

Block-corrected modularity for community detection

Hasti Narimanzadeh ^{*}, Takayuki Hiraoka , and Mikko Kivelä
Department of Computer Science, Aalto University, 00076 Espoo, Finland



(Received 26 February 2025; revised 24 June 2025; accepted 29 July 2025; published 27 August 2025)

Unknown node attributes in complex networks may introduce community structures that are important to distinguish from those driven by known attributes. We propose a *block-corrected* modularity that discounts given block structures present in the network to reveal communities masked by them. We show analytically how the proposed modularity finds the community structure driven by an unknown attribute in a simple network model. Further, we observe that the block-corrected modularity finds the underlying community structure on a number of simple synthetic network models while methods using different null models fail. We develop an efficient spectral method as well as two Louvain-inspired fine-tuning algorithms to maximize the proposed modularity and demonstrate their performance on several synthetic network models. Finally, we assess our methodology on various real-world citation networks built using the OpenAlex data by correcting for the temporal citation patterns.

DOI: [10.1103/PhysRevE.112.024307](https://doi.org/10.1103/PhysRevE.112.024307)

I. INTRODUCTION

Complex systems typically consist of elements interacting with one other, commonly represented by networks of nodes connected by edges. Many such systems comprise multiple types of elements that differ in their characteristics, affiliations, roles, or functions. For example, proteins in cells accomplish different biological functions; species in an ecosystem occupy different niches and habitats; individuals in society can be categorized by gender, age, educational background, or political beliefs; scientific articles vary in topics, fields of study, and institutional affiliations of their authors. These node-level *attributes* can be known, i.e., directly observable, or unknown, i.e., some characteristics of the nodes which are not directly measured. The likelihood that nodes are connected is often influenced, to different extents, by their attributes through mechanisms such as homophily or assortativity. That is, two elements that share functions or characteristics are more likely to interact. When mechanisms are present that translate attributes into network structure, those attributes can be inferred from the network topology even when they are not directly observable—a task commonly referred to as community detection. Indeed, the communities are typically interpreted as proxies for some unknown attributes, such as functional roles of genes or social groups of individuals [1–6].

Among the many community detection methods that aim to uncover unknown attributes, many of the most popular

approaches are based on *modularity* [7]. Built upon maximizing the modularity quality measure across all possible partitions, modularity optimization methods comprise a vast collection of algorithms, ranging from greedy algorithms [8,9] to spectral methods [7,10,11]. Modularity, in essence, measures the abundance of edges within each community as compared to that predicted by a randomized *null model* of the network with no community structure. Despite its prevalence, modularity maximization is not without its shortcomings and has faced criticism over the recent years [12,13]. Nonetheless, it is a versatile method easily adaptable to various network types such as multilayer, temporal multilayer, spatial, and motif networks, to name a few [14–19].

Detecting communities in networks whose every node is annotated by metadata comprises another vast arena. These metadata are observable (known) node attributes that can either be continuous, discrete or categorical. On many occasions, it has been shown that such node metadata can be correlated with the connectivity patterns of networks [3,20–22]. In light of these observations, a widespread assumption emerged that these known attributes intrinsically align with the unknown attributes, which has led to the development of methods that ensure homogeneity of known attributes in the community structures that have yet to be uncovered [23–26]. A common approach, for example, is to construct a similarity network based on the known node attributes, then perform community detection, e.g., using modularity-based methods, on its union with the original network [23,24,27].

It is not, however, always the case that known attributes are the sole drivers of the network structure, or that the unknown attributes that are not directly observed in data align with the known ones. If such unknown attributes induce a structure that is distinct from that driven by the known ones, focusing on the latter may eclipse the former. That is, conventional community detection methods, in such scenarios, would either return the known attributes or yield communities that are

^{*}Contact author: hasti.narimanzadeh@aalto.fi

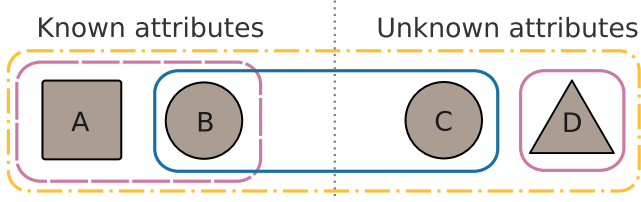


FIG. 1. A schematic of the relation of community structures found by three types of community detection methods to the attributes that induce assortative structures in a network. In this example, nodes are labeled with two known attributes (A and B) and two unknown attributes (C and D) where attributes B and C are closely aligned. Using a conventional community detection method with no consideration for metadata, the discovered structure is affected by all known and unknown attributes A, B, C, and D (dot-dashed, yellow). If a method assumed a specific known attribute, in this example B, was aligned with the community structure, the found communities would resemble the effect of this known attribute and the unknown attribute C with which it is aligned (dark solid, blue). Our proposed method would nullify the effect of the known attributes A and B (dashed, purple) as well as the unknown attribute C to arrive at a community structure solely driven by the unknown attribute D (light solid, purple).

distorted versions of the unknown attributes. Thus excluding the influence of known attributes on the structure and retrieving the community structure underpinned by unknown attributes will lead to more meaningful findings.

Academic citation networks provide an example of how known attributes can obscure potentially intriguing unknown ones. The publication time of each article imposes a rather strong constraint on its citation behavior, as papers generally cannot cite future publications, nor do they cite too recent or too old papers [28]. Therefore, what we discover by applying off-the-shelf community detection methods may strongly reflect the temporal connectivity of the network, as opposed to other driving forces that may be at play, such as academic disciplines. At times we may be able to use the known attributes to bias the discovered communities towards what we are interested in—for example, the journal in which an article is published can be treated as node metadata that aligns with the academic field. This approach, however, relies heavily on the assumption that there is an available known attribute that aligns with the unknown one, which is not necessarily the case. If publication date and journal are the only known attributes, previous methods cannot uncover the community structure caused by other attributes, such as the countries where authors are based. We illustrate this schematically in Fig. 1.

Here we take the opposite approach: for a given network data accompanied by node metadata, we discount the effect of known attributes on network structure in order to isolate the effect of unknown attributes. This amounts to asking whether there exists any community structure beyond the known attributes. Factoring out the known attributes has been previously accomplished in spatial networks, for example, by adapting the definition of modularity to discount the effects of spatial embedding [17]. In a mobile phone network where calls are predominantly placed among persons

in close physical proximity, nullifying this physical proximity of callers came to reveal linguistically homogeneous communities, communities of callers speaking the same language. Similar approaches have been proposed for temporal networks by accounting for an ordering on the nodes, in either modularity-based [14] or Bayesian inference methods [29].

In this work we propose a modularity measure with a stochastic block model as a null model that can factor out any categorical attributes. We illustrate that the model isolates and captures the effects of unknown attributes on the network structure by incorporating arbitrary connectivity patterns induced by known attributes. We also derive a fast, vectorized and efficient extension of the spectral modularity maximization heuristic [7], as well as two greedy fine-tuning algorithms. We demonstrate the utility and scalability of our method on real-world citation networks containing millions of nodes, with publication time as the known attribute, showing that null models proposed in prior studies often impose specific assumptions of connectivity which might not align with the observed empirical citation patterns, especially considering how they have evolved in more recent times. In contrast, our proposed method offers the flexibility to accommodate arbitrary connectivity patterns, providing a more adaptable framework for community detection.

First, in Sec. II we formally define our proposed null model and modularity. Next, in Sec. III we show with two sets of examples how this null model can capture the unknown community structure even in the presence of known attributes that significantly affect the connectivity of the network. Next, in Sec. IV we discuss the practical application of our null model on large networks, specifically the intricacies of efficiently applying our proposed null model to sparse networks. Finally, in Sec. V we demonstrate the utility and the efficacy of our proposed null model on real-world citation networks ranging from hundreds of thousands to tens of millions of edges.

II. BLOCK-CORRECTED MODULARITY

In the following, we focus on unweighted, directed networks characterized by their asymmetric adjacency matrix $\mathbf{A}$. Element A_{ij} of the adjacency matrix $\mathbf{A}$ equals 1 if there is an edge from node i to node j and 0 otherwise. V is the set of nodes in the network. The number of edges and nodes in the network is denoted by m and $|V|$, respectively.

Behind most community detection methods lies the fundamental idea that communities of nodes in the network are densely connected within and sparsely connected between communities. Associated with this notion is often a mathematical definition that assesses the quality of a candidate partition of the network. These are associated with optimization methods aiming to maximize the quality function across all possible node partitions. One of the most widely recognized quality functions is the modularity function [7] defined mathematically as

$$Q = \frac{1}{m} \sum_{C \in \mathcal{P}} \sum_{i,j \in C} (A_{ij} - P_{ij}), \quad (1)$$

where the summation $i, j \in C$ is over all pairs of nodes i and j in a community C in a candidate partition of the network

denoted by $\mathcal{P}$. This summation applied to the corresponding elements of the adjacency matrix $\mathbf{A}$ equates the number of edges in community C . Modularity value Q tells how salient the communities in the candidate partition are compared to a random reference network. The notion of a random reference network is captured by the parameter P_{ij} , which is the expected probability of an edge from node i to j across an ensemble of random networks, a generative *null model* with certain constraints assimilated into edge probability P_{ij} . These constraints embody specific features of the network under study which must be preserved in the null model, such that generated random networks are maximally similar to the original network without containing any community structure other than what is induced by the known attributes.

Multitudes of choices for null models have been developed over the years, ranging from Erdős-Rényi $G(n, p)$ model where number of nodes and the edge probability are constants, preserving the expected number of edges [30,31], the configuration models [32–34] wherein degrees of nodes are preserved, to models designed for bipartite networks [35], spatial networks [17], directed acyclic networks [14,36], models designed for uncovering communities in networks with signed edges [8,37], or generating dynamic networks with temporal motifs [38,39]. All such models are characterized by preserving core qualities present in the observed network in their respective null models, which is integral to uncovering unknown community structures as we will expound shortly.

Block null model

Widely used as an inference framework, stochastic block models [34,40] also serve as effective generative models, much like a null model is. They are characterized by their block connectivity matrix which can be interpreted as an array of discrete node-level attributes. In the study of real-world networks, it is often the case that some attributes are known *a priori*, while some are unknown. Unknown attributes may impose a community structure that can be of interest to retrieve and may be largely independent of communities borne out of the known attributes. Addressing this interplay, we devise a general *block* null model for uncovering unknown community structures, generalized to preserve any form of block structure inherent to the network.

We propose a *block-corrected* modularity and its associated *block* null model for detecting communities in networks with any form of *a priori* known block structure. The probability of an edge from node i to node j under the block null model is

$$P_{ij}^{\text{bc}} = \frac{k_i^{\text{out}} k_j^{\text{in}}}{K_r^{\text{out}} K_s^{\text{in}}} L_{rs}, \quad (2)$$

where r denotes the *a priori* block of node i and s the block of node j , k_i^{in} and k_i^{out} the in- and out-degree of node i , K_r^{in} and K_r^{out} the sum of in- and out-degrees of all nodes in block r , and L_{rs} the number of out-going edges from block r to s . In a citation network, for example, scientific papers are nodes, their publication dates are *a priori* blocks and their citation relationships are embodied through edges. The block null model preserves, on average, both the in- and out-degree distribution of nodes as well as the edge density between every

pair of blocks. The model is generalizable to any existing block structure in networks. Any *a priori* block structure that a network may contain can be captured through L_{rs} . Elements of matrix $\mathbf{L}$ are determined from the data, where L_{rs} is the number of edges from block r to s .

An effective null model preserves the properties of the observed network and, yet, is divested of any community structure. Retrieving block structures of the known properties (attributes) might not always pass for an interesting question as discussed before. In many applications, the uncovering of communities induced by unknown attributes or properties can make for a compelling find. To this end, the known block structure needs to be discounted by the null model. Such nullification occurs when the *a priori* block structure is captured by the model.

In the absence of such discounting, the associated modularity-based method may be biased by the *a priori* structure, detecting erroneous or suboptimal communities. We illustrate a simple example of this in Sec. III A.

III. ILLUSTRATIVE EXAMPLES

In what follows, we present two example network models to illustrate (1) how the block-corrected modularity reveals unknown community structures in a simplified network with two node attributes, one known and one unknown, and (2) how it fares under slightly more complex settings where the network is temporal and the connectivity pattern follows specific distributions.

A. Simple network model with hidden communities

We devise a simplified example to show the effect of the known and unknown attributes on modularity-based community detection under our block-corrected null model according to Eq. (2) and the directed configuration null model [32,41].

Null model in the directed modularity [41], is defined as

$$P_{ij}^{\text{dir}} = k_i^{\text{out}} k_j^{\text{in}} / \sum_i k_i^{\text{out}}, \quad (3)$$

where edge probability is proportional to the in-degree and out-degree of nodes, normalized by the total number of edges. We analytically compute modularity values of specific partitions with respect to both known and unknown structures to assess which partition is preferred by the block-corrected modularity and the directed modularity.

