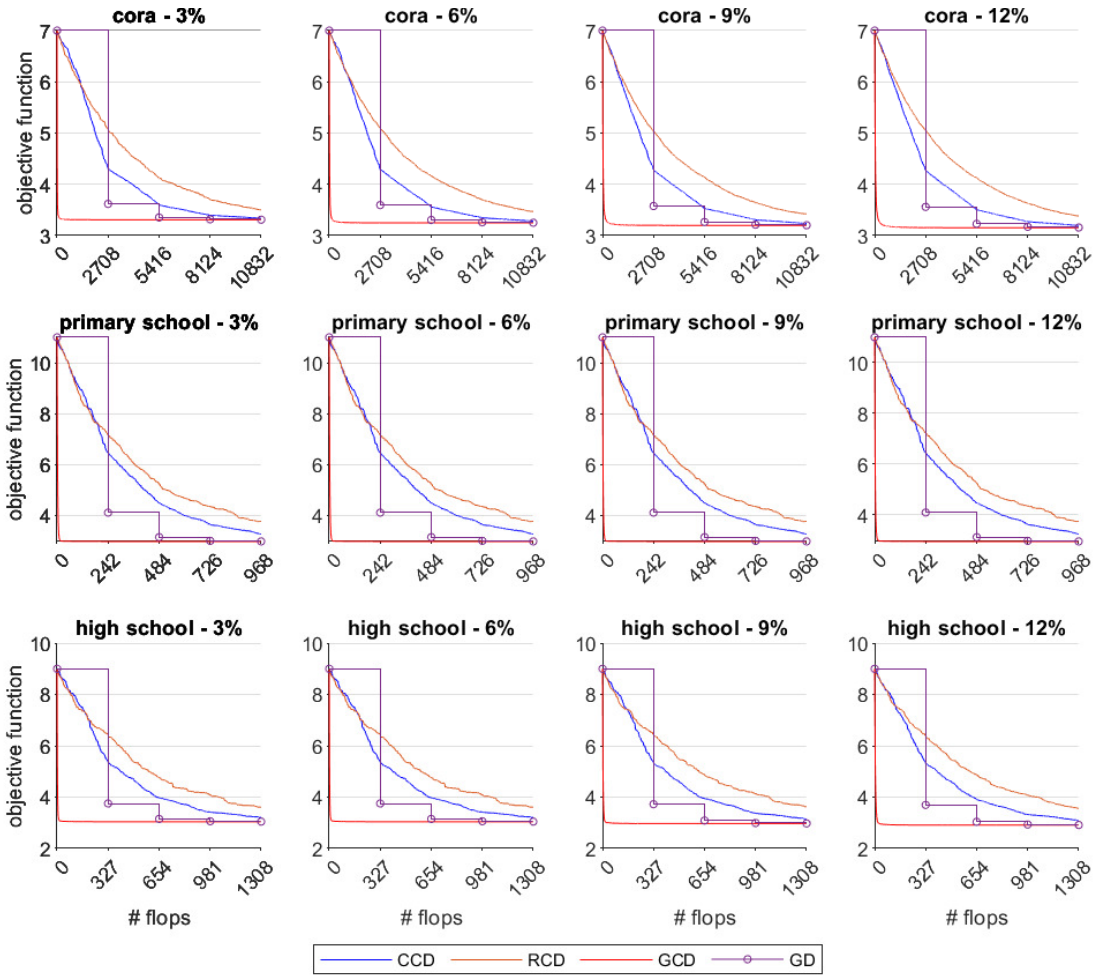


Figure 4.4: Average values of the objective function over 5 sampling of know labels, referring to 1 multilayer real-world dataset (cora) and 2 real-world hypergraphs (primary school and high school) with quadratic regularizer. $perc \in [3\%, 6\%, 9\%, 12\%]$ varies in the columns.

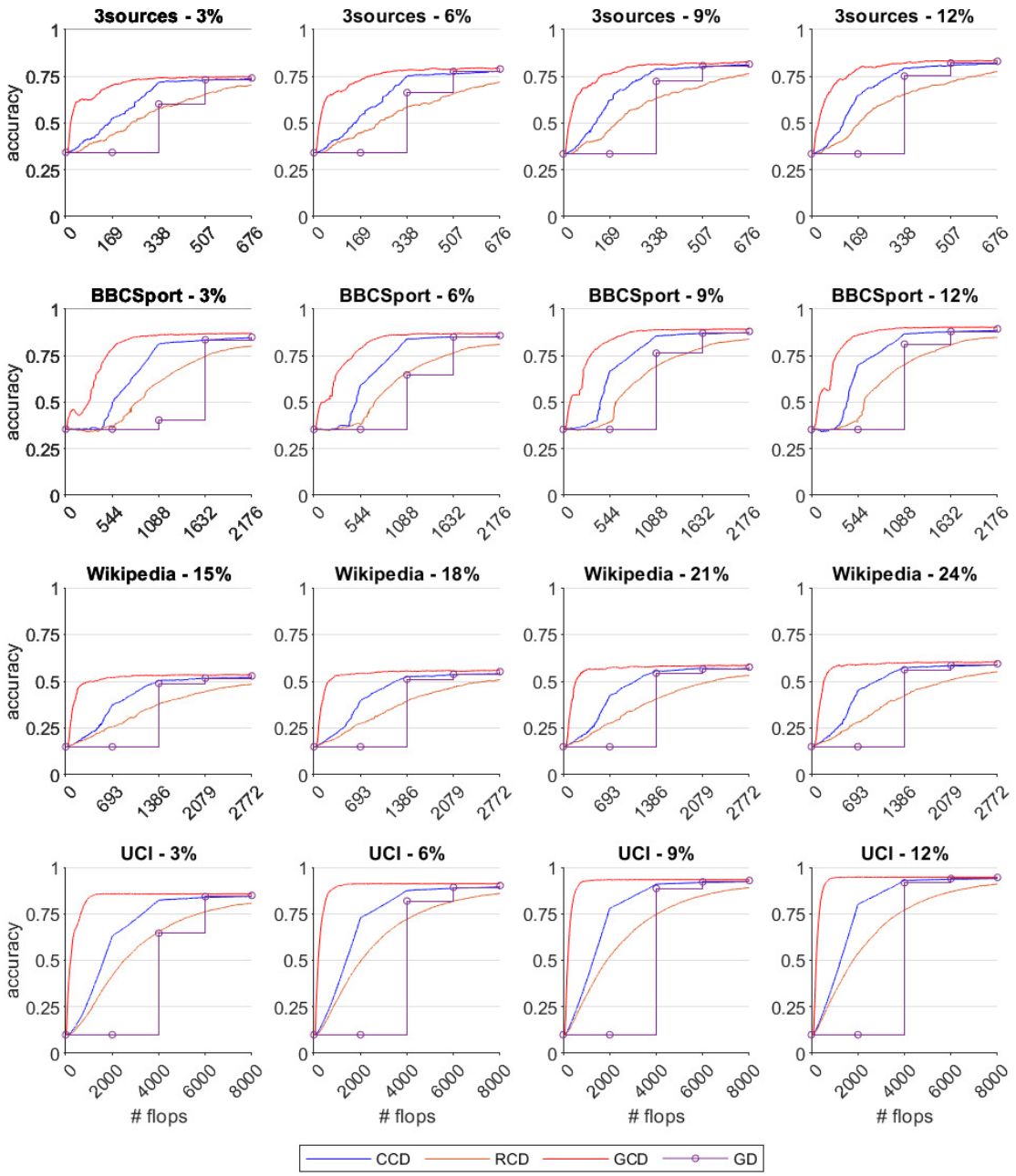


Figure 4.5: Average values of the accuracy over 5 sampling of know labels, referring to 4 multilayer real-world datasets (3sources, BBCSport, Wikipedia, UCI) with quadratic regularizer. $perc \in [3\%, 6\%, 9\%, 12\%]$ (resp. $perc \in [15\%, 18\%, 21\%, 24\%]$ for Wikipedia) varies in the columns.

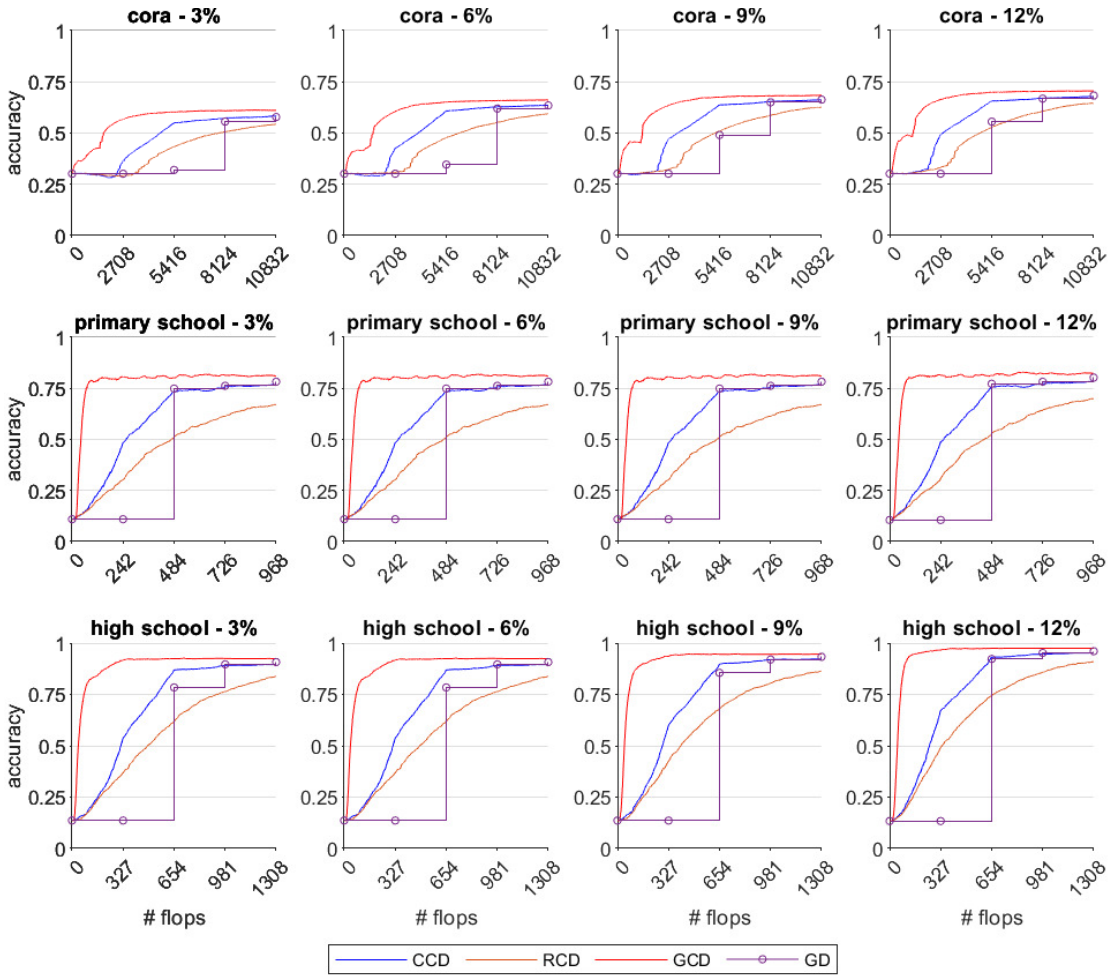


Figure 4.6: Average values of the accuracy over 5 sampling of know labels, referring to 1 multilayer real-world dataset (cora) and 2 real-world hypergraphs (primary school and high school) with quadratic regularizer. $perc \in [3\%, 6\%, 9\%, 12\%]$ varies in the columns.

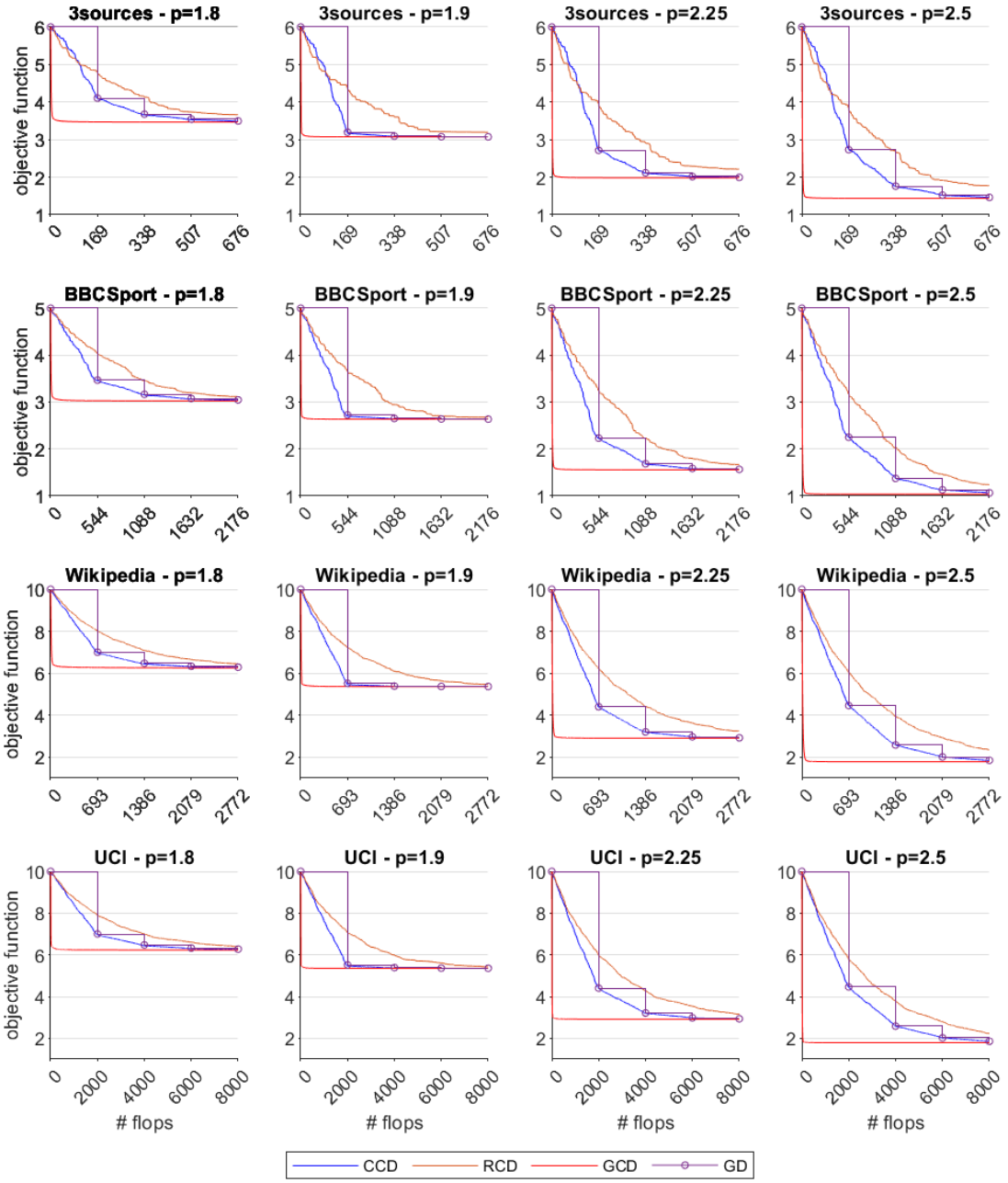


Figure 4.7: Average values of the objective function over 5 sampling of know labels, referring to 4 multilayer real-world datasets (3sources, BBCSport, Wikipedia, UCI) with $perc = 6\%$ (resp. $perc = 18\%$ for Wikipedia). $p \in [1.8, 1.9, 2.25, 2.5]$ in the regularization term varies in the columns.

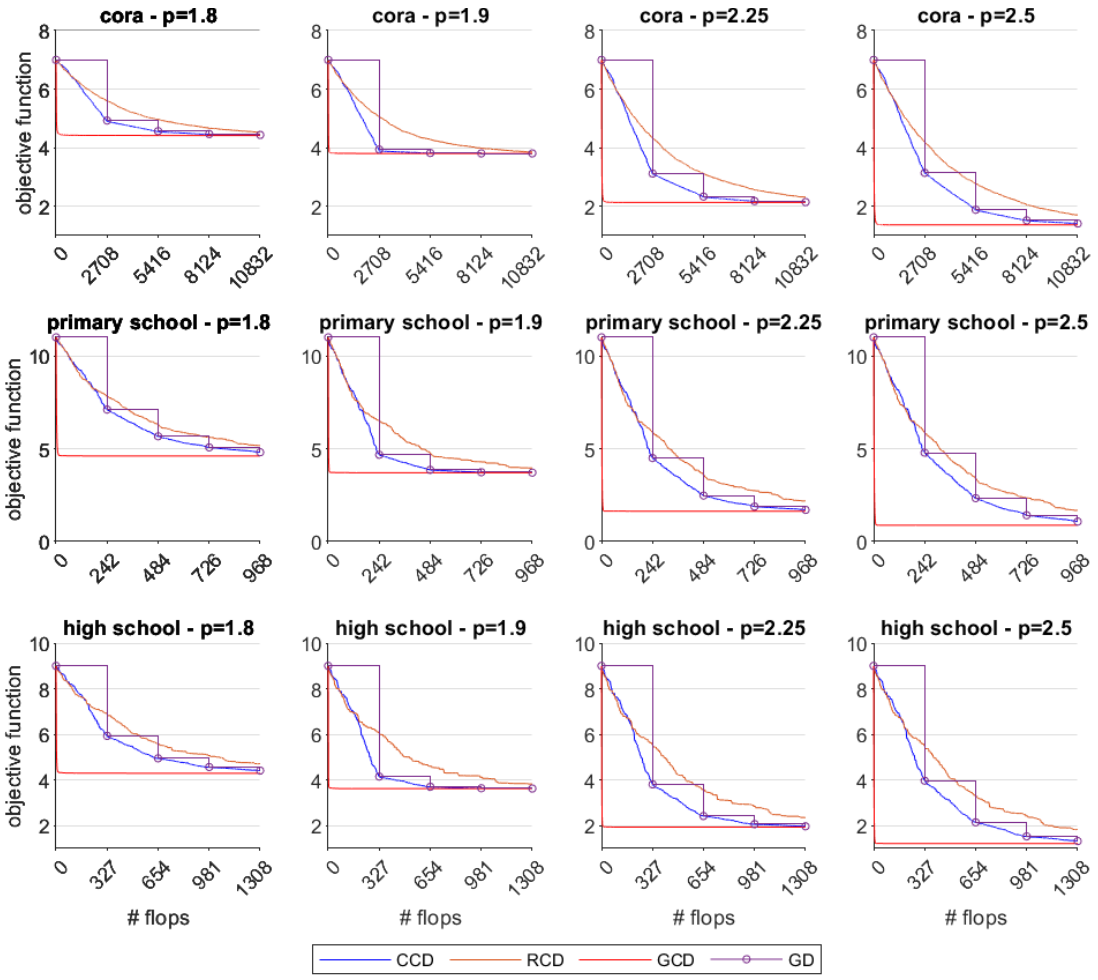


Figure 4.8: Average values of the objective function over 5 sampling of know labels, referring to 1 multilayer real-world dataset (cora) and 2 real-world hypergraphs (primary school and high school) with $perc = 6\%$. $p \in [1.8, 1.9, 2.25, 2.5]$ in the regularization term varies in the columns.

