RESEARCH ARTICLE

Social Network Analysis: A Novel Paradigm for Improving Community Detection

Rodrigo Hernández[1] · Inmaculada Gutiérrez[1,2] · Javier Castro[1,2]

Received: 15 January 2025 / Revised: 14 March 2025 / Accepted: 26 March 2025

Abstract
Social network analysis has become increasingly important across a wide range of fields, offering valuable insights into complex systems of interconnected entities. One of the fundamental challenges in this field is the community detection problem, which involves identifying groups within networks. Multiple algorithms have been proposed, exploring new approaches to finding solutions for cohesive partitions of the graph. One of the most considered philosophies when defining this type of technique is the use of the graph's adjacency matrix as input and the consideration of modularity as the function to be optimized. We propose an enhancement to this approach to community detection by incorporating high-order relationships between nodes, allowing for a more comprehensive capture of network structure. By modifying the algorithm's input, our method improves community detection accuracy. Moreover, our proposed approach is universal, applicable to any algorithm that utilizes a matrix as input. Its value is further validated through a comprehensive set of results, comparing the original problem with the enhanced method we present. We also present a tourism case study.

Keywords Social network analysis · Community detection · Graphs · Machine learning

1 Introduction

Social networks are instrumental in understanding complex and interconnected systems that influence human behavior and interaction in numerous domains. Consequently, social network analysis (SNA) has emerged as one of the most influential fields of study in recent decades, enabling the exploration of networked structures in areas ranging from sociology and biology to marketing and communication [1–5]. A core task within SNA is the community detection problem (CDP), which seeks to identify communities/groups of nodes within a graph that are more densely connected internally than with the rest of the network [6]. Accurate detection of these communities is crucial, as it uncovers underlying structural patterns that provide valuable information on the functioning of complex systems [7].

Complex networks-or graphs characterized by non-trivial properties such as hierarchical or overlapping community structures and large scale-pose unique challenges to the CDP. These networks typically represent real-world systems which require sophisticated methods to capture their intricate connectivity patterns effectively.

CDP has been widely studied, leading to numerous algorithms and evaluation methods [8–12]. Fortunato and Hric's comprehensive review [6] has become a foundational reference, providing a broad overview of existing

R. Hernández, I. Gutiérrez and J. Castro have contributed equally to this work

https://doi.org/10.1007/s44196-025-00812-9

Published online: 22 April 2025

techniques and their applications. Additional classic contributions include [13], where key methods for network analysis were introduced; modularity-based approaches by Girvan and Newman [14] or algorithms in [15–17]. Among these, the Louvain algorithm stands out due to its popularity, widespread application, and quality of its results. Developed by Blondel et al. [18], it is popular for its efficiency in large networks, although it can yield suboptimal partitions. In the literature, several modifications of the Louvain algorithm can be found [19, 20], such as the proposal that optimizes computational time [21].

Despite its notable importance, while the Louvain algorithm is widely regarded for its efficiency and effectiveness in modularity optimization, it often fails to escape local optima, particularly in complex networks with nuanced community structures [8, 19]. This issue was partially addressed by Traag et al. [22], who proposed a refinement of the Louvain algorithm to avoid the creation of disconnected partitions. Nevertheless, the problem of greediness was not fully resolved with this proposal. Then, in [23], the authors introduced a flexible methodology that systematically modifies the input to the Louvain algorithm, thus improving the quality of its output in terms of modularity and community partitioning, specifically addressing greediness.

Inspired by Ref. [23], in this paper, we introduce a novel enhancement to improve and optimize any CDP algorithm. Our approach involves altering the initial adjacency matrix input to any CDP algorithm in a way that captures more intricate relationships between nodes, effectively expanding the algorithm's perspective beyond immediate connections. This input modification can be implemented in multiple ways, including through higher-order adjacency matrices or by inputting the intrinsic information of the graphs provided by using different resolution parameters in its aggregation. In this broader framework, however, we introduce a variety of potential transformations, such as applying the aggregation of higher-order matrices and adjusting it to fine-tune the influence of longer-range connections on community detection.

By introducing these adjustments, our method enables any algorithm to detect deeper and more meaningful community structures, ultimately leading to improved modularity scores and more accurate partitions in complex networks. This flexible input-modification strategy offers a powerful extension to any technique, enhancing its capability to detect communities in intricate and large-scale networks. This idea is easily illustrated with a small example, the well-known case of the chain, non-optimally divided into two clusters by the *Louvain* algorithm. Apart from a thorough experimental analysis done to assess the goodness of our proposal, we apply it to a real case concerning tourist movements in Spain. This case study highlights the quality and realism of the results provided by the proposed methodology.

The main valuable contribution of this paper is that, unlike other proposals in the literature that focus on individually improving specific algorithms [19–21], our methodology is defined in a general way and is suitable for application to any community detection algorithm whose input is a matrix related to a graph (square and with non-negative values). In this way, without altering the computational cost of the community search process, our methodology will provide more cohesive communities, often increasing the modularity value

Therefore, our work presents two main goals. On one hand, from a modeling perspective, we propose the characterization of the multistructural communication graph (MCG) and some of its variants, detailing the possibilities this tool offers in the context of SNA. On the other hand, focusing on the community detection problem, we propose a new methodology that integrates the use of the MCG into community detection. This approach addresses the issue of greediness that causes many well-known and widely used algorithms to get stuck in local solutions. Furthermore, we evaluate the performance of the proposed methodology through a thorough assessment process and demonstrate its effectiveness in a real-world problem.

The structure of this paper is organized as follows. Section 2 includes the theoretical background and essential concepts necessary to understand our proposed approach. We define a new representation model in Sect. 3, applied to a new community detection methodology in Sect. 4. In Sect. 5, we detail our experimental setup and present the results, demonstrating the effectiveness of our method. A case study is presented in Sect. 7. Some conclusions and further research steps are detailed in Sect. 8.

2 Preliminaries

In the study of complex networks, identifying meaningful substructures or communities is a fundamental task. CDP allows for the identification of densely connected groups of nodes that may represent important clusters, such as social groups, functional modules in biological systems, or tightly-knit groups in communication networks.

Graphs or networks are used to model such systems. Formally, a graph is a pair $G = (V, E)$, where $V = \{1, \ldots, n\}$ represents the set of nodes and $E = \{\{i, j\} : i, j \in E\}$ the set of undirected edges connecting them [6, 13]. A graph can be univocally characterized by its adjacency matrix, A, which represents direct connections between nodes ($a_{ij} = 1$ if $\exists \{i, j\} \in E$ and $a_{ij} = 0$ otherwise). If there is a function $w : E \rightarrow \mathbb{R}^+$ which assigns a weight w_{ij} to edge $\{i, j\} \in E$, G is known as weighted graph.

Solutions to split these networks into meaningful communities often leverage graph-theoretic concepts, such as modularity [14], which measures the strength of division by comparing the density of edges within communities to the expected density of edges in a random network. Due to its balance of computational efficiency and interpretability, modularity has been widely used as a function to optimize in the development of community detection algorithms. By maximizing modularity, these methods aim to find partitions that reveal the underlying structures of the network, identifying groups of nodes that are more interconnected with each other than with the rest of the network. This approach is fundamental in community detection, providing a framework for distinguishing significant subgroups within complex systems.

Definition 1 (*Modularity* [14]) Let $G = (V, E)$ denote a graph with adjacency matrix A, let P denote a partition of the set of nodes, V and Let $\mathbb{R}^+$ denote the resolution parameter. The modularity of P in G considering $\gamma \in \mathbb{R}^+$ is defined as

$$Q(A, P, \gamma) = \frac{1}{2m} \sum_{ij} \left[a_{ij} - \gamma \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{1}$$

where k_i and k_j are the degrees of $i, j \in V$; m is the sum of all of the edge weights in the network; c_i and c_j are the communities of the nodes i and j, respectively, in partition P, and δ is the Kronecker delta, which is 1 if $c_i = c_j$ and 0 otherwise.

Modularity optimization in CDP typically involves iteratively reassigning nodes to different communities to maximize the overall modularity. At each step, the algorithm considers moving a node to a neighboring community and evaluates the modularity.

Definition 2 (*Variation of modularity* $\Delta Q_i(j, \gamma)$ [18]) Let $G = (V, E)$ denote a graph with adjacency matrix A, let $\gamma \in \mathbb{R}^+$ denote the resolution parameter and let P denote a partition of V. Given a node i and one of its neighbours, j, let $P_i(j)$ denote a partition of V equal to P but moving i to the community assigned in P, C_i, to the community to which j belongs in P, C_j. Then, the variation of modularity obtained when moving i from C_i to C_j is calculated as

$$\Delta Q_i(j, \gamma) = Q(A, P_i(j), \gamma) - Q(A, P, \gamma). \tag{2}$$

3 An Advance Representation Model

Understanding and modeling real-world problems often involve navigating vast amounts of information, encompassing both structural and non-structural aspects of the system under study. SNA plays a critical role in unraveling the complexities of such systems by examining the intricate web of relationships, interactions, and dependencies among their components [6, 10]. While the structural properties of networks, typically captured through adjacency matrices, provide valuable insights, real-world systems frequently exhibit richer dynamics that transcend direct connections.

In such real-world networks, nodes often carry additional attributes-such as age, gender, or interests-that influence their interactions. Recent works have incorporated machine learning (ML) to integrate these attributes into community detection algorithms [24–26], improving the accuracy of the results. In networks where relationships are driven by motivations or capacities, authors have incorporated fuzzy measures [27] or cooperative games [28] among others to capture the soft, uncertain information that naturally exists in real-world scenarios, providing a more realistic representation of network dynamics [3, 29–34].

Our proposal in this paper is based on the assumption of certain communication sources inherent to the nodes, regardless of their nature (it could be adapted to the specific contexts of fuzzy logic or cooperative games previously mentioned). Incorporating nuanced aspects such as indirect relationships, multi-level interactions or higher-order paths is essential for a comprehensive understanding of these systems [23]. By considering communication patterns, latent connections, and additional attributes, we can construct models that better reflect the multifaceted nature of real-world interactions. Such enriched models enable a deeper analysis of the network's topology and behavior, offering a more complete representation of the underlying phenomena. Specifically, this idea can be useful for representing non-local information, such as direct communications between nodes, and for capturing potential 'future' communications between nodes, for instance, long-distance paths.

In this paper, we assume the existence of one or more communication structures within the graph. Unlike assumptions made in previous works, which rely on external information sources beyond the graph-thereby defining a problem distinct from the original graph-based one [35, 36]-, the approach detailed here focuses solely on modeling intrinsic graph characteristics. When appropriately modeled, these inherent features enhance the quality of the resulting partitions.