Each node has a known binary attribute, x , and an unknown binary attribute y , taking values zero or one. The probability of connection from node i to j is $p_{ij} = p_{\delta_{ix}\delta_{jy}}^{(x)} p_{\delta_{iy}\delta_{jx}}^{(y)}$, where δ denotes the Kronecker delta function. Thus the parameters describing the system are $p_1^{(x)}$, $p_1^{(y)}$, $p_0^{(x)}$, and $p_0^{(y)}$, where the former two affect the connectivity probability of when two nodes have the same values for attributes x and y , respectively, and the latter two when values of x and y differ. For simplicity, we assume the network is divided into equally sized quadrants of nodes for every possible combination of the (x, y) pair. Figure 2 shows a schematic of the network. We refer to this network model as the *intersecting* model in the following discussions.

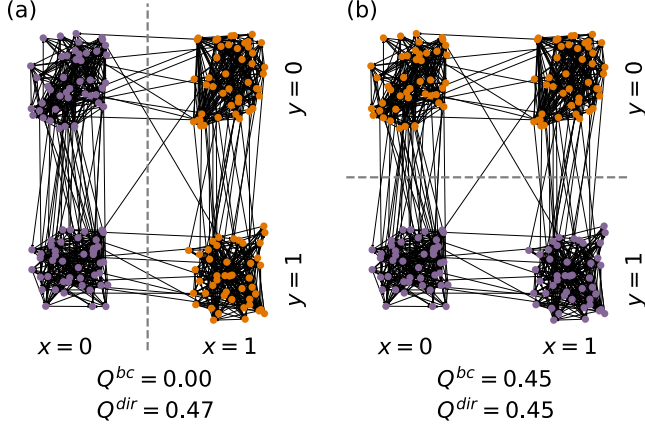


FIG. 2. Network whose nodes are labeled with a known binary attribute x and a unknown binary attribute y . Panel (a) shows the partitioning of x and (b) that of y , for which modularity values are computed for the block-corrected and the directed models. The network is constructed from 180 nodes equally split between four possible combinations of x and y values, with parameter $p_1^x = 0.6$, $p_1^y = 0.2$, $p_0^x = 0.02$, and $p_0^y = 0.01$.

For such a network, we can define two candidate partitions: one that divides nodes into two classes where $x = 0$ or $x = 1$, whose modularity value is denoted by Q_x , and one with two groups of nodes where $y = 0$ and $y = 1$ with its respective modularity value being Q_y . Modularity values Q_y and Q_x of a null model are, therefore, quality measures reflecting the superior partition. Here the aim is to find the unknown partition defined by y but masked by the known partition given by x . To do so, the partitioning across the unknown attribute y needs to be preferred over that across the known attribute x , that is, Q_y needs to be larger than Q_x .

Deriving the modularity values Q_x and Q_y for our block null model shows that partitioning along the known attribute x results in a negligible modularity gain ($Q_x^{bc} \approx 0$ when $|V| \gg 0$, where Q_x^{bc} denotes Q_x calculated using the block null model). For the directed modularity, using the directed null model [Eq. (3)], we likewise compute the modularity value along x to be

$$Q_x^{\text{dir}} = \frac{1}{2} \frac{p_1^{(x)} - p_0^{(x)}}{p_1^{(x)} + p_0^{(x)}}. \quad (4)$$

Splitting the network with respect to the unknown attribute y yields the same modularity value for both null models:

$$Q_y^{\text{dir}} = Q_y^{\text{bc}} = \frac{1}{2} \frac{p_1^{(y)} - p_0^{(y)}}{p_1^{(y)} + p_0^{(y)}}. \quad (5)$$

The derived modularity expressions for Q_y^{dir} , Q_y^{bc} , and Q_x^{dir} have a similar form to the E-I index [42] which equates to the number of internal links of group members subtracted from the number of external links divided by their summation. Similar to the E-I index, modularity values Q_x and Q_y are measures of the dominance of in-group over out-group edges. As per Eq. (4) and Eq. (5) for a given binary attribute the more segregated its two membership groups are, the larger its corresponding modularity value is. The block-corrected modularity evidently rejects the known attribute community structure

($Q_x^{\text{bc}} = 0$) regardless of how well separated and embedded its two comprising groups, i.e., the nodes with $x = 0$ and those with $x = 1$, are. In contrast, the directed null model's ability to recover the unknown structure depends on the separatedness of the corresponding attribute's groups. If the separation of the two groups under known attribute x is more pronounced than those of the unknown attribute y , the modularity value Q_x is larger than Q_y , in which case a modularity optimization partitioning algorithm will dismiss the community structure driven by the unknown attribute y in favor of partitioning the network based on the known attribute x . The details of the derivations are outlined in Appendix A.

While it may seem plausible to approach this problem by partitioning the network into four communities, i.e., one community for each combination of x and y , it is noteworthy that such a partition would only be preferred under certain combinations of parameters p_0^x , p_1^x , p_0^y and p_1^y . In Appendix A we present a combination of parameters where a four-community partition would not be the solution, while the method presented in this paper provides a more general solution for nullifying any pattern of connectivity imposed by known attributes.

B. Communities in temporal networks

A quintessential example of networks that have directionality with a set of known attributes is citation networks, commonly modeled as directed acyclic graphs (DAGs) [14]. Citation networks are temporal networks in the sense that scientific publications represented by nodes (and citations represented by edges) are timestamped. The timestamps impose structures onto the network that may be classified by community detection methods as genuine communities. In what follows, we use citation networks as our primary class of networks under study.

A null model is characterized by its edge probability P_{ij} , which describes a generative network model for networks maximally similar to an observed network with no community structure present. Null models may nonetheless exhibit structures different from those existing in the observed network which can speak to their efficacy as a generative model. With citation networks in mind, we evaluate the performance of our proposed block null model against two previously proposed null models: the directed null model [41], which we already introduced in the previous subsection, and the DAG null model [14,36]. The directed and the DAG null models are designed to operate on directed networks, with the latter specifically devised with citation networks in mind as they are often represented by directed acyclic graphs.¹ While the directed null model preserves only in- and out-degrees of nodes, the DAG null model additionally encodes two parameters μ_t and λ_t per time layer t describing the edge connectivity between layers:

$$P_{ij}^{\text{DAG}} = \begin{cases} 0 & t_i \geq t_j, \\ k_i^{\text{in}} k_j^{\text{out}} \frac{\prod_{r=t_i+1}^{t_j-1} \lambda_r}{\prod_{r=t_i+1}^{t_j} \mu_r} & t_i < t_j, \end{cases} \quad (6)$$

¹Real-world citation networks, however, are not always acyclic, requiring preprocessing measures to ensure the acyclic nature holds.

where

$$\mu_t = \sum_{i:t_i < t} k_i^{\text{in}} - \sum_{i:t_i < t} k_i^{\text{out}}$$

and

$$\lambda_t = \sum_{i:t_i < t} k_i^{\text{in}} - \sum_{i:t_i \leq t} k_i^{\text{out}}.$$

1. Generative network models

We define here a family of generative models for citation networks with specific edge connectivity patterns and timestamped nodes. The realizations of these generative models can be thought of as simplified citation networks, with characteristics of directed acyclic graphs and specified edge length distributions, that is, the time differences of the edge endpoints. The generated networks can thus be used to test the performance of the three definitions of modularity based on different null models.

We assume that each node in the network is assigned two attributes: a discrete timestamp represented by integers $1, \dots, t_{\text{max}}$, and the membership in a planted community, which is a categorical variable represented by integers $1, \dots, b$. We call the set of nodes that share the same timestamp value a time layer. Each generative model is characterized by three matrices: $\mathbf{N}$ is a $b \times t_{\text{max}}$ matrix denoting the number of nodes in each planted community in each time layer; $\mathbf{T}$ is a $t_{\text{max}} \times t_{\text{max}}$ matrix denoting the effect of time layers on the probability of connectivity; and $\mathbf{B}$ is $b \times b$ matrix specifying the effect of community memberships on the probability of connectivity. Each node i belongs to exactly one planted community b_i and one time layer t_i and the probability of connectivity between nodes i and j is thus simply

$$p_{ij} = B_{b_i b_j} T_{t_i t_j}. \quad (7)$$

We first define the “skewed” variant of our generative model. This variant limits connections to only the temporally adjacent and those in the first and last blocks. The probability of two nodes being connected by an edge is greater if both belong to the same planted community, i.e.,

$$N_{b_i t_i} = N/b, \quad (8)$$

$$B_{b_i b_j} = \begin{cases} \frac{\langle k^{\text{ing}} \rangle}{N/b} & b_i = b_j, \\ \frac{\langle k^{\text{outg}} \rangle}{N/b} & \text{otherwise,} \end{cases} \quad (9)$$

$$T_{t_i t_j}^{\text{skw.}} = \begin{cases} 1 & t_j = t_i - 1, \\ 1 & t_j = 1 \text{ and } t_i = t_{\text{max}}, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

with node rate N determining the number of nodes in each layer,² b determining the number of planted communities, and $\langle k^{\text{ing}} \rangle$ and $\langle k^{\text{outg}} \rangle$ parametrizing the expected in- and out-group degree of the nodes. The distribution of edge length, i.e., the time difference between adjacent nodes, is heavily skewed,

with many short edges between adjacent temporal blocks, and a handful of very long ones.

Figure 3 illustrates a realization of the skewed model. The example synthetic network has $b = 2$ planted (horizontal) unknown communities, and *a priori* known (vertical) time layers (or temporal blocks). The connections between the first and the last layer are not visualized.

To contrast this, we construct an “exponential” generative model with a probability of connectivity exponentially decaying with edge lengths, i.e.,

$$T_{t_i t_j}^{\text{exp.}} = \begin{cases} d(1-d)^{t_i - t_j} & \text{for } t_i > t_j, \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

where d is a parameter controlling the rate of decay.

Lastly, we construct a “power-law” model with a smooth yet heavy-tail probability distribution of edge lengths defined as

$$T_{t_i t_j}^{\text{plw}} = \begin{cases} \frac{(t_i - t_j)^\gamma}{\zeta(-\gamma)} & \text{for } t_i > t_j, \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

where $\zeta(\cdot)$ is the Riemann zeta function. This provides a more realistic scenario compared to the skewed model, where the temporal edge lengths follow a smoother, heavy-tail probability distribution which, as we will show in Sec. V, is corroborated by edge length distributions observed in real-world citation data [28].

Note that for all three models, the matrices $\mathbf{N}$ and $\mathbf{B}$ remain the same, and only the connectivity pattern along the known attribute (time) is changed. This allows us to compare the robustness of the three null models with respect to changes in connectivity patterns across time.

2. Temporal connectivity pattern in null models

Each null model, in essence, constructs a function P_{ij} by observing and encoding key values from the given network. These null models can, in turn, function as generative models. Ideally, a null model will reproduce as much of the effects of the known attribute from the original network without reproducing any of the effects of the unknown attribute. In this way, the term $\mathbf{A} - \mathbf{P}$ of modularity [Eq. (1)] will reflect the connectivity present (or missing) due to the effect of the unknown attributes.

To assess this, we define a single generalized $(bt_{\text{max}}) \times (bt_{\text{max}})$ connectivity matrix $\mathbf{M}$ for each temporal network, where $M_{(bt_i + b_i)(bt_j + b_j)}$ is the fraction of existing edges (out of all possible edges) from community b_i at time t_i to community b_j at time t_j . We then compare this to the connectivity matrix of the realizations of the same network passed through block-corrected, DAG, and directed null models.

Figure 3(b) depicts the connectivity matrix $\mathbf{M}$ of a network drawn from the skewed network model with two unknown communities, along with connectivity matrices of the three null models. The effect of the known attribute (time) is encoded in the 2×2 blocks of nonzero probability of connectivity near the diagonal representing adjacent time layers, as well as the 2×2 block at the corner representing connections between the first and the last layer. The structure induced by the unknown attribute is encoded in the difference of colors

²Node rate N is selected such that it is divisible by the number of planted communities b .