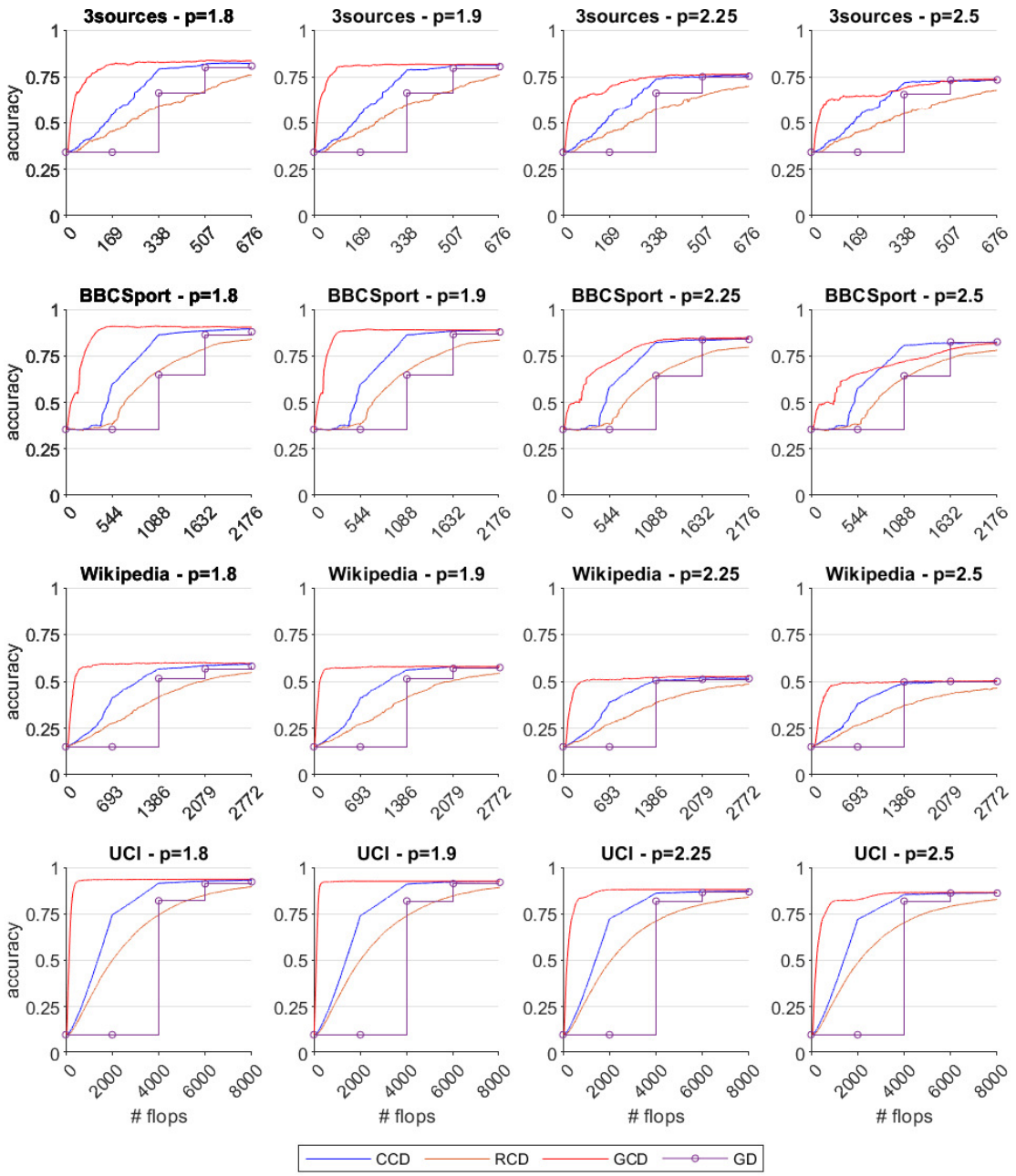


Figure 4.9: Average values of the accuracy over 5 sampling of know labels, referring to 4 multilayer real-world datasets (3sources, BBCSport, Wikipedia, UCI) with $perc = 6\%$ (resp. $perc = 18\%$ for Wikipedia). $p \in [1.8, 1.9, 2.25, 2.5]$ in the regularization term varies in the columns.

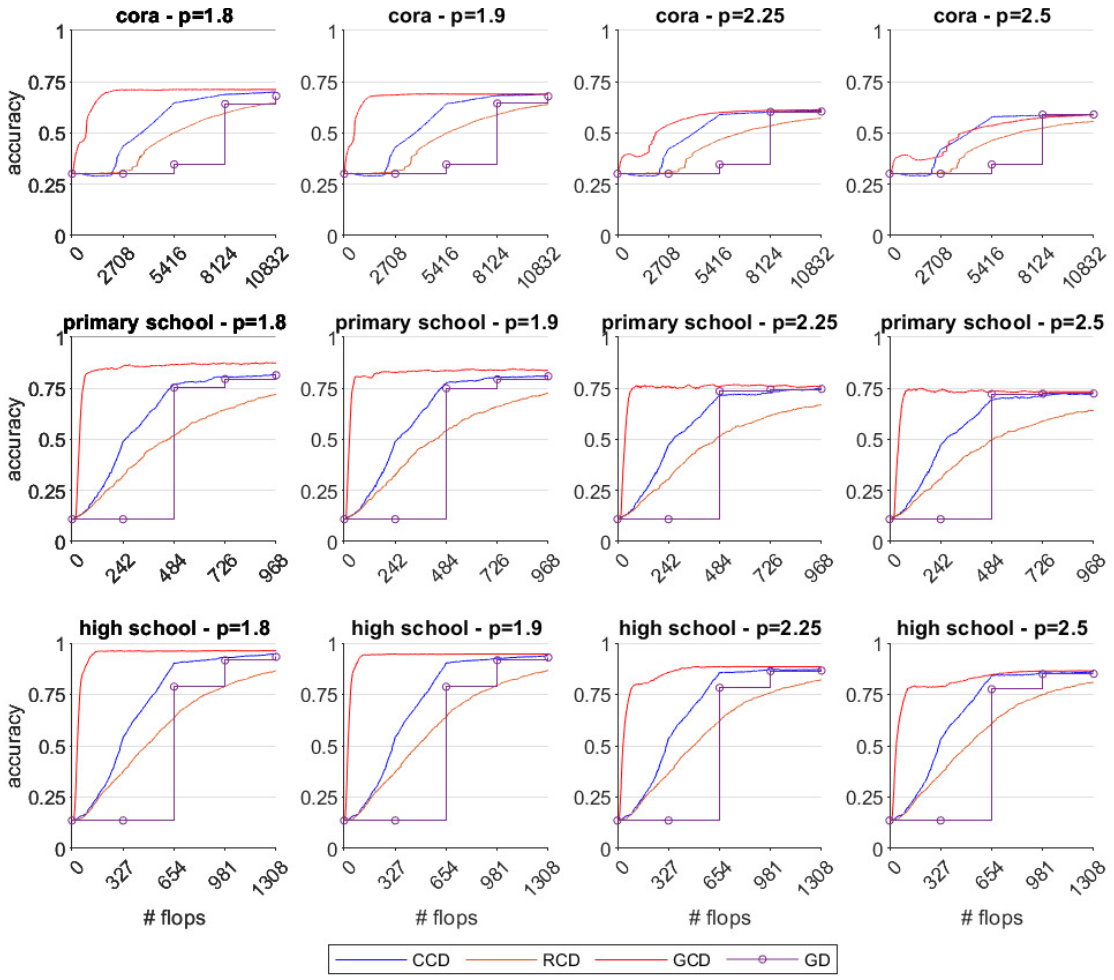


Figure 4.10: Average values of the accuracy over 5 sampling of know labels, referring to 1 multilayer real-world dataset (cora) and 2 real-world hypergraphs (primary school and high school) with $perc = 6\%$. $p \in [1.8, 1.9, 2.25, 2.5]$ in the regularization term varies in the columns.

# Chapter 5

# Collaboration and topic switches in science

In this chapter, we focus on an application to the science of science. Firstly, we report an overview of this relatively new but highly promising and rapidly developing field. Secondly, we present our work regarding the relation between collaboration and topic switches in science. Collaboration plays a pivotal role in advancing science and driving innovation. Stemming primarily from the necessity to harness diverse skills and expertise to address complex scientific challenges, the collaboration also serves as a valuable wellspring of insights into the future activities and contributions of researchers. More specifically, collaboration enables us to deduce the probability with which scientists opt for future research directions through the interconnected mechanisms of selection and social influence. Here we thoroughly investigate the interplay between collaboration and topic switches. Our findings reveal that the likelihood of a researcher engaging in a new research topic rises in conjunction with the number of prior collaborators. Furthermore, we observe a discernible pattern indicating that the impacts of individual collaborators are interconnected and not independent. As authors demonstrate greater productivity and influence, their colleagues are more inclined to embark on new research subjects. Additionally, there exists an inverse correlation between the average count of coauthors per paper and the probability of transitioning to new topics, suggesting a dilution of this effect as the number of collaborators increases.

## 5.1 Overview of the science of science

The increasing availability of digital data in the 21st century has opened up unprecedented opportunities to model and analyze complex social systems and interactions. One emerging and rapidly developing field in this realm is the science of science (SOS). At its core, the SOS aims to uncover the intricate patterns and mechanisms that characterize scientific discovery. This involves understanding how various components within the system of science, such as research papers, authors, and research fields, interact and evolve over time. These interactions are often represented as evolving networks, capturing the dynamic relationships among different elements

of the scientific landscape. The ultimate objective of the SOS is to derive insights and knowledge that can inform the development of tools, policies, and strategies to accelerate scientific progress. Unlike traditional scientometrics, which primarily focuses on quantitatively measuring aspects of science, the SOS delves deeper into uncovering the underlying principles and dynamics that drive the advancement of knowledge. In the subsequent sections, we provide a concise overview to offer a general understanding of the diverse topics and challenges that the SOS addresses. This field encompasses a wide range of areas, each contributing to a deeper comprehension of the intricacies of scientific discovery and the factors influencing it.

**Scientific publication data**

Scientific publication data serve as a cornerstone for research in the SOS. These datasets contain comprehensive information about individual research papers, including attributes like title, authors, affiliations, publication date, journal name, abstract, keywords, topics, and references. These are some of the widely used datasets for SOS research:

- American Physical Society (APS) provides a collection of physics-related publications, making it a valuable source for studying the evolution of physics research.
- OpenAlex (successor to Microsoft Academic Graph) offers a rich database of scholarly publications and their associated metadata, encompassing a wide range of research domains.
- MEDLINE and PubMeddatasets focus on medical literature and contain valuable information for investigating trends and developments in medical research.
- Web of Science (WoS) and Scopus databases offer comprehensive coverage of scholarly literature, including a wide range of research fields.
- arXivis is a preprint repository used by researchers in physics, mathematics, computer science, and related fields, providing a platform to share research before formal publication.
- Google Scholar aggregates scholarly articles from various sources, serving as a widely accessed resource for researchers worldwide.

A fundamental challenge in SOS research, as well as other scientometric studies, is name *disambiguation* [203, 204, 205, 206]. It refers to the problem of associating publications with the correct authors, considering the prevalence of common names and variations in how authors are referenced. This challenge has two key dimensions:

- Homonym Resolution: multiple scholars may share the same name, leading to instances where different authors are mistakenly merged into a single identity.
- Synonym Resolution: the same author might be referred to using different name variations or affiliations, causing the system to create distinct identities for the same individual.

Addressing this challenge requires sophisticated techniques that leverage various aspects of the data, such as co-authorship patterns, affiliations, publication history, and co-citation relationships.

Advances in natural language processing, machine learning, and network analysis have contributed to more effective name disambiguation methods, helping researchers accurately attribute publications to the correct authors and create more reliable research profiles.

**Scientific networks**

The collaboration and citation networks are among the most extensively studied networks in the field of SOS.
In a collaboration network, scientists are represented as nodes, and a connection between two scientists is established if they have collaborated on a paper together. The edge weight reflects the number of coauthored papers shared by the two scientists. The pioneering work by [27] marked the initial investigation into the structural attributes of collaboration networks. Noteworthy features of these networks, as highlighted by [207, 208, 209], include:

- Power-law distributions for both the collaborators of scientists and the papers they coauthor.
- The presence of a *small-world* phenomenon within each discipline, where the average shortest path between scientists is relatively short, typically around five or six steps.
- The emergence of community structure, where communities correspond to different scientific disciplines.
- Assortative mixing, signifying a preference for high-degree nodes to connect to other high-degree nodes and low-degree nodes to connect to other low-degree nodes.
- The *friendship paradox*, where collaborators of a scientist tend to have more coauthors, higher citation counts, and more publications.

Citation networks, on the other hand, are characterized by papers as nodes, and directed, unweighted edges represent citations from one paper to another. Key features of citation networks, as described by [28], encompass:

- Power-law distribution of in-degree (number of incoming citations) and an exponential distribution of out-degree (number of outgoing citations) of nodes.
- Evident community structure, with communities often corresponding to specific subfields within a discipline.
- Weak disassortativity, implying a tendency for high-degree nodes to connect to low-degree nodes.
- Shortest path lengths that are relatively small within the network.
- Frequent occurrence of the feed-forward loop pattern, where paper A cites paper B, paper B cites paper C, and paper A also cites paper C.
- The acyclic nature of citation networks, where links point from newer papers to older papers.

These characteristics provide insights into the structure and dynamics of scientific collaboration and citation networks, shedding light on patterns of knowledge dissemination and interdisciplinary interactions.

### Dynamical properties of science

In addition to studying the static structural properties of collaboration and citation networks, the SOS also investigates their dynamic characteristics.

The Barabasi–Albert (BA) model [210] is one of the earliest and fundamental models used to describe the dynamics of citation networks. It is based on the concept that nodes with higher degrees attract new links at a higher rate than nodes with lower degrees. This model can account for the emergence of power-law distribution in citation counts; however, it does not fully capture all the intricate features observed in real citation networks. Another dynamic mechanism that has garnered attention is the *preferential attachment mechanism*, as discussed by [211]. This mechanism predicts that highly cited papers tend to be older, attributing this phenomenon to a first-mover advantage. However, in real-world scenarios, there are instances of recent papers achieving high citation rates. This discrepancy is due to the aging effect on early papers and the varying suitability of papers for garnering citations. Conversely, the concept of a *sleeping beauty* paper, explored by [212, 213], refers to papers that initially receive only a few citations after publication but subsequently experience a sudden surge in their citation count. This phenomenon highlights the potential for scientific impact to be recognized at a later stage, illustrating the complex interplay between the timing of publication and the recognition of contributions.

The dynamics of citation networks have been addressed by [214], who proposed a model that considers two distinct behaviors within citation networks. Specifically, a paper may directly cite an older paper or indirectly cite it by referencing a more recent publication that includes the older paper as a reference. Collaboration networks also play a pivotal role in SOS dynamics, with a particular focus on team formation. The growth of scientific collaboration networks is characterized by preferential attachment, wherein the likelihood of collaboration between two researchers is positively correlated with the number of coauthors they share. The significance of teamwork in contemporary science has been increasingly recognized, leading to investigations into the size, multi-university nature, and interdisciplinary composition of research teams. Various models have been developed to gain insights into team dynamics [215, 216]. Notably, a study by [217] revealed that both smaller and larger research teams play essential roles in the scientific landscape. Smaller teams tend to introduce novel and disruptive ideas, while larger teams often contribute to the development and refinement of existing concepts. As a result, both types of teams contribute to the advancement of science and technology. In a recent study, [218] introduced a metric designed to quantify the distinction between papers that contribute to the enhancement of existing streams of knowledge and those that introduce disruptive new streams of knowledge. The underlying idea is that if a paper is disruptive, subsequent works that cite it are less likely to also cite its predecessors. Conversely, if a paper is consolidating, subsequent works citing it are more likely to cite its predecessors. Their findings reveal that, across various fields, the nature of science and technology is trending toward being less disruptive over time. However, this conclusion has been met with criticism by [219]. They argue that the observed decline in disruption is largely influenced by what they refer to as *citation inflation*. The phenomenon of citation inflation is characterized by the increasing length of reference lists in papers over time. This leads to a higher density of links within citation networks and a rise in self-citations, thereby enhancing the rate of triadic closure in these networks. Consequently, the metric for measuring

disruption becomes temporally biased due to these changes in citation patterns.
These dynamics within both citation and collaboration networks underscore the intricate interplay between individual researchers, their contributions, and the evolving structures of scientific communities. Understanding these dynamics can shed light on the emergence of new ideas, the propagation of knowledge, and the collaborative mechanisms that drive scientific progress. They also underscore the need to go beyond static network representations and delve into the temporal dynamics that shape the patterns of scientific discovery and impact.
The study of the evolution of scientific disciplines is a significant aspect of understanding how fields arise, evolve, and eventually fade away [220]. Researchers in this area aim to model the underlying processes that govern these dynamics. One perspective, put forth by [221], proposes that the evolution of disciplines is primarily shaped by the social interactions among scholars. To explore this idea, they developed the Social Dynamics of Science agent-based model. In this model, collaboration networks are constructed, with nodes representing scholars and edges indicating coauthored papers. They find that the emergence or decline of a discipline is associated with an increase in the network's modularity. Accordingly, they model the formation of new scientific disciplines by manipulating communities within the collaboration network through splitting and merging processes. Another perspective, proposed by [222], suggests that the progress of larger scientific fields might be impeded. When the volume of papers published annually within a field becomes substantial, authors tend to repeatedly cite the same well-cited papers. This behavior can result in new papers struggling to attain high citation counts or disrupt existing research. In another study, [223] introduces a quantitative framework to analyze the temporal evolution of all scientific fields. They identify a rise-and-fall pattern common to all fields, characterized by a two-parameter right-tailed Gumbel distribution. They divide a field's lifetime into distinct evolutionary phases: creation, adoption, peak, and decay. Early phases are characterized by disruptive works that challenge existing paradigms, while later phases witness the emergence of specialized, large research teams building upon previous works.
Overall, these studies shed light on the intricate dynamics governing the lifespan of scientific disciplines, the role of social interactions and collaboration networks, as well as the emergence of disruptive and specialized research in shaping the trajectory of various fields.