Since these communication sources are associated with the graph, they can be represented in matrix form. These matrices are subsequently aggregated with the adjacency matrix of the graph to encode the enhanced communication dynamics. This approach allows us to integrate any communication-related information into the network, thereby enriching the representation of its topology and interaction patterns. To do so, we first define a new Multiestructural Communication Graph (MCG).

Definition 3 (*Multiestructural Communication Graph (MCG)*) Let $G = (V, E)$ represent a network with n nodes and m edges. Let C denote a set of communication structures among the nodes of G. Specifically, we assume $C = \{C_1, \dots, C_r\}$, where each $C_i : V \times V \to \mathbb{R}^+$ is a communication source related to the nodes in V. The tuple $\widehat{G} = (G, (C_1, \dots, C_r))$ is called Multistructural Communication Graph (MCG).

Definition 3 establishes a formal framework for representing enhanced communication dynamics in networks, laying the groundwork for deeper analytical exploration. Normally, the most natural form of representation and analysis of graphs is the adjacency matrix. Therefore, to facilitate the integration of communication sources with the structural characterization of a graph, it seems natural to consider the communication sources in C to be represented as squared and non-negative matrices, each linked to a communication source and related to the graph. That is, being $\Pi_{n\times n}$ the set of $n \times n$ non-negative matrices each communication, we assume $C_i := C_{Ai} \in \Pi_{n\times n}$, where each row (and column, respectively) represents a node, with non-negative values. Then, if A denotes the adjacency of G, the tuple $\widehat{G} = (A, (C_{A1}, \dots, C_{Ar}))$ is said to be a *Multistructural Communication Graph Adjacency* (MCGA).

These communication structures can be highly diverse, ranging from the use of paths of various lengths, such as $A^2, A^3, \dots, A^k$, to the inclusion of partition matrices provided by other community detection algorithms [23], incidence matrices or common neighbors matrix. They may also encompass adjacency matrices derived from the same set of nodes in a different context or represent any type of connection or synergy within the network [1, 5, 12, 36, 37].

Example 1 Let $G = (V, E)$ denote the chain with 8 nodes represented in the Fig. 1. In this case, we assume the communication structures represent different-length paths. Thus, we define the MCGA $\widehat{G} = (A, (A^2, A^3))$ represented in the Fig. 2, whose matrices are defined in the Fig. 3 where $C_{A1} = A^2$ and $C_{A2} = A^3$.

Once the representation model MCG is defined, specifically in its matrix representation MCGA, an aggregation is proposed for further analysis or application by means of an aggregator $\Phi : \Pi_{n\times n} \times \cdots \times \Pi_{n\times n} \to \Pi_{n\times n}$. The

Fig. 1 Graph $G = (V, E)$

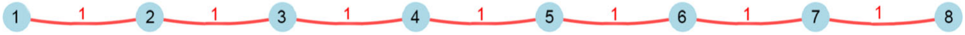


Fig. 2 MCGA $\widehat{G} = (A, (A^2, A^3))$. Graph representation

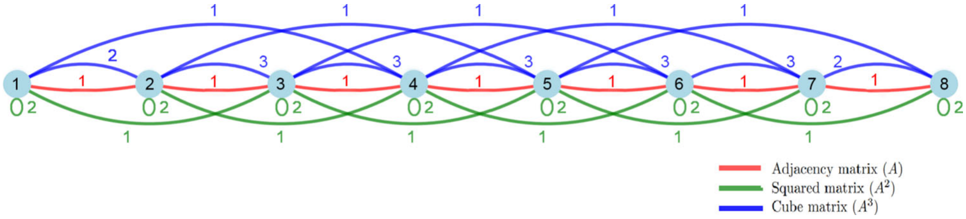


Fig. 3 MCGA $\widehat{G} = (A, (A^2, A^3))$

$$A = \begin{pmatrix} 0&1&0&0&0&0&0&0\\ 1&0&1&0&0&0&0&0\\ 0&1&0&1&0&0&0&0\\ 0&0&1&0&1&0&0&0\\ 0&0&0&1&0&1&0&0\\ 0&0&0&0&1&0&1&0\\ 0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&1&0 \end{pmatrix} \quad A^2 = \begin{pmatrix} 1&0&1&0&0&0&0&0\\ 0&\mathbf{2}&0&1&0&0&0&0\\ 1&0&\mathbf{2}&0&1&0&0&0\\ 0&1&0&\mathbf{2}&0&1&0&0\\ 0&0&1&0&\mathbf{2}&0&1&0\\ 0&0&0&1&0&\mathbf{2}&0&1\\ 0&0&0&0&1&0&\mathbf{2}&0\\ 0&0&0&0&0&1&0&1 \end{pmatrix} \quad A^3 = \begin{pmatrix} 0&\mathbf{2}&0&1&0&0&0&0\\ \mathbf{2}&0&\mathbf{3}&0&1&0&0&0\\ 0&\mathbf{3}&0&\mathbf{3}&0&1&0&0\\ 1&0&\mathbf{3}&0&\mathbf{3}&0&1&0\\ 0&1&0&\mathbf{3}&0&\mathbf{3}&0&1\\ 0&0&1&0&\mathbf{3}&0&\mathbf{3}&0\\ 0&0&0&1&0&\mathbf{3}&0&\mathbf{2}\\ 0&0&0&0&1&0&\mathbf{2}&0 \end{pmatrix}$$

choice of this aggregator Φ is, in itself, an interesting problem whose analysis could extend beyond the scope of this paper. Information fusion or aggregation refers to the application of mathematical or computational methods to combine and condense available data into meaningful representations that capture various aspects of the analyzed model [38–42]. This process is essential for effectively utilizing network information for further analysis. In the context of SNA, and particularly within MCGA, aggregation can be understood as a procedure that takes MCGA data as input and produces a single matrix summarizing its key aspects. The selection and interpretation of Φ depend on the specific elements of the aggregation process, which may emphasize different properties such as connectivity, node influence, relationships, or other relevant dimensions. Moreover, it is inherently linked to the intended use of the aggregated model and the specific objectives of the problem under analysis.

Definition 4 (*Aggregated MCG (AMCG)*) Let $\widehat{G} = (G, (C_1, \ldots, C_r))$ denote a MCG, and let Φ any aggregator of $\widehat{G}$. We define the aggregated MCG (AMCG) as

$$A^* = \Phi(G, C_1, \ldots, C_r), \tag{3}$$

being A^* an aggregated representation of all the communication structures in $\widehat{G}$, resulting in an adjacency matrix of some network.

Note that, depending on the specific characteristics of the network or the requirements of the problem, the operator could take various forms, ranging from a simple weighting scheme to the consideration of more complex OWA operators [40]. Some intuitively interpretable aggregation approaches include parametric aggregation, where different weights are assigned to each component based on their importance, or the use of maximum- or minimum-based aggregators to emphasize specific types of relationships. For instance, one might define a connection between two nodes based on the weakest observed relationship among the sources in the MCGA, using the minimum as the aggregator. Conversely, the strongest observed relationship can be considered by applying the maximum aggregator, highlighting the most prominent connection.

Similarly, we could assume Φ as a specific aggregation involving a matrix aggregation that assigns, for example, degrees of 'importance' or 'relevance' to each component of MCGA. Specifically, this type of aggregation, which will be analyzed in detail, is denoted as Aggregated-MCGA (AMCGA), resulting in a square matrix A^*.

Example 2 We recall the MCGA introduced in the Example 1. To illustrate the calculation of A^*, we select different aggregation methods, ranging from a weighted aggregation with equal importance assigned to all communication sources, including the adjacency matrix (Φ_1), to an aggregation that transforms the MCGA into an unweighted

Fig. 4 AMCGAs obtained from aggregations, Φ_1, Φ_2, Φ_3, and Φ_4, respectively

$$A_1^* = \frac{1}{3}\begin{pmatrix} 1 & \mathbf{3} & 1 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{3} & 2 & \mathbf{4} & 1 & 1 & 0 & 0 & 0 \\ 1 & \mathbf{4} & 2 & \mathbf{4} & 1 & 1 & 0 & 0 \\ 1 & 1 & \mathbf{4} & 2 & \mathbf{4} & 1 & 1 & 0 \\ 0 & 1 & 1 & \mathbf{4} & 2 & \mathbf{4} & 1 & 1 \\ 0 & 0 & 1 & 1 & \mathbf{4} & 2 & \mathbf{4} & 1 \\ 0 & 0 & 0 & 1 & 1 & \mathbf{4} & 2 & \mathbf{3} \\ 0 & 0 & 0 & 0 & 1 & 1 & \mathbf{3} & 1 \end{pmatrix}$$

(a) $A_1^* = \Phi_1(A, A^2, A^3)$, with $\alpha_1 = \alpha_2 = \frac{1}{3}$.

$$A_2^* = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

(b) $A_2^* = \Phi_2(A, A^2, A^3)$, with $\delta_A = \delta_{A^2} = \delta_{A^3} = 1$.

$$A_3^* = \begin{pmatrix} 0 & \mathbf{\frac{3}{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{\frac{3}{2}} & 0 & \mathbf{\frac{7}{4}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{\frac{7}{4}} & 0 & \mathbf{\frac{7}{4}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{\frac{7}{4}} & 0 & \mathbf{\frac{7}{4}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{\frac{7}{4}} & 0 & \mathbf{\frac{7}{4}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{\frac{7}{4}} & 0 & \mathbf{\frac{7}{4}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{\frac{7}{4}} & 0 & \mathbf{\frac{3}{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{\frac{3}{2}} & 0 \end{pmatrix}$$

(c) $A_3^* = \Phi_3(A, A^2, A^3)$, with $\alpha = \frac{1}{2}$ and $\beta = \frac{1}{4}$.

$$A_4^* = \frac{1}{3}\begin{pmatrix} 1 & \mathbf{2} & 1 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{2} & 1 & \mathbf{2} & 1 & 1 & 0 & 0 & 0 \\ 1 & \mathbf{2} & 1 & \mathbf{2} & 1 & 1 & 0 & 0 \\ 1 & 1 & \mathbf{2} & 1 & \mathbf{2} & 1 & 1 & 0 \\ 0 & 1 & 1 & \mathbf{2} & 1 & \mathbf{2} & 1 & 1 \\ 0 & 0 & 1 & 1 & \mathbf{2} & 1 & \mathbf{2} & 1 \\ 0 & 0 & 0 & 1 & 1 & \mathbf{2} & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & \mathbf{2} & 1 \end{pmatrix}$$

(d) $A_4^* = \Phi_4(A, A^2, A^3)$, with $\alpha_1 = \alpha_2 = \frac{1}{3}$.