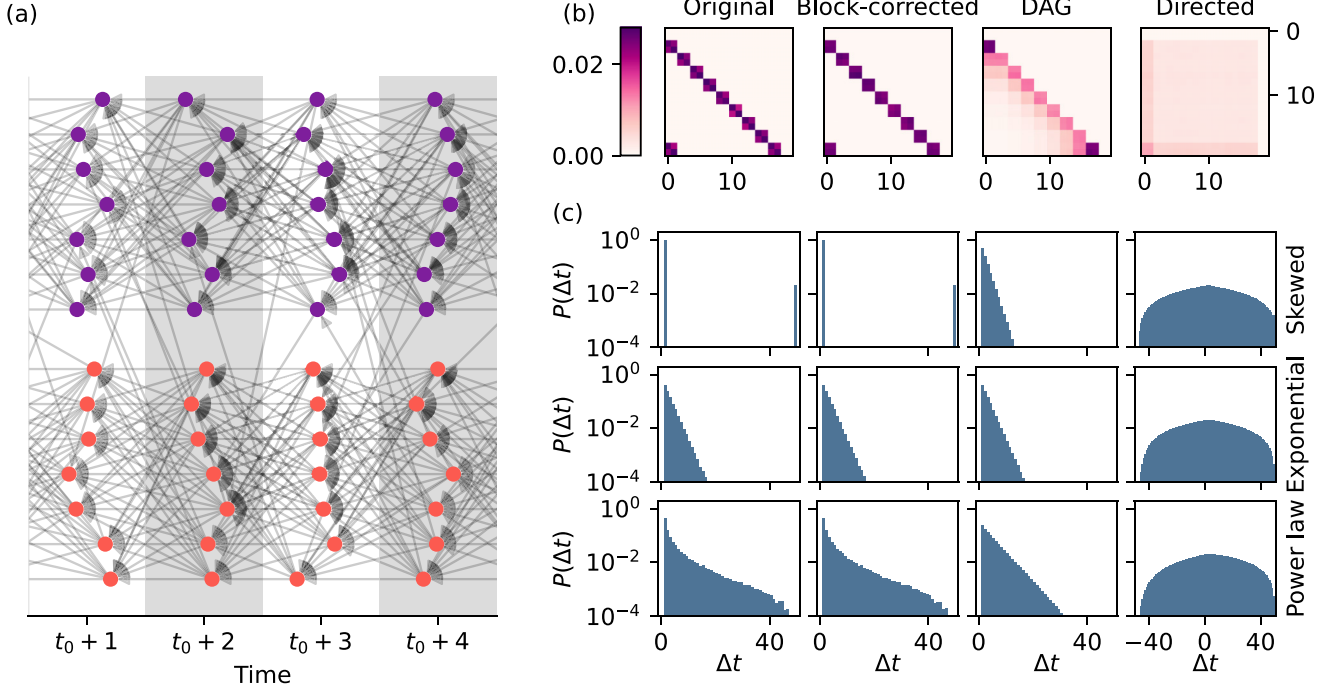


FIG. 3. (a) Schematic of the skewed generative network model with two planted communities and a temporal structure where only nodes in adjacent blocks are connected. In addition, there are long links from the last block to the first. (b) Averaged connectivity matrices $\mathbf{M}$ of an ensemble of 100 network realizations of the block-corrected, DAG, and directed null models. The connectivity matrix of the original network, generated using the skewed model [Eq. (10)], is labeled “Original.” The skewed model network has $b = 2$ planted communities, $t_{\max} = 10$ temporal layers, $N = 100$ nodes per temporal layer, mean in-group degree $\langle k^{\text{ing}} \rangle = 10$ and out-group degree $\langle k^{\text{outg}} \rangle = 8$. (c) The distribution of temporal edge lengths for all three network models: one with a skewed distribution of edge lengths [Eq. (10)], one with an exponential distribution [Eq. (11)], and one with a power-law edge length distribution [Eq. (12)]. The three observed networks for skewed, exponential, and power law in panel (c) have $t_{\max} = 50$ temporal blocks, i.e. time layers, with a fixed number of nodes ($N = 100$) per temporal block. The number of planted communities is $b = 2$ and each node has a mean in-group degree $\langle k^{\text{ing}} \rangle = 10$ and out-group degree $\langle k^{\text{outg}} \rangle = 8$. Edge length distributions for the drawn networks from the three null models are shown in their respective columns.

between diagonal and antidiagonal elements in each 2×2 block.

Averaged connectivity matrices of the realizations for the block-corrected, DAG, and directed null models are shown in the following three panels. The temporal connectivity patterns produced in our block-corrected model closely resemble those of the original skewed network, whereas the DAG model smooths the temporal connectivity pattern into an exponential decay. The directed model, on the other hand, assumes almost all community-time pairs are equally likely to be connected.

Models’ capacity to generate networks close to the observed network is further affirmed by comparing the temporal edge length distribution of the generated ensemble of networks by the null models to that of the observed network. We construct a network with each of the three generative models described in Sec. III B 1, namely, a skewed network [Eq. (10)], a network with exponential edge length distribution [Eq. (11)] with decay $d = 0.4$, and a network with power-law length distribution [Eq. (12)] with exponent $\gamma = -1.4$. For each network, we draw an ensemble of networks by passing it to each of the three null models. We then compare the edge length distribution of these *nullified* networks to that of the original of the three network models.

Figure 3(c) shows that both the block-corrected and the DAG models recover the exponential edge length distribution

observed in the original network. This is not the case, however, when the observed edge lengths follow a more heavily skewed distribution. The DAG model falls short of reproducing the distribution, producing an exponential edge length decay instead, while the block-corrected model generates an ensemble whose edge lengths closely follow the observed distribution. Similarly, under a power-law distribution for the edge lengths, the DAG model generates networks with exponential edge length distributions instead, whereas the original power-law distribution is recreated by the block-corrected model. In all three generative network models with exponential and skewed distributions, the directed model proves ineffective at capturing the nature of the distributions, producing almost constant temporal edge length distributions only affected by the finite number of temporal blocks $t_{\max}$.

3. Susceptibility to erroneous partitioning

For an ideal modularity measure, the modularity value of the true partition [the two horizontal color-coded communities of nodes in Fig. 3(a)] is larger than that of any other erroneous candidate partition. Otherwise, a modularity maximization algorithm will find the erroneous candidate partition instead of the true partition. To assess whether this holds, for each of the three modularities, we compare the modularity value

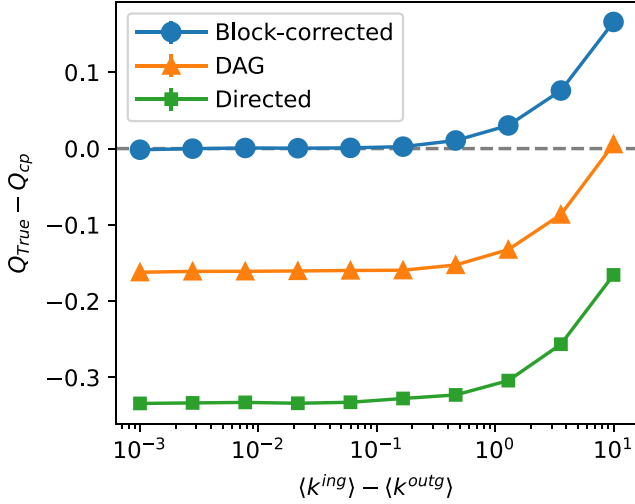


FIG. 4. Difference between modularity values for the erroneous candidate partitions, where the first and last one-fourth of temporal blocks are placed in one community and the rest in another, and that of the true partition over an ensemble of random networks with varying average in-group degrees. All networks are realizations of the skewed model [Eq. (10)], with $t_{\max} = 12$ temporal blocks and $N = 100$ nodes per block. $Q_{\text{true}} > Q_{\text{cp}}$ indicates that a modularity prefers the true partition to the erroneous candidate. Error bars show 95% confidence intervals.

of the true planted partition Q_{true} to that of an erroneous candidate partitioning Q_{cp} , in which the first and last quarter of all nodes in the temporally ordered network are in one community and the rest in another. This candidate partition places most temporally long edges in one community and the short ones in another. Such partitioning can potentially portray how effective null models are in the face of temporally long edges.

Our results in Fig. 4 illustrate that, under the skewed synthetic network model, the block-corrected null model proves to be impervious to the erroneous candidate partitioning ($Q_{\text{true}} > Q_{\text{cp}}$). It correctly deems the ground-truth horizontal partition preferable, even in networks whose ground-truth communities are only slightly distinguishable. The other two null models, however, erroneously prefer the incorrect partitioning over the ground-truth communities, even more so in networks whose true communities are not abundantly distinct from each other as quantified by the difference between average in-group and out-group degrees, i.e., as $\langle k^{\text{ing}} \rangle - \langle k^{\text{outg}} \rangle$ approaches zero.

As we will describe in Sec. V, heavy-tail distributions of edge lengths is observed in certain real-world systems, including in citation networks. As we saw in Fig. 3(c), while our block null model can replicate the temporal connectivity pattern induced by a power-law edge length distribution, the two other null models do not reproduce the underlying distribution. To understand the implications of this, we directly measure the ability of each modularity definition to recover the planted partitions in the space of all possible bipartitions of the network. To simplify this comparison, we assume the number of planted communities is known in all cases.

The quality of a partition can be quantified by the adjusted Rand score between that and the planted partition. The Rand score in its original form is a similarity measure for two partitionings, defined as

$$R = \frac{\sum_{i < j} \eta_{ij}}{\binom{|V|}{2}}, \quad (13)$$

where η_{ij} is equal to one if both partitionings agree that i and j are in the same or different communities [43]. The adjusted Rand score modifies this definition so that a score of zero represents performance no better than random labeling of nodes and one is a complete match between the candidate partitioning and the true one [44]:

$$\frac{R - E[R_{\text{random}}]}{R_{\max} - E[R_{\text{random}}]}, \quad (14)$$

where $E[R_{\text{random}}]$ is the expected Rand score of random labeling and $R_{\max}$ the maximum possible value for the Rand score corresponding to a perfect match.

Figure 5(a) shows that as the edge length distribution tail becomes heavier, that is, as the absolute value of the exponent γ in Eq. (12) becomes smaller, the block-corrected modularity retains its ability to distinguish the ground-truth partition, whereas a sharp drop is observed in the case of the DAG modularity. The directed modularity fails to capture the planted temporal community structure in the networks altogether and its adjusted Rand score remains at zero across all exponents. The sharp drop in performance of the DAG modularity, ultimately, stems from the fact that its implicit connectivity assumption, i.e., exponential edge length distribution, does not match the connectivity patterns present in the realized networks.

In the *intersecting* network model we studied in Sec. III A, where nodes come with exactly one *a priori* and one unknown attribute, we observed how the directed modularity is likely to divide the network with respect to the known attribute. Under the *power-law* random network model where node timestamps operate as the *a priori* or known attribute, it stands to reason to speculate that the low adjusted Rand scores for directed modularity, resembling those of close-to-random labeling, are rooted in the network being divided into two temporally separated clusters.

In such a case, community entropies quantifying how uniformly each community is distributed across time layers should be relatively small. For the same ensemble of networks in Fig. 5(a), we plot the entropy of one of the two discovered communities over a range of power-law exponents. For a community, the probability P_t that a randomly selected node belongs to time layer t is computed and the entropy associated with the community is thus $-\sum_t P_t \log_2 P_t$. As evidenced by Fig. 5(b) the entropies for the directed modularity are notably smaller than that of the block-corrected and the DAG modularities, remaining close to the entropy for equal vertical cut $S = \log_2(t_{\max}/2) \approx 6.64$ for $t_{\max} = 200$. Entropy of the block-corrected modularity keeps at the analytical entropy for a uniform distribution $S = \log_2(t_{\max}) \approx 7.64$ for all values of exponents. It is noteworthy that the block-corrected and the DAG modularities begin to diverge in entropy values starting from the same threshold of $\gamma \geq -2$ as in Fig. 5(a)