### Quantification of scientific impact

With the exponential growth of scientific publications and researchers, the need for effective metrics to assess the impact of scholarly work and individual scholars has become more pronounced. Several methods have been devised to gauge the influence of scientific publications and researchers, addressing various aspects of their output. Simple metrics often rely on basic statistics such as the total number of publications, total citations received, average citations per publication, number of highly cited papers (often defined by a predefined citation threshold), and the proportion of highly cited papers in a researcher's portfolio. One of the most widely known and used metrics is the h-index introduced by [224]. The h-index seeks to quantify both the quantity and quality of a researcher's work. Specifically, a scholar's h-index is h if they have h papers each cited at least h times. While the h-index provides a single numerical summary of a researcher's impact, it does have limitations, including its inability to fairly compare researchers across different disciplines

and career stages. In response to these limitations, various alternative metrics have been proposed to assess researchers' impact more comprehensively. [225] devised a weighted citation network between authors and introduced a ranking method called SARA (System for Allocating Credits). SARA utilizes a diffusion algorithm to simulate the flow of credits across the network, providing improved predictions for award assignments in the field of physics compared to local measures. [226] evaluated the effectiveness of the h-index by correlating it with rankings based on community awards. They observed that evolving authorship patterns and hyperauthorship have reduced the effectiveness of traditional scientometric measures. Fractional allocation of citations among coauthors was found to enhance the predictive power of research metrics. The quest for a more nuanced impact assessment led [227] to propose the E-index. This metric accounts for the distribution of citations and emphasizes consistently producing high-quality work over time. This approach aims to address the limitations of the h-index and provide a more sensitive measure of impact. An additional challenge in evaluating authors' impact is the fair allocation of credit among coauthors of a paper. Various approaches have been suggested to determine the proportional contribution of each author [228]. On the other side, in the context of evaluating the impact of scientific journals, the impact factor (IF) is a well-known metric. Proposed by Garfield [229], the impact factor quantifies the average number of citations to recent articles published in a journal. The 5-year impact factor (IF5), which averages the IF over the last five years, is a widely utilized variant of this metric.
Overall, the development of impact metrics has been driven by the need to accurately capture the influence of scholarly work and researchers, taking into account the evolving landscape of scientific publishing and collaboration.

### Inequalities in science

A significant portion of research within the SOS domain focuses on analyzing factors that contribute to disparities in scientific publications, particularly about gender inequalities. Various studies have shed light on the nuanced dynamics underlying these disparities and their implications. [230] investigated gender disparities in scholarly publications. Their findings highlighted several key points: articles authored by women in dominant positions receive fewer citations compared to those authored by men in similar positions; women's representation in fractional authorship is smaller than men's; women are underrepresented in first authorships; regions like South America and Eastern Europe show relatively better gender parity, although this might be linked to lower scientific output. A comprehensive study by [231] spanned disciplines and countries from 1955 to 2010. Their results revealed intriguing trends: as the proportion of women in science increases, gender differences in productivity and impact also increase paradoxically; annual publication rates and impact per work are two consistent gender invariants; disparities in career lengths and dropout rates contribute to gender disparities. Research by [232] delved into gender diversity within research teams in medical sciences. They observed that mixed-gender teams are growing but remain underrepresented compared to a null model. Publications from mixed-gender teams are more novel and impactful, with a stronger gender balance on a team correlating with better performance measures. Examining gender inequality in scholarly mobility, [233] conducted a cross-national study. Their work revealed that female researchers tend to be less geographically

mobile, migrating over shorter distances with lower origin and destination diversity. Diversity within academic communities was expanded upon by [234], who investigated ethnicity, discipline, gender, affiliation, and academic age. They found evidence of homophily in ethnicity, gender, and affiliation. Ethnic diversity showed the strongest correlation with scientific impact. Another insightful study by [235] explored gender differences in productivity and prominence among researchers. Their research indicated that these differences can largely be explained by variations in coauthorship networks. Additionally, they observed that collaborative networks act as a form of social capital that transfers from senior to junior collaborators. In the academic job market, [236] scrutinized faculty hiring processes and found evidence of deep-rooted social inequalities. Prestige in doctoral programs significantly predicts placement outcomes, with women often faring worse than male graduates from the same institution. Greater institutional prestige leads to enhanced faculty production, better placement, and a more influential position within the academic discipline.

The complex and pervasive nature of inequalities within the scientific community underscores the importance of continued research to better understand and address these issues across various dimensions.

### Prediction

The SOS field has also delved into the intriguing question of whether it's possible to forecast success in scientific endeavors. Numerous studies have been conducted to uncover the key factors influencing the impact of publications and the trajectories of successful scientific careers. [237] presented a mechanistic model for citation dynamics of individual papers, revealing a universal temporal pattern in citation histories. This model introduces three fundamental parameters that characterize a paper's citation history: immediacy (time to reach citation peak), longevity (decay rate), and fitness (perceived novelty and importance). Analyzing the impact and productivity change across scientific careers, [238] discovered the *random-impact rule* indicating that influential publications are distributed randomly within a scientist's sequence of publications. They constructed a null model of scientific careers based on this rule, where scientists select projects with potential and enhance them uniquely, resulting in impactful papers. A personal parameter $Q_i$ predicts a scientist's impact evolution and external recognitions. [239] demonstrated that junior researchers who collaborate with top scientists enjoy a persistent competitive advantage in their careers, suggesting a prediction of future success based on early career indicators. Forecasting new scientific collaborations was addressed by [24], who employed a multiplex network encompassing scientific credit (citations) and common interests (shared keywords) in distinct layers. The dynamics of failure were explored by [240], who developed a one-parameter model illustrating how future attempts build upon past failures. Their model indicated a phase transition separating failure dynamics into progression or stagnation regions, with agents who share similar characteristics experiencing diverse outcomes near the critical threshold.

These studies collectively contribute to unraveling the intricate factors that drive success in science, ranging from the evolution of paper impact to predicting influential collaborations and understanding the mechanisms behind both success and failure in scientific endeavors.

## 5.2 Introduction

Over the past few decades, modern science has witnessed a significant shift towards collaboration, with larger teams becoming essential for addressing complex problems across various disciplines [241]. These teams are often necessary due to the need for a diverse range of knowledge and expertise. However, it's worth noting that small teams can also play a role in introducing novel paradigms and innovative approaches [217].
A powerful representation of the collaborative nature of science is provided by collaboration networks, where nodes represent authors and edges connect authors who have coauthored at least one paper. The availability of bibliometric data has led to extensive studies of collaboration networks, revealing their structural characteristics and patterns [27, 216, 242, 243].
Collaboration networks highlight the concept of *homophily* among scholars. This phenomenon suggests that individuals with similar research interests or working on related topics tend to collaborate. This can be seen as an example of *selection*, where like-minded individuals naturally form collaborations.
However, collaboration also introduces the notion of *social influence*. Coauthors can expose each other to new tools, methods, and theories, even if they are not directly relevant to the current project. This relationship between knowledge diffusion and collaboration has been studied extensively. Research has shown that knowledge tends to flow more readily between scholars who have collaborated before [244] or who are closely connected in the network [245].
In the context of switching research interests, scholars may decide to work on new topics based on their collaborative experiences. These switches have become more frequent over time [246] and have been quantitatively explored [247, 248]. Such decisions may be influenced by coauthors in a process analogous to social contagion [249, 250, 251, 252, 253], where one scholar influences another to adopt a new research area. This idea draws parallels with epidemic models used to describe the spread of ideas [254, 255, 256], where an "infected" individual introduces a "susceptible" individual to a new concept, leading to its adoption.
In this study, we conduct an empirical analysis of the relationship between scholars' topic switches and their collaboration patterns. We distinguish between active authors who have published on the new topic and inactive authors who have not. We focus on the immediate collaboration network of authors, examining their first-order neighbors. The results suggest that the probability of an inactive scholar switching topics increases with the productivity and impact of their active coauthors. Moreover, the effect is influenced by the average number of inactive coauthors of active scholars. Additionally, the probability of an inactive scholar switching topics is correlated with the number of active coauthors they have, indicating a potential interdependence of coauthors' contributions.

## 5.3 Results

The scientific publication dataset OpenAlex [29] serves as the basis for our study. We analyze twenty topics from three distinct disciplines: Physics, Computer Science, and Biology & Medicine. The specifics of this dataset can be found in the Methods section (5.5.1).
Our methodology draws inspiration from the groundbreaking work conducted by Kossinets and

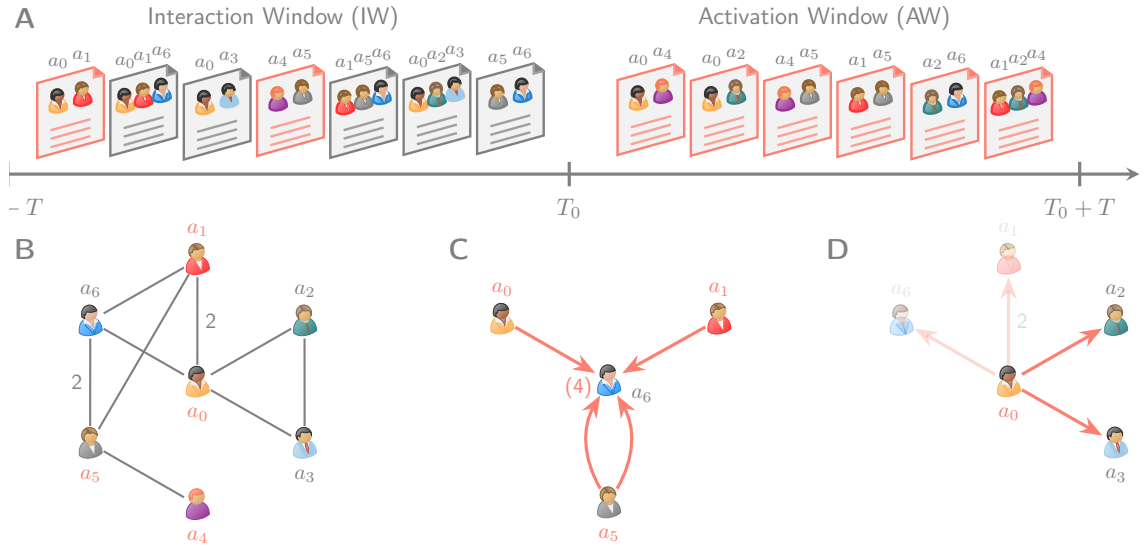


Figure 5.1: Schematic setup for our analysis. (A) Stream of papers across interaction (IW) and activation (AW) windows. Papers tagged with the focal topic $t$ are marked in red. (B) Author collaboration graph at the end of IW. Authors $a_i$ and $a_j$ are linked by an edge of weight $k$ if $a_i$ coauthored $k$ papers with $a_j$ within the IW. The authors active in the focal topic by the end of IW are marked in red. (C) Focus: inactive authors. Inactive author $a_6$ has four active contacts from three sources $\{a_0, a_1, a_5\}$ derived from the collaboration graph in (B). (D) Focus: active authors. Active author $a_0$ has four coauthors $\{a_1, a_2, a_3, a_6\}$, of whom $a_1$ is already active, and $a_6$ also collaborated with $a_1$ in the IW. This leaves the subset of exclusive inactive coauthors $\{a_2, a_3\}$. Within this subset, only $a_2$ becomes active in the AW, resulting in $a_0$'s source activation probability of $\frac{1}{2} = 0.50$. Additionally, $a_2$ writes their first paper with $a_0$ in the AW.

Watts on the evolution of social networks [257]. In their study, they explored the concept of *triadic closure* concerning two individuals, referred to as $a$ and $b$. This term denotes the likelihood that $a$ and $b$ establish a connection based on the number of mutual friends they share. Their approach involved capturing two snapshots of the network during consecutive time periods: the initial snapshot tracked all pairs of disconnected individuals, while the subsequent snapshot tallied how many of those pairs eventually formed connections. A similar framework has been employed to compute *membership closure*, a concept centered on the probability that an individual opts to participate in an activity after being connected to $k$ others who are already involved in that activity [258]. Building upon this approach, we adapt the framework to analyze how collaborations influence shifts in research topics, allowing us to explore how scholars transition between different scientific interests.

Here's an overview of our methodology.

Given a specific scientific topic denoted as $t$, a reference year $T_0$, and a window size $T$, we establish two successive and non-overlapping time intervals covering the years $[T_0 - T, T_0)$ and $[T_0, T_0 + T)$ respectively. The initial interval is termed the *interaction window* (IW), during which we monitor

author interactions within the collaboration network. The subsequent interval is referred to as the *activation window* (AW), where we quantify occurrences of researchers switching their research topics.
During the IW, we identify the cohort of *active* authors $A$ who have published papers $P$ related to topic $t$. To illustrate, in Figure 5.1A, $A$ would be represented as $a_0, a_1, a_4, a_5$. We construct the collaboration network $G$ by considering all papers $P'$ authored by individuals $a \in A$ after their activation. Notably, $P'$ encompasses papers outside of $P$, as indicated by the gray representations in Figure 5.1A.
Within $G$, we classify authors who are not active as *inactive* authors, and these are potential candidates for topic switches within the AW. They transition to active status when they publish their initial paper on topic $t$. In Figure 5.1B, authors $a_2$, $a_3$, and $a_6$ are deemed inactive, with both $a_2$ and $a_6$ converting to active status during the AW.
Furthermore, we assess each active author $a \in A$ using two metrics that gauge their scientific prominence: *productivity* and *impact*. The specifics of these metrics are outlined in Methods 5.5.3, and they are calculated after the IW to capture the contemporaneous perception of the author's scholarly contributions. Subsequently, we identify and designate authors who rank within the top and bottom 10% for each metric.
With this setup, we carry out two distinct experiments:

- In Experiment I, we analyze membership closure among inactive authors, evaluating how past collaborations with active authors correlate with topic switches.
- In Experiment II, the focus shifts to active authors, investigating how their inactive coauthors are inclined to start working on the same research topic.

More detailed information about these experiments and their outcomes can be found in Sections 5.3.1 and 5.3.2.

### 5.3.1 Experiment I

In this section, we delve into the concept of membership closure among inactive authors. We aim to address the following inquiries:

- How does the probability of topic switches correlate with the parameter $k$, which signifies the number of interactions with active authors?
- Does this probability display a dependence on the relative prominence of the active authors?

To effectively compute this metric, we initially need to establish what qualifies as an interaction with an active author within the IW. We explore two distinct definitions, outlined below:

- Counting the number of active coauthors while accounting for repeated collaborations with the same coauthor by considering their number of collaborations. In the context of the collaboration network, this equates to the weighted degree when exclusively considering active coauthors.
- Tallying the number of papers jointly authored with active coauthors.

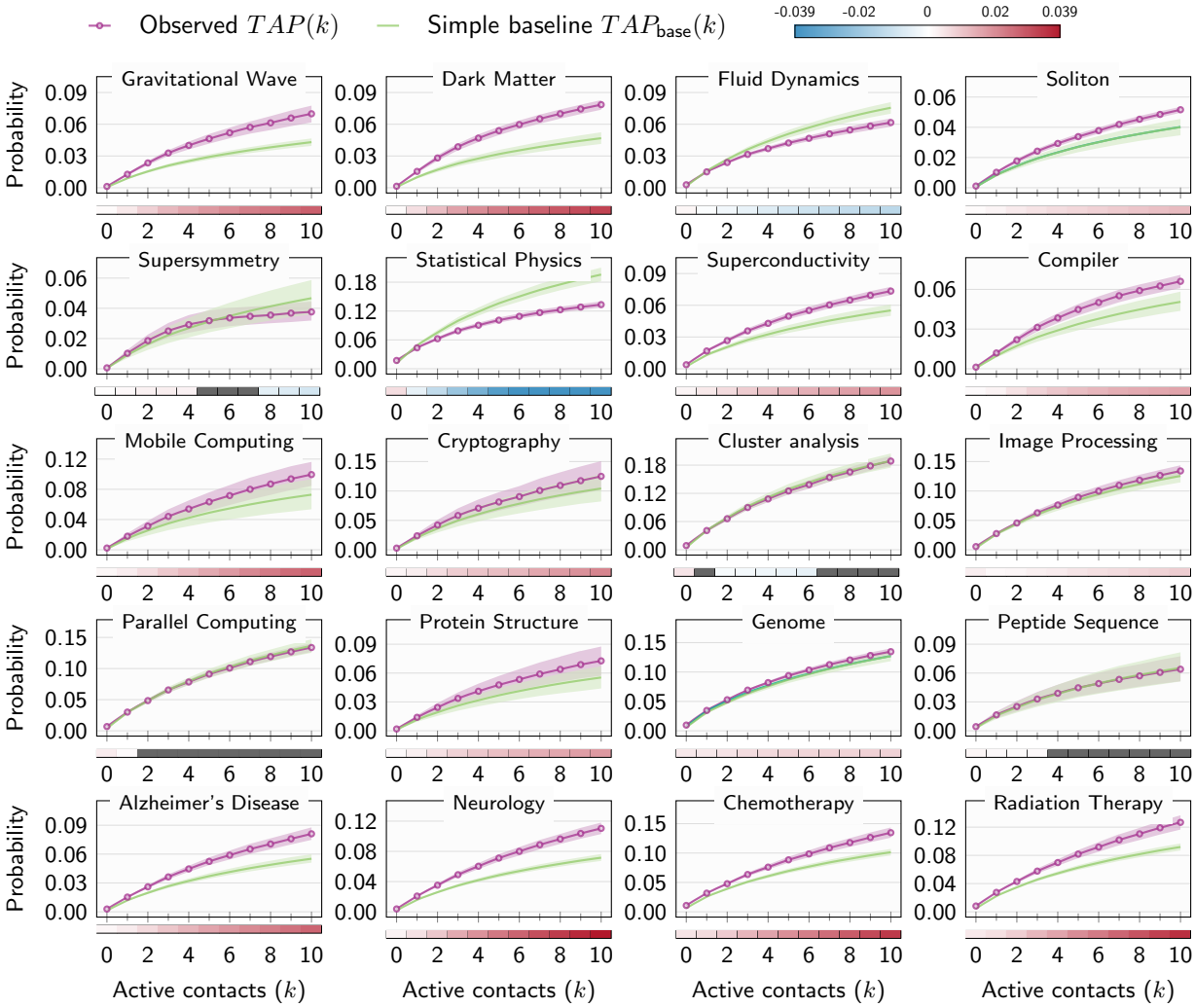


Figure 5.2: Experiment I. Cumulative target activation probability (in purple) for inactive authors in the AW with shaded 95% confidence intervals. For each $k$, the $y$-value indicates the fraction of inactive authors with at least $k$ active contacts in the IW who became active in the AW. The green solid line with shaded errors represents the baseline described in the text, corresponding to independent effects from the coauthors. The heatmap below the $x$-axis shows the mean difference between the observed and baseline curves for each $k$ value. It is gray if the 95% confidence interval contains 0, denoting the $k$-values where the points are statistically indistinguishable at $p$-value 0.05. Positive and negative deviations from the baseline are in red and blue, respectively.