network, establishing the existence or absence of a relationship in any of the information sources (Φ_2). Additionally, we consider a multiplicative aggregation (Φ_3) that modifies the weights of direct connections in A depending on the presence of higher-order paths, and a final function that establishes the weakest relationships among the considered sources (Φ_4). Then, the elements a_{ij}, a_{ij}^2 and a_{ij}^3 denote the values in the row i and the column j of matrices A, A^2 and A^3 respectively

- $A_1^* = \Phi_1(A, A^2, A^3) = \alpha_1 a_{ij} + \alpha_2 a_{ij}^2 + (1 - \alpha_1 - \alpha_2) a_{ij}^3, \forall i, j \in V$ with $\alpha_1, \alpha_2, (1 - \alpha_1 - \alpha_2) \in [0, 1]$. Φ_1 is a parametric aggregation to assign weights to information sources.
- $A_2^* = \Phi_2(A, A^2, A^3) = \max\{\min\{1, \delta_A a_{ij}\}, \min\{1, \delta_{A^2} a_{ij}^2\}, \min\{1, \delta_{A^3} a_{ij}^3\}\}$ $\forall i, j \in V$ with $\delta_A, \delta_{A^2}, \delta_{A^3} \in \{0, 1\}$. Φ_2 aims to establish the existence of at least one path in any of the sources, A, A^2 and A^3, if the corresponding $\delta_A, \delta_{A^2}, \delta_{A^3}$ is non null.
- $A_3^* = \Phi_3(A, A^2, A^3) = a_{ij} + \alpha a_{ij} a_{ij}^2 + \beta a_{ij} a_{ij}^3$, with $\alpha, \beta \in \mathbb{R}^+$. Φ_3 is somehow related with a Yager's weighted power mean [40].
- $A_4^* = \Phi_4(A, A^2, A^3) = \alpha_1 \min\{a_{ij}, \max\{1, a_{ij}\}\} + \alpha_2 \min\{a_{ij}^2, \max\{1, a_{ij}^2\}\} + (1 - \alpha_1 - \alpha_2) \min\{a_{ij}^3, \max \{1, a_{ij}^3\}\}$ with $\alpha_1, \alpha_2, (1 - \alpha_1 - \alpha_2) \in [0, 1]$. Φ_4 seeks to represent, somehow, the weakest relationship between nodes among sources considered.

In any aggregation, and particularly in the ones suggested here, only A and A^2, A and A^3, or any other combination with higher-order paths could be used. It could also be assumed that loops are eliminated by nullifying the main diagonal of the matrices.

4 Application of $\widehat{G}$ to Improve Community Detection

Previous studies have explored enhancing community detection algorithms by incorporating information beyond the structural data provided by the adjacency matrix [3, 35, 43, 44]. Specifically, the greediness of some of the most commonly used methods, such as the Louvain algorithm, while computationally efficient, has resulted in suboptimal partitions that only consider local information-specifically, the direct connections between nodes as represented by the adjacency matrix [23]. Intuitively, one might argue that feeding algorithms with broader information beyond the purely local could help prevent them from falling into their weaknesses, such as failing to analyze alternative solutions. To address this, one of the proposals we put forward is to require community detection algorithms to take into account future, non-local information, such as paths-sequences of connections that reveal deeper structural insights.

For example, second-order matrices identify whether two nodes share a common path involving a third node, uncovering nodes that connect groups structurally without direct links [23]. A common case is the second-order adjacency matrix A^2, where a_{ij}^2 represents paths of length 2, with higher powers A^k encoding paths of length k.

By leveraging communication sources inherent to the graph, our methodology offers a generalized approach that can be applied across various CDP algorithms. This flexibility ensures that improvements in partition quality are not limited to a specific algorithm but can enhance the performance of multiple techniques in diverse contexts.

Inspired by these approaches, we propose a framework to incorporate additional communication structures into a given graph G. Then we define our community detection methodology on the basis of $\widehat{G}$ and Φ. Finally, let ALG represent any existing community detection algorithm whose input is a matrix related to a graph. In this paper, a methodology for community detection in G is proposed, which, using the information in C, improves the modularity (i.e. considering the density of the groups and the separation between them) of the partition provided by any classical community detection algorithm, ALG, when evaluating the result in G with $\gamma = 1$ (the base case of modularity calculation). The step-by-step explanation is detailed below.

Step 1. Given the network $G = (V, E)$ with adjacency A, and having a set of communication structures related to G and summarized into $C = \{C_1, \dots, C_r\}$, define $\widehat{G} = (A, (C_1, \dots, C_r))$.
Step 2. Aggregate $\widehat{G}$ into matrix A^* by the aggregator Φ as $A^* = \Phi(A, C_1, \dots, C_r)$.
Step 3. Let ALG be a community detection algorithm that takes a matrix as input. Apply $ALG(A^{)}$ to find the communities in G, resulting in P^*.
Step 4. Evaluate P^* in G, i.e., calculate $Q(A, P^*, 1)$.

We find that, as will be demonstrated in the Sect. 5, that our methodology improves the partition P provided by classical $ALG(A)$ for any ALG, when it is evaluated in G, i.e., $Q(A, P, 1) \leq Q(A, P^*, 1)$.

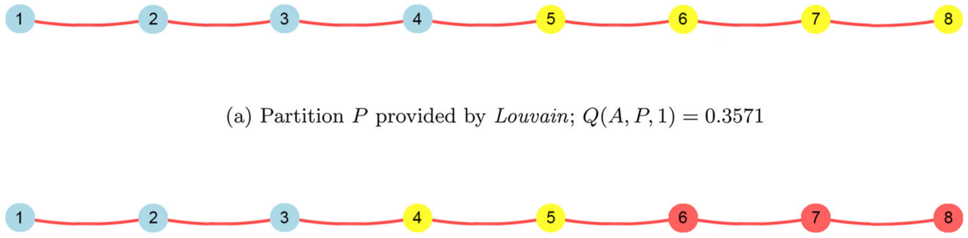
Algorithm 1 MCG-CDP

```
Input: $A \leftarrow G = (V, E)$; $C = \{C_1, \dots, C_r\}$; $\Phi$; $ALG$
Output: $P^*$
function MCG- CDP($(A, C)$, $\Phi$, $ALG$)
    $A^* = \Phi(A, C_1, \dots, C_r)$
    $ALG(A^*) = P^*$
    return $P^*$
end function
```

Example 3 Let us recall the MCG introduced in the Example 1, and let us consider the four AMCGAs characterized in the Example 2 by means of Φ_1, Φ_2, Φ_3 and Φ_4. We show below the difference between the partition provided by Louvain and that provided by $MCG - CDP$ using A_1^*, A_2^*, A_3^*, A_4^* and $ALG = Louvain$. Considering the aforementioned matrices as input, A_1^*, A_2^*, A_3^*, A_4^*, the application of *MCG-CDP* yields the same partition P^*, shown in Fig. 5b, whose modularity, when evaluated in A, improves upon the performance of the original solution P, shown in Fig. 5a.

This example provides insight into the potential benefits of incorporating additional communication sources, specifically higher-order paths and their various aggregations. We then proceed to evaluate this approach.

Fig. 5 Partitions for G obtained with the *Louvain* and the *MCG-CDP* algorithms



(a) Partition P provided by *Louvain*; $Q(A, P, 1) = 0.3571$

(b) Partition P^* provided by $MCG-CDP$ with A_1^*, A_2^*, A_3^* or A_4^*; $Q(A, P^*, 1) = 0.3775$

5 Computational Results

5.1 Benchmark, Comparison Methods and Evaluation Metrics

In this section we assess the performance of methodology proposed in the paper. The primary objective of this study is to improve the partitioning outcomes of existing algorithms applied to the original network. Our methodology aims to achieve this enhancement by incorporating communication sources derived from the graph, including their analyses and/or transformations. To evaluate the effectiveness of the modified algorithm, we compare the partitions it generates with those produced by the original algorithm. Specifically, we compare the modularity of the partitions in the original network (A_ι) as defined by the original algorithm with the modularity calculated for the partitions generated by our methodology, which uses different $A^*_{\iota\ell} = \Phi^\ell(A_\iota, C_{A_\iota 1}, \dots, C_{A_\iota r})$ as input matrix.

In this experimental analysis, we improve the structural information provided to CDP algorithms by combining adjacency matrices raised to different powers. The process begins with the adjacency matrix of a graph, which encodes direct connections between nodes. By raising this matrix to successive powers, higher-order relationships are introduced, where each power represents the number of steps in a walk between nodes. These matrices, corresponding to different walk lengths, are then weighted and aggregated into a composite matrix.

This approach incorporates both local and global connectivity patterns into a single representation. For each network, we consider the original adjacency matrix (A_ι (paths of distance 1), A_ι^2 (paths of distance 2), and A_ι^3 (paths of instance 3). So, in recalling the Step 2 of our methodology, $A_\iota^* = \Phi(A_\iota, A_\iota^2, A_\iota^3)$.

Then, we need to choose the aggregation Φ. Although many other options can be applied, including more sophisticated ones like OWA operators [40], for better understanding of the results, we consider different aggregations of the matrices in terms of importance by using different weighting coefficients $\alpha_1^\ell, \alpha_2^\ell \in [0, 1]$ with $\alpha_1^\ell + \alpha_2^\ell \leq 1$:

$$A^*_{\iota\ell} = \Phi^\ell(A, A_\iota^2, A_\iota^3) = \alpha_1^\ell A_\iota + \alpha_2^\ell A_\iota^2 + (1 - \alpha_1^\ell - \alpha_2^\ell) A_\iota^3. \tag{4}$$

From a mathematical standpoint, the comparison can be formulated as: $Q(A_\iota, P^\iota, 1) < \max_{\alpha_1^\ell, \alpha_2^\ell \in [0,1]} Q(A_\iota, P^{\iota\ell})$, where P^ι and $P^{\iota\ell}$ represent the partitions obtained by the corresponding algorithm, considering A_ι and $A^*_{\iota\ell}$ as input, respectively.

The weighting scheme is a crucial aspect of this methodology. It assigns relative importance to walks of different lengths, allowing for tailored representations that emphasize either local or global connectivity. By tuning these weights, the composite matrix can be adapted to the specific characteristics of the network under analysis.

$A^*_{\iota\ell}$ provides a richer representation of the network's structure compared to the original A_ι. To evaluate its effectiveness, $A^*_{\iota\ell}$ is compared to A_ι to evaluate whether the inclusion of higher-order information yields more meaningful partitions.

This process is applied to several well-known networks: G_1: *Zachary Karate Club*, G_2: *Dolphins*, G_3: *Les Miserables*, G_4: *PolBooks*, G_5: *American College Football*, G_6: *Jazz Musicians*, G_7: *Caenorhabditis elegans*, G_8: *Cora*, G_9: *CiteSeer*, and G_{10}: *Power Grid*, which serve as benchmark references due to their established results.