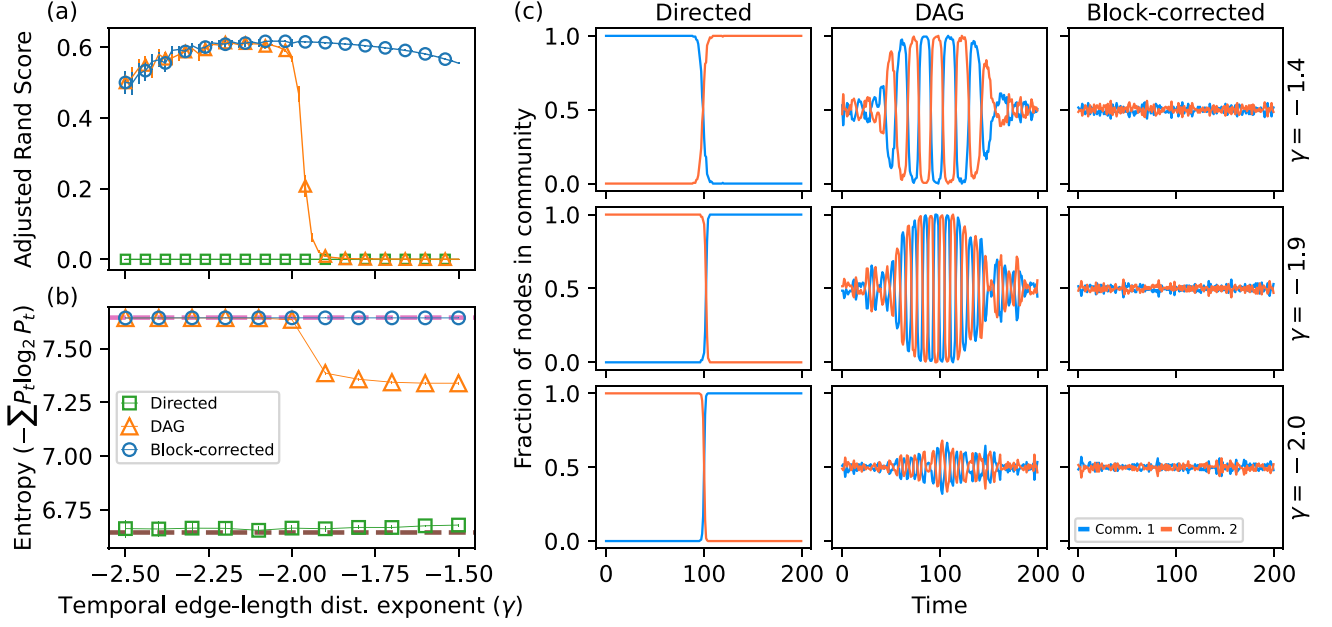


FIG. 5. Temporal homogeneity and labeling accuracy for the temporal planted partition model with power-law edge length distribution. Adjusted Rand scores between discovered communities and the true partitions (a) and the Shannon entropy of the discovered communities (b) for an ensemble of 50 synthetic networks with two planted communities over a range of exponents of power-law edge length distributions (see Eq. (12)). An adjusted Rand score of zero indicates the performance is no better than random chance, and an adjusted Rand score of 0.6 indicates misplacement of around 11.2% of $|V| = 40\,000$ nodes. Higher values of entropy indicate more even distribution of communities across time layers. The horizontal dashed pink line represents the analytical entropy for a uniform distribution $S = \log_2(t_{\max}) \approx 7.64$ while the horizontal dashed brown line corresponds to the analytical entropy for an equal vertical cut $S = \log_2(t_{\max}/2) \approx 6.64$. In panel (c) fraction of nodes in each of the two discovered communities is plotted across time layers for three networks with $\gamma = -1.4$, $\gamma = -1.9$, $\gamma = -2.0$. The synthetic networks are generated with a fixed number of nodes ($N = 200$) per time layer ($t_{\max} = 200$). Each node has an in-group degree $\langle k^{\text{in}} \rangle = 8$ and out-group degree $\langle k^{\text{out}} \rangle = 4$. Edges probabilities decrease as $P(\Delta t) \propto \Delta t^\gamma$. All three models are coded to find exactly two communities. Error bars in panels (a) and (b) show 95% confidence interval.

where the adjusted Rand score plummets for the DAG model. That is to say, as the edge length distribution becomes more heavy-tailed, the entropy level falls for the DAG modularity, while the block-corrected modularity remains unaffected.

This hypothesized vertical cut preferred by the directed modularity, dividing the network into two communities of older and younger nodes, is further substantiated by Fig. 5(c) which illustrates the fraction of nodes in the two discovered communities over the time layers. Three networks are realized from the aforementioned power-law generative network model [see Eq. (12)] each parametrized by exponent γ . With the block-corrected modularity, nodes in both communities are evenly distributed across time layers for all values of γ , as corroborated by the entropy value continuously remaining close to the analytical value for a homogeneous distribution of the community nodes across time as seen in Fig. 5(b). This more closely resembles the distribution of the nodes in the planted communities, where there are $N/2$ nodes of each community in each layer.

The DAG modularity, however, alternatively assigns the majority of the nodes of groups of time layers to one of the two communities for smaller values of γ . This behavior, which is not reminiscent of the planted (original) communities, can already be seen at $\gamma = -2.0$ and seems to aggravate for more heavy-tailed distributions. This, again, corroborates the observations of a drop in entropy in Fig. 5(b). As hypothesized before, the directed modularity makes a vertical cut for all

three values of γ visualized in Fig. 5(c). The cut seems to become more gradual as the magnitude gets smaller, i.e., the heavy tail characteristic of the edge length distribution augments.

IV. HEURISTICS FOR MODULARITY MAXIMIZATION

Maximizing the modularity function over all possible partitions of the network is computationally hard, an NP-hard problem [45], hence the development of an array of heuristic algorithms [9,46]. Here we opt for the spectral method [7], the so-called leading eigenvector method. Initially designed for finding exactly two groups of nodes in the network in the simplest setting, spectral bipartitioning can easily be extended to an arbitrary number of communities by repeatedly dividing the communities into two until no modularity gain is to be achieved by so doing.

While in many scenarios the leading eigenvector produces satisfactory outcomes [41], the resulting partition can nonetheless be improved by reassigning nodes to alternate communities. This has prompted the application of fine-tuning steps in many implementations of the spectral method [14,41] where nodes are moved among communities for possible gains in the modularity value. In Sec. IV C and Sec. IV B we present two slightly distinct, commonly used greedy fine-tuning algorithms, along with their vectorized implementation optimized for sparse representations.

A. Spectral partitioning

Given a network, finding two communities is equivalent to discovering a membership vector, $\mathbf{s}$, whose elements are community labels of $+1$ and -1 of all nodes, $\mathbf{s} = (s_1, \dots, s_{|V|})^\top$, $s_i \in \{+1, -1\}$. Modularity value in Eq. (1) can be thus rewritten as a function of membership vector $\mathbf{s}$ and modularity matrix $\mathbf{B}$, whose element B_{ij} is equal to $A_{ij} - P_{ij}$, and

$$Q = \frac{1}{2m} \mathbf{s}^\top \mathbf{B} \mathbf{s}. \quad (15)$$

Note that if rows and columns of $\mathbf{B}$ do not sum to zero, the modularity value Q for an undivided network would not be zero. In such a case sum of rows must be subtracted from the diagonal of $\mathbf{B}$ [47].

An approximate solution to this bipartitioning problem can be found by relaxing the integral constraint $s_i \in \{-1, +1\}$ to the real domain [48]. An effective heuristic for the problem is setting the values of s_i based on the leading eigenvector of $\mathbf{B}$ [41], that is, $s_i = \text{sgn}(u_i)$, $\forall i \in [n]$, where $\mathbf{u}$ is the leading eigenvector of $\mathbf{B}$.

The network is divided into two communities $\{C_1, C_2\}$ as above and subsequently, in an iterative fashion, the subgraph induced by community $C \in \{C_1, C_2\}$ is then bipartitioned into two smaller communities. The contribution to the modularity value of breaking up C into two partitions can be calculated as

$$\Delta Q = \frac{1}{2m} \tilde{\mathbf{s}}^\top \tilde{\mathbf{B}}^{(C)} \tilde{\mathbf{s}}, \quad (16)$$

where $\tilde{\mathbf{s}}$ is the membership vector of community C to be solved, and $\tilde{\mathbf{B}}^{(C)}$ a generalized modularity matrix of size $|C| \times |C|$ where $\tilde{B}_{ij}^{(C)} = B_{ij} - \delta_{ij} \sum_{k \in C} B_{ik}$ for $i, j \in C$ [7, 14]. From this point forward the superscript (C) for a matrix denotes a submatrix containing rows and columns in C . The bipartition of community C is executed only when the corresponding ΔQ is positive, indicating that performing the bipartitioning results in a gain in modularity value. The iterative bipartitioning halts when modularity value Q stops increasing, meaning ΔQ is nonpositive. In the very first step of the algorithm, community C corresponds to the entire network ($C = V$) in which case, assuming no self-loops are present, $\tilde{\mathbf{B}}^{(V)} = \mathbf{B}$ and $\tilde{\mathbf{s}} = \mathbf{s}$, resulting in ΔQ in Eq. (16) being equal to Q in Eq. (15).

In the more general case of directed networks, $\tilde{\mathbf{B}}^{(C)}$ is not necessarily symmetric. This asymmetry might cause issues when finding the leading eigenvector [41]. This can be resolved by restoring symmetry to the $\tilde{\mathbf{B}}^{(C)}$ by adding to its transpose [14, 41]. Therefore ΔQ becomes

$$\Delta Q = \frac{1}{4m} \tilde{\mathbf{s}}^\top (\tilde{\mathbf{B}}^{(C)} + (\tilde{\mathbf{B}}^{(C)})^\top) \tilde{\mathbf{s}}.$$

Computing the leading eigenvector of matrix $\tilde{\mathbf{B}}^{(C)} + (\tilde{\mathbf{B}}^{(C)})^\top$ can be done efficiently via the power method, which amounts to iteratively calculating

$$\mathbf{x}^{(i+1)} = (\tilde{\mathbf{B}}^{(C)} + (\tilde{\mathbf{B}}^{(C)})^\top) (\mathbf{x}^{(i)} / |\mathbf{x}^{(i)}|) \quad (17)$$

in every step i starting with an arbitrary vector $\mathbf{x}^{(0)}$ in the first step $i = 0$, until $\mathbf{x}^{(i)}$ converges to the leading eigenvector.

The modularity matrix is, however, dense and, for example, the explicit matrix-vector multiplication $\mathbf{B}\mathbf{x}$ requires $O(|V|^2)$ operations. Reference [7] shows how the matrix-vector multiplication $\mathbf{B}\mathbf{x}$ with a simpler definition of modularity can be

rewritten in vector notation to allow for fast computation. This derivation is not directly applicable to our block-corrected modularity, and we next derive a similar procedure for it. We can take advantage of the definition of B and the fact that typically adjacency matrix $\mathbf{A}$ is sparse to write it as $\mathbf{B}\mathbf{x} = \mathbf{A}\mathbf{x} - \mathbf{P}\mathbf{x}$. Recalling edge probability in Eq. (2), we define

$$\kappa_i^{\text{out/in}} = \frac{k_i^{\text{out/in}}}{K_i^{\text{out/in}}}$$

and write out element i of $\mathbf{P}\mathbf{x}$ as

$$\begin{aligned} (\mathbf{P}\mathbf{x})_i &= \sum_{j=1}^n P_{ij} x_j = \sum_{j=1}^n \kappa_j^{\text{in}} \kappa_i^{\text{out}} L_{t_i t_j} x_j \\ &= \kappa_i^{\text{out}} \sum_{t=1}^{t_{\max}} L_{t_i t} \sum_{j \in t} \kappa_j^{\text{in}} x_j \\ &= \kappa_i^{\text{out}} \sum_{t=1}^{t_{\max}} L_{t_i t} \langle \kappa^{\text{in}(t)}, \mathbf{x}^{(t)} \rangle. \end{aligned}$$

Writing this in terms of the element-wise multiplication operator $\odot$ gives

$$\mathbf{P}\mathbf{x} = \kappa^{\text{out}} \odot (\mathbf{L}\mathbf{v}^{\text{in}})_{t_{\max} \times 1 \mapsto |V| \times 1}, \quad (18)$$

where $\mathbf{v}_t^{\text{in}} = \langle \kappa^{\text{in}(t)}, \mathbf{x}^{(t)} \rangle$. Here, $(\mathbf{L}\mathbf{v}^{\text{in}})_{t_{\max} \times 1 \mapsto |V| \times 1}$ denotes vector $\mathbf{L}\mathbf{v}^{\text{in}}$ extended to $|V|$ elements by repeating each value in proportion to the number of nodes in the corresponding block. This can be done only if the nodes are sorted by blocks. If not, the nodes are relabeled to satisfy the condition.

Matrix $\mathbf{L}$ is of dimension $t_{\max} \times t_{\max}$, with $t_{\max}$ denoting the number of blocks in the network, and $\mathbf{v}^{\text{in}}$ is a $t_{\max} \times 1$ vector. Multiplication $\mathbf{L}\mathbf{v}^{\text{in}}$ is easily mapped from a $t_{\max}$ -dimensional space to an $|V|$ -dimensional one since all nodes in block t would have the same values for both $\mathbf{L}$ and $\mathbf{v}^{\text{in}}$. Calculating $\mathbf{B}\mathbf{x}$ using sparse matrix operations therefore has an overall time complexity of $O(m + |V| + t_{\max}^2)$.

This formulation, however, is only useful as it is for the first bipartitioning of the network. In the general case expressed in Eq. (17), we need to calculate $\tilde{\mathbf{B}}^{(C)}\mathbf{x}$ and $(\tilde{\mathbf{B}}^{(C)})^\top \mathbf{x}$ for any $C \subseteq V$ in a vectorized manner. As a first step, we can calculate $\mathbf{B}^{(C)}\mathbf{x} = \mathbf{A}^{(C)}\mathbf{x} - \mathbf{P}^{(C)}\mathbf{x}$. We can carefully modify Eq. (18) to derive the second term $\mathbf{P}^{(C)}\mathbf{x}$ by (i) subsetting occurrences of $\kappa^{\text{in/out}}$ to include only nodes in C , and (ii) extending term $(\mathbf{L}\mathbf{v}^{\text{in}})$ from $t_{\max} \times 1$ to $|C| \times 1$ instead of $|V| \times 1$.