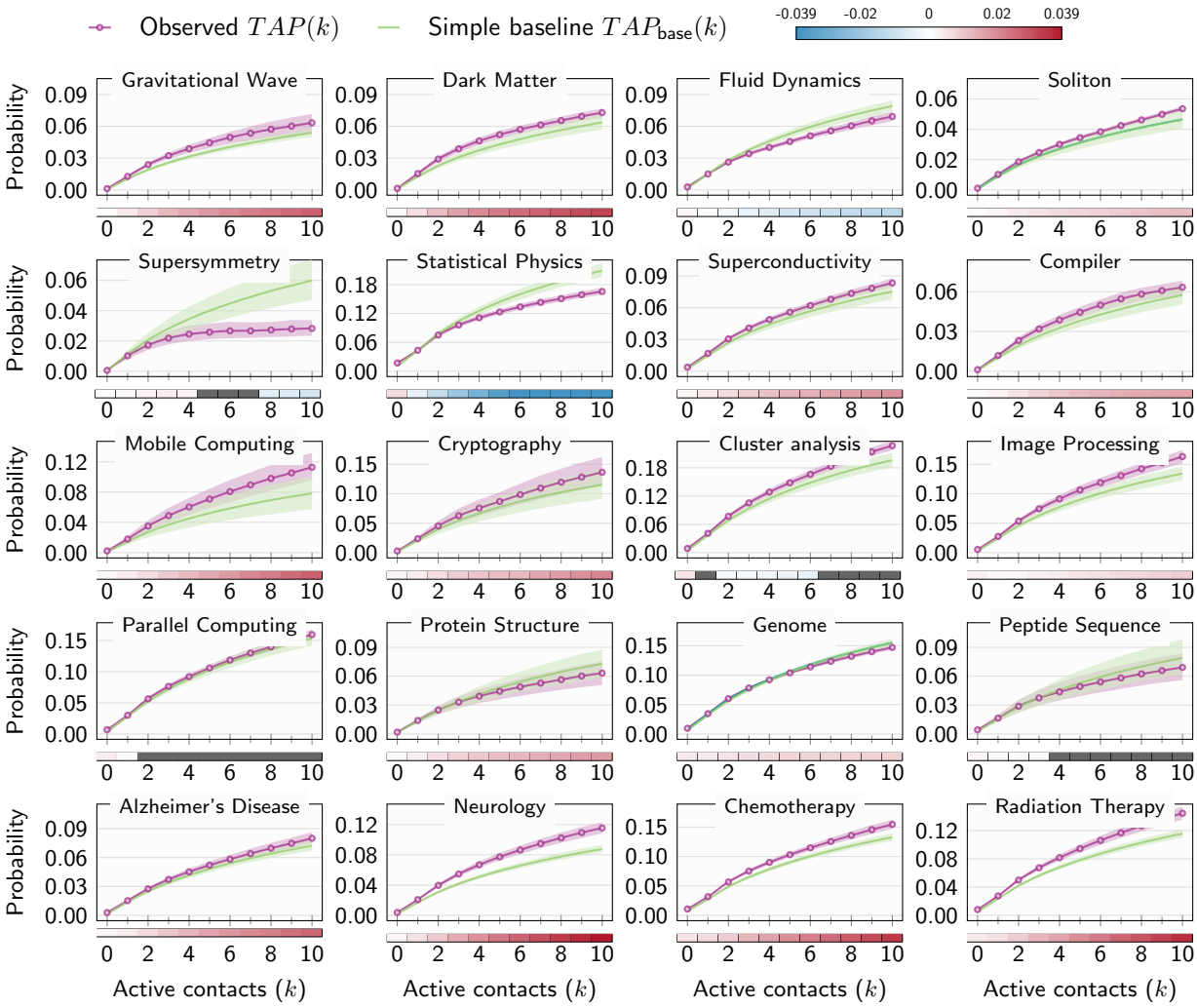


Figure 5.3: Experiment I. Same as 5.2, but here the number of contacts is the number of papers written with active coauthors in the IW. Cumulative target activation probability (in purple) for inactive authors in the AW with shaded 95% confidence intervals. For each $k$, the $y$-value indicates the fraction of inactive authors with at least $k$ active contacts in the IW who became active in the AW. The green solid line with shaded errors represents the baseline described in the text, corresponding to independent effects from the coauthors. The heatmap below the $x$-axis shows the mean difference between the observed and baseline curves for each $k$-value. It is gray if the 95% confidence interval contains 0, denoting the $k$-values where the points are statistically indistinguishable at $p$-value 0.05. Positive and negative deviations from the baseline are in red and blue, respectively.

For instance, in Figure 5.1C, author $a_6$ would have four interactions under the first definition (two from $a_5$ and one each from $a_0$ and $a_1$) and two interactions under the second definition (arising from the second and fourth papers within the IW).

In our main analysis, we present the results based on the first definition. It is worth noting that the conclusions drawn from the second definition do not significantly alter the primary findings.

To address the first question, we have calculated the cumulative *target activation probability* $TAP(k)$, which represents the fraction of inactive authors that transition to becoming active during the AW based on the number of contacts $k$ they have (for more details, refer to Methods 5.5.5). Figure 5.2 showcases the plotted $TAP(k)$ values (in purple) across the twenty studied topics. The error bars are derived from averaging results over different time windows for each field (as detailed in Methods 5.5.4).

As anticipated, a discernible upward trend is evident in the plot. Specifically, the transition from $k = 0$ to $k = 1$ is particularly prominent, indicating that the likelihood of *spontaneous* activation in the absence of prior contacts ($k = 0$) is notably lower compared to activation resulting from collaborative efforts ($k \geq 1$). Notably, the probability increases as the number of contacts grows. The majority of this increase transpires for low $k$ values.

To provide context to these findings, we establish a simple baseline denoted as $TAP_{\text{base}}(k)$ (as detailed in Methods 5.5.6). In this baseline, we assume that each contact has a consistent and independent probability of resulting in a topic switch. This baseline serves as a reference point for comparison with the observed results.

Within each topic, we calculate the difference (as outlined in Methods 5.5.4) between the observed curves for different values of $k$ across all reference years. These differences are then plotted below the $x$-axis. Except Cluster Analysis, Parallel Computing, and Peptide Sequence topics, the observed curves consistently deviate from the baseline. This observation provides empirical evidence suggesting that the baseline fails to capture the intricate details present in the actual data.

The positive deviations observed for most topics suggest a cumulative effect, indicating that interactions with active authors indeed play a substantial role in inducing topic switches. However, Fluid Dynamics and Statistical Physics stand out as exceptions, undershooting the baseline. This anomaly might be attributed to these fields being broad and interdisciplinary, unlike the other more focused topics. Collaborations spanning across diverse fields may have a dampening effect on topic switches, reducing the impact of the compounding mechanism observed in other topics. Furthermore, we verified that our conclusions also hold considering the second definition of interaction, as shown in Figure 5.3.

Continuing our investigation, we turn to the second research question, examining whether the prominence of the contact source influences the probability of topic activation. In our analysis, we consider the two subsets of active authors: those in the top 10% and those in the bottom 10% based on their productivity and impact. This categorization effectively segregates the most prominent active authors from the least prominent ones.

To ensure a clear evaluation of the impact of contact source prominence, we limit our analysis to the subset of inactive authors who are neighbors with only one of the two categories of active authors (top 10% or bottom 10%). This approach helps mitigate potential confounding effects, allowing us to draw more accurate conclusions regarding the relationship between the prominence

of the contact source and the likelihood of topic activation.
In Figure 5.4, we present a visual representation of our assessment of the significance of differences between cumulative target activation probabilities for inactive authors who are in contact with active authors in the top 10% and the bottom 10% in terms of productivity and impact.
Each row of the heatmaps corresponds to a specific topic, and the color of each square within the row indicates whether the difference in activation probabilities is positive (red), negative (blue), or statistically non-significant (grey). The two columns represent the prominent authors selected based on their productivity (left column) and their impact (right column).
In the case of productivity, we observe that all the differences are both significant and positive. This suggests that contacts with highly productive active authors result in higher probabilities of target activation among inactive authors. For the impact column, there are a few exceptions where the differences are not significant or where negative differences are present. Nonetheless, the overall trend indicates that having prominent contacts, based on either productivity or impact, tends to increase the probability of target activation.
Results deriving from the use of the second definition of interaction, which confirms this trend, can be found in Figure 5.5.

### 5.3.2 Experiment II

Here our focus is on active authors and their collaborators. For each active author, denoted as $a$, we examine the subset of their inactive coauthors who have engaged in *exclusive* collaborations with $a$ during the interaction window (IW). We refer to this subset as the "exclusive inactive coauthors" of $a$". To clarify, let's consider an example depicted in Figure 5.1D. Active author $a_0$ has four coauthors, namely $a_1, a_2, a_3, a_6$. Among these, only $a_2$ and $a_3$ have exclusively collaborated with $a_0$ during the IW. We focus on this specific subset of coauthors because it helps us isolate the effects that are solely attributed to the active author $a$, eliminating potential confounding factors introduced by interactions with other active authors.
We measure a significant metric known as the "source activation probability", denoted as $P_s^a$. This probability represents the fraction of the exclusive inactive coauthors of active author $a$ who transition to becoming active in the activation window (AW). The use of a fraction takes into account variations in collaboration network sizes that could differ greatly among different scholars. In the case of the example in Figure 5.1D, the source activation probability $P_s^{a_0}$ is $\frac{1}{2}$, or 0.5, because only $a_2$ among the exclusive inactive coauthors becomes active in the AW.
We proceed as follows: for a given group of active authors, we calculate $C_s$, which is the complementary cumulative distribution of their source activation probabilities. This involves computing the probability that an exclusive inactive coauthor of an active author transitions to becoming active in the activation window (AW). The details of this calculation are provided in Methods 5.5.7.
We divide the active authors into two groups: the most prominent (top 10% in productivity or impact) and the least prominent (bottom 10% in productivity or impact), as explained in Experiment I. We then compare the effects of these two groups by analyzing the *cumulative source activations*. This refers to the points on their respective cumulative distributions at a specific threshold denoted as $f^*$. In other words, we're interested in how the source activation

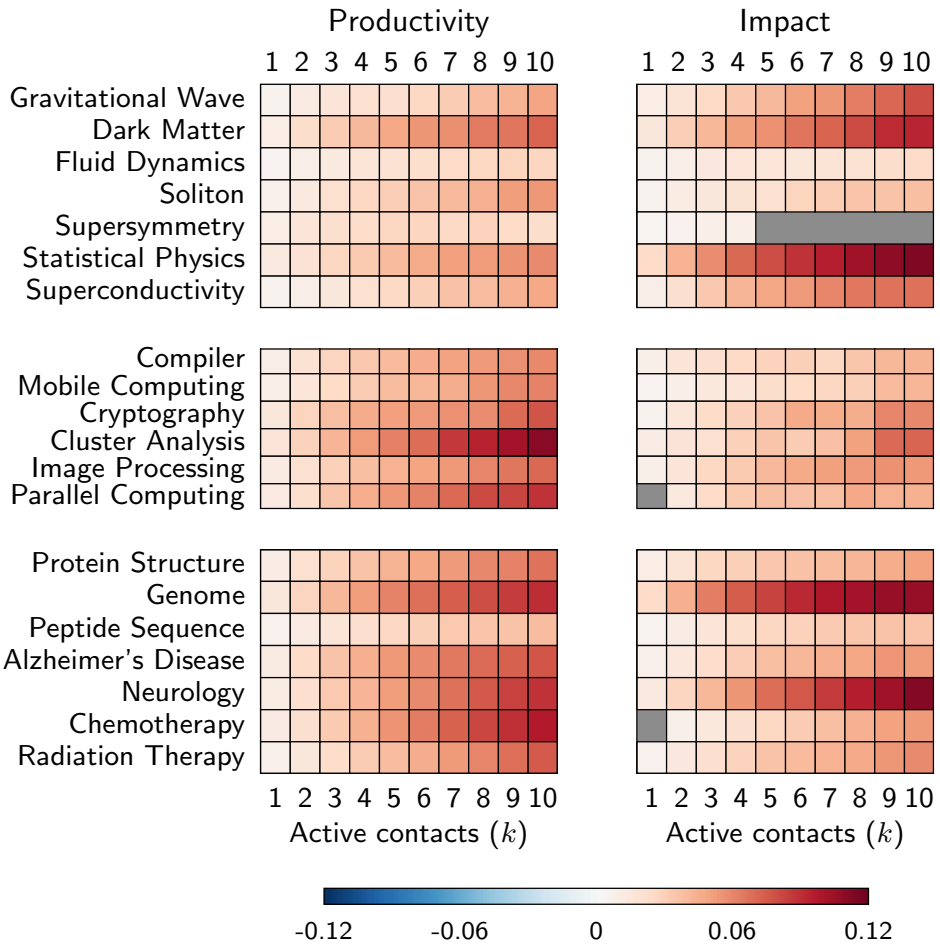


Figure 5.4: Heatmaps showing the mean difference between the cumulative target activation probabilities of the inactive authors in the AW who had exclusive contacts with the top 10% and bottom 10% of active authors, respectively, selected according to productivity (left) and impact (right) in the IW. The cells are gray if the 95% confidence interval contains 0. The majority of red cells indicate that the cumulative target activation probabilities for contacts with the top 10% are higher than those with the bottom 10%.

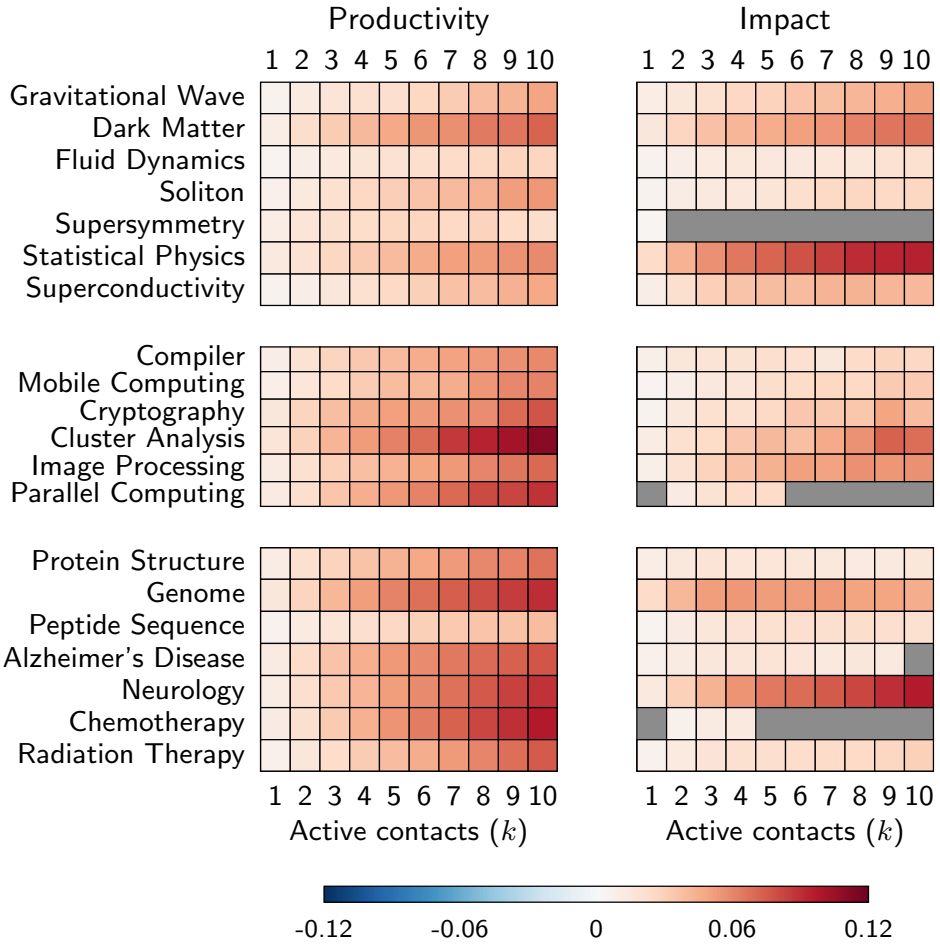


Figure 5.5: Experiment I. Same as 5.4, but here the number of contacts is the number of papers written with active coauthors in the IW. Heatmaps showing the mean difference between the cumulative target activation probabilities of the inactive authors in the AW who had exclusive contacts with the top 10% and bottom 10% of active authors, respectively, selected according to productivity (left) and impact (right) in the IW. The cells are gray if the 95% confidence interval contains 0. The majority of red cells indicate that the cumulative target activation probabilities for contacts with the top 10% are higher than those with the bottom 10%.

probabilities of these two groups vary based on this threshold.

The outcomes of our analysis are presented in Figure 5.6A for a threshold value of $f^* = 0.10$. Furthermore, we have verified that our conclusions hold even when considering a threshold of $f^* = 0.20$, as shown in Figure 5.7A.

In Figure 5.6, each row corresponds to a specific scientific topic. The green and purple ranges in the plot represent the 95% confidence intervals of the mean difference between the cumulative source activations of the two pools of authors (most prominent vs. least prominent) for both productivity and impact, respectively.

In terms of productivity, we observe that the difference is statistically significant for all topics except Superconductivity. This suggests that having more prominent active authors in terms of productivity increases the likelihood of their exclusive inactive coauthors becoming active on the same topic. When considering impact, the differences are somewhat less pronounced compared to productivity, but they are still statistically significant for most topics.

In Figure 5.6B, we investigate the chaperoning propensity of active authors [259], and we define the measure in Methods 5.5.8. It refers to the likelihood that exclusive coauthors of a given active author also publish their first paper on the same topic during the activation window. This measure helps us assess whether prominent active authors play a role in guiding their coauthors to work on new topics.

The green and purple ranges in the plot represent the 95% confidence intervals of the mean difference between the chaperoning propensities of the most prominent and the least prominent active authors for both productivity and impact, respectively. As with the previous analysis, we observe that the more productive/impactful an active author is, the more likely their coauthors are to transition and work with them on a new topic during the activation window.

Results for $f^* = 0.20$, which confirm this trend, can be found in 5.7B.

Although our analysis effectively demonstrates the influence of prominence, one might inquire whether the number of coauthors also contributes. Our hypothesis suggests that, on average, a greater number of collaborators could lead to weaker connections with each of them, potentially resulting in decreased source activation probabilities. To explore this idea, we select the top and bottom 20% from each group of most prominent authors based on the average number of coauthors present on papers published with exclusive inactive coauthors. Notably, this selection process intentionally omits any papers written on the specific focal topic.