Additionally, to work with a large and varied benchmark that allows us to assess the quality of our methodology, we organize our working database based on two criteria. On one hand, we leverage a broad collection of networks from the Konect repository [45], encompassing 80 more distinct networks. To mitigate computational complexity associated with larger networks in this repository, we limit our analysis to those containing no more than 13.000 nodes and 168.000 edges. This subset is deemed sufficiently representative for the evaluation. On the other hand, we work on the random generation of networks of various types. Specifically, we consider networks of the small-world [46] with high clustering and short path lengths, resembling real-world networks, hierarchical networks [47] with nested, tree-like structures, common in organizational and biological systems, and scale-free networks [48] with a power-law degree distribution, where a few nodes have high degrees, providing 20 random networks for each type. The random values for the parameters of each network type are as follows:

- **Small world networks.** The number of nodes n takes five different values, randomly selected with equal probability within the range of values from the networks considered in [45]. The parameter k, which represents the number of initial connections each node has with its nearest neighbors in a regular graph, takes two values: $k = \max(2, \min(5\%n, 50))$ and $k = \max(4, \min(10\%n, 100))$. The parameter p, which denotes the probability that each edge in the initial regular graph is randomly rewired to another node, is set to $p = 0.1$ and $p = 0.3$.
- **Hierarchical networks.** This type of networks has two different parameters: the number of *levels* (values 4, 5, 6, 7 and 8) and the maximum number of *children* of each node (values 2, 3, 4, and 5).
- **Scale-free networks.** The generation of this type of network primarily depends on two parameters: the number of nodes, n, which takes five equally distributed values within the range of networks from [45], and the number of edges each new node adds to the graph upon incorporation, m, which takes four values defined as a function of the randomly chosen number of nodes: $m = \min(1\%n, 10)$, $\min(5\%n, 50)$, $\min(10\%n, 100)$, and $\min(20\%n, 200)$.

Thus, our framework includes the 10 well-known networks, an additional 80 from Konect, and 60 synthetic networks.

For each of these 150 networks, $G_\iota = (V_\iota, E_\iota)$ with adjacency A_ι, we consider $\widehat{G_\iota} = (A_\iota, (A_\iota^2, A_\iota^3))$. We assume the parametric aggregation method $\Phi^\ell(A_\iota, A_\iota^2, A_\iota^3) = A^*_{\iota\ell} = \alpha_1^\ell A_\iota + \alpha_2^\ell A_\iota^2 + (1 - \alpha_1^\ell - \alpha_2^\ell)A_\iota^3$, with 5151 different combinations of $\alpha_1^\ell, \alpha_2^\ell \in [0, 1]$, taken by 0.01 steps with $\alpha_1^\ell + \alpha_2^\ell$. Then, for each network in our benchmark repository we obtain 25.755 partitions (5 algorithms and 51.51 different inputs $A^*_{\iota\ell}$), including the base case, $\alpha_1^\ell = 1$ and $\alpha_2^\ell = 0$.

Specifically, some classic CDP algorithms have been implemented using the R package *igraph*: Louvain [18] ($Louvain$), Leiden [22] ($Leiden$), *Walktrap* [15] ($Walktrap$), *Infomap* [16] ($Infomap$) and *Fast Greedy* [17] ($Fast\_Greedy$).

Each algorithm is run on both the original network and the modified structure derived from the MCGA, producing partitions P^ι (original) and $P^{\iota\ell}$ (*MCG-CDP*). The modularity of both partitions is computed using the original adjacency matrix. This approach ensures a fair comparison of the partitions produced by both techniques, enabling the identification of the method that yields better results. To ensure reproducibility, random factors were controlled by fixing elements such as the initial node order and using 12345 as the random seed.

5.2 Results and Discussion

Then we assess the performance of our proposed algorithm across 150 networks, comparing it to several widely used CDP algorithms. The primary metric for comparison is the percentage improvement of our algorithm over each baseline algorithm, evaluated across various combinations of network features and configurations.

For each network, we apply our CDP approach, referred to as *MCG-CDP*, alongside 5 classic CDP algorithms: *Louvain, Leiden, Walktrap, Infomap* and *Fast Greedy*. In Table 1, for each network $G_1, \ldots, G_{10}$ and each algorithm, we present $Q(A_\iota, P^\iota, 1)$ (from the baseline algorithm, denoted as Q) and $\max_{\alpha_1^\ell, \alpha_2^\ell \in [0,1]} Q(A_\iota, P^{\iota\ell}), 1$ (from *MCG-*

Table 1 Modularity comparison of the 10 well-known networks, considering the original method (Q) and the *MCG-CDP* algorithm (Q^*)

ALG	MOD	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
Louvain	Q	0.4057	0.5188	0.5583	0.5270	0.6106	0.4471	0.4379	0.8261	0.8916	0.9396
	Q^*	0.4057	**0.5269**	**0.5720**	**0.5343**	**0.6107**	0.4471	**0.4461**	**0.8264**	**0.9028**	**0.9411**
Leiden	Q	0.4415	0.5241	0.5600	0.5269	0.6106	0.4500	0.4447	0.8343	0.8993	0.9460
	Q^*	**0.4470**	0.5241	**0.5706**	**0.5416**	**0.6207**	**0.4575**	**0.4485**	**0.8425**	0.8993	0.9460
Walktrap	Q	0.3719	0.4888	0.5214	0.5070	0.6128	0.4447	0.3897	0.7868	0.8326	0.8578
	Q^*	**0.3726**	**0.5032**	**0.5533**	**0.5260**	**0.6174**	**0.4524**	**0.3998**	**0.7913**	0.8326	**0.8586**
Infomap	Q	0.4185	0.5247	0.5513	0.5228	0.6067	0.2847	0.4232	0.7630	0.8202	0.8706
	Q^*	**0.4258**	**0.5348**	0.5513	**0.5432**	**0.6153**	**0.2908**	**0.4294**	**0.7704**	**0.8220**	**0.8732**
Fast Greedy	Q	0.4345	0.5093	0.5006	0.5020	0.6080	0.4361	0.4177	0.8272	0.8946	0.9381
	Q^*	0.4345	0.5093	0.5006	0.5020	**0.6085**	**0.4408**	**0.4229**	**0.8346**	**0.8950**	**0.9436**

The bold values indicates the results improving compared to the base cases

CDP, denoted as Q^*). The scenarios where the partitions obtained with the*MCG-CDP* exhibit higher modularity than those provided by the baseline algorithms are highlighted in bold in Table 1. It is important to note that our results are at least as good as those of the original algorithm, as the partition provided by an original algorithm is part of the set of partitions over which this maximum is calculated, for example, when $\alpha_1^\ell = 1$.

The descriptive analysis provides a comprehensive view of the improvements in modularity, reinforcing the idea that incorporating preprocessed data leads to better outcomes. However, to substantiate this claim with empirical statistical evidence, we need to perform inferential testing. To achieve this, we apply this procedure to the 150 networks from the repository [45], with the complete results available on GitHub.[1] This repository, which contains the supplementary materials, includes both code and datasets. To facilitate a better understanding of the process, we have included a primary file that provides further explanations on how to reproduce the experimental results, particularly the parameter settings and experimental procedures. Once the experiments have been conducted, we apply binomial test-based confidence intervals to evaluate whether using preprocessed data results in superior outcomes compared to traditional input.

To do so, confidence intervals based on the binomial test are applied to test the hypothesis of whether the use of preprocessed input yields better results than the use of classical input. Due to computational constraints, we apply the CDP algorithms to all 10 well-known networks, the additional 80 networks obtained from [45] and the 60 randomly generated networks. Specifically, for each of the 750 experiments conducted, we define a Bernoulli distribution to model the percentage improvement of using preprocessed input compared to the use of classical adjacency [49]. In this context, the parameter p of the Bernoulli distribution represents the likelihood that $Q(A_\iota, P^\iota, 1) < max_{\forall \alpha_1^\ell, \alpha_2^\ell \in [0,1]} Q(A_\iota, P^{\iota \ell}, 1)$. Afterwards, we sum these Bernoulli variables based on the algorithm used, and all of them, giving rise to 5 binomial distributions (one for each CDP algorithm) with parameters n_k, representing the number of networks, and p. In Table 2, we present the confidence intervals (CI) obtained with confidence levels of 95%, 99% and 99.9% for the parameter p.

The results demonstrate a significant probability of modularity improvement for most community detection methods (CDP). *Louvain* (0.8000) and *Walktrap* (0.7733) emerge as the most positively impacted methods, consistently achieving high probabilities of improvement. In contrast, *Fast Greedy* exhibits a lower probability (0.6067), indicating less pronounced benefits from the methodology. Confidence intervals for the top-performing methods are robust, with all intervals, underscoring the reliability of these improvements. When considering all methods collectively, the aggregated improvement probability is 0.7666, demonstrating the effectiveness of the approach while highlighting variability in its impact across different algorithms. These findings suggest that the methodol-

[1] https://github.com/rodrhern-ucm/Flins-Inske.

Table 2 CI for the probability of modularity improvement in each CDP algorithm

CDP Method	n_k	p	95% CI	99% CI	99.9% CI
Louvain	150	0.8000	[0.7360, 0.8640]	[0.7159, 0.8841]	[0.6925, 0.9075]
Leiden	150	0.7733	[0.7063, 0.8403]	[0.6853, 0.8614]	[0.6608, 0.8858]
Walktrap	150	0.7733	[0.7063, 0.8403]	[0.6853, 0.8614]	[0.6608, 0.8858]
Infomap	150	0.7600	[0.6917, 0.8283]	[0.6702, 0.8498]	[0.6452, 0.8748]
Fast Greedy	150	0.6067	[0.5285, 0.6848]	[0.5039, 0.7094]	[0.4754, 0.7379]
All methods	750	0.7666	[0.7041,0.8378]	[0.6754, 0.8719]	[0.6537, 0.8789]

ogy is particularly beneficial for certain algorithms like *Louvain* and *Walktrap*. This variability underscores the importance of tailoring the approach to specific algorithms for optimal performance.

Overall, introducing transformations to input data led to variations in partitions and had a clear positive effect on modularity. These findings provide statistical evidence that applying the method described in Sect. 4 is beneficial, offering significant improvements in results while maintaining manageable computational costs.