Based on these building blocks, we calculate a vectorized notation for $\tilde{\mathbf{B}}^{(C)}\mathbf{x}$. The value for $\tilde{\mathbf{B}}^{(C)}\mathbf{x}$ can be calculated by decomposing it as

$$\tilde{\mathbf{B}}^{(C)}\mathbf{x} = \mathbf{B}^{(C)}\mathbf{x} - (\mathbf{B}^{(C)}\tilde{\mathbf{1}}) \odot \mathbf{x}, \quad (19)$$

where $\tilde{\mathbf{1}}$ is a column vector of ones. Vector $\mathbf{B}^{(C)}\tilde{\mathbf{1}}$, whose i th element is simply the sum of i th row of $\mathbf{B}^{(C)}$, can be efficiently calculated using the same method described in Eq. (18). This results in

$$\tilde{\mathbf{B}}^{(C)}\mathbf{x} = \mathbf{A}^{(C)}\mathbf{x} - \mathbf{A}^{(C)}\tilde{\mathbf{1}} \odot \mathbf{x} - \mathbf{P}^{(C)}\mathbf{x} + \mathbf{P}^{(C)}\tilde{\mathbf{1}} \odot \mathbf{x}. \quad (20)$$

$\tilde{\mathbf{B}}^\top \mathbf{x}$ can be similarly calculated through a substitution of $\tilde{\mathbf{B}}$ with $\tilde{\mathbf{B}}^\top$ in Eq. (20). Notably, following the same pattern as derivation of Eq. (18), for $\mathbf{P}^\top \mathbf{x}$ we get

$$\mathbf{P}^\top \mathbf{x} = \kappa^{\text{in}} \odot (\mathbf{L}^\top \mathbf{v}^{\text{out}})_{t_{\max} \times 1 \mapsto n \times 1}. \quad (21)$$

Note that the power iteration method as described converges to the eigenvector corresponding to the largest eigenvalue by absolute value, while the bipartitioning heuristic requires finding the sign of the eigenvector corresponding to the largest eigenvalue. If the largest eigenvalue by absolute value (λ) returned by the power iteration method is negative, the power iteration process can be repeated for the matrix $\tilde{\mathbf{B}}^{(C)} + (\tilde{\mathbf{B}}^{(C)})^\top - \lambda \mathbf{I}$ to find the latter.

B. Split-Louvain fine-tuning

The algorithm, based on the implementation in Ref. [14], can optimize the outcome of a single bipartitioning step from an initial heuristic by calculating the effect of moving each node to the opposite community on modularity and performing the most beneficial move. Our implementation enhances the overall performance by minimizing the amount of redundant calculations and maximizing the use of vectorized operations.

Every bipartition is characterized by membership vector $\mathbf{s}$ where $s_i \in \{+1, -1\}$. The change in Q for a hypothetical change in community membership of node k we calculate to be $[-4s_k(\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)\mathbf{s} + 4(\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)_{kk}]/4m$. This can be derived by calculating the modularity for the new membership vector $\mathbf{s}' = \mathbf{s} - 2s_k\mathbf{e}_k$ where $\mathbf{e}_k$ is the k th standard basis vector.

Note that, for readability we will describe the derivations in this section for the case of bipartitioning the entire network ($C = V$). For the more general case of fine-tuning a bipartition of $C \subseteq V$, we can simply replace $\mathbf{s}$ with $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{B}}$ with $\tilde{\mathbf{B}}^{(C)}$.

In a similar fashion, we can compute a vector of all possible improvements $\mathbf{d}$ where d_k is the change in ΔQ for the hypothetical movement of node k to the other community:

$$\mathbf{d} = \frac{1}{m}[\text{diag}(\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top) - \mathbf{s} \odot (\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)\mathbf{s}], \quad (22)$$

where $\odot$ indicates the Hadamard product. $(\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)\mathbf{s}$ can be efficiently computed using the sparse matrix operations discussed in Sec. IV A.

Upon selecting the node with the largest value in vector $\mathbf{d}$ ($k = \text{argmax} \mathbf{d}$), a running variable for the accumulation of all ΔQ values is incremented by d_k , after which the improvements vector $\mathbf{d}$ is recomputed and the selecting of a node is repeated for the updated improvements vector $\mathbf{d}$. This is repeated until no modularity-increasing moves remain on unmoved nodes. For compatibility reasons with the implementation in Ref. [14], we limit the maximum number of movements for every node to one.

We find it more efficient to apply the effect of changing the membership of node k directly to $\mathbf{d}$ instead of recomputing the entire vector $\mathbf{d}$ from scratch in every iteration:

$$\mathbf{d}' = \mathbf{d} + \frac{2}{m}\mathbf{s}_k\mathbf{s} \odot (\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)_k, \quad (23)$$

with the exception element k where $\mathbf{d}'_k = -\mathbf{d}_k$. This computation avoids the repetition of the sparse matrix-vector product in Eq. (22).

C. Final-Louvain fine-tuning

In addition to the split-Louvain fine-tuning, we implement another method of fine-tuning akin to phase one of the

Louvain algorithm [9], that is applicable to a partition with an arbitrary number of communities. Unlike the split-Louvain fine-tuning nodes can be moved between communities multiple times, so long as the corresponding ΔQ values are positive. While in split-Louvain fine-tuning, the best possible move among all nodes is selected in each iteration, in the multicomunity final-Louvain fine-tuning, we perform the most beneficial move for every node in an arbitrary order of nodes in each round and repeat it until no modularity-gaining move remains for any of the nodes. Unlike the split-Louvain fine-tuning which is applied after every bipartition, we only perform the final-Louvain fine-tuning on the final output partition of the community detection algorithm.

In order to leverage the sparse matrix operations outlined in Sec. IV A, we define matrix $\mathbf{D}$ of dimension $n \times b$, n being the number of nodes and b the number of communities in the partition, where element D_{kc} is defined to be the modularity contribution of node k to community c . This is by definition equal to $(\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)\mathbf{G}$ where $\mathbf{G} \in \{0, 1\}^{n \times b}$ denotes the community-membership matrix whose element G_{kc} is one if node k belongs to community c and zero otherwise. Column j in $\mathbf{D}$ is denoted by $\mathbf{D}_{*j}$ that equates $(\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)\mathbf{G}_{*j}$ where likewise $\mathbf{G}_{*j}$ is column j of matrix $\mathbf{G}$, and this matrix-vector multiplication can be computed efficiently as described in Sec. IV A. Thus, this allows for an efficient populating of matrix $\mathbf{D}$.

To decide whether node k be moved to a different community, a ΔQ_k is computed, that is the gain in modularity when detaching node k from its original community b_k , subtracted from that of moving it to a candidate community c where the gain is maximum across all possible communities. These two quantities are immediately accessed via matrix $\mathbf{D}$:

$$\Delta Q_k = D_{kc} - D_{kb_k}. \quad (24)$$

Node k is moved to community c if this value is larger than zero. In the case of a move, the k th element of the respective membership vectors in $\mathbf{G}$ are updated accordingly, and so is matrix $\mathbf{D}$:

$$\begin{aligned} D_{kb_i} &= D_{kb_i} - (\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)_{*k}, \\ D_{kc} &= D_{kc} + (\tilde{\mathbf{B}} + \tilde{\mathbf{B}}^\top)_{*k}. \end{aligned}$$

This update rule allows for an efficient recomputation of the changed columns of $\mathbf{D}$ without having to recalculate the entire matrix.

V. REAL-WORLD CITATION DATA: OPENALEX

To assess the performance and scalability of our modularity method, we apply it to a set of large real-world citation networks varying in size and academic field. For our purposes, we rely on the extensive OpenAlex citation dataset [49]. Publications in the OpenAlex database are accompanied by their scientific fields, with different levels of granularity, ranging from topic and subfield to scientific domain, as well as information about publication venues, author affiliations, and funding sources.

TABLE I. Community detection results of the block-corrected modularity-based algorithm on a list of OpenAlex networks, along with their respective sizes and academic fields. “Duration” specifies running time, “Comms.” specifies the total number of communities and the smallest number of communities that contain at least 90% of the nodes in parentheses. The split-Louvain and final-Louvain are abbreviated as sLouv and fLouv respectively. Results are averaged on an ensemble of 100 independent runs.

		Veterinary	Dentistry	Energy	Economics	Psychology	Neuroscience	Phys. and astro.	CS
Nodes		50.6K	126.8K	311.6K	353.1K	781.3K	890.1K	1.1M	1.2M
Edges		470.5K	1.7M	7.9M	4.1M	12.6M	21.0M	23.1M	14.0M
Comms.	No fine-tune	7.2	5.5	8.6	37	12	13	20	14
	sLouv	53	44	25	46	38	31	40	58
	fLouv	6.9	5.5	8.5	36	13	14	20	16
	Both	51	42	24	46	36	30	41	59
Large comms.	No fine-tune	3.1	1.0	1.0	15	2.5	5.8	6.1	5.1
	sLouv	24	15	8.8	15	16	12	18	21
	fLouv	3.3	3.6	4.8	13	5.3	7.5	9.4	7.3
	Both	23	16	8.4	13	15	13	18	21
Q	No fine-tune	0.487	0.369	0.346	0.228	0.287	0.327	0.394	0.228
	sLouv	0.824	0.729	0.682	0.657	0.659	0.64	0.791	0.744
	fLouv	0.659	0.661	0.632	0.657	0.62	0.638	0.73	0.641
	Both	0.834	0.751	0.699	0.706	0.692	0.671	0.807	0.769
Duration	No fine-tune	93s	66s	8.6m	30m	30m	39m	2.0h	57m
	sLouv	14m	19m	37m	58m	2.7h	3.0h	6.5h	7.6h
	fLouv	1.9m	3.3m	19m	70m	3.7h	3.5h	6.6h	6.5h
	Both	13m	20m	38m	62m	3.5h	4.2h	7.4h	9.8h

A. Field networks

We select eight networks ranging in size and scientific discipline as summarized in Table I. Nodes in the networks are articles whose publication dates span from the beginning of the year 2000 to 2025, removing all whose in-degrees are less than 10. We bin nodes in time by year, resulting in 24 time layers in total. Since the networks need not be directed acyclic, we allow for edges within time layers or against the arrow of time.

For each network, we run our community detection method and report the mean and the standard error of the mean of the number discovered of communities (comms.), the number of discovered communities comprising in total more than 90% of the network (Large comms.), modularity value (Q), and runtime (Duration) across 50 runs. We execute additional fine-tuning steps in two slightly different forms, split-Louvain and final-Louvain. Split-Louvain fine-tuning is performed following every bipartition step and is akin to that performed in [41,47]. Final-Louvain fine-tuning is the multicommodity counterpart to split-Louvain fine-tuning applied to the discovered partition by the community detection method, resembling the first phase of the Louvain algorithm [9]. To achieve improved runtime we leverage the vectorized matrix operations described in Sec. IV A in computing the changes in modularity values for both fine-tuning processes, the details of which we presented in Sec. IV B and Sec. IV C.

Our implementation of spectral partitioning, detailed in Sec. IV A, involves computing leading eigenvectors by way of the power iteration method, parameterized by the tolerance within which the leading eigenvalue is converged. Tolerance affects the runtime and dictates the precision of the converged eigenvector. Producing Table I we observe that a tolerance of 10^{-10} is a reasonable enough choice, yielding eigenvalues containing six to eight significant decimal digits on average

across all runs. There are limitations imposed by the tolerance parameter that account for the nonzero errors of the number of communities in Table I. What we observe first-hand on both empirical and synthetic networks is the existence of nodes whose values in the leading eigenvector are zero or notably small. The signs of these values can flip across runs, resulting in different group assignments for nodes, and subsequently different partitionings of the network altogether. As the spectral partitioning process is iterative, these alternative group assignments can compound over subsequent iterations, resulting in larger deviations.

As we illustrate in Appendix B, the tolerance value required for achieving a specific level of accuracy in a single bipartitioning step is intrinsically linked with the network structure. We show empirically, using synthetic network models, that the required tolerance scales with network properties such as the number of temporal layers as well as the average temporal edge lengths. Such dependence on the network structure, especially in empirical settings where errors can accumulate over the many iteration steps of the spectral partitioning, can incur errors on the final output and therefore lead to fluctuations across runs.

For all networks in Table I, the split-Louvain fine-tuning results in larger modularity values than the final-Louvain. One way to explain this is by the same phenomenon of error accumulation that, we posit, accounts for the fluctuations across community detection outcomes of a network. Split-Louvain fine-tuning is executed after every bipartition and thus diminishes the error that might compound over all subsequent iterative steps. In contrast, final-Louvain fine-tuning is performed only on the final outcome partition. Although applying both types of fine-tuning only marginally improves the results of split-Louvain, it typically adds a negligible amount of extra computation time compared to using split-Louvain alone.

TABLE II. Community detection results of the block-corrected modularity-based algorithm on a list of OpenAlex networks consisting of two subfields. Results are averaged on an ensemble of 100 independent runs.