In Figures 5.8 and 5.9, we present a similar analysis to Figures 5.6A and 5.7A, focusing on the two distinct groups of authors as previously described. Notably, the confidence intervals of the differences in this case are positioned to the left of zero, indicating negative values. In terms of productivity, all of these values are statistically significant. For impact, there are only two topics (Chemotherapy and Radiation Therapy) where the significance is not observed. In general, we find that inactive coauthors of prominent authors who collaborate with more individuals tend to have a lower probability of switching topics. This aligns with the notion that interactions with each coauthor may be less frequent or robust in these cases, thus having a reduced effectiveness in inducing topic switches.

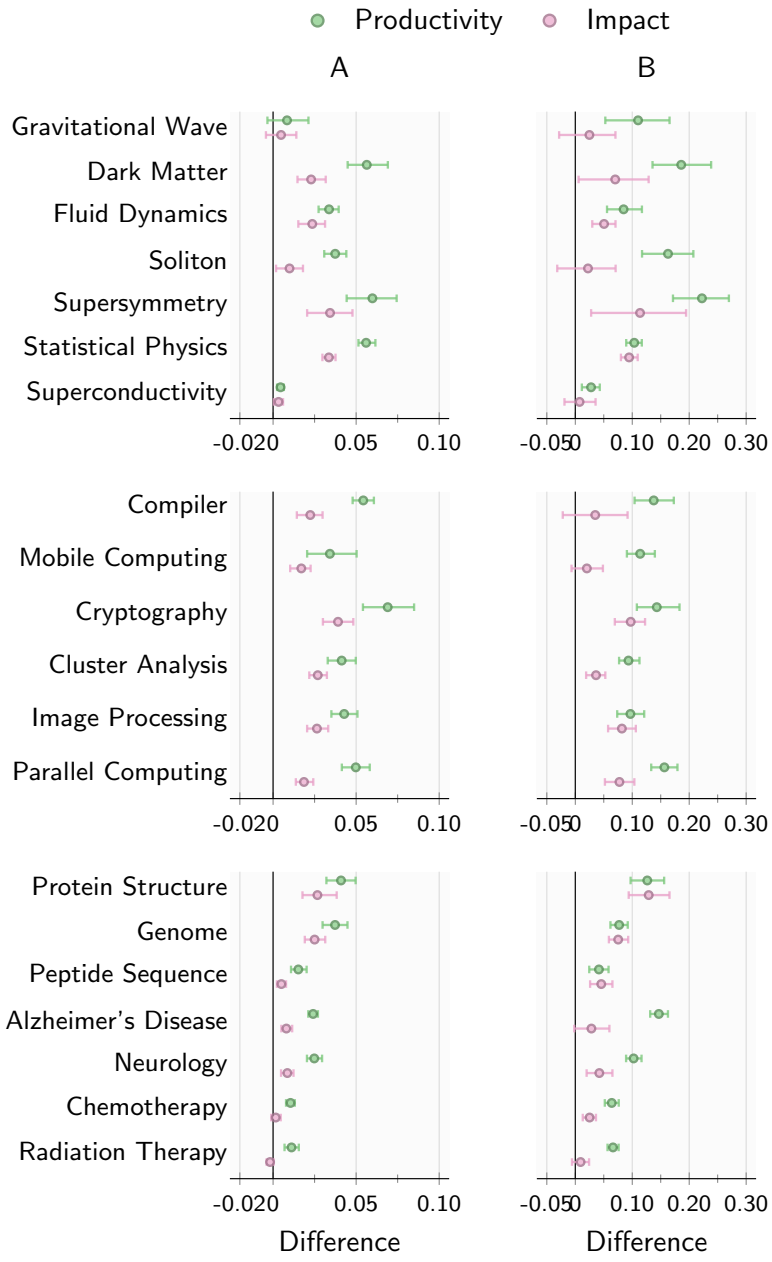


Figure 5.6: Experiment II results for $f^* = 0.10$. (A) The mean and 95% confidence interval of the means of the difference between the cumulative source activations of active authors in the top 10% and bottom 10% based on productivity (green) and impact (pink). (B) The mean and 95% confidence interval of the means of the difference between the chaperoning propensities of active authors in the top 10% and bottom 10% based on productivity (green) and impact (pink). A positive difference indicates that the effect is stronger for the top 10% active authors.

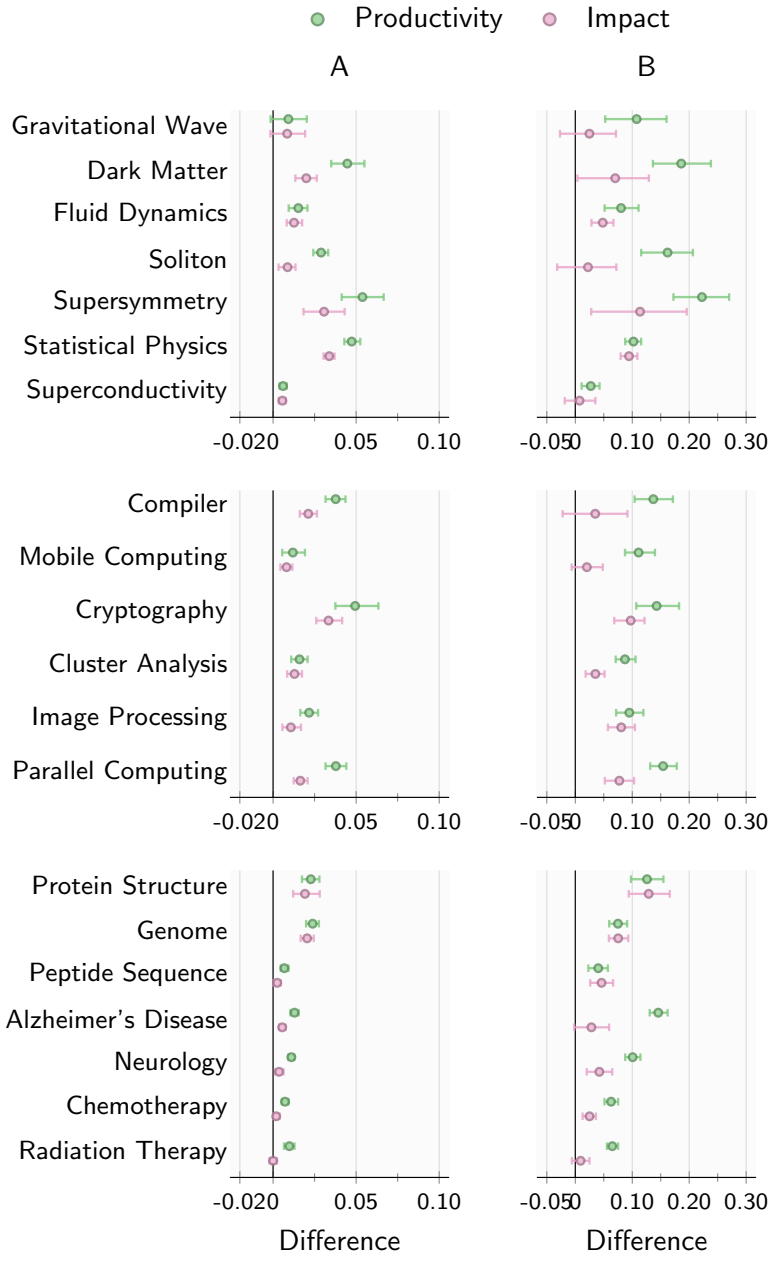


Figure 5.7: Experiment II. Same as 5.6, with threshold $f^\star = 0.20$. (A) Mean and 95% confidence interval of the means of the difference between the cumulative source activations of active authors in the top 10% and bottom 10% based on productivity (green) and impact (pink). (B) Mean and 95% confidence interval of the means of the difference between the cumulative chaperoning propensities of active authors in the top 10% and bottom 10% based on productivity (green) and impact (pink). A positive difference indicates that the effect is stronger for the top 10% active authors.

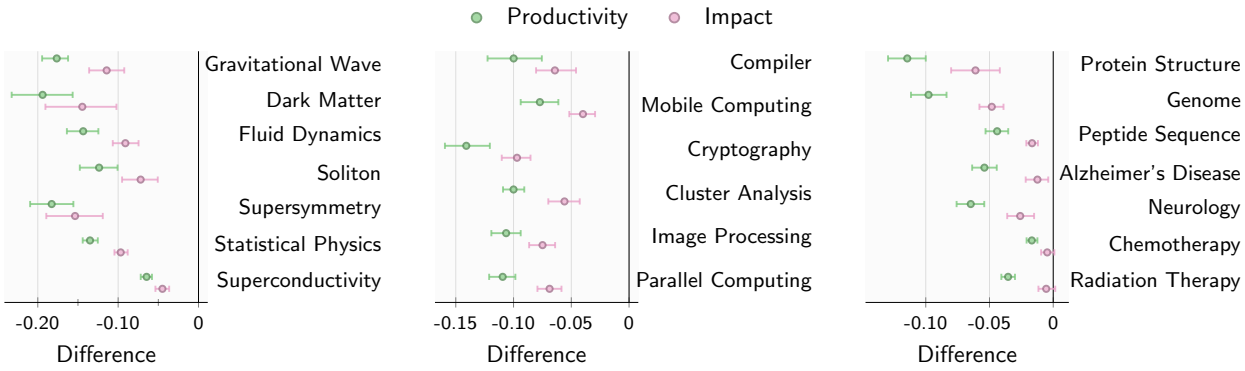


Figure 5.8: Dilution effect results for $f^* = 0.10$. The mean and 95% confidence interval of the mean of the difference between the cumulative source activations of active authors in the top 20% and bottom 20% bins, based on the average number of coauthors, among the top 10% active authors in productivity (green) and impact (pink). A negative difference across the topics indicates a *dilution* effect, wherein coauthors of prominent active scholars with fewer collaborators are more likely to switch topics.

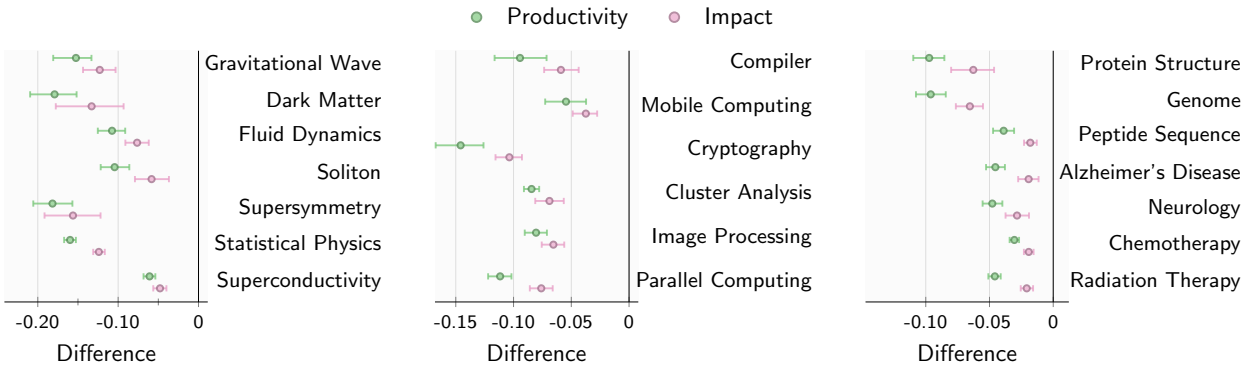


Figure 5.9: Experiment II. Same as Fig. 5.8, with threshold $f^\star = 0.20$. Dilution effect. The mean and 95% confidence interval of the mean of the difference between the cumulative source activations of active authors in the top 20% and bottom 20% bins, based on the average number of coauthors, from the set of top 10% active authors in productivity (green) and impact (pink). A negative difference across the topics indicates a *dilution* effect, wherein coauthors of prominent active scholars with fewer collaborators (on average) are more likely to switch topics.

## 5.4 Discussion

Collaboration serves as a means for scholars to delve deeper into existing knowledge and to be exposed to novel ideas. In the course of this chapter, we have thoroughly examined whether and how collaboration patterns influence the likelihood of researchers transitioning to new research topics. Our investigation has revealed that a scholar's propensity to embark on a new topic is influenced by their prior interactions with individuals who are already active in that particular field. This impact is directly proportional to the number of these interactions, such that a higher number of contacts corresponds to an elevated probability of topic transition. Our findings deviate from a basic baseline assumption that suggests independent effects arising from these contacts. This divergence from the baseline underscores the presence of complex, non-dyadic interactions that warrant further exploration and analysis.

Likewise, we quantified the likelihood that authors who are inactive in a given field but have collaborated with active authors in that field, will eventually publish papers on the new topic. This approach allows us to isolate the influence of the connection with the active author in the process of topic activation. It is important to emphasize that our study is structured such that prior collaborations between inactive and active authors pertain exclusively to subjects other than the focal topic. Thus, our analysis reveals that an active author can introduce an inactive author to a new research area, even when their past interactions did not directly involve that topic. This underscores the social nature of scientific interactions, where conversations and collaborations may venture beyond the immediate context that initially spurred them.

Furthermore, we investigated whether the probability of topic activation is influenced by certain characteristics of the active authors. Our findings revealed that authors who are more productive and have a greater impact tend to possess higher probabilities of inspiring their coauthors to switch topics and participate as coauthors in their initial publication on the new subject. This observation underscores the significant role that prominent authors play in shaping the research directions and collaborations within the scientific community.

Additionally, we demonstrated that as the number of coauthors collaborating with an active author increases, the probability of a topic switch decreases. This phenomenon aligns with the concept of *dilution*, where the influence or impact of an individual's input diminishes due to the challenge of maintaining strong interactions within a larger group. Notably, this effect is being brought to light for the first time through our research.

A plausible interpretation of our findings is that topic switches are driven by a process of social contagion, similar to the way new products are adopted [251, 260] or political propaganda spreads [253]. However, we must acknowledge that in observational studies like ours, there is the potential for selection effects [261]. The presence of a large number of active coauthors in a specific topic might be correlated with underlying homophily between the authors, which could facilitate the adoption of the topic even without direct influence from the active authors.

Our study leverages the OpenAlex database, a valuable open-access bibliometric resource. We utilize their author disambiguation and topic classification algorithms to carry out our analyses. However, it's important to acknowledge that these processes inherently introduce noise and potential biases. Additionally, there seem to be some gaps in citation coverage, which could contribute to the comparatively less robust results for impact metrics. We anticipate that future

updates to OpenAlex might help address these issues.
To mitigate these challenges, we conducted our analysis across multiple topics spanning three distinct scientific disciplines. While the magnitude of the observed effects may vary based on the specific topic, our core conclusions remain consistent across all topics, with only a few exceptions. This broader approach enhances the reliability and generalizability of our findings.

## 5.5 Methods

### 5.5.1 Data

In our analysis, we focus on papers from the February 2023 snapshot of the bibliometric dataset OpenAlex, which is the successor to Microsoft Academic Graph (MAG). We narrow our scope to papers published between 1990 and 2022 and limit them to a maximum of thirty authors per paper. Each paper is labeled with specific *concepts* (topics) using a classifier trained on MAG data.
Our investigation is based on constructing snapshots for three distinct fields: Physics, Computer Science (CS), and Biology and Medicine (BioMed). The Physics dataset comprises 19.7 million papers, while CS and BioMed contain 27.6 million and 43.52 million papers, respectively. Within each field, we have selected a subset of topics for analysis: seven topics in Physics, six topics in Computer Science, and seven topics in Biology and Medicine.
To ensure transparency and reproducibility, we have made our code and related data available on GitHub at the following link: GitHub Repository.
In our study, we focus on individual topics within each field and consider reference years spanning from 1995 to 2018. For each topic, we establish interaction and activation windows that encompass a minimum of 3000 papers. This threshold is essential to ensure that we have a substantial amount of data and a sufficiently large number of authors to conduct meaningful analyses.
Specifically, each topic we have selected has at least 10 reference years that satisfy the mentioned constraint. The statistical tests presented in the manuscript are aggregated across these different reference years, providing a comprehensive view of the topic dynamics.
For further details and specific information about each topic in Physics, Computer Science, Biology, and Medicine, you can refer to Tables 5.1 - 5.3. For each topic, we report: the number of considered reference years, the average number of papers written in the IW and in the EW, and the corresponding average number of active authors.

### 5.5.2 Overlap coefficient

We use the overlap coefficient to measure the degree of overlap between the different sets of authors picked based on productivity and impact.

$$\text{Overlap}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}.$$

In our case, the two sets are the same size, so a score of 10% implies that both sets share 10% of the elements.

Table 5.1: Summary information for Physics topics. #Papers: average number of Papers. #Authors: average number of active authors. Averages are computed over all time windows selected for a topic.

| Topic | #Windows | Interaction Window | | Activation Window | |
|---|---|---|---|---|---|
| | | #Papers | #Authors | #Papers | #Authors |
| Gravitational Wave | 10 | 3,613.70 | 5,745.20 | 5,486.40 | 9,160.30 |
| Dark Matter | 13 | 6,433.69 | 8,348.23 | 9,203.38 | 12,346.00 |
| Fluid Dynamics | 16 | 5,290.75 | 11,950.38 | 7,231.25 | 16,960.50 |
| Soliton | 18 | 4,004.39 | 5,715.61 | 4,700.89 | 7,014.89 |
| Supersymmetry | 20 | 5,328.85 | 4,827.45 | 5,470.75 | 5,361.25 |
| Statistical Physics | 23 | 88,147.52 | 109,702.70 | 105,018.87 | 137,680.65 |
| Superconductivity | 23 | 24,038.35 | 33,606.04 | 23,218.52 | 34,874.74 |

Table 5.2: Summary information for Computer Science topics. #Papers: average number of Papers. #Authors: average number of active authors. Averages are computed over all time windows selected for a topic.

| Topic | #Windows | Interaction Window | | Activation Window | |
|---|---|---|---|---|---|
| | | #Papers | #Authors | #Papers | #Authors |
| Compiler | 13 | 3,786.31 | 7,869.23 | 4,208.46 | 9,701.92 |
| Mobile Computing | 13 | 6,356.00 | 13,844.77 | 6,828.77 | 15,827.85 |
| Cryptography | 15 | 9,706.47 | 15,181.93 | 14,865.13 | 25,218.93 |
| Cluster Analysis | 21 | 18,585.57 | 36,645.95 | 30,996.52 | 63,910.10 |
| Image Processing | 23 | 13,149.65 | 28,191.35 | 16,617.65 | 38,089.70 |
| Parallel Computing | 23 | 31,453.30 | 48,006.87 | 38,271.61 | 61,960.22 |

Table 5.3: Summary information for Biology & Medicine topics. #Papers: average number of Papers. #Authors: average number of active authors. Averages are computed over all time windows selected for a topic.

| Topic | #Windows | Interaction Window | | Activation Window | |
|---|---|---|---|---|---|
| | | #Papers | #Authors | #Papers | #Authors |
| Protein Structure | 19 | 6,379.95 | 17,583.68 | 7,149.11 | 20,967.63 |
| Genome | 23 | 28,066.09 | 71,481.87 | 44,089.78 | 114,696.48 |
| Peptide Sequence | 23 | 12,347.48 | 43,348.96 | 9,733.04 | 37,330.09 |
| Alzheimer's Disease | 23 | 9,313.78 | 22,628.30 | 11,723.22 | 31,624.61 |
| Neurology | 23 | 9,260.17 | 26,046.57 | 12,795.70 | 39,515.00 |
| Chemotherapy | 23 | 36,280.48 | 104,649.39 | 47,760.09 | 143,505.65 |
| Radiation Therapy | 23 | 30,926.39 | 76,314.48 | 43,963.57 | 110,397.96 |

Table 5.4: Physics average Overlap Coefficient between the top and the bottom 10% of active authors selected based on Productivity and two different definitions of Impact. The first definition uses $C_{\text{avg}}$ and is used in the main text. The second definition uses $C_{\text{tot}}$. The degree of overlap is significantly greater for $C_{\text{tot}}$.

| Topic | Top 10% | | Bottom 10% | |
|---|---|---|---|---|
| | $C_{\text{avg}}$ | $C_{\text{tot}}$ | $C_{\text{avg}}$ | $C_{\text{tot}}$ |
| Gravitational Wave | 0.33 | 0.59 | 0.14 | 0.14 |
| Dark Matter | 0.31 | 0.56 | 0.15 | 0.15 |
| Fluid Dynamics | 0.24 | 0.38 | 0.11 | 0.11 |
| Soliton | 0.30 | 0.54 | 0.14 | 0.13 |
| Supersymmetry | 0.30 | 0.58 | 0.17 | 0.16 |
| Statistical Physics | 0.32 | 0.56 | 0.13 | 0.13 |
| Superconductivity | 0.26 | 0.60 | 0.16 | 0.15 |

$C_{\text{avg}}$: Average of incoming citations from papers on the topic.