5.3 Parameters Analysis

The inferential analyses presented in Sect. 5.2 confirm that using the matrix $A^{*}_{\iota\ell}$, as considered in the *MCG-CDP* algorithm, leads to improvements; however, they do not provide specific insights into the values of α_1^ℓ and α_2^ℓ. In this section we assess the impact of these aggregation parameters by calibrating the improvement ratio across different values of α_1^ℓ and α_2^ℓ, where both parameters range from 0 to 1 in increments of 0.01, while ensuring that $\alpha_1^\ell + \alpha_2^\ell \leq 1$. The analysis reveals specific patterns in the combinations of α_1 and α_2 that are more likely to improve modularity. These patterns vary depending on the baseline algorithm, but several general observations can be made regarding the ranges that contribute the most.

These findings highlight the importance of tuning the parameters α_1 and α_2 to optimize the modularity improvement process, especially in experimental designs involving community detection algorithms.

Regarding the difficulty of finding combinations of α_1 and α_2 that improve the base algorithms ($\alpha_1 = 1$ and $\alpha_2 = 0$), it should be noted that it is not challenging to identify such combinations. Table 3 presents the percentage of the conducted experiments different to the base combination, p^*, ($n_k^* = 5.150 * 150 = 772.500$) in which the combination of α_1 and α_2 values outperformed the base case. We also include the confidence interval values under the binomial distribution hypothesis. As shown, in all cases, more than the 10% of the combinations lead to improvements, indicating that extensive searches for optimal α_1 and α_2 values are not necessary to achieve enhancements. To provide a more detailed sensitivity analysis of these parameters and illustrate how small changes in them may affect the performance of a CDP algorithm, Fig. 6 represents the percentage of cases in which each specific combination of α_1 and α_2 improves upon the base case, where $\alpha_1 = 1$ and $\alpha_2 = 0$ (i.e., the classic algorithm's performance). To achieve this, for each of the 5.150 parameter combinations, a heatmap represents the percentage of the 150 networks in the benchmark for which each combination outperforms the base case. We provide an individual analysis for each of the five considered algorithms, as well as a combined analysis that aggregates their performance when integrated with our methodology. A detailed breakdown of all combinations across all examples is available on GitHub.

In addition to specifying how many cases our methodology improves upon classical results, we find it interesting to further analyze the percentage of modularity improvement (compared to the baseline case) achieved by different numbers of parameter combinations. For each baseline model used, as well as considering their aggregation, Fig. 7 illustrates how, depending on the desired gain, it is possible to assert that virtually any parameterization can provide some improvement.

In order to gain a deeper understanding of how the parameters α_1^ℓ and α_2^ℓ affect aggregation, and to determine which values to consider, we extend our analysis beyond the descriptive statistics previously presented by adopting a supervised ML approach. For this purpose, we use several techniques such as logistic regression (logit), SVM,

Table 3 CI for the probability of modularity improvement in each CDP method

CDP Method	n_k	p	95% CI	99% CI	99.9% CI
Louvain	772500	0.1460	[0.1452, 0.1468]	[0.1449, 0.1470]	[0.1447, 0.1473]
Leiden	772500	0.1454	[0.1446, 0.1462]	[0.1443, 0.1464]	[0.1440, 0.1467]
Walktrap	772500	0.1420	[0.1413, 0.1428]	[0.1410, 0.1431]	[0.1407, 0.1433]
Infomap	772500	0.2816	[0.2805, 0.2826]	[0.2802, 0.2829]	[0.2799, 0.2832]
Fast Greedy	772500	0.1091	[0.1085, 0.1098]	[0.1082, 0.1101]	[0.1080, 0.1103]
All methods	3862500	0.1648	[0.1644, 0.1652]	[0.1643, 0.1653]	[0.1642, 0.1654]

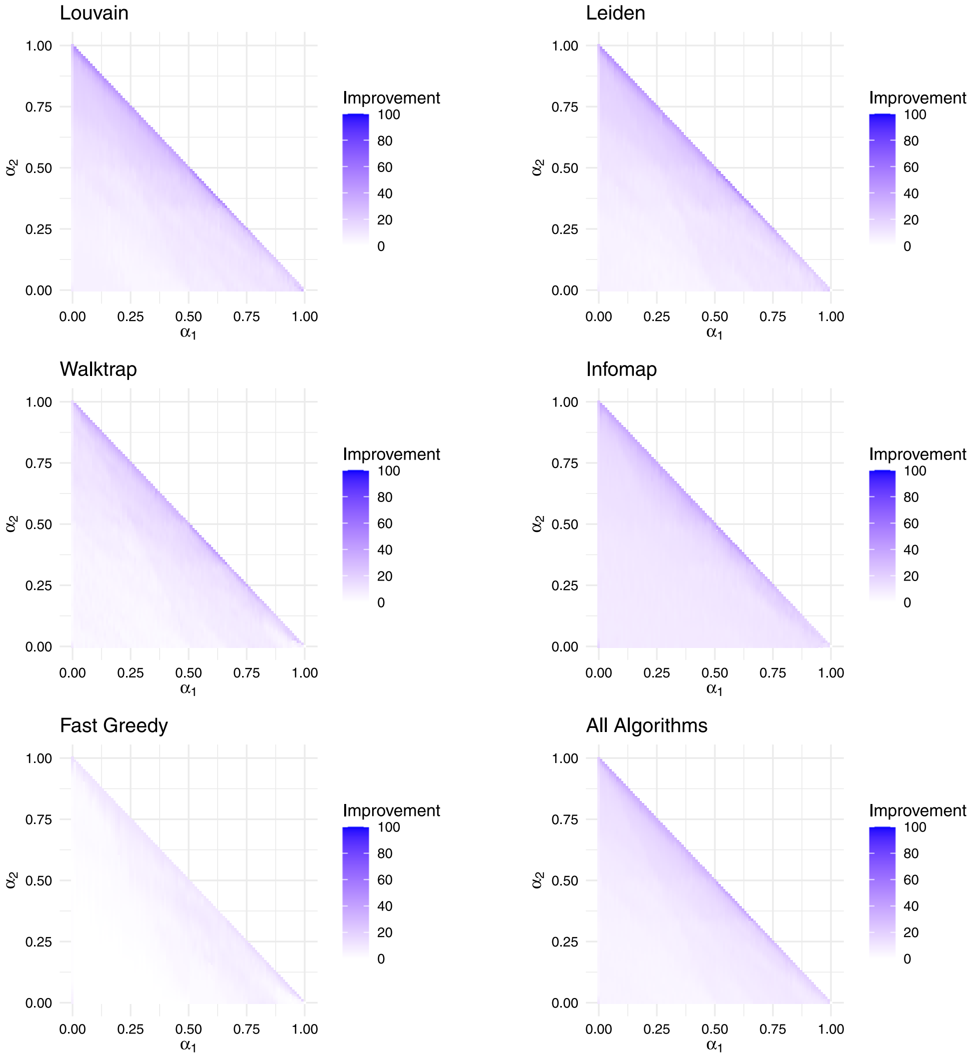


Fig. 6 Sensitivity analysis of α_1 and α_2: representation of percentage of cases for which each combination of α_1 and α_2 improves the base case, $\alpha_1 = 1$ and $\alpha_2 = 0$

random forests, decision trees and XGBoost [50], trained with 80% of data. Our objective is to explore the relationship between modularity improvement, the CDP baseline algorithm, and the values of α_1^ℓ and α_2^ℓ. This is framed as a binary classification problem, where the independent variables are α_1, α_2, and the CDP algorithms. Each combination of α_1^ℓ, α_2^ℓ, and $algorithm_h$ corresponds to an instance, with the dependent variable being 1 if this combination leads to an improvement in modularity over the classical approach.

Thus, binary classification methods are applied to predict the effect of the aggregation parameters α_1^ℓ and α_2^ℓ. ML algorithms aim to estimate the probability that our method outperforms classical techniques (represented by 1) or not (represented by 0). It is crucial to note that we are predicting a probability (a continuous value between 0

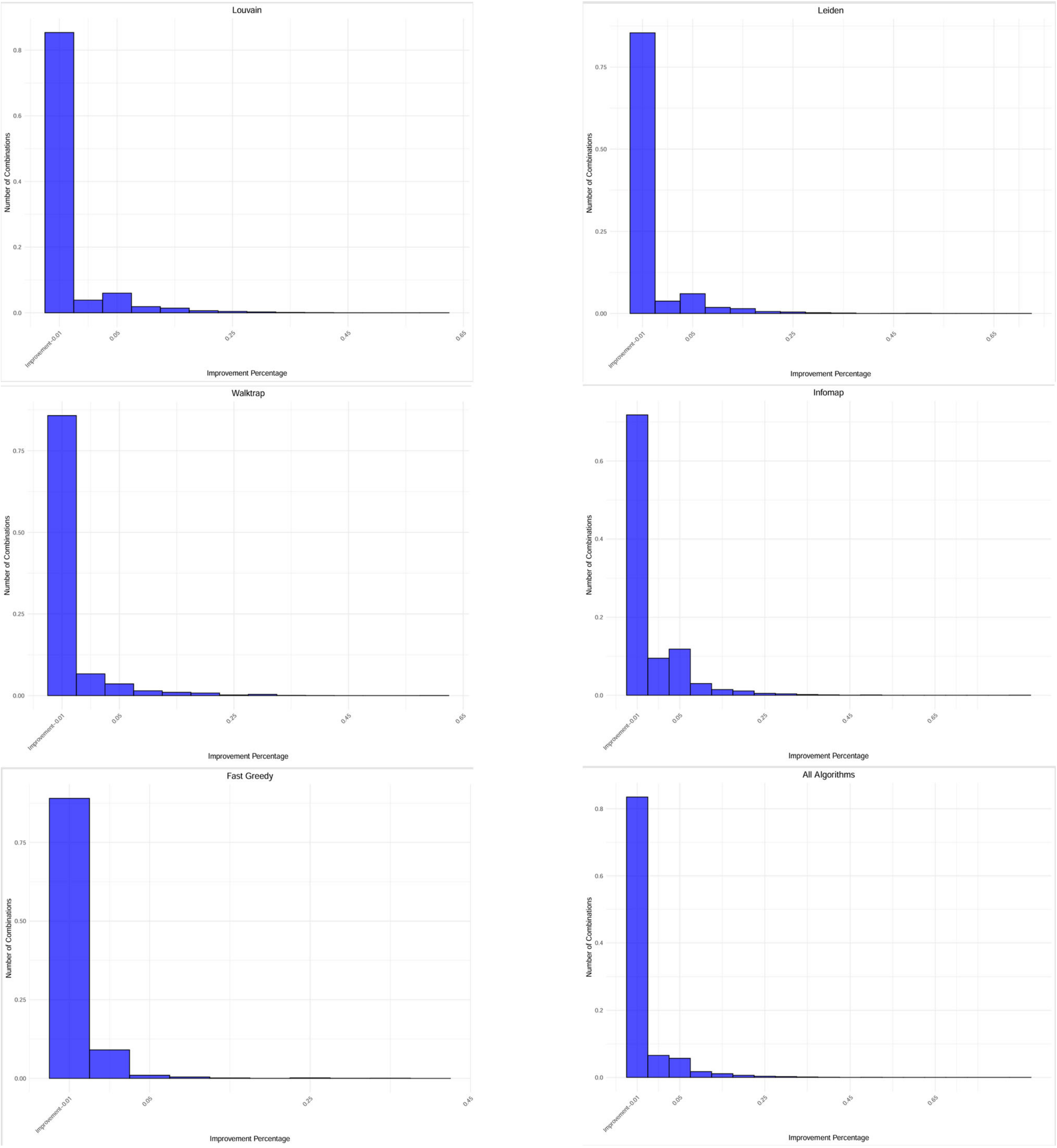


Fig. 7 Histogram: number of parameter combinations that provide a certain percentage of modularity gain compared to the baseline case, $\alpha_1 = 1, \alpha_2 =$