		Biochem.–clinical biochem.	Environmental chem.–analytical chem.	Environmental chem.–ocean eng.	Environmental eng.–pollution
Nodes		73.8K	179.3K	209.2K	333.2K
Edges		951.6K	2.6M	2.6M	5.4M
Metadata Q		0.428	0.451	0.432	0.420
Q	No fine-tune	0.186	0.140	0.138	0.143
	sLouv	0.429	0.434	0.438	0.462
	fLouv	0.450	0.437	0.450	0.453
$Q_{\text{randomized}}$	No fine-tune	0.048	0.032	0.032	0.026
	sLouv	0.118	0.113	0.120	0.106
	fLouv	0.120	0.115	0.121	0.107
Duration	No fine-tune	3.9s	14s	14s	32s
	sLouv	29s	5.4m	5.3m	21m
	fLouv	51s	4.5m	9m	19m
		Pharmacology–social psych.	Polymers and plastics–biomater.	Aerosp. eng.–mech. of material	Physiology–cardio. med.
Nodes		389.2K	407.5K	491.2K	772.0K
Edges		5.1M	7.9M	5.4M	13.4M
Metadata Q		0.461	0.370	0.427	0.417
Q	No fine-tune	0.165	0.160	0.0591	0.287
	sLouv	0.440	0.445	0.457	0.449
	fLouv	0.468	0.417	0.449	0.454
$Q_{\text{randomized}}$	No fine-tune	0.027	0.033	0.025	0.022
	sLouv	0.117	0.097	0.128	0.102
	fLouv	0.118	0.099	0.129	0.103
Duration	No fine-tune	32s	43s	35s	1.8m
	sLouv	27m	28m	42m	90m
	fLouv	25m	25m	37m	56m

Also notable in the result is that the application of the split-Louvain method (or the simultaneous application of both fine-tuning methods) often results in a larger number of detected communities, as well as a notable increase in the number of large detected communities, whereas the spectral method often results in only one or very few large communities. This, combined with the noteworthy gain in modularity, suggests that an effective fine-tuning step is often necessary to alleviate the shortcomings of the spectral method.

B. Subfield networks and metadata alignment evaluation

In a similar fashion to field networks, subfield networks can be constructed, containing scientific works sharing one or more closely related subfields. Subfields are the second-most granular level of the “domain, field, subfield, topic” system that tags publications. Limiting the number of subfields to two, we build networks where only a single bipartition can not only evade the complexities of the recursive spectral bipartitioning, but additionally provides a simpler setting to inspect the alignment of the discovered communities with the two subfields.

To do this, we construct eight pairs of subfield networks where every node has at least one of the select subfields in its list of all subfields it belongs to. Our criterion for choosing the select pairs of subfields is such that the pairs are not too overlapping, nor completely disparate both in terms of subject

matter and network connectivity. We furthermore ensure the subfield communities are comparable in size.

In addition to the information reported in Table I, we include the modularity of the bipartition based on metadata labels (metadata Q) and the modularity of the best bipartition of a randomized realization of the same network drawn from the block null model. The significant gap between Q and $Q_{\text{randomized}}$ shows that the community structure present in the network is well beyond the detectability limit of this model. See Appendix E for more information on this issue.

On every subfield network, we perform exactly one bipartition followed by either the split-Louvain or the final-Louvain fine-tuning. As shown in Table II the final-Louvain fine-tuning outperforms its split-Louvain counterpart by a slight margin, possibly due to allowing for multiple movements for every node. Table II also shows that for all networks, the modularity value of the outcome partition with no fine-tuning is significantly smaller than that of the metadata partition. This shows that as a heuristic approximation to the modularity maximization problem [48], the leading eigenvector spectral bipartitioning method can strongly benefit from additional optimization steps. This also reaffirms the idea of applying fine-tuning at every bipartitioning step, as opposed to a single application of a fine-tuning step after partitioning.

Unlike synthetic networks, there are generally no planted communities in real-world networks, nor is the true generative process known [50]. Absent a ground-truth partition, metadata, typically node attributes, are often treated as such.

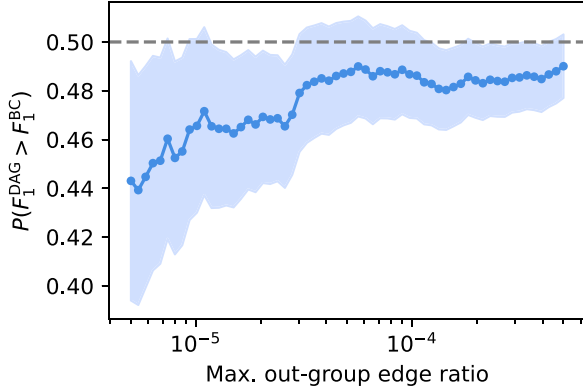


FIG. 6. Probability of the DAG model producing outcomes closer to the metadata compared to the block-corrected, as measured by F_1 scores, decreases with the minimum threshold on separatedness of the metadata communities. The F_1 scores are produced by performing spectral bipartitioning, followed by split-Louvain fine-tuning (see Sec. IV B). Note that the separatedness on the horizontal axis is plotted cumulatively. Shaded area shows 95% confidence interval.

The assumption, nevertheless, that metadata are inherently linked with the community structure cannot be readily made [50–52]. Alongside the deep-seated reasons that metadata do not always correlate with network structure, the process of generating metadata labels can be noisy and/or biased. In the OpenAlex database, for instance, citation relationship is a feature for the classifier that produces the node labels. With said caveats in mind, for randomly selected networks we evaluate whether partitions found by our model and the DAG [14] bear any resemblance to that of the metadata partitions.

For a random sample of 200 OpenAlex subfields, we construct all possible networks containing exactly two subfields. We exclude networks where one subfield contributes less than 3% of the largest weakly connected component, totaling 13 361 networks. The networks thus have two innate communities with varying degrees of separatedness. We then perform a spectral bipartitioning followed by the greedy split-Louvain fine-tuning step. We thereafter measure how much the fine-tuned membership vectors differ from the metadata subfield bipartition under the DAG and block-corrected modularities.

For each modularity, we compute a global F_1 score (based on the total number of true positives, false negatives, and false positives) indicating how well they recover the two original subfields, to inspect how the F_1 scores of the two models compare for networks with different levels of separatedness, defined as the fraction of out-group edges to all edges. To illustrate this, in Fig. 6, we plot the probability that block-corrected performs better than DAG for different maximum thresholds of separatedness. As the separatedness of the two subfields accentuates, the labeling advantage of block-corrected compared to DAG increases. This shows that the block-corrected modularity is more effective at recovering the partitions imposed by the omitted (unknown) subfield metadata from the network structure compared to DAG, especially in networks where innate communities imposed by subfields are more distinct. Here the choice of F_1 score is justified since the problem at hand is a binary classification.

However, in a more general setting, a more established metric for partition comparison such as the normalized mutual information is more fitting. We additionally repeated this analysis using normalized mutual information as the measure of performance with similar results, presented in Appendix D.

VI. CONCLUSION AND DISCUSSION

Complex networks often include node attributes that can impact their community structure. We showed that modularity-based community detection methods can fail to reveal connectivity patterns driven by unknown attributes if known attributes and their effects on connectivity are not explicitly assimilated into the modularity framework. We introduced a modularity measure with its associated null model, which isolates the effect of known attributes to uncover underlying structures shaped by unknown ones. This premise runs counter to most literature on community detection methods, which typically treat the attributes as ground truth or leverage them for improved results in terms of their alignment with the discovered structures.

We demonstrated the effectiveness of our proposed modularity on select temporal network models, varying in temporal connectivity patterns. We took temporal networks, specifically citation networks, as our primary application as they display an innate block structure due to the constraints imposed by the arrow of time, as well as the heavy-tail citation patterns they may exhibit. Our proposed block-corrected modularity, a variant adaptable to arbitrary temporal connectivity distributions, addresses the limitations of existing modularity-based approaches when dealing with networks characterized by the aforementioned heavy-tail edge decay.

Using the OpenAlex dataset we further showed the efficacy of our method on a range of citation networks representing select fields, and similarly on a wide array of subfield networks consisting of exactly two subfields. In the absence of an all-encompassing ground truth community structure [50] and given the complex relationship of available metadata to structural communities [53,54], it remains a challenge to assess and compare the quality of detected communities across methods in real-world settings. Construction of the subfield networks allowed for a more controlled experiment for metadata evaluation where we could also control for the amount of separatedness between the two subfields.

We also outlined some shortcomings of the commonly employed leading eigenvector heuristic, demonstrating in simulations how it produces suboptimal outcomes under certain conditions. In such circumstances, methods such as greedy fine-tuning algorithms as a postprocessing step can notably improve the final results. The implementation and the necessary instructions for reproducing the results of the paper are provided in Ref. [55].

More fundamentally, community detection methods based on modularity maximization suffer from a systematic flaw, namely, that in finite networks, a modularity maximization method reliably finds partitions with nonzero modularity in realizations of its own null model [57,58]. In Appendix E we investigate the performance of our block-corrected modularity in the face of randomized realizations drawn from the block null model. Bayesian inference methods for community

detection, for example those based on exponential random graph models [59], can be inherently resistant to this issue at the cost of implementation complexity. These methods can be adapted to discount the effect of known attributes on connectivity, similar to what we discussed in this paper.

Networks are often accompanied by not one, but a multitude of known attributes. The method presented here allows us to incorporate any subset of the known attributes into the null model as necessitated by the research question. We then treat the resulting communities as what mirrors the unknown attributes. Likewise, the resulting community structure is often affected by a combination of multiple unknown attributes. The distilling of the discovered structures into their attribute constituents lies beyond the capabilities of our model. We treat the structures we find as an all-encompassing outcome of what is likely an amalgam of unknown attributes, ones we cannot disentangle within our model definition.

ACKNOWLEDGMENTS

We acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LUMI, hosted by CSC (Finland) and the LUMI consortium. We also acknowledge the computational resources provided by the Aalto Science-IT project. M.K. acknowledges Grant No. 349366 from the Research Council of Finland.

DATA AVAILABILITY

Data that are not openly available are available upon reasonable request from the authors. The dataset used in the production of this work is available through the OpenAlex database [55,56].

APPENDIX A: COMPUTING MODULARITY VALUES ACROSS KNOWN AND UNKNOWN ATTRIBUTES

Under the framework discussed in Sec. III A, the modularity value of the partition splitting the network into two groups of nodes with attributes (labels) $x = 0$ and $x = 1$ denoted by Q_x is

$$Q_x = \frac{1}{m} \left(\sum_{\substack{i,j \\ x_i=x_j}} A_{ij} - \sum_{\substack{i,j \\ x_i=x_j}} \tilde{p}_{ij} \right), \quad (\text{A1})$$

where $\tilde{p}_{ij}$ is the edge probability between nodes i and j under a null model. Modularity value of partition splitting the network into nodes with attribute $y = 0$ and $y = 1$, Q_y is defined similarly as above, with x_i (resp. x_j) replaced by y_i (resp. y_j).

In what follows, we derive the modularity values Q_x and Q_y of the block-corrected modularity and the directed modularity.

Under the block null model as defined in Eq. (2), the term $k_i^{\text{out}}/K_{x_i}^{\text{out}}$ in $\tilde{p}_{ij}$ becomes $2/|V|$ since there are $|V|/2$ nodes in community x_i , the known attribute. Thus, $\tilde{p}_{ij}$ transforms into $4/|V|^2 L_{x_i x_j}$ where $L_{x_i x_j}$ is the number of edges between nodes with label x_i and nodes with label x_j .

We assume edge densities in each community of $x = 0$ and $x = 1$ are equal, as well as between the two, that is, $L_{00} = L_{11}$ and $L_{01} = L_{10}$. Node i can be connected to a node whose label

for y could be the same with probability p_1^y , or different with probability p_0^y . There are $|V|/4 - 1$ possible nodes tagged with the same y as node i and $|V|/4$ with a different value for y . Since there are $|V|/2$ possible nodes with the same value for x as node i we get

$$\begin{aligned} L_{00} = L_{11} &= \frac{|V|}{2} p_1^x \left[\left(\frac{|V|}{4} - 1 \right) p_1^y + \frac{|V|}{4} p_0^y \right] \\ &\approx \frac{|V|^2}{8} p_1^x (p_1^y + p_0^y). \end{aligned}$$

Following a similar logic, $\sum_{\substack{i,j \\ x_i=x_j}} A_{ij}$ is approximately equal to $|V|^2 p_1^x (p_1^y + p_0^y)/4$. We substitute this and $\tilde{p}_{ij} = p_1^x (p_1^y + p_0^y)/2$ into Q_x to get

$$\begin{aligned} m Q_x^{\text{bc}} &= \frac{|V|^2}{4} p_1^x (p_1^y + p_0^y) - \sum_{\substack{i,j \\ x_i=x_j}} \frac{p_1^x (p_1^y + p_0^y)}{2} \\ &= \frac{|V|^2}{4} p_1^x (p_1^y + p_0^y) - p_1^x (p_1^y + p_0^y) \frac{|V|}{2} \left(\frac{|V|}{2} - 1 \right) \\ &\approx 0. \end{aligned}$$

This approximation holds for $|V| \gg 0$.