$C_{\text{tot}}$: Sum of incoming citations from papers on the topic over all windows.

### 5.5.3 Author ranking metrics

Let $P$ be the set of papers published on topic $t$ authored by the set of active authors $A$ during the interaction window IW. Let $a$ be an active author who wrote $P_a$ papers during the IW. We define the following metrics to rank active authors and select the top and bottom 10%.

*Productivity:* quantifies the output of an active author $a$ in terms of the number of papers they have authored on the given topic $t$ during the interaction window (IW). More formally, it is the cardinality of the set $P \cap P_a$.

*Impact:* measures the influence or significance of an active author's work on the given topic. It's determined by the average number of citations received by the papers authored by $a$ during the IW from the papers in the set $P$ published on the same topic. The average number of citations is a better indicator of excellence than the total citation count [227].

Also, considering the average instead of the sum lowers its correlation with productivity, here measured by the overlap coefficient of Methods 5.5.2, as often the most productive authors are also the most cited ones [248]. The independence of these metrics is important because it enables you to analyze their effects separately and draw conclusions without the concern of confounding factors. For a more detailed view of the correlation statistics and how they relate to the topics in Physics, Computer Science, Biology, and Medicine, you can refer to Tables 5.4 - 5.6. For each topic, we report the average Overlap Coefficient between the top and the bottom 10% of active authors selected based on Productivity and two different definitions of Impact. The first definition uses $C_{\text{avg}}$ and is used in the main text. The second definition uses $C_{\text{tot}}$. The degree of overlap is significantly greater for $C_{\text{tot}}$.

### 5.5.4 Statistical test for difference of samples

To test whether two independent samples $X_1$ and $X_2$ are different concerning their means $\mu_1$ and $\mu_2$, we assume the null hypothesis $H_0 : \mu_1 = \mu_2$. We compute the mean and 95% confidence interval of $\mu_1 - \mu_2$ using bootstrapping and reject the null hypothesis $H_0$ at $p < 0.05$ if the

Table 5.5: Computer Science average Overlap Coefficient between the top and the bottom 10% of active authors selected based on Productivity and two different definitions of Impact. The first definition uses $C_{\text{avg}}$ and is used in the main text. The second definition uses $C_{\text{tot}}$. The degree of overlap is significantly greater for $C_{\text{tot}}$.

| **Topic** | **Top 10%** | | **Bottom 10%** | |
|---|---|---|---|---|
| | Mean | Sum | Mean | Sum |
| Compiler | 0.27 | 0.46 | 0.12 | 0.11 |
| Mobile Computing | 0.25 | 0.41 | 0.12 | 0.12 |
| Cryptography | 0.28 | 0.51 | 0.12 | 0.12 |
| Cluster Analysis | 0.25 | 0.41 | 0.12 | 0.12 |
| Image Processing | 0.25 | 0.42 | 0.12 | 0.11 |
| Parallel Computing | 0.24 | 0.53 | 0.13 | 0.13 |

$C_{\text{avg}}$: Average of incoming citations from papers on the topic.

$C_{\text{tot}}$: Sum of incoming citations from papers on the topic over all windows.

Table 5.6: Biology & Medicine average Overlap Coefficient between the top and the bottom 10% of active authors selected based on Productivity and two different definitions of Impact. The first definition uses $C_{\text{avg}}$ and is used in the main text. The second definition uses $C_{\text{tot}}$. The degree of overlap is significantly greater for $C_{\text{tot}}$.

| **Topic** | **Top 10%** | | **Bottom 10%** | |
|---|---|---|---|---|
| | Mean | Sum | Mean | Sum |
| Protein Structure | 0.22 | 0.46 | 0.13 | 0.13 |
| Genome | 0.22 | 0.50 | 0.13 | 0.13 |
| Peptide Sequence | 0.18 | 0.41 | 0.12 | 0.12 |
| Alzheimer's Disease | 0.19 | 0.55 | 0.13 | 0.14 |
| Neurology | 0.16 | 0.37 | 0.12 | 0.12 |
| Chemotherapy | 0.22 | 0.54 | 0.13 | 0.13 |
| Radiation Therapy | 0.24 | 0.54 | 0.13 | 0.13 |

$C_{\text{avg}}$: Average of incoming citations from papers on the topic.

$C_{\text{tot}}$: Sum of incoming citations from papers on the topic over all windows.

confidence interval *does not* contain 0 [262]. In other words, $X_1$ and $X_2$ are considered statistically different at $p < 0.05$ if the 95% confidence interval of the difference of their respective means does not contain 0. Furthermore, a positive mean of the difference indicates that $X_1 > X_2$, while a negative mean indicates $X_1 < X_2$.

### 5.5.5 Target activation probability

Let $n(k)$ be the number of inactive authors with exactly $k$ contacts during the exposure window, of whom $m(k)$ becomes active in the observation window. The *target activation probability* $P(k)$ is the probability of becoming active after having exactly $k$ contacts, defined as

$$P(k) = \frac{m(k)}{n(k)}. \tag{5.1}$$

The *cumulative target activation probability* $TAP(k)$ with $k$ or more contacts is given by

$$TAP(k) = \frac{\sum_k^\infty m(k)}{\sum_k^\infty n(k)}. \tag{5.2}$$

### 5.5.6 Simple baseline for membership closure

Let $p$ represent the probability of activation from a single contact. The probability of activation having $k$ contacts, acting independently of each other, is $P_{\text{base}}(k) = 1 - (1-p)^k$. We compute $p$ from the observed data using Eq. (5.1) as $p = P(1) = \frac{m(1)}{n(1)}$. This is the fraction of inactive authors with *exactly* one contact who became active as $P_{\text{base}}(1) = 1 - (1-p)^1 = p$. Like before, we calculate the cumulative target activation probability for the baseline $TAP_{\text{base}}(k)$ with $k$ or more contacts as

$$TAP_{\text{base}}(k) = \frac{\sum_k^\infty P_{\text{base}}(k) \cdot n(k)}{\sum_k^\infty n(k)}. \tag{5.3}$$

The denominator is the same as in Eq. (5.1) and comes from the observed data. The numerator represents the expected number of active authors if the contacts affect the activation independently.

### 5.5.7 Source activation probability

Let $n_a$ be the number of exclusive inactive coauthors of an active author $a$ in the IW. Let $m_a$ be the number of those exclusive inactive coauthors who become active in the AW. The *source activation probability* of scholar $a$ is thus

$$P_s^a = \frac{m_a}{n_a}. \tag{5.4}$$

We want to highlight that, for the probability to be meaningful, the value of $n_a$ must be greater than zero. As a result, our calculations are centered on active authors who have at least one exclusive inactive coauthor.
For any value within the range $0 \leq f \leq 1$, we calculate the fraction $C_s(f)$, which represents the proportion of all active authors whose source activation probability is equal to or exceeds

$f$. In essence, $C_s(f)$ forms the complementary cumulative probability distribution of the source activation probability $P_s^a$. It is to be expected that as $f$ increases, $C_s(f)$ rapidly declines towards 0. It's worth noting that the curves can become somewhat noisy when $f$ approaches 1 due to limited data statistics in that region.
To facilitate comparison between two sets of active authors, we concentrate on specific points at a chosen threshold $f^*$. This point is represented by the value $C_s(f^*)$, and we refer to it as the *cumulative source activation*. The reason we focus on this particular value is that the curves about the two sets of active authors are essentially indistinguishable at the tail end.
The selection of the threshold $f^*$ holds significance. Opting for a value of 0 or 1 would essentially yield the same probability for both sets of authors. Conversely, setting the value too low could lead to numerical issues. For instance, if there are only five inactive coauthors, the smallest nonzero fraction cannot be less than $1/5 = 0.20$. Conversely, selecting an excessively high value would result in weaker statistical support.
To address this, we have fixed the threshold at 0.10 for the outcomes presented in Figures 5.6 and 5.8. Furthermore, we provide results for the threshold of 0.20 in Figures 5.7 and 5.9. We analysed the results in Sections 5.3.1.

### 5.5.8 Chaperoning propensity

Let $m_a$ be the number of exclusive inactive coauthors of an active author $a$ who become active in the AW, which is the same as the numerator of Eq. (5.4). Let $i_a$ be the number of those authors who write their first paper on topic $t$ with $a$ in the AW. The *chaperoning probability* of $a$ is defined as

$$P_c^a = \frac{i_a}{m_a}. \tag{5.5}$$

We define the *chaperoning propensity* $P_c(f)$ corresponding to a specific threshold $f \in [0, 1]$ as the fraction of all active authors with $P_c^a \geq f$. We use the aforementioned values of 0.10 (5.6,5.8) and 0.20 (5.7,5.9) for the threshold $f$. We analysed the results in Sections 5.3.1.

## 5.6 Conclusion

In conclusion, our study provides a foundation for further explorations into the mechanisms underlying homophily within the scientific community. To gain a comprehensive understanding of these mechanisms, it's essential to effectively integrate all potential factors that might influence the phenomenon. Beyond productivity and impact metrics, topic switches could also be influenced by factors such as institutional affiliations. Collaborations within the same institution could be facilitated by increased opportunities for interaction and behavioral influence, while collaborations with researchers from prestigious institutions might carry more weight in the process.
Another relevant factor could be the number of citations received by a collaborator's papers. The higher the citation count, the stronger the connection between collaborators might be. Additionally, scientific affinity between coauthors could be considered by examining the similarity of their research outputs. Modern neural language models [263, 264], such as those based on

embeddings like Word2Vec or BERT, offer the capability to embed papers and authors in high-dimensional vector spaces. This allows us to quantify the similarity between authors based on the distance between their embeddings, providing a proxy for the similarity of their research outputs. Integrating these factors into future analyses could provide a more comprehensive picture of the dynamics shaping scientific collaborations and topic switches.

# Chapter 6

# Conclusion

In this thesis, we dealt with both the computational analysis of complex networks and their application to the real world. In the first chapters, we faced important issues in network science taking into consideration higher-order interactions. On one side, multilayer networks can treat multiple types of information; on the other side, simplicial complexes and hypergraphs can model group interactions. Empirical evidence has shown these more sophisticated representations significantly enhance modeling capabilities compared to standard single-layer graphs. However, dealing with more complex structures brings additional challenges and harder treatability. We tackled them by applying and adapting suited and tailored modern optimization methods.

Firstly, we dealt with the community detection problem over multiplex networks. We proposed an innovative approach that simultaneously consider the information contained in the different layers and takes into consideration the possible presence of noisy layers. These two properties are often not considered by many of the already proposed methods and they can lead to an improvement in the clustering prediction. The method extends the popular Louvain heuristic for single-layer graphs by introducing a quality function that takes modularity variance into account. We reformulated the problem as a multiobjective optimization problem, where the modularities of the single layers need to be optimized. Therefore, the algorithm employs a vector-valued modularity optimization process based on a specialized Pareto search. We conducted a thorough investigation of various versions of this method to analyze scenarios involving informative and noisy cases. In the informative case, each layer of the multiplex reflects the same community structure, while in the noisy case, some layers contain communities while others are purely noisy. Our evaluation encompassed extensive experiments comparing our method with nine baseline methods drawn from both network science and machine learning fields. We tested our approach on synthetic networks generated using the LFR and stochastic block models, and on five real-world multilayer datasets, considering both informative and noisy settings for each case. The results of our experiments highlighted the competitive performance of our multiobjective approach in comparison to the baseline algorithms.

In this thesis, we also dealt with an important issue in network science strictly related to community detection, that is the graph semi-supervised learning problem. Given a set of input labels, the goal is to build a classifier that takes into account both labeled and unlabeled observations, by considering a suitable loss function and the underlying graph structure of the observations. We wanted to extend it to multiplex networks. However, one of the main issues is related to the fact that not all the layers are equally informative. Therefore, a standard aggregation of the layers can lead to misleading results. We proposed an innovative method with the ability to learn a nonlinear aggregation function that adapts the weights assigned to each network based on the available labeled data. We formulated this problem as a bilevel optimization task and addressed it using an inexact Frank-Wolfe algorithm, coupled with a parametric Label Propagation strategy. We also provided a comprehensive convergence analysis of our method to ensure its mathematical robustness. Through extensive experimentation, we systematically compared our approach with single-layer methods and various baseline techniques, using both synthetic and real-world datasets. The consistent findings across these experiments demonstrated our method's proficiency in identifying informative layers. This leads to reliable and robust performance across diverse clustering scenarios, especially when certain layers contain substantial noise. Furthermore, our approach gives us also a better interpretability of the multiplex network under analysis.

Another important challenge in dealing with higher-order interactions is their additional complexity and harder treatability. We took into consideration an optimization-based formulation of the graph semi-supervised learning problem within the context of multilayer hypergraphs. In particular, we conducted an extensive comparison of various coordinate descent methods against the conventional gradient descent approach. Our finding highlights the potential for developing specialized coordinate methods tailored to address semi-supervised learning challenges within this specific context. Furthermore, we explored the impact of substituting the standard quadratic regularization term in the objective function with a more generalized $p$-regularizer. Our results in this context demonstrated the possibility of achieving improved performance through this modification.

In the last part of this thesis, we dealt with a specific complex network coming from the science of science. Science of science is a quite recent field that uses bibliometric data to study the evolution of science. Firstly, we gave a brief overview of the topics it deals with. Then, we focused on the analysis of collaboration networks evolving over time to analyze how collaborations influence scholars' transitions to new research topics. We categorized scholars into two distinct groups: active authors, who have actively contributed to the new topic by publishing in it, and inactive authors, who have not made such contributions. Our findings revealed several noteworthy insights. Firstly, the probability of an inactive scholar switching topics is also correlated with the number of active coauthors they collaborate with. Secondly, the probability of an inactive scholar transitioning to a new topic increases when they collaborate with active coauthors who exhibit high productivity and impact on that topic. Finally, the effect on an inactive scholar's likelihood of switching topics is influenced by the average number of inactive coauthors that active scholars have. We show the robustness of our findings considering different topics in the three disciplines of physics, computer science, and biology & medicine, although our analysis heavily depends on the bibliometric dataset OpenAlex. These findings collectively shed light on

the multifaceted relationship between collaboration patterns and scholars' transitions to new research topics. Understanding these dynamics is crucial for gaining insights into the underlying mechanisms that drive scholarly evolution and the formation of research communities.

# Acknowledgements

Inizialmente non volevo scrivere ringraziamenti. Non perché non avessi nessuno da ringraziare, anzi. Penso che tutte le persone che ho incontrato, per tanto o poco tempo, affini e meno affini, abbiano fatto la differenza in me. Inoltre, non sono brava con le parole, quindi sono sicura non riuscirei a trasmettere la mia gratitudine. Tuttavia, eccomi qui.

Ringrazio i miei supervisori Francesco Rinaldi e Francesco Tudisco i quali, oltre ad avermi trasmesso la loro passione in ambito scientifico, mi hanno sempre mostrato molta umanità. Mi hanno appoggiata, incoraggiata e accompagnata in ogni progetto e non mi hanno mai fatto sentire giudicata per i miei errori. Ringrazio i revisori Caterina De Bacco e Bissan Ghaddar per i commenti costruttivi. È stato un privilegio che due esperte abbiano letto questa tesi.

Ringrazio tutti i miei collaboratori. Da ognuno ho potuto imparare qualcosa e ogni confronto è stato prezioso. Ringrazio Andrea Cristofari che, con la sua esperienza e il suo supporto, ha dato un significante contributo a questa ricerca; Santo Fortunato per la sua ambizione e la sua dedizione al lavoro, per aver creduto in me dall'inizio e non avermi mai fatto sentire inferiore; Satyaki Sikdar per la sua gentilezza, passione e precisione, e per essere stato un prezioso collaboratore sempre disponibile ad aiutarmi.

Ringrazio AccelNet-Multinet per avermi dato la possibilità di confrontarmi con diversi ricercatori e di andare in visiting all'Indiana University a Bloomington. Ringrazio tutto il dipartimento, studenti e professori, che mi hanno accolta calorosamente.

Ringrazio i dottorandi di matematica, con i quali ho condiviso le difficoltà e le gioie di questo complicato percorso.

Sono grata alle comunità di reti complesse e ottimizzazione, nelle quali mi sono riconosciuta e sentita accolta. Penso che questo percorso mi abbia dato la grande possibilità di conoscere e confrontarmi con persone molto diverse ma accumunate dal vivere seguendo una passione.

Ringrazio di cuore la mia famiglia, che c'è sempre stata, mi ha incoraggiata e ha assecondato tutte le mie scelte. Ringrazio Agnese, Natascia, Nausicaa, Rio e Veronica, con le quali posso veramente essere me stessa. Abbiamo condiviso momenti e decisioni importanti e sono sicura che ne affronteremo molti altri insieme.

# Bibliography


[1] Herbert A Simon. The architecture of complexity. *Proceedings of the American philosophical society*, 106(6):467–482, 1962.

[2] Mark Newman. *Networks*. Oxford University Press, 2018.