Table 4 Exploring parameters relevance: a comparative analysis through ML

Parameters	MSE: 0.1268 Logit	MSE: 0.1494 SVM	MSE: 0.1248 Random forest	MSE: 0.1240 XGBoost	MSE: 0.1299 Decision tree
Leiden	0.94	1.22	0.80	0.40	2.46
Louvain	2.91	2.17	2.63	2.84	0.00
Walktrap	0.00	2.35	0.00	0.00	3.45
Infomap	19.80	15.86	20.00	16.66	41.49
Fast Greedy	9.90	7.93	10.98	8.33	20.74
α_1	32.60	32.65	32.88	35.35	11.67
α_2	33.85	37.82	32.71	36.41	20.19

and 1), not a discrete classification, 0 or 1. This distinction requires using a continuous metric to evaluate error. We have chosen the Mean Squared Error (MSE), which penalizes larger errors more heavily. Although there are other metrics to assess model performance, we selected MSE because it penalizes significant mispredictions by squaring the differences between predicted and actual values. This characteristic aligns with our objective of minimizing large errors to achieve more accurate model performance, and offers a clear, interpretable measure that fits our research goals.

In Table 4 we present the results of the best parameterization for each model, achieved using the *R* library *caret*. Specifically, we report the goodness of fit measured by the MSE and the permutation importance (in mean) of each independent feature — *ALG*, α_1, and α_2 — for each ML model. Permutation importance is a model-agnostic method that evaluates the importance of each feature by observing the change in model performance when the values of a particular feature are randomly shuffled. To obtain it, we use five dummy variables representing different algorithms and two numerical features corresponding to α_1 and α_2.

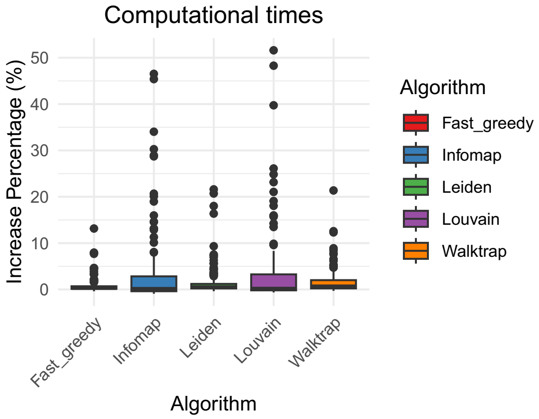
The evaluation outlined in this Section confirms the value of the proposed methodology, reinforcing its effectiveness. The findings emphasize the critical role of selecting the right parameter, which proves to be more important than the choice of the community detection technique itself.

5.4 About the Computational Complexity

Our proposed methodology functions as a preprocessing step to enhance any CDP algorithm (denoted as *ALG*). This process is divided into two phases:

- **Phase 1.** Construction of the MCGA. Specifically, in the proposed computational results, this involves calculating the matrices A^2 and A^3, representing the number of 2-step and 3-step walks between nodes, respectively. The computational complexity for calculation A^2 is $O(n < k >^2)$, and for A^3, it is $O(n < k >^3)$, where n is the number of nodes and $< k >$ is the average degree. Since $< k >$ is generally independent of n (it is not true that larger networks necessarily have a higher average degree), this phase is computationally feasible for large networks. Moreover, this calculation is performed once, even when testing different weight parameterizations.
- **Phase 2.** Application of the community detection algorithm *ALG* to the aggregated matrix AMCGA. The complexity of this phase depends on the specific algorithm *ALG* used. The aggregated network is similar to the previous one in terms of the number of nodes (so algorithms whose complexity depends on the number of nodes are not affected), but the number of edges differs (so algorithms whose complexity depends on the number of edges are affected). For instance, algorithms like Louvain and Leiden have a complexity of $O(m \log n)$, where m is the number of edges. In the aggregated matrix, the number of edges is bounded by $n(< k > + < k >^2 + < k >^3)$ if A^3 is weighted positively, or by $n(< k > + < k >^2)$ otherwise. The real average value $< k >$ is much lower than these bounds since most neighborhood relationships are repetitive and therefore do not reach new nodes. In any case, given that the value of $< k >$ does not depend on n, the

Fig. 8 Average computational cost per algorithm for the benchmark



increase in the number of edges will be a constant factor independent of the network size. Consequently, this does not change the order of the computational complexity of the solution, making it efficient and feasible for large networks.

In the Fig. 8, we represent the average execution time of the 5.150 combinations of α_1 and α_2 (excluding the base case) for each network in the benchmark across each base algorithm, showing the percentage variation relative to the execution time of the original algorithm. We excluded one of the networks (*arenas-jazz*) because the percentual increment represented an outlier for two of the algorithms, *Louvain* and *Infomap* (in absolute terms, computational times are low in both algorithms, 0.0031 and 0.0613 respectively). Results are available in GitHub.

So, this methodology is computationally feasible for large networks, as the preprocessing step introduces a manageable increase in complexity, and the subsequent application of community detection algorithms operates within their established computational bounds.

6 Advances in Deep Learning for CDP

In this section, we explore the connection between our proposed approach and recent advancements in deep learning models for community detection. We highlight how the enhanced network representation provided by an MCG can improve the performance of deep learning-based algorithms in identifying communities.

Machine learning and deep learning models have recently garnered significant attention in addressing community detection challenges within complex networks [26, 51–53] These approaches typically involve transforming graph data into node embeddings or graph representations, effectively capturing the structural and community-related properties of the network. This transformation facilitates the application of advanced analytical tasks, such as link prediction or node classification, especially when there is supplementary information available. These methodologies often utilize inputs derived from both the network's topology and its attributes. The topological component, lastly represented by nodes and edges, can be characterized y matrices like the adjacency matrix A, signed adjacency matrix $A^{(+,-)}$, or the modularity matrix, among others. Similar to traditional spectral methods [54], after this embedding transformation, any classical clustering techniques as the k-means can be applied to derive a final partitioning of the graph's nodes. Consequently, these contemporary methods can also be regarded as algorithms addressing classical community detection problems.

In [26], an in-depth review of various methods, scenarios, and options is presented. Specifically, it discusses the challenges that traditional embedding methods, such as spectral clustering and statistical inference, often face when handling the complexity and high dimensionality inherent in real-world networks.

To face these issues, deep learning-based techniques enable more flexible and accurate solutions by learning non-linear relationships, embedding networks in lower-dimensional spaces, and improving the detection of community structures. Several approaches have been proposed as those based on convolutional networks (CNN and GCN) [51], graph attention networks (GAT) [52], generative adversarial networks (GAN) [53], autoencoders

Table 5 Comparison between CSEA + *MCG-CDP* and CSEA

Network	Modularity CSEA + *MCG-CDP*	Modularity CSEA
G1	0.4020	0.4020
G2	0.5254	0.5254
G3	0.5672	0.5672
G4	0.5193	0.5193
G5	**0.6059**	0.6048
G6	**0.4492**	0.4427
G7	**0.4453**	0.4346
G8	**0.8186**	0.8181
G9	**0.8801**	0.8718
G10	**0.9381**	0.9341

The bold values indicates the results improving compared to the base cases

(AE) [55], deep nonnegative matrix factorization (DNMF) [56] or deep sparse filtering (DSF) [57]. This network representation proves particularly valuable for tackling more complex tasks beyond community detection, with practical applications in social, biological, and citation networks, highlighting the significant role of deep learning in these fields.

Specifically, deep learning methods have been demonstrated to effectively address challenges such as managing sparse or noisy data, identifying intricate structural patterns, and enhancing clustering performance.

The methodology proposed in this article focuses on an advanced representation of a network, going beyond solely considering local information about direct connections between nodes and not incorporating, for example, node attributes. In this section, we aim to integrate this perspective with deep learning models for community detection. These modern community detection techniques often rely on initial partitions created by conventional methods, such as Louvain [18], to initiate their processes.

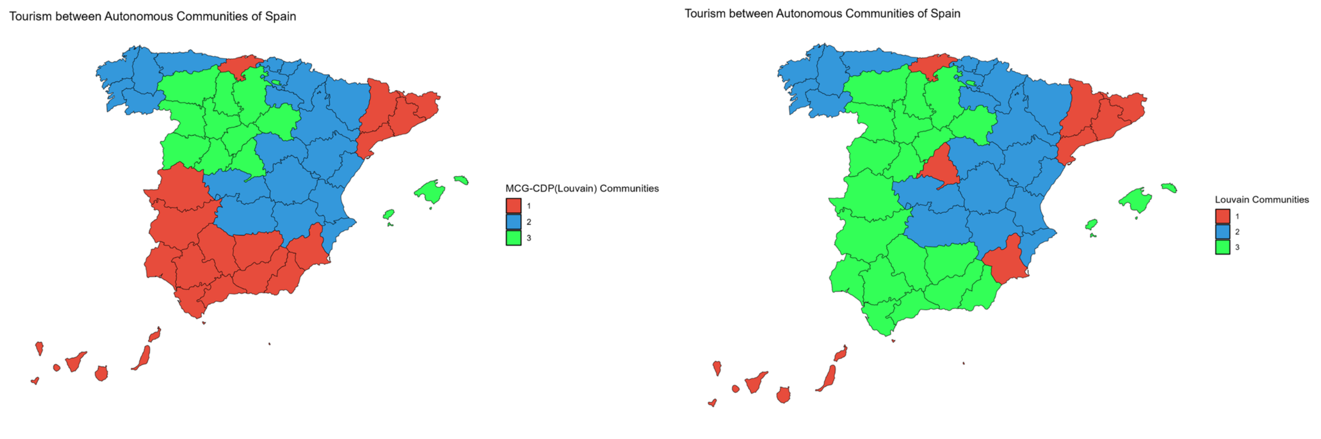
Our approach aims to use the results of the *MCG-CDP* as the initial input for these advanced deep learning frameworks. By doing so, we provide these algorithms with a stronger starting point for their training, as our method has shown improvements over traditional partitioning techniques (as demonstrated in Sect. 5). This integration allows us to connect our methodology with cutting-edge models, offering a foundation that enhances accuracy. Specifically, we have conducted a comparative analysis using a recent and competitive supervised deep learning algorithm presented in [25]: the ‘CSEA algorithm’. This new generation of supervised methods, which improves upon classical spectral clustering techniques, relies significantly on the quality of the initial network partitions. Our approach enhances these partitions, leading to more accurate and context-specific graph representations.