Using our proposed modularity, the partition dividing the network with respect to the known attribute x thus provides no improvement over a single partition covering the entire network in terms of modularity maximization, making such bipartitioning undesirable.

Computing Q_y^{bc} , the term $\sum_{\substack{i,j \\ y_i=y_j}} A_{ij}$ is the same as its Q_x^{bc} equivalent, with x and y replaced:

$$Q_y^{\text{bc}} = \frac{1}{m} \left(\frac{|V|^2}{4} p_1^y (p_1^x + p_0^x) - \sum_{\substack{i,j \\ y_i=y_j}} \tilde{p}_{ij} \right). \quad (\text{A2})$$

Elements $L_{01} = L_{10}$ of matrix $\mathbf{L}$ are equal to $|V|/2(|V|p_0^x/4 + p_1^y + |V|p_0^y p_0^x/4) = |V|^2 p_0^x (p_0^y + p_1^y)/8$ since there are $|V|/2$ nodes in each community with respect to x and for every node the destination node is in the same community with respect to y with probability p_1^y and probability p_0^y of being in different communities with respect to y .

We can write out $\sum_{\substack{i,j \\ y_i=y_j}} \tilde{p}_{ij}$ as

$$\sum_{\substack{i,j \\ y_i=y_j}} \tilde{p}_{ij} = \sum_{i=1}^{|V|} \sum_{j: y_j=y_i} \tilde{p}_{ij}. \quad (\text{A3})$$

The sum $\sum_{j: y_j=y_i} \tilde{p}_{ij}$ is divided into cases where x_i and x_j are equal or otherwise, i.e.,

$$\left(\frac{|V|}{4} - 1 \right) \left(\frac{4}{|V|^2} L_{00} \right) + \frac{|V|}{4} \frac{4}{|V|^2} L_{01} \approx \frac{1}{|V|} (L_{00} + L_{01}).$$

Plugging this in Eq. (A3) we get

$$\sum_{\substack{i,j \\ y_i=y_j}} \tilde{p}_{ij} = \sum_{i=1}^{|V|} \frac{1}{|V|} (L_{00} + L_{01}) = L_{00} + L_{01}. \quad (\text{A4})$$

We substitute the right-hand side of Eq. (A3) in Eq. (5) and plug in $2(L_{00} + L_{01}) = |V|^2(p_1^x + p_0^x)(p_1^y + p_0^y)/4$ for m to arrive at

$$Q_y^{\text{bc}} = \frac{1}{2} \frac{p_1^y - p_0^y}{p_1^y + p_0^y}. \quad (\text{A5})$$

Modularity value Q_y^{bc} in Eq. (A5) increases as the probability of two nodes having the same values for label y approaches 1 ($p_1^y \rightarrow 1$) and the probability of two nodes having different values for y approaches 0 ($p_0^y \rightarrow 0$). In other words, the more separated the two communities of $y = 0$ and $y = 1$, the larger the value for Q_y^{bc} , thus rendering such a division a more desirable one.

The probability of an edge in the directed model is defined as $\tilde{p}_{ij} = \frac{k_i^{\text{out}} k_j^{\text{in}}}{m}$, solely dependent on in- and out-degrees of the two nodes of an edge. This indicates modularity values Q_x^{dir} and Q_y^{dir} are calculated symmetrically, i.e., Q_y^{dir} can be derived through replacing p^x with p^y in Q_x^{dir} . Deriving Q_x^{dir} alone should thus suffice. The average in- or out-degree would be $|v|/4(p_0^x p_0^y + p_0^x p_1^y + p_1^x p_0^y + p_1^x p_1^y)$ which we refer to as $|V|^2 p^{xy}/4$ for brevity, and the number of edges m is computed similarly as before, equal to $|V|^2(p_1^x + p_0^x)(p_1^y + p_0^y)/4 = |V|^2 p^{xy}/4$. Modularity value Q_x^{dir} is thus

$$\begin{aligned} Q_x^{\text{dir}} &= \frac{1}{m} \frac{|V|^2}{4} p_1^x (p_1^y + p_0^y) - \frac{1}{m^2} \sum_{\substack{i,j \\ x_i = x_j}} \frac{|v|^2}{16} (p^{xy})^2 \\ &= \frac{1}{m} \frac{|V|^2}{4} p_1^x (p_1^y + p_0^y) - \frac{1}{m} \sum_{\substack{i,j \\ x_i = x_j}} \frac{p^{xy}}{4} \\ &\approx \frac{1}{m} \frac{|V|^2}{4} p_1^x (p_1^y + p_0^y) - \frac{1}{m} \frac{|V|^2}{2} \frac{p^{xy}}{4} \\ &= \frac{1}{2} \frac{p_1^x - p_0^x}{p_1^x + p_0^x}. \end{aligned}$$

Modularity value Q_y^{dir} therefore is

$$Q_y^{\text{dir}} = \frac{1}{2} \frac{p_1^y - p_0^y}{p_1^y + p_0^y}.$$

It might be argued that a bipartitioning based on known attribute x by the directed modularity might still help in recovering the unknown attribute, if the modularity is then allowed to subsequently bipartition the two discovered communities into four communities, based on values of both x and y . While this four-community solution certainly works for the example presented in Fig. 2, it is not generally the case. Starting from Eq. (1), calculating the modularity of the four-community solution with directed and block-corrected modularity similar to the derivations above, we get

$$Q_{xy}^{\text{dir}} = \frac{p_1^y p_1^x}{(p_1^y + p_0^y)(p_1^x + p_0^x)} - \frac{1}{4} \quad (\text{A6})$$

and

$$Q_{xy}^{\text{bc}} = \frac{1}{2} \frac{p_1^x (p_1^y - p_0^y)}{(p_1^y + p_0^y)(p_1^x + p_0^x)}. \quad (\text{A7})$$

When the inequalities

$$Q_y^{\text{bc}} > Q_x^{\text{bc}} \text{ and } Q_y^{\text{bc}} > Q_{xy}^{\text{bc}} \quad (\text{A8})$$

are held, the modularity maximization method using the proposed block-corrected modularity will successfully recover hidden attribute y , but does not continue to needlessly split the found communities based on (x, y) . Since $Q_x^{\text{bc}} \approx 0$, the first part of the inequality simplifies into $p_1^y > p_0^y$, i.e., that the hidden attribute must be assortative. The second inequality translates to

$$\frac{1}{2} \frac{p_1^y - p_0^y}{p_1^y + p_0^y} > \frac{1}{2} \frac{p_1^x (p_1^y - p_0^y)}{(p_1^y + p_0^y)(p_1^x + p_0^x)},$$

from which we obtain $p_0^x > 0$.

Similarly, when inequalities

$$Q_x^{\text{dir}} > Q_y^{\text{dir}} \text{ and } Q_x^{\text{dir}} > Q_{xy}^{\text{dir}} \quad (\text{A9})$$

are held, the modularity maximization method using the directed modularity will not split the initial (erroneous) bipartitioning based on the known attribute into four communities based on (x, y) values. From the first inequality, we have

$$\frac{1}{2} \frac{p_1^x - p_0^x}{p_1^x + p_0^x} > \frac{1}{2} \frac{p_1^y - p_0^y}{p_1^y + p_0^y},$$

which simplifies to $p_1^x p_0^y > p_0^x p_1^y$. As we assumed $p_1^y > p_0^y$ before, the first inequality is held if known attribute x is more strongly assortative than unknown attribute y . From the second inequality, we obtain

$$\frac{1}{2} \frac{p_1^x - p_0^x}{p_1^x + p_0^x} > \frac{p_1^y p_1^x}{(p_1^y + p_0^y)(p_1^x + p_0^x)} - \frac{1}{4},$$

which leads to $p_1^x (3p_0^y - p_1^y) > p_0^x (p_1^y + p_0^y)$. Since the left-hand side is positive, this implies that $3p_0^y - p_1^y > 0$ should hold and

$$\frac{p_1^x}{p_0^x} > \frac{p_1^y + p_0^y}{3p_0^y - p_1^y}. \quad (\text{A10})$$

For example, with $p_1^x = 0.9$, $p_0^x = 0.05$, $p_1^y = 0.35$, and $p_0^y = 0.25$, we get $Q_x^{\text{dir}} = 0.45$, $Q_x^{\text{bc}} \approx 0$, $Q_y^{\text{dir}} = Q_y^{\text{bc}} = 0.084$, $Q_{xy}^{\text{dir}} = 0.30$, and $Q_{xy}^{\text{bc}} = 0.078$, which satisfies both inequalities above. In this scenario, directed modularity splits the network into two communities based on the known attribute x but does not proceed to split it again into four communities based on (x, y) , while block-corrected modularity correctly identifies the unknown attribute y and does not further divide the network.

APPENDIX B: POWER ITERATION TOLERANCE SCALING WITH NETWORK PARAMETERS

Spectral clustering through calculating exact values for the leading eigenvectors of the dense modularity matrix is a computationally expensive task. Employing the power iteration method for finding the leading eigenvector can not only help utilize sparse matrix operations but also provide a vehicle for vectorisation, which in turn significantly improves computation time. This is described in detail in Sec. IV A.

Different implementations of the power iteration algorithm rely on different convergence criteria. Often the algorithm

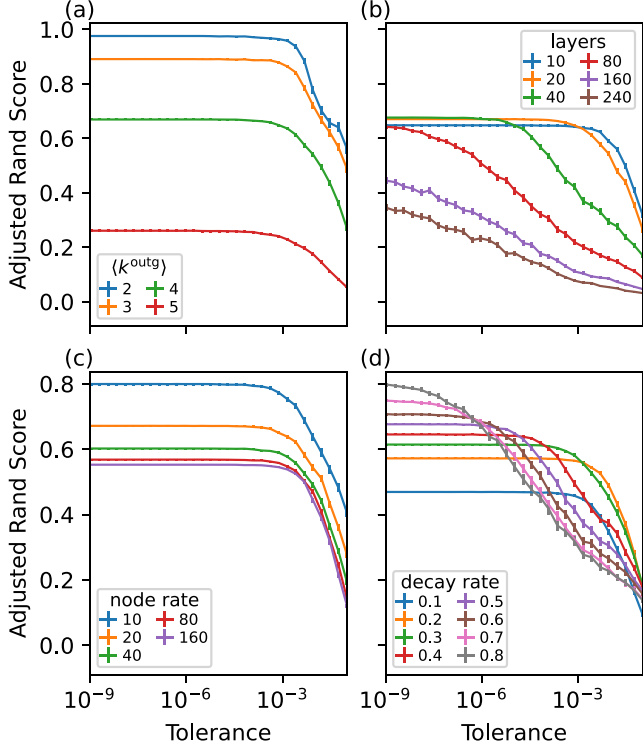


FIG. 7. Quality of bipartitioning proposed by the spectral method for random networks containing two planted communities as a function of the power iteration tolerance. Average in-group degree (a) and number of nodes in each time layer (c) do not change the required tolerance threshold, while an increase in the number of time layers (b) or higher prevalence of longer edges (d) necessitates smaller values for tolerance. Unless stated in the plot, the generated networks have 20 nodes per time layer, 20 time layers [40 for (d)], in-group degree 8, out-group degree 4, and an edge decay rate of 0.5.

halts when the estimated eigenvalue remains within a certain threshold of that of the previous iteration. This threshold is here denoted as tolerance. A smaller value for tolerance is a stricter condition on convergence for the eigenvector.

The choice for the tolerance value affects both runtime and the precision of the converged eigenvector. It thus raises the question of what value we must choose for tolerance. We, moreover, observe in our study of empirical networks in Sec. V that fluctuations in the number of communities occur across multiple runs of the community detection process, indicating variance among the power iteration outcomes across runs. Motivated by these considerations, we attempt to study the interplay between tolerance and the structural properties of the network and the impact it may have on the community detection outcomes.

Our goal is to explore the relationship between various network parameters and the required tolerance value to obtain a certain level of accuracy. The ensemble of networks is drawn from the temporal network model with two planted communities and exponentially distributed edge lengths, details of which are discussed in Sec. III B.