[3] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

[4] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics reports*, 544(1):1–122, 2014.

[5] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[6] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 874:1–92, 2020.

[7] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[8] Santo Fortunato and Mark EJ Newman. 20 years of network community detection. *Nature Physics*, 18(8):848–850, 2022.

[9] Matteo Magnani, Obaida Hanteer, Roberto Interdonato, Luca Rossi, and Andrea Tagarelli. Community detection in multiplex networks. *ACM Computing Surveys (CSUR)*, 54(3): 1–35, 2021.

[10] Sara Venturini, Andrea Cristofari, Francesco Rinaldi, and Francesco Tudisco. A variance-aware multiobjective louvain-like method for community detection in multiplex networks. *Journal of Complex Networks*, 10(6), 11 2022. ISSN 2051-1329. doi: 10.1093/comnet/cnac048. URL `https://doi.org/10.1093/comnet/cnac048`.

[11] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. Semi-supervised Learning. Adaptive computation and machine learning. *Methods*, 1(1):4–8, 2010.

[12] Zixing Song, Xiangli Yang, Zenglin Xu, and Irwin King. Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[13] Amarnag Subramanya and Partha Pratim Talukdar. Graph-based semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(4):1–125, 2014.

[14] Pedro Mercado, Francesco Tudisco, and Matthias Hein. Generalized matrix means for semi-supervised learning with multilayer graphs. In *Advances in Neural Information Processing Systems*, pages 14877–14886, 2019.

[15] Sarvenaz Choobdar, Mehmet E Ahsen, Jake Crawford, Mattia Tomasoni, Tao Fang, David Lamparter, Junyuan Lin, Benjamin Hescott, Xiaozhe Hu, Johnathan Mercer, et al. Assessment of network module identification across complex diseases. *Nature methods*, 16(9): 843–852, 2019.

[16] Koji Tsuda, Hyunjung Shin, and Bernhard Schölkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21(suppl_2):ii59–ii65, 2005.

[17] Andreas Argyriou, Mark Herbster, and Massimiliano Pontil. Combining graph laplacians for semi–supervised learning. *Advances in Neural Information Processing Systems*, 18, 2005.

[18] Junting Ye and Leman Akoglu. Robust semi-supervised learning on multiple networks with noise. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 196–208. Springer, 2018.

[19] Sara Venturini, Andrea Cristofari, Francesco Rinaldi, and Francesco Tudisco. Learning the right layers a data-driven layer-aggregation strategy for semi-supervised learning on multilayer graphs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35006–35023. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/venturini23a.html`.

[20] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1): 3–34, 2015.

[21] Sara Venturini, Andrea Cristofari, Francesco Rinaldi, and Francesco Tudisco. Laplacian-based semi-supervised learning in multilayer hypergraphs by coordinate descent. *EURO Journal on Computational Optimization*, page 100079, 2023. ISSN 2192-4406. doi: https://doi.org/10.1016/j.ejco.2023.100079.

[22] An Zeng, Zhesi Shen, Jianlin Zhou, Jinshan Wu, Ying Fan, Yougui Wang, and H Eugene Stanley. The science of science: From the perspective of complex systems. *Physics reports*, 714:1–73, 2017.

[23] Santo Fortunato, Carl T Bergstrom, Katy Börner, James A Evans, Dirk Helbing, Staša Milojević, Alexander M Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, et al. Science of science. *Science*, 359(6379):eaao0185, 2018.

[24] Marta Tuninetti, Alberto Aleta, Daniela Paolotti, Yamir Moreno, and Michele Starnini. Prediction of new scientific collaborations through multiplex networks. *EPJ data science*, 10(1):25, 2021.

[25] Leo Torres, Ann S Blevins, Danielle Bassett, and Tina Eliassi-Rad. The why, how, and when of representations for complex systems. *SIAM Review*, 63(3):435–485, 2021.

[26] Jonas L Juul, Austin R Benson, and Jon Kleinberg. Hypergraph patterns and collaboration structure. *arXiv preprint arXiv:2210.02163*, 2022.

[27] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409, 2001.

[28] Filippo Radicchi, Santo Fortunato, and Alessandro Vespignani. Citation networks. *Models of science dynamics: encounters between complexity theory and information sciences*, pages 233–257, 2011.

[29] Jason Priem, Heather Piwowar, and Richard Orr. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv preprint arXiv:2205.01833*, 2022.

[30] Sara Venturini, Satyaki Sikdar, Francesco Rinaldi, Francesco Tudisco, and Santo Fortunato. Collaboration and topic switches in science. *arXiv preprint arXiv:2304.06826, submitted to Scientific Reports*, 2023.

[31] Riccardo Gallotti and Marc Barthelemy. The multilayer temporal network of public transport in great britain. *Scientific Data*, 2(1):1–8, 2015.

[32] Marya Bazzi, Mason A Porter, Stacy Williams, Mark McDonald, Daniel J Fenn, and Sam D Howison. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling & Simulation*, 14(1):1–41, 2016.

[33] Dane Taylor, Rajmonda S Caceres, and Peter J Mucha. Super-resolution community detection for layer-aggregated multilayer networks. *Physical Review X*, 7(3):031056, 2017.

[34] Dane Taylor, Saray Shai, Natalie Stanley, and Peter J Mucha. Enhanced detectability of community structure in multilayer networks through layer aggregation. *Physical Review Letters*, 116(22):228301, 2016.

[35] Joao Sedoc, Jean Gallier, Dean Foster, and Lyle Ungar. Semantic word clusters using signed spectral clustering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 939–949, 2017.

[36] Xiaochun Cao, Changqing Zhang, Chengju Zhou, Huazhu Fu, and Hassan Foroosh. Constrained multi-view video face clustering. *IEEE Transactions on Image Processing*, 24(11): 4381–4393, 2015.

[37] Niko Kiukkonen, Jan Blom, Olivier Dousse, Daniel Gatica-Perez, and Juha Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*, 68, 2010.

[38] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological Review*, 63(5):277, 1956.

[39] Wei Tang, Zhengdong Lu, and Inderjit S Dhillon. Clustering with multiple graphs. In *2009 Ninth IEEE International Conference on Data Mining*, pages 1016–1021. IEEE, 2009.

[40] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[41] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[42] Pedro Mercado, Antoine Gautier, Francesco Tudisco, and Matthias Hein. The power mean laplacian for multilayer graph clustering. *arXiv preprint arXiv:1803.00491*, 2018.

[43] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, 2008.

[44] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Finding and characterizing communities in multidimensional networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 490–494. IEEE, 2011.

[45] Jungeun Kim, Jae-Gil Lee, and Sungsu Lim. Differential flattening: A novel framework for community detection in multi-layer graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–23, 2016.

[46] Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data Mining and Knowledge Discovery*, 25(1):1–33, 2012.

[47] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[48] Soumajit Pramanik, Raphael Tackx, Anchit Navelkar, Jean-Loup Guillaume, and Bivas Mitra. Discovering community structure in multilayer networks. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 611–620. IEEE, 2017.

[49] Clara Pizzuti and Annalisa Socievole. Many-objective optimization for community detection in multi-layer networks. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 411–418. IEEE, 2017.

[50] Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, 5(1):011027, 2015.

[51] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[52] Caterina De Bacco, Eleanor A Power, Daniel B Larremore, and Cristopher Moore. Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, 95(4):042317, 2017.

[53] James D Wilson, John Palowitch, Shankar Bhamidi, and Andrew B Nobel. Community extraction in multilayer networks with heterogeneous community structure. *The Journal of Machine Learning Research*, 18(1):5458–5506, 2017.

[54] Zhiping Zeng, Jianyong Wang, Lizhu Zhou, and George Karypis. Coherent closed quasi-clique discovery from large dense graph databases. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–802, 2006.

[55] Jian Pei, Daxin Jiang, and Aidong Zhang. On mining cross-graph quasi-cliques. In *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 228–238, 2005.

[56] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. Clustering with multi-layer graphs: A spectral perspective. *IEEE Transactions on Signal Processing*, 60(11):5820–5831, 2012.

[57] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260. SIAM, 2013.

[58] Subhadeep Paul and Yuguo Chen. Spectral and matrix factorization methods for consistent community detection in multi-layer networks. *The Annals of Statistics*, 48(1):230–250, 2020.

[59] Dengyong Zhou and Christopher JC Burges. Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1159–1166, 2007.

[60] Pin-Yu Chen and Alfred O Hero. Multilayer spectral graph clustering via convex layer aggregation: Theory and algorithms. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):553–567, 2017.

[61] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds. *IEEE Transactions on Signal Processing*, 62(4):905–918, 2013.

[62] Kun Zhan, Changqing Zhang, Junpeng Guan, and Junsheng Wang. Graph learning for multiview clustering. *IEEE Transactions on Cybernetics*, 48(10):2887–2895, 2017.

[63] Kun Zhan, Chaoxi Niu, Changlu Chen, Feiping Nie, Changqing Zhang, and Yi Yang. Graph structure fusion for multiview clustering. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1984–1993, 2018.

[64] Kun Zhan, Feiping Nie, Jing Wang, and Yi Yang. Multiview consensus graph clustering. *IEEE Transactions on Image Processing*, 28(3):1261–1270, 2018.

[65] Feiping Nie, Lai Tian, and Xuelong Li. Multiview clustering via adaptively weighted procrustes. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2022–2030, 2018.

[66] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 28, 2014.

[67] Youwei Liang, Dong Huang, Chang-Dong Wang, and S Yu Philip. Multi-view graph learning by joint modeling of consistency and inconsistency. *IEEE transactions on neural networks and learning systems*, 2022.

[68] Robert Winkler. *An Introduction to Bayesian Inference and Decision.* Probabilistic Publishing, Gainesville, 2003.

[69] Tiago P Peixoto. Bayesian stochastic blockmodeling. *Advances in Network Clustering and Blockmodeling*, pages 289–332, 2019.

[70] Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26. Citeseer, 2004.

[71] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[72] Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 393–400, 2011.

[73] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1413–1421, 2011.

[74] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[75] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2007.

[76] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[77] Mingming Chen, Konstantin Kuzmin, and Boleslaw K Szymanski. Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems*, 1(1):46–65, 2014.

[78] Benjamin H Good, Yves-Alexandre De Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[79] Vilfredo Pareto. Cours d'économie politique, rouge. *Lausanne, Switzerland*, 1896.

[80] Lei Tang, Xufei Wang, and Huan Liu. Uncoverning groups via heterogeneous interaction analysis. In *2009 Ninth IEEE International Conference on Data Mining*, pages 503–512. IEEE, 2009.

[81] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583–617, 2002.

[82] Derek Greene and Pádraig Cunningham. A matrix factorization approach for integrating multiple data views. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 423–438. Springer, 2009.

[83] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2): 127–163, 2000.

[84] Dheeru Dua and Casey Graff. UCI machine learning repository – multiple features data set, 2017. URL `https://archive.ics.uci.edu/ml/datasets/Multiple+Features`.

[85] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 251–260, 2010.

[86] Vincent A Traag. Faster unfolding of communities: Speeding up the louvain algorithm. *Physical Review E*, 92(3):032801, 2015.

[87] Vikas Kumar, Anubhav Sisodia, Umesh Maini, and A Pankaj Anand. Comparing algorithms of community structure in networks. *Indian Journal of Science and Technology*, 9(44):1–5, 2016.

[88] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

[89] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pages 624–638. Springer, 2004.

[90] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[91] Matthias Hein, Simon Setzer, Leonardo Jost, and Syama Sundar Rangapuram. The total variation on hypergraphs: learning on hypergraphs revisited. *Advances in Neural Information Processing Systems*, 26, 2013.

[92] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[93] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*, 2018.

[94] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. Combining Label Propagation and Simple Models out-performs Graph Neural Networks. In *International Conference on Learning Representations*, 2020.

[95] Francesco Tudisco, Austin R Benson, and Konstantin Prokopchik. Nonlinear higher-order label spreading. In *Proceedings of the Web Conference 2021*, pages 2402–2413, 2021.

[96] Konstantin Prokopchik, Austin R Benson, and Francesco Tudisco. Nonlinear feature diffusion on hypergraphs. In *International Conference on Machine Learning*, pages 17945–17958. PMLR, 2022.

[97] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

[98] Jianxi Gao, Sergey V Buldyrev, H Eugene Stanley, and Shlomo Havlin. Networks formed from interdependent networks. *Nature physics*, 8(1):40–48, 2012.

[99] Zhen Wang, Lin Wang, Attila Szolnoki, and Matjaž Perc. Evolutionary games on multilayer networks: a colloquium. *The European physical journal B*, 88(5):1–15, 2015.

[100] Manlio De Domenico, Albert Solé-Ribalta, Sergio Gómez, and Alex Arenas. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences*, 111(23):8351–8356, 2014.

[101] Kyle Higham, Martina Contisciani, and Caterina De Bacco. Multilayer patent citation networks: A comprehensive analytical framework for studying explicit technological relationships. *Technological Forecasting and Social Change*, 179:121628, 2022.

[102] Mark E Dickison, Matteo Magnani, and Luca Rossi. *Multilayer social networks*. Cambridge University Press, 2016.

[103] Barry Bentley, Robyn Branicky, Christopher L Barnes, Yee Lian Chew, Eviatar Yemini, Edward T Bullmore, Petra E Vértes, and William R Schafer. The multilayer connectome of Caenorhabditis elegans. *PLoS computational biology*, 12(12):e1005283, 2016.

[104] Giuseppe Mangioni, Giuseppe Jurman, and Manlio De Domenico. Multilayer flows in molecular networks identify biological modules in the human proteome. *IEEE Transactions on Network Science and Engineering*, 7(1):411–420, 2018.

[105] Ginestra Bianconi. *Multilayer networks: structure and function*. Oxford university press, 2018.

[106] Tsuyoshi Kato, Hisahi Kashima, and Masashi Sugiyama. Robust label propagation on multiple networks. *IEEE Transactions on Neural Networks*, 20(1):35–44, 2008.

[107] Masayuki Karasuyama and Hiroshi Mamitsuka. Multiple graph label propagation by sparse integration. *IEEE transactions on neural networks and learning systems*, 24(12):1999–2012, 2013.

[108] Feiping Nie, Jing Li, Xuelong Li, et al. Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification. In *IJCAI*, pages 1881–1887, 2016.

[109] Raul J Mondragon, Jacopo Iacovacci, and Ginestra Bianconi. Multilink communities of multiplex networks. *PloS one*, 13(3):e0193821, 2018.

[110] Krishnamurthy Viswanathan, Sushant Sachdeva, Andrew Tomkins, and Sujith Ravi. Improved semi-supervised learning with multiple graphs. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3032–3041. PMLR, 2019.

[111] Kai Bergermann, Martin Stoll, and Toni Volkmer. Semi-supervised learning for aggregated multilayer graphs using diffuse interface methods and fast matrix-vector products. *SIAM Journal on Mathematics of Data Science*, 3(2):758–785, 2021.

[112] Ekta Gujral and Evangelos E Papalexakis. Smacd: Semi-supervised multi-aspect community detection. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 702–710. SIAM, 2018.

[113] Reza Ghorbanchian, Vito Latora, and Ginestra Bianconi. Hyper-diffusion on multiplex networks. *arXiv:2205.10291*, 2022.

[114] Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, Disha Makhija, and Mohit Kumar. Zoobp: Belief propagation for heterogeneous networks. *Proceedings of the VLDB Endowment*, 10(5):625–636, 2017.

[115] Mahsa Ghorbani, Mahdieh Soleymani Baghshah, and Hamid R Rabiee. Mgcn: semi-supervised classification in multi-layer graphs with graph convolutional networks. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 208–211, 2019.

[116] Marco Grassia, Manlio De Domenico, and Giuseppe Mangioni. mgnn: Generalizing the graph neural networks to the multilayer case. *arXiv preprint arXiv:2109.10119*, 2021.

[117] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.

[118] Karl Kunisch and Thomas Pock. A bilevel optimization approach for parameter learning in variational models. *SIAM Journal on Imaging Sciences*, 6(2):938–983, 2013.

[119] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

[120] Stephan Dempe and Alain Zemkoho. Bilevel optimization. In *Springer optimization and its applications. Vol. 161*. Springer, 2020.

[121] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[122] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435. PMLR, 2013.

[123] Immanuel M Bomze, Francesco Rinaldi, and Damiano Zeffiro. Frank–wolfe and friends: a journey into projection-free first-order optimization methods. *4OR*, 19(3):313–345, 2021.

[124] Robert M Freund and Paul Grigas. New analysis and results for the frank–wolfe method. *Mathematical Programming*, 155(1):199–230, 2016.

[125] Immanuel M Bomze, Francesco Rinaldi, and Damiano Zeffiro. Active set complexity of the away-step frank–wolfe algorithm. *SIAM Journal on Optimization*, 30(3):2470–2500, 2020.

[126] Francesco Rinaldi and Damiano Zeffiro. Avoiding bad steps in frank-wolfe variants. *Computational Optimization and Applications*, pages 1–40, 2022.

[127] Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22(2):507–560, 2022.

[128] Anit Kumar Sahu, Manzil Zaheer, and Soummya Kar. Towards gradient free and projection free stochastic optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3468–3477. PMLR, 2019.

[129] Richard S Varga. *Geršgorin and his circles*, volume 36. Springer Science & Business Media, 2010.

[130] Rafael Martí, Mauricio GC Resende, and Celso C Ribeiro. Multi-start methods for combinatorial optimization. *European Journal of Operational Research*, 226(1):1–8, 2013.

[131] Sara Venturini, Andrea Cristofari, Francesco Rinaldi, and Francesco Tudisco. A variance-aware multiobjective louvain-like method for community detection in multiplex networks. *Journal of Complex Networks*, 10(6):cnac048, 2022.

[132] Derek Greene and Pádraig Cunningham. A matrix factorization approach for integrating multiple data views. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 423–438. Springer, 2009.

[133] Derek Greene and Pádraig Cunningham. Producing accurate interpretable clusters from high-dimensional data. In *European conference on principles of data mining and knowledge discovery*, pages 486–494. Springer, 2005.

[134] D Dua and KT Efi. Uci machine learning repository multi-objective particle swarm optimization: Theory, 2017.

[135] Q Lu and L Getoor. Link-based classification using labeled and unlabeled data. In *ICML 2003 workshop on The Continuum_from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.

[136] Obaida Hanteer, Luca Rossi, Davide Vega D'Aurelio, and Matteo Magnani. From interaction to participation: The role of the imagined audience in social media community detection and an application to political communication on twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 531–534. IEEE, 2018.

[137] Luca Rossi and Matteo Magnani. Towards effective visual analytics on multiplex and multilayer networks. *Chaos, Solitons & Fractals*, 72:68–76, 2015.

[138] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.