By improving the quality of the initial partitions, our *MCG-CDP* methodology greatly boosts the performance of supervised autoencoders. In the Table 5, we present the experiment we conducted over networks $G1, ..., G10$ introduced in Sect. 5.1.

As outlined in [25], we implemented the network core structure extraction algorithm using variational autoencoders, further adapted by incorporating the initial partition generated by the *MCG-CDP* algorithm (using Louvain as base algorithm) instead of the traditional Louvain partition. Subsequently, we applied the classical k-means algorithm to derive the final network partition. Table 5 displays the k-partition that results in the highest modularity for each network. As shown, integrating advanced information into the problem by means of the MCG characterization improves the majority of the problems addressed.

As demonstrated by the results in Sect. 5, our method consistently produces partitions with greater modularity compared to traditional algorithms. This highlights that our approach not only improves the initial graph representation but also significantly boosts the overall performance of supervised deep learning models.

It can be seen that the advanced network representation through the MCG effectively captures the information from the graph and its communities, outperforming the original algorithm, which learns from partitions with lower


Tourism between Autonomous Communities of Spain
MCG-CDP(Louvain) Communities
1
2
3
Tourism between Autonomous Communities of Spain
Louvain Communities
1
2
3

(a) $MCP - CDP$; $Q(A, P^*, 1) = 0.77235$ (b) $Louvain(A)$; $Q(A, P, 1) = 0.76558$

Fig. 9 Results of CDP algorithms to analyze tourism in Spain

modularity. Consequently, when applied to more complex tasks, such as node classification using node attribute data or link prediction, this approach proves to be more effective.

7 Case Study

In this section we present a case study to illustrate the utility of our *MCG-CDP* algorithm in real problems. We work with a database provided by the National Institute of Statistic in Spain[2] that represents the number of overnight stays between emitting/receiving Autonomous Communities. Spain has 17 Autonomous Communities and 2 Autonomous Cities, which define the nodes of the analyzed network. Undirected edges are established between two communities if there are overnight stays by tourists originating from one and traveling to the other during the year 2024. The weight of the edges depends on the number of tourists (visit GitHub for further information).

Due to its general acceptance, speed, and quality of results, we focus on the *Louvain* algorithm, whose results are compared with the application of our methodology. Specifically, being A_ι the adjacency of each Spain-constructed network, we apply *MCG-CDP*$((A_\iota, (A_\iota^2, A_\iota^3)), \Phi, Louvain)$, being $\Phi(A_\iota, A_\iota^2, A_\iota^3) = \alpha_1 a_{\iota ij} + \alpha_2 a_{\iota ij}^2 + (1 - \alpha_1 - \alpha_2) a_{\iota ij}^3$, for 5151 different combinations of α_1 and α_2, as explained in Sect. 5.1.

We propose two types of analysis. On one hand, we consider all the overnight stays between regions throughout the year 2024, constructing a single network with adjacency A_{anual} that captures all the movements. For this problem, we show that our method provides better results than other classical algorithms while ensuring that these improvements do not increase the complexity of interpretation or practical application in real-world cases. Specifically, $MCG - CDP$ reaches the optimal partition P^* with values $\alpha_1 = 0.64$ and $\alpha_2 = 0.23$, with a modularity $Q(A_{\text{anual}}, P^*, 1) = 0.77235$ (Fig. 9a), which is higher than the modularity of the partition provided by *Louvain*, $Q(A_{\text{anual}}, P, 1) = 0.76558$ (Fig. 9b).

In the annual solution provided in the Fig. 9, it can be observed that the regions of Castilla y León, Extremadura, and Andalucía are grouped into the same community, which may seem counterintuitive when considering interregional transportation in Spain. Specifically, Castilla y León does not have the best connectivity with the southern capital, in addition to being relatively distant and geographically extensive, leading to a predominance of internal movements.

[2] https://www.ine.es/dynt3/inebase/es/index.htm?padre=8578&capsel=8580.

Table 6 Comparison of *Louvain* and *MCG-CDP* per month

Month	$MCG - CDP + Louvain$	$Louvain$	% of improvement
January	0.6200	0.6200	–
February	**0.5734**	0.5278	8.64%
March	**0.5892**	0.4776	23.37%
April	**0.5827**	0.4801	21.37%
May	0.4612	0.4612	–
June	**0.5901**	0.4890	20.76%
July	**0.5796**	0.4705	23.19%
August	**0.6013**	0.4987	20.57%
September	**0.5678**	0.4523	25.54%
October	0.4759	0.4759	–
November	0.4698	0.4698	–
December	**0.5721**	0.4698	21.78%

Moreover, the Louvain partition defines Madrid-Spain's central region, home to the capital, the country's main transportation hub, the largest airport, and key railway lines-as an isolated community.

On the other hand, we incorporate a dynamic analysis of network evolution over time, specifically through a monthly breakdown of the dataset. This extension provides a deeper understanding of community structures and their stability, allowing us to observe how groups form, dissolve, or evolve over time, rather than treating them as static entities. By capturing temporal variations, this approach enhances the interpretability of community detection results and offers a more nuanced perspective on the underlying processes within the network.

While community detection is typically performed on static snapshots, our methodology naturally extends to sequential analyses, where community evolution can be studied through a series of aggregated representations. To achieve this, we consider 12 monthly and independent networks-$A_{January}$, $A_{February}$, and so on-and apply our *MCG-CDP* methodology for 1.151 different combinations of α_1 and α_2, as previously explained. The comparison results are presented in Table 6, where, for month ι, we apply the *MCG-CDP* and Louvain methods, and evaluate the obtained partition by calculating the modularity in the matrix A_ι. All the partitions can be found in GitHub.

It is particularly evident in our case study, where we observe that the modularity of detected communities is significantly higher during the seasonal months (Easter, Summer and Christmas) compared to other periods of the year. This improvement aligns with the increase in travel activity and overnight stays, which reinforces natural groupings within the network. The seasonal effect highlights that communities are not merely artifacts of the detection algorithm but rather emerge from meaningful behavioral patterns.

By leveraging higher-order connections and adaptive aggregation strategies, our approach ensures that results reflect not only structural density but also real-world interaction intensity, making the results more interpretable and contextually relevant.

8 Conclusions and Future Research

This article presents two main contributions. First, we conduct a modeling study, exploring the representational possibilities offered by different sources of information within the same network. Second, we propose a specific application of the proposed models, focusing on the problem of community detection. Specifically, we have introduced a methodology aimed at enhancing the effectiveness of CDP techniques by enriching the input data with communication-related information inherent to the graph. By integrating this additional data, our approach provides a more nuanced understanding of the network's structure, enabling the identification of more cohesive and meaningful partitions.

The central idea of this work is to enhance the representation of a network's for further community detection algorithm application. Incorporating additional information derived from the graph itself, such as communication patterns or other relational data, can provide a more nuanced understanding of the network's structure.

Our first goal, in terms of representativeness, is to propose a new model, the so-called *MCG*, for which we introduce a specific characterization in matrix form, referred to as *MCGA*. This enriched characterization of a graph, not limited to local information, such as direct connections between nodes, allows community detection algorithms to identify more cohesive and meaningful partitions, reflecting the underlying relationships more accurately.

This new representation models (*MCG-MCGA*) are particularly relevant for tasks where the interplay between local and global structures is critical, such as community detection, network clustering, and structural embedding. By integrating information across multiple scales, the proposed approach offers a robust framework for network analysis that extends beyond the limitations of single-scale adjacency matrices.

To validate the performance of the proposed community detection methodology based on complex representation models, another contribution of this work is the evaluation conducted in the experimental results presented in Sect. 5. These results demonstrate that incorporating communication-based data directly into the graph's structure leads to improved modularity scores compared to traditional methods relying solely on the adjacency matrix. This enhancement underscores the importance of considering intrinsic communication patterns within the network to achieve more accurate community detection outcomes. The crowning achievement of this work is the application of the proposed methodology to a case study that demonstrates the utility of our method in solving real-world problems.

On the other hand, the methodological contribution of this paper is of notable relevance for consideration in the application of new community detection methods based on deep learning [25, 26]. These methodologies typically rely on an initial partition obtained through the prior application of classical methods. As demonstrated, these initial partitions can be improved through the application of the *MCG-CDP* algorithm. Enhancing this classical partition inevitably leads to improvements in the final solution of the associated Deep Learning model [44]. To illustrate that this philosophy can be integrated into state-of-the-art deep learning techniques and also contributes to improving their performance, we have conducted a brief analysis in Sect. 6.

Future research should focus on applying this methodology to more real-world problems, where the interplay between local and global structures is critical. Moreover, although the effectiveness of the model has been demonstrated through computational results for a specific family of Φ values, it would be interesting to explore in detail the behavior and performance of community detection algorithms under other families of Φ. This could include, for instance, some of the aggregators proposed in Sect. 3 or, more generally, any operator from the OWA family. Another interesting approach to explore in detail is the representation in matrix form of the results from previous CDP algorithm applications. Subsequently, aggregating these results with the adjacency matrix may offer additional insights into the network's community structure. Such explorations could lead to more robust and adaptable community detection frameworks, enhancing their applicability across different domains. The introductory work presented in Sect. 6 will be expanded upon and thoroughly analyzed, focusing on the impact of our new CDP methodology when integrated with the latest and most advanced deep learning techniques applied to the context of group detection in networks.

Another promising avenue for future research is to extend the analysis of MCGA beyond modularity-based optimization approaches and explore its applicability in other community detection techniques, such as spectral algorithms [54]. While our study has primarily focused on methods that maximize modularity, integrating MCGA into alternative frameworks could provide further insights into its versatility and potential advantages. Investigating how MCGA interacts with spectral properties of networks and its integration to enhance the effectiveness of spectral methods in capturing community structures, represents an interesting future research line. The approach presented in Sect. 6, which addresses the community detection problem through state-of-the-art Deep Learning techniques, undoubtedly warrants an in-depth analysis.

On the other hand, and not necessarily related to the CDP (although it could be considered part of the objective), another very interesting context in which we believe the philosophy proposed in this paper could be exploited is in the analysis of node influence, in an effort to enhance the local information on which the heuristic models proposed in this regard are typically based [58–60].