[139] Ahmed El Alaoui, Xiang Cheng, Aaditya Ramdas, Martin J Wainwright, and Michael I Jordan. Asymptotic behavior of\ell_p-based laplacian regularization in semi-supervised learning. In *Conference on Learning Theory*, pages 879–906. PMLR, 2016.

[140] Boaz Nadler, Nathan Srebro, and Xueyuan Zhou. Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. *Advances in neural information processing systems*, 22:1330–1338, 2009.

[141] Rasmus Kyng, Anup Rao, Sushant Sachdeva, and Daniel A Spielman. Algorithms for Lipschitz learning on graphs. In *Conference on Learning Theory*, pages 1190–1223. PMLR, 2015.

[142] Xueyuan Zhou and Mikhail Belkin. Semi-supervised learning by higher order regularization. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 892–900. JMLR Workshop and Conference Proceedings, 2011.

[143] Andrea Cristofari, Francesco Rinaldi, and Francesco Tudisco. Total variation based community detection using a nonlinear optimization approach. *SIAM Journal on Applied Mathematics*, 80(3):1392–1419, 2020.

[144] Thomas Bühler and Matthias Hein. Spectral clustering based on the graph p-laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 81–88, 2009.

[145] Francesco Tudisco, Pedro Mercado, and Matthias Hein. Community detection in networks via nonlinear modularity eigenvectors. *SIAM J. Applied Mathematics*, 78:2393–2419, 2018.

[146] F. Tudisco and D. Zhang. Nonlinear Spectral Duality. *arxiv:2209.06241*, 2022.

[147] Jeff Calder. The game theoretic p-Laplacian and semi-supervised learning with few labels. *Nonlinearity*, 32(1):301, 2018.

[148] Mauricio Flores, Jeff Calder, and Gilad Lerman. Analysis and algorithms for $\ell_p$-based semi-supervised learning on graphs. *Applied and Computational Harmonic Analysis*, 60: 77–122, 2022.

[149] Dejan Slepcev and Matthew Thorpe. Analysis of p-Laplacian regularization in semisupervised learning. *SIAM Journal on Mathematical Analysis*, 51(3):2085–2120, 2019.

[150] Francesco Tudisco and Matthias Hein. A nodal domain theorem and a higher-order Cheeger inequality for the graph $p$-Laplacian. *EMS Journal of Spectral Theory*, 8:883–908, 2018.

[151] Uthsav Chitra and Benjamin Raphael. Random walks on hypergraphs with edge-dependent vertex weights. In *International Conference on Machine Learning*, pages 1172–1181. PMLR, 2019.

[152] Rania Ibrahim and David F Gleich. Local hypergraph clustering using capacity releasing diffusion. *Plos one*, 15(12):e0243485, 2020.

[153] Nate Veldt, Austin R Benson, and Jon Kleinberg. Minimizing localized ratio cut objectives in hypergraphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1708–1718, 2020.

[154] Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 555–564, 2017.

[155] Chenzi Zhang, Shuguang Hu, Zhihao Gavin Tang, and TH Hubert Chan. Re-revisiting learning on hypergraphs: confidence interval and subgradient method. In *International Conference on Machine Learning*, pages 4026–4034. PMLR, 2017.

[156] Joyce Jiyoung Whang, Rundong Du, Sangwon Jung, Geon Lee, Barry Drake, Qingqing Liu, Seonggoo Kang, and Haesun Park. MEGA: Multi-view semi-supervised clustering of hypergraphs. *Proceedings of the VLDB Endowment*, 13(5):698–711, 2020.

[157] Ayhan Demiriz and Kristin P Bennett. Optimization approaches to semi-supervised learning. In *Complementarity: Applications, Algorithms and Extensions*, pages 121–141. Springer, 2001.

[158] Hao-Chen Dong, Yu-Feng Li, and Zhi-Hua Zhou. Learning from semi-supervised weak-label data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[159] Piero Deidda, Mario Putti, and Francesco Tudisco. Nodal domain count for the generalized graph p-Laplacian. *Applied and Computational Harmonic Analysis*, 64:1–32, 2023.

[160] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

[161] Sameer Agarwal, Jongwoo Lim, Lihi Zelnik-Manor, Pietro Perona, David Kriegman, and Serge Belongie. Beyond pairwise clustering. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 838–845. IEEE, 2005.

[162] Martina Contisciani, Federico Battiston, and Caterina De Bacco. Inference of hyperedges and overlapping communities in hypergraphs. *Nature communications*, 13(1):7229, 2022.

[163] Nate Veldt, Austin R Benson, and Jon Kleinberg. Hypergraph cuts with general splitting functions. *SIAM Review*, 64(3):650–685, 2022.

[164] Francesco Tudisco and Desmond J Higham. Core-periphery detection in hypergraphs. *SIAM Journal on Mathematics of Data Science*, 5(1):1–21, 2023.

[165] Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 1999.

[166] Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72 (1):7–35, 1992.

[167] Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[168] Luigi Grippo and Marco Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization methods and software*, 10(4):587–637, 1999.

[169] RWH Sargent and DJ Sebastian. On the convergence of sequential minimization algorithms. *Journal of Optimization Theory and Applications*, 12(6):567–575, 1973.

[170] EG Birgin and JM Martínez. Block coordinate descent for smooth nonconvex constrained minimization. *Computational Optimization and Applications*, 83(1):1–27, 2022.

[171] Andrea Cassioli, David Di Lorenzo, and Marco Sciandrone. On the convergence of inexact block coordinate descent methods for constrained optimization. *European Journal of Operational Research*, 231(2):274–281, 2013.

[172] Andrea Cristofari. An almost cyclic 2-coordinate descent method for singly linearly constrained problems. *Computational Optimization and Applications*, 73(2):411–452, 2019.

[173] Luigi Grippo and Marco Sciandrone. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Operations research letters*, 26(3):127–136, 2000.

[174] Stefano Lucidi, Laura Palagi, Arnaldo Risi, and Marco Sciandrone. A convergent decomposition algorithm for support vector machines. *Computational Optimization and Applications*, 38(2):217–234, 2007.

[175] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[176] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[177] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1): 1–38, 2014.

[178] Alireza Ghaffari-Hadigheh, Lennart Sinjorgo, and Renata Sotirov. On convergence of a $q$-random coordinate constrained algorithm for non-convex problems. *arXiv preprint arXiv:2210.09665*, 2022.

[179] Ion Necoara, Yurii Nesterov, and François Glineur. Random block coordinate descent methods for linearly constrained optimization over networks. *Journal of Optimization Theory and Applications*, 173(1):227–254, 2017.

[180] Ion Necoara and Andrei Patrascu. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. *Computational Optimization and Applications*, 57(2):307–337, 2014.

[181] Andrei Patrascu and Ion Necoara. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *Journal of Global Optimization*, 61(1):19–46, 2015.

[182] Sashank Reddi, Ahmed Hefny, Carlton Downey, Avinava Dubey, and Suvrit Sra. Large-scale randomized-coordinate descent methods with non-separable linear constraints. *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

[183] Andrea Cristofari. A decomposition method for lasso problems with zero-sum constraint. *European Journal of Operational Research*, 306(1):358–369, 2023.

[184] Marianna De Santis, Stefano Lucidi, and Francesco Rinaldi. A Fast Active Set Block Coordinate Descent Algorithm for $\ell_1$-Regularized Least Squares. *SIAM Journal on Optimization*, 26(1):781–809, 2016.

[185] Amir Beck. The 2-coordinate descent method for solving double-sided simplex constrained minimization problems. *Journal of Optimization Theory and Applications*, 162(3):892–919, 2014.

[186] Chih-Jen Lin. On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*, 12(6):1288–1298, 2001.

[187] Julie Nutini, Issam Laradji, and Mark Schmidt. Let's Make Block Coordinate Descent Converge Faster: Faster Greedy Rules, Message-Passing, Active-Set Complexity, and Superlinear Convergence. *Journal of Machine Learning Research*, 23(131):1–74, 2022.

[188] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.

[189] Paul Tseng and Sangwoon Yun. Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *Journal of optimization theory and applications*, 140(3):513–535, 2009.

[190] Mert Gürbüzbalaban, Asuman Ozdaglar, Nuri Denizcan Vanli, and Stephen J Wright. Randomness and permutations in coordinate descent methods. *Mathematical Programming*, 181(2):349–376, 2020.

[191] Francesco Tudisco, Austin R Benson, and Konstantin Prokopchik. Nonlinear higher-order label spreading. In *Proceedings of The Web Conference*, page to appear, 2021.

[192] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL `https://proceedings.neurips.cc/paper/2003/file/87682805257e619d49b8e0dfdc14affa-Paper.pdf`.

[193] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[194] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 795–811. Springer, 2016.

[195] Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the Gauss-Southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641. PMLR, 2015.

[196] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling ii: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

[197] Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. *The Journal of Machine Learning Research*, 17(1):2657–2681, 2016.

[198] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156:433–484, 2016.

[199] Saverio Salzo and Silvia Villa. Parallel random block-coordinate forward–backward algorithm: a unified convergence analysis. *Mathematical Programming*, 193(1):225–269, 2022.

[200] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[201] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91:201–213, 2002.

[202] Philip S Chodrow, Nate Veldt, and Austin R Benson. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances*, 7(28):eabh1303, 2021.

[203] Sarah Elliott. Survey of author name disambiguation: 2004 to 2010. *Library philosophy and practice*, 473:1–11, 2010.

[204] Anderson A Ferreira, Marcos André Gonçalves, and Alberto HF Laender. A brief survey of automatic methods for author name disambiguation. *Acm sigmod record*, 41(2):15–26, 2012.

[205] Ijaz Hussain and Sohail Asghar. A survey of author name disambiguation techniques: 2010–2016. *The knowledge engineering review*, 32:e22, 2017.

[206] Sadamori Kojaku, Xiaoran Yan, Jisung Yoon, Filipi N Silva, Vincent Larivière, and Yong-Yeol Ahn. DisamBERT: Author name disambiguation with BERT. 2023. `https://drive.google.com/file/d/1zKcrX4Gi-aVFG_2A7-eUxSpl4THNyilQ/view`.

[207] Mark EJ Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E*, 64(1):016131, 2001.

[208] Mark EJ Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132, 2001.

[209] Mark EJ Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the national academy of sciences*, 101(suppl_1):5200–5205, 2004.

[210] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[211] Mark EJ Newman. The first-mover advantage in scientific publication. *Europhysics letters*, 86(6):68001, 2009.

[212] Eugene Garfield. Premature discovery or delayed recognition-why. *Current contents*, pages 5–10, 1980.

[213] Anthony FJ Van Raan. Sleeping beauties in science. *Scientometrics*, 59(3):467–472, 2004.

[214] Zheng Xie, Zhenzheng Ouyang, Qi Liu, and Jianping Li. A geometric graph model for citation networks of exponentially growing scientific papers. *Physica A: statistical mechanics and its applications*, 456:167–175, 2016.

[215] Staša Milojević. Principles of scientific research team formation and evolution. *Proceedings of the national academy of sciences*, 111(11):3984–3989, 2014.

[216] Roger Guimera, Brian Uzzi, Jarrett Spiro, and Luis A Nunes Amaral. Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308 (5722):697–702, 2005.

[217] Lingfei Wu, Dashun Wang, and James A Evans. Large teams develop and small teams disrupt science and technology. *Nature*, 566(7744):378–382, 2019.

[218] Michael Park, Erin Leahey, and Russell J Funk. Papers and patents are becoming less disruptive over time. *Nature*, 613(7942):138–144, 2023.

[219] Alexander M Petersen, Felber Arroyave, and Fabio Pammolli. The disruption index is biased by citation inflation. *arXiv preprint arXiv:2306.01949*, 2023.

[220] Mario Coccia. The evolution of scientific disciplines in applied sciences: dynamics and empirical properties of experimental physics. *Scientometrics*, 124(1):451–487, 2020.

[221] Xiaoling Sun, Staša Kaur, Jasleenand Milojević, Alessandro Flammini, and Filippo Menczer. Social dynamics of science. *Scientific reports*, 3(1):1069, 2013.

[222] Johan SG Chu and James A Evans. Slowed canonical progress in large fields of science. *Proceedings of the national academy of sciences*, 118(41):e2021636118, 2021.

[223] Chakresh Kumar Singh, Emma Barme, Robert Ward, Liubov Tupikina, and Marc Santolini. Quantifying the rise and fall of scientific fields. *PloS one*, 17(6):e0270131, 2022.

[224] Jorge E Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the national academy of sciences*, 102(46):16569–16572, 2005.

[225] Filippo Radicchi, Santo Fortunato, Benjamin Markines, and Alessandro Vespignani. Diffusion of scientific credits and the ranking of scientists. *Physical review E*, 80(5):056103, 2009.

[226] Vladlen Koltun and David Hafner. The h-index is no longer an effective correlate of scientific reputation. *PLoS One*, 16(6):e0253397, 2021.

[227] Şirag Erkol, Satyaki Sikdar, Filippo Radicchi, and Santo Fortunato. Consistency pays off in science. *Quantitative Science Studies*, 4(2):491–500, 2023.

[228] Louis de Mesnard. Attributing credit to coauthors in academic publishing: The 1/n rule, parallelization, and team bonuses. *European Journal of Operational Research*, 260(2): 778–788, 2017.

[229] Eugene Garfield. Citation analysis as a tool in journal evaluation: Journals can be ranked by frequency and impact of citations for science policy studies. *Science*, 178(4060):471–479, 1972.

[230] Vincent Larivière, Chaoqun Ni, Yves Gingras, Blaise Cronin, and Cassidy R Sugimoto. Bibliometrics: Global gender disparities in science. *Nature*, 504(7479):211–213, 2013.

[231] Junming Huang, Alexander J Gates, Roberta Sinatra, and Albert-László Barabási. Historical comparison of gender inequality in scientific careers across countries and disciplines. *Proceedings of the national academy of sciences*, 117(9):4609–4616, 2020.

[232] Yang Yang, Tanya Y Tian, Teresa K Woodruff, Benjamin F Jones, and Brian Uzzi. Gender-diverse teams produce more novel and higher-impact scientific ideas. *Proceedings of the national academy of sciences*, 119(36):e2200841119, 2022.

[233] Xinyi Zhao, Aliakbar Akbaritabar, Ridhi Kashyap, and Emilio Zagheni. A gender perspective on the global migration of scholars. *Proceedings of the national academy of sciences*, 120 (10):e2214664120, 2023.

[234] Bedoor K AlShebli, Talal Rahwan, and Wei Lee Woon. The preeminence of ethnic diversity in scientific collaboration. *Nature communications*, 9(1):5163, 2018.

[235] Weihua Li, Sam Zhang, Zhiming Zheng, Skyler J Cranmer, and Aaron Clauset. Untangling the network effects of productivity and prominence among scientists. *Nature communications*, 13(1):4907, 2022.

[236] Aaron Clauset, Samuel Arbesman, and Daniel B Larremore. Systematic inequality and hierarchy in faculty hiring networks. *Science advances*, 1(1):e1400005, 2015.

[237] Dashun Wang, Chaoming Song, and Albert-László Barabási. Quantifying long-term scientific impact. *Science*, 342(6154):127–132, 2013.

[238] Roberta Sinatra, Dashun Wang, Pierre Deville, Chaoming Song, and Albert-László Barabási. Quantifying the evolution of individual scientific impact. *Science*, 354(6312):aaf5239, 2016.

[239] Weihua Li, Tomaso Aste, Fabio Caccioli, and Giacomo Livan. Early coauthorship with top scientists predicts success in academic careers. *Nature communications*, 10(1):5170, 2019.

[240] Yian Yin, Yang Wang, James A Evans, and Dashun Wang. Quantifying the dynamics of failure across science, startups and security. *Nature*, 575(7781):190–194, 2019.

[241] Stefan Wuchty, Benjamin F Jones, and Brian Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.

[242] Raj Kumar Pan, Kimmo Kaski, and Santo Fortunato. World citation and collaboration networks: uncovering the role of geography in science. *Scientific reports*, 2(1):1–7, 2012.

[243] Alexander Michael Petersen. Quantifying the impact of weak, strong, and super ties in scientific careers. *Proceedings of the National Academy of Sciences*, 112(34):E4671–E4680, 2015.

[244] Jasjit Singh. Collaborative networks as determinants of knowledge diffusion patterns. *Management science*, 51(5):756–770, 2005.

[245] Olav Sorenson, Jan W Rivkin, and Lee Fleming. Complexity, networks and knowledge flow. In *Academy of Management Proceedings*, volume 2004, pages R1–R6. Academy of Management Briarcliff Manor, NY 10510, 2004.

[246] An Zeng, Zhesi Shen, Jianlin Zhou, Ying Fan, Zengru Di, Yougui Wang, H Eugene Stanley, and Shlomo Havlin. Increasing trend of scientists to switch between topics. *Nature communications*, 10(1):1–11, 2019.

[247] Tao Jia, Dashun Wang, and Boleslaw K Szymanski. Quantifying patterns of research-interest evolution. *Nature Human Behaviour*, 1(4):1–7, 2017.

[248] An Zeng, Ying Fan, Zengru Di, Yougui Wang, and Shlomo Havlin. Impactful scientists have higher tendency to involve collaborators in new topics. *Proceedings of the National Academy of Sciences*, 119(33):e2207436119, 2022.

[249] Damon Centola and Michael Macy. Complex contagions and the weakness of long ties. *American journal of Sociology*, 113(3):702–734, 2007.

[250] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379, 2007.

[251] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es, 2007.

[252] Damon Centola. The spread of behavior in an online social network experiment. *science*, 329(5996):1194–1197, 2010.

[253] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012.

[254] William Goffman and Vaun A Newill. Generalization of epidemic theory: An application to the transmission of ideas. *Nature*, 204(4955):225–228, 1964.

[255] William Goffman. Mathematical approach to the spread of scientific ideas—the history of mast cell research. *Nature*, 212(5061):449–452, 1966.

[256] Luís MA Bettencourt, Ariel Cintrón-Arias, David I Kaiser, and Carlos Castillo-Chávez. The power of a good idea: Quantitative modeling of the spread of ideas from epidemiological models. *Physica A: Statistical Mechanics and its Applications*, 364:513–536, 2006.

[257] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[258] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, 2006.

[259] Vedran Sekara, Pierre Deville, Sebastian E Ahnert, Albert-László Barabási, Roberta Sinatra, and Sune Lehmann. The chaperone effect in scientific publishing. *Proceedings of the National Academy of Sciences*, 115(50):12603–12607, 2018.

[260] Frank M Bass. A new product growth for model consumer durables. *Management science*, 15(5):215–227, 1969.

[261] Cosma Rohilla Shalizi and Andrew C. Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods & Research*, 40(2): 211–239, 2011.

[262] Martin J Gardner and Douglas G Altman. Confidence intervals rather than p values: estimation rather than hypothesis testing. *Br Med J (Clin Res Ed)*, 292(6522):746–750, 1986.

[263] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[264] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.