Supplementary information The complete calculations of the entire experimental process, the case study, the networks used, and the applied code can be found in the Supplementary Materials accompanying this paper and on GitHub.

Funding This research has been partially supported by the Government of Spain, Grant Plan Nacional de I+D+i, PID2021-122905NB-C21.

Data availability The datasets generated and/or analyzed during the current study are available in the Github repository, [https://github.com/rodrhern-ucm/Social-network-analysis-a-novel-paradigm-for-improving-community-detection].

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

1. Rehman, S.U., Liu, K., Ali, T., Nawaz, A., Fong, S.J.: A graph mining approach for ranking and discovering the interesting frequent subgraph patterns. Int. J. Comput. Ingell. Syst. (1) (2021)
2. Besharatnia, F., Talebpour, A., Aliakbary, S.: Metaheuristic multi-objective method to detect communities on dynamic social networks. Int. J. Comput. Ingell. Syst. **14**(1), 1356–1372 (2021)
3. Gutiérrez, I., et al.: Community detection problem based on polarization measures. An application to Twitter: the COVID-19 case in Spain. Mathematics **9**(4) (2021)
4. Li, H., Gan, W.: A decomposition-based multiobjective chemical reaction optimization algorithm for community detection in complex networks. Int. J. Comput. Intell. Syst. **13**(1), 524–537 (2020)
5. Ahmadian, S., Rostami, M., Jalali, S.M.J., Oussalah, M., Farrahi, V.: Healthy food recommendation using a time-aware community detection approach and reliability measurement. Int. J. Comput. Intell. Syst. **15**(1) (2022)
6. Fortunato, S., Hric, D.: Community detection in networks: a user guide. Phys. Rep. **659**, 1–44 (2016)
7. Yang, Z., Algesheimer, R., Tessone, C.: A comparative analysis of community detection algorithms on artificial networks. Sci. Reports **6**(30750) (2016)
8. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. Phys. Rev. E **80**(5), 056117 (2009)
9. Marchese, E., Calderelly, G., Squartini, T.: Detecting mesoscale structures by surprise. Commun. Phys. **5**(132) (2022)
10. Luca, M., Fasolino, A.R., Ferraro, A., Moscato, V., Sperlí, G., Tramontana, P.: A community detection approach based on network representation learning for repository mining. Expert Syst. Appl. **120597** (2023)
11. Zhuo, Z., Chen, B., Yu, S., Cao, L.: Overlapping community detection using expansion with contraction. Neurocomputing **565**(126989) (2024)
12. Liu, B., Wang, D., Gao, J.: A multi-objective community detection algorithm with a learning-based strategy. Int. J. Comput. Intelli. Syst. **17**(1) (2024)
13. Newman, M.: Networks: An Introduction. Oxford Univ. Press, Oxford (2010)
14. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
15. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Turkey, 2005. Proceedings 20, pp. 284–293 (2005)

16. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. Eur. Phys. J. Special Topics **178**(1), 13–23 (2009)
17. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)
18. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exp. **2008**(10), 10008 (2008)
19. Zhang, H., Fan, C., Ren, Y., Jin, Y.: An improved Louvain algorithm for community detection. Math. Probl. Eng. **2021**, 1–11 (2021)
20. Do, D., Phan, T.: An improvement on the Louvain algorithm using random walks. arXiv:2108.13482 (2024)
21. Traag, V.: Faster unfolding of communities: speeding up the Louvain algorithm. Phys. Rev. E **92**(3), 032801 (2015)
22. Traag, V., Waltman, L., Eck, N.: From Louvain to Leiden: guaranteeing well-connected communities. Sci. Reports **9**(5233) (2019)
23. Hernández, R., Gutiérrez, I., Castro, J.: Social network analysis: beyond the greediness in community detection methods. In: Intelligent and Fuzzy Systems, pp. 714–721 (2024)
24. Cai, J., Hao, J., Yang, H., Yang, Y., Zhao, X., Xun, Y., Zhang, D.: A new community detection method for simplified networks by combining structure and attribute information. Expert Syst. Appl. **246**(123103) (2024)
25. Rong, F., Yuxin, W., Bo, H., Qian, L.: A novel network core structure extraction algorithm utilized variational autoencoder for community detection. Expert Syst. Appl. **222**, 119775 (2023)
26. Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D., Sheng, Q., Yu, P.: A comprehensive survey on community detection with deep learning. IEEE Trans. Neural Netw. Learn. Syst. **35**(4), 4682–4702 (2024)
27. Sugeno, M.: Fuzzy measures and fuzzy integrals-a survey. In: Readings in Fuzzy Sets for Intelligent Systems, pp. 251–257. Elsevier, NY (1993)
28. Shapley, L.: A value for $n-$person games. Ann. Math. Stud. **2**, 307–317 (1953)
29. Barroso, M., Gutiérrez, I., Gómez, D., Castro, J., Espínola, R.: Group definition based on flow in community detection. In: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 524–538 (2020)
30. Carnivali, G.S., Vieira, A.B., Ziviani, A., Esquef, P.A.: Covec: Coarse-grained vertex clustering for efficient community detection in sparse complex networks. Inf. Sci. **522**, 180–192 (2020)
31. Jonnalagadda, A., Kuppusamy, L.: A survey on game theoretic models for community detection in social networks. Soc. Netw. Anal. Min. **6**, 1–24 (2016)
32. Jonnalagadda, A., Kuppusamy, L.: A cooperative game framework for detecting overlapping communities in social networks. Physica A **491**, 498–515 (2018)
33. Gutiérrez, I., Gómez, D., Castro, J., Espínola, R.: Fuzzy measures: a solution to deal with community detection problems for networks with additional information. J. Intell. Fuzzy Syst. **39**(5), 6217–6230 (2020)
34. Zhou, X., Cheng, S., Liu, Y.: A cooperative game theory-based algorithm for overlapping community detection. IEEE Access **8**, 68417–68425 (2020)
35. Gutiérrez, I., et al.: Fuzzy measures: a solution to deal with community detection problems for networks with additional information. J. Intell. Fuzzy Syst. **39**(5), 6217–6230 (2020)
36. Gutiérrez, I., Gómez, D., Castro, J., Espínola, R.: Multiple bipolar fuzzy measures: an application to community detection problems for networks with additional information. Int. J. Comput. Intell. Syst. **13**(1), 1636–1649 (2020)
37. Lu, Z., Dong, Z.: A gravitation-based hierarchical community detection algorithm for structuring supply chain network. Int. J. Comput. Intell. Syst. **16**(1) (2023)
38. Calvo, T., Kolesárová, A., Komorníková, M., Mesiar, R.: Aggregation operators: properties, classes and construction methods. In: Studies in Fuzziness and Soft Computing, pp. 3–104 (2002)
39. Messiar, R., Kolesárová, A., Calvo, T., Komorníková, M.: A review of aggregation functions **20**, 121–144 (2008)
40. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans. Syst. Man Cybern. **18**(1), 183–190 (1988)
41. Gómez, D., Rodríguez, J.T., Montero, J., Bustince, H., Barrenechea, E.: N-Dimensional overlap functions. Fuzzy Sets Syst. **287**, 57–75 (2016)
42. Bustince, H., Barrenechea, E., Pagola, M., Fernandez, J.: The notions of overlap and grouping functions. Stud. Fuzziness Soft Comput. **336**, 137–156 (2016)
43. Gutiérrez, I., Barroso, M., Gómez, D., Castro, J.: Social networks analysis and fuzzy measures: a general approach to improve community detection in directed graphs. In: The 20th World Congress of the International Fuzzy Systems Association, pp. 246–253 (2023)
44. Gutiérrez, I., Barroso, M., Gómez, D., Castro, C.: Improving community detection algorithms in directed graphs with fuzzy measures. An application to mobility networks. Expert Syst. Appl. **269**(126305) (2025)
45. Kunegis, J.: KONECT – The Koblenz Network Collection. In: Proceedings of the International Conference Companion on World Wide Web, pp. 1343–1350 (2013). http://konect.cc/
46. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. Nature **393**, 440–442 (1998)
47. Ravasz, E., Barabási, A.: Hierarchical organization in complex networks. Phys. Rev. E **67**, 026112 (2003)
48. Barabási, A., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)

49. Casella, G., Berger, R.: Statistical Inference. Duxbury Press, Pacific Grove (2002)
50. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction (2009)
51. De Santo, A., Galli, A., Moscato, V., Sperlì, G.: A deep learning approach for semi-supervised community detection in online social networks. Knowl.-Based Syst. **229**, 107345 (2021)
52. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
53. Wang, J., Cao, J., Li, W., Wang, S.: CANE: community-aware network embedding via adversarial training. Knowl. Inf. Syst. **63**(2), 411–438 (2021)
54. Amini, A., Chen, A., Bickel, P., Levina, E.: Pseudo-likelihood methods for community detection in large sparse networks. Ann. Stat. **41**(4), 2097–2122 (2013)
55. Liu, F., Wu, J., Xue, S., Zhou, C., Yang, J., Sheng, Q.: Detecting the evolving community structure in dynamic social networks. World Wide Web **23**(2), 715–733 (2020)
56. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
57. Ngiam, J., Chen, Z., Bhaskar, S.A., Koh, P.W., Ng, A.Y.: Sparse filtering. In: Advances in Neural Information Processing Systems 24 (NIPS 2011) (2011)
58. Sheng, J., Dai, J., Wang, B., Duan, G., Long, J., Zhang, J., Guan, K., Hu, S., Chen, L., Guan, W.: Identifying influential nodes in complex networks based on global and local structure. Physica A **541**, 123262 (2020)
59. Samie, M.E., Behbood, E., Hamzeh, A.: Local community detection based on influence maximization in dynamic networks. Appl. Intell. **53**(15), 18294–18318 (2023)
60. Zhao, Z., Zhang, N., Xie, J., Hu, A., Liu, X., Yan, R., Wan, L., Sun, Y.: Detecting network communities based on central node selection and expansion. Chaos Solitons Fractals **188** (2024)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Rodrigo Hernández[1] · Inmaculada Gutiérrez[1,2] · Javier Castro[1,2]

✉ Rodrigo Hernández
rodrhern@ucm.es

Inmaculada Gutiérrez
inmaguti@ucm.es

Javier Castro
jcastroc@estad.ucm.es

[1] Faculty of Statistical Studies, Complutense University of Madrid, Avda. Complutense s/n, Madrid 28040, Madrid, Spain

[2] Instituto Universitario de Estadística y Ciencia de Datos UCM, Avda. Complutense s/n, Madrid 28040, Madrid, Spain