Illustrated in Fig. 7, it is evident that the tolerance has a clear relationship with the number of temporal layers. As networks become temporally longer [Fig. 7(b)] or as longer edges

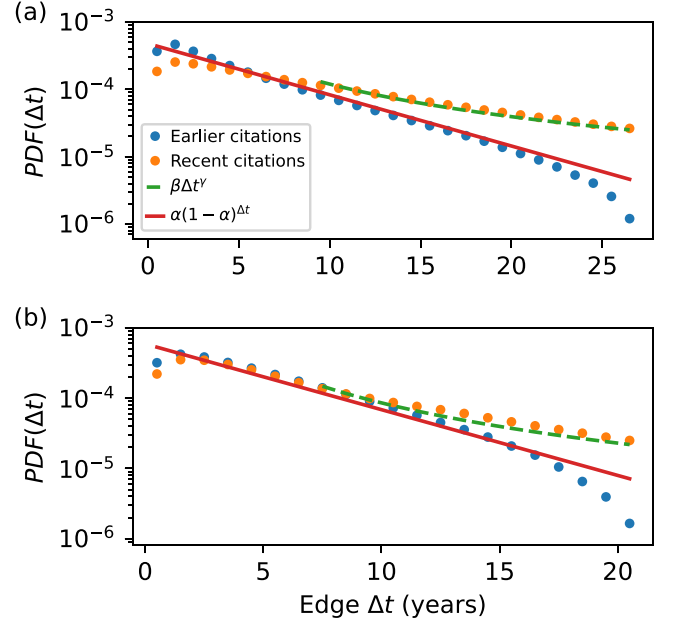


FIG. 8. Distribution of temporal edge lengths in years for citation networks of (a) physics and astronomy and (b) computer science fields in the OpenAlex database. Earlier citations include references made by papers published before the year 2000. Recent citations include references spanning nine years January 2015–March 2024. An exponential is fitted to the earlier citation distributions and a power law to the tail of the recent citation distributions. Fitted curve parameters are (a) $\beta = 63.42$, $\gamma = -1.6$, $\alpha = 4.8 \times 10^{-4}$ and (b) $\beta = 5.0 \times 10^3$, $\gamma = -1.9$, $\alpha = 5.9 \times 10^{-4}$. The probability distribution of recent citations as a function of Δt has a heavier tail compared to that of the earlier citations.

become more prevalent [Fig. 7(d)], the required tolerance to maintain the accuracy level tightens. For larger and longer networks, this can pose an obstacle as the accuracy of computing the difference in eigenvalues of consecutive iterations is bounded by floating-point errors. In practical terms and for the case of real-world citation networks, we observe that tolerance values smaller than 10^{-13} fail to reliably converge within a reasonable number of iterations.

APPENDIX C: TEMPORAL EDGE LENGTH DISTRIBUTION IN OPENALEX CITATION NETWORKS

As shown in Fig. 8, the distributions of temporal edge lengths in both networks seem to be better fitted by an exponential distribution for earlier publications, whereas for more recent publications, published in the last roughly 10 years (2015–2024), they are more suitably fitted by a power-law decay. The results indicate that connectivity patterns induced by time in citation networks have undergone significant change, signaling that, similar to the examples studied in Sec. III B, the applied community detection method should be able to (1) take into account this temporal connectivity pattern and (2) be able to flexibly handle this changing connectivity pattern for it to be effective.

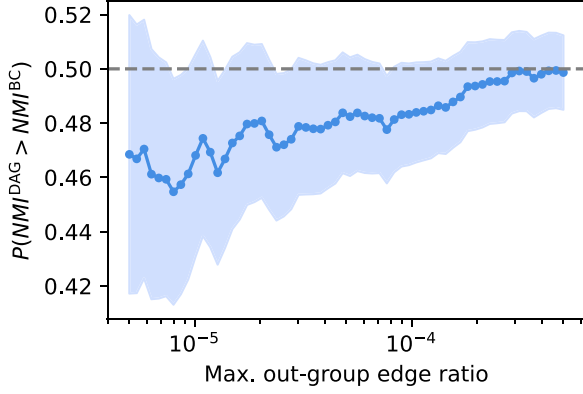


FIG. 9. Probability of the DAG model producing outcomes closer to the metadata compared to the block-corrected, as measured by normalized mutual information (NMI), decreases with the minimum threshold on separatedness of the metadata communities. This confirms the similar observation in Fig. 6 based on F_1 scores. The NMIs are produced by performing spectral bipartitioning, followed by split fine-tuning (see Sec. IV B). Note that the separatedness on the horizontal axis is plotted cumulatively. Shaded area shows 95% confidence interval.

APPENDIX D: SUBFIELD NETWORKS EVALUATION USING NORMALIZED MUTUAL INFORMATION

Figure 6 reports the agreement between the automatically detected communities and the subfield labels through the F_1 score. The choice of F_1 score is adequate as long as we treat the task as binary classification, i.e., if we know *a priori* that every article belongs to one of exactly two known subfields and we care equally about precision and recall. Once we step outside this binary framework, however, the use of a classification metric becomes problematic because (1) the community labels obtained from an unsupervised algorithm are arbitrary up to permutation, and (2) the number of inferred communities need not equal the number of ground-truth classes. Normalized mutual information (NMI) sidesteps both issues by quantifying information overlap between two partitions independently of label identities and cardinalities, making it the standard choice for partition-level evaluation in community detection research.

Despite its widespread use, normalized mutual information is not, however, without shortcomings. Chiefly, it exhibits bias in finite samples, that is, partitions with many small communities can achieve deceptively high normalized mutual information against a reference partition with heterogeneous community sizes, and it effectively ignores part of the information encoded in the contingency table [60].

Figure 9 provides a reproduction of Fig. 6, using normalized mutual information instead of F_1 scores for comparing labels inferred from the data. This confirms that our presented method proposes communities that are closer to the metadata communities compared to the DAG model, as measured by larger normalized mutual information between metadata and detected communities.

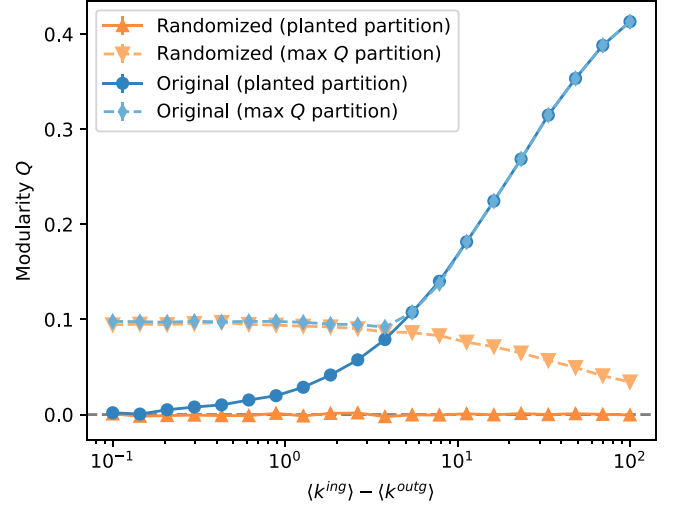


FIG. 10. Modularity value of the planted partition (dark blue) and the maximum modularity partition (light blue) under the block-corrected modularity over an ensemble of random networks with varying average in-group degrees. All networks are realizations of the exponential model [Eq. (11)], with $t_{\max} = 12$ time layers, $N = 100$ nodes per layer, $b = 2$ planted communities, and an edge decay rate of $d = 0.5$. Error bars show 95% confidence interval. When observing the modularity values of the same partitions in the null model realizations (dark and light orange) we see that below a threshold of distinctness, the true partition is not detectable.

APPENDIX E: DETECTABILITY LIMIT AND OVERFITTING

Any network drawn from the block null model preserves layer structure and layer connectivity of the original network, as well as the in- and out-degree distributions (on expectation), but without any community structure other than what is induced by the layer structure. A randomized realization of a null model is a network drawn from the null model with its edges shuffled and the same properties preserved. Ideally, a modularity-based community detection method should detect no significant community structure on realizations of its own corresponding null model. In actuality, nevertheless, modularity maximization methods often find communities with non-negligible modularity values where none exist [57,58].

For an array of random networks drawn from the exponential model described by Eq. (11), Fig. 10 inspects how the block-corrected modularity would treat the randomized versions of the networks and whether it detects the true partitions. For both the original network and its randomized counterpart, we compute the modularity of the true planted partition and the partition resulting from maximizing the modularity function.

As shown in Fig. 10, below a certain threshold of distinctiveness between communities, the modularity values of the maximum modularity partition on both original and the randomized networks are on par, an indication that the true partition would not be detectable and that the maximum modularity partition is explained by the tendency of the modularity maximization method to overfit. However, as the difference between the in-group and out-group degrees becomes greater the maximum modularity partition converges to that of the

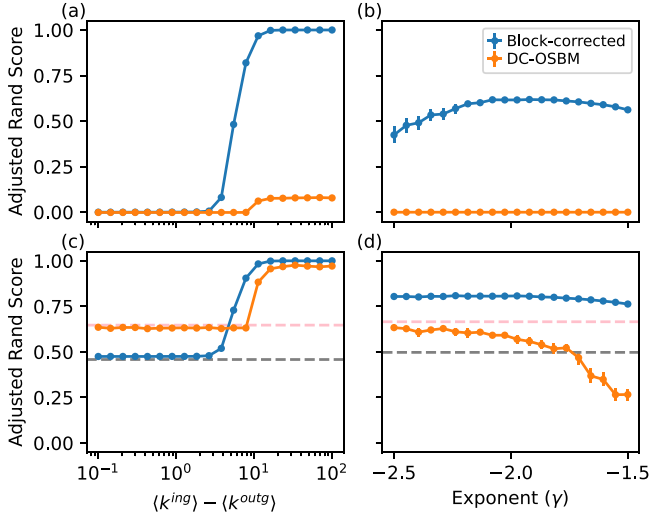


FIG. 11. Adjusted Rand scores between discovered communities and the true partitions. Realizations of the exponential model [Eq. (11)] are networks with $t_{\max} = 12$ time layers, $N = 100$ nodes per layer, $b = 2$ planted communities, and an edge decay rate of $d = 0.5$ [panels (a) and (c)]. Realizations of the power-law model (Eq. (12)) are networks with $t_{\max} = 200$ time layers, $N = 100$ nodes per layer, $b = 2$ planted communities, in-group degree 8, and out-group degree 4 [panels (b) and (d)]. For panels (c) and (d) the outcome communities detected by block-corrected modularity are split based on the time layers for parity with DC-OSBM. The gray dashed line shows the expected adjusted Rand score for correctly inferring all node-times while randomly assigning the unknown attribute. The pink dashed line indicates expected adjusted Rand score corresponding to correctly inferring only the $t_{\max}$ time layers. In panels (a) and (b) the comparison is done on the raw output communities. Error bars show 95% confidence intervals.

true planted partition, indicating that the true partition becomes detectable.

APPENDIX F: COMPARISON WITH ORDERED STOCHASTIC BLOCK MODEL

Thus far we compared the performance of our proposed block-corrected modularity to other modularity-based approaches. There are, however, other methods not necessarily based on modularity that take an inherent ordering present in the network into account. A primary example is Ordered Stochastic Block Model (OSBM) and its degree-corrected variant (DC-OSBM) [29]. A major difference between the approach presented here and that of Ref. [29], however, is that the block-corrected modularity takes any inherent ordering imposed on nodes as an *a priori* attribute, readily available as an input to the model, while the latter attempts to infer

the ordering, as an additional layer on top of the inferred communities.

Despite this crucial difference we seek to investigate the models' performances on an ensemble of synthetic networks drawn from the exponential generative model [Eq. (11)] across varying average in-group degrees, and the power-law model [Eq. (12)] across exponents. We show the comparison of performance of the block-corrected model and DC-OSBM in Figs. 11(a) and 11(b). The direct comparison of the resulting partitions based on adjusted Rand score with the planted partition shows that the block-corrected modularity is more successful at directly inferring the hidden attribute.

Considering that the OSBM and DC-OSBM attempt to infer the time layers as well as the planted partition, on our synthetic network models they ideally infer $t_{\max} \times b$ distinct communities. To make a comparison that does not disadvantage the methods that split communities in time, we adapt the ground-truth and the discovered communities of the block-corrected method by dividing them into their constituent time layers, forming $t_{\max} \times b$ communities. This should somewhat balance the fact that OSBM-based methods are at a disadvantage due to the method not being designed for this particular problem. However, it should be noted that in real-world use cases, we would want to infer communities that are not split in time. The results are illustrated in Figs. 11(c) and 11(d).

In the exponential model based on Eq. (11), the block-corrected modularity consistently outperforms the DC-OSBM when comparing the raw partitions [Figs. 11(a) and 11(b)]. In the case where ground-truth communities are split into $t_{\max} \times b$ communities, shown in Fig. 11(c), for networks with only a small difference between in- and out-group degrees the block-corrected modularity fails to distinguish anything other than the time layers, as they coincide with the dashed gray line indicating correctly inferring the time layers and random assignment for the ground-truth communities. This can be ascribed to the detectability limit described in Appendix E. The DC-OSBM, however, levels at the analytical expected adjusted Rand score for inferring the time layers only (dashed pink line), forming a total of $t_{\max}$ communities. As the planted communities become more distinct, the classification performance of both models rise toward 1.0, while the block-corrected achieves slightly better outcomes.

For the case of power-law edge length model based on Eq. (12), shown in Fig. 11(d), the block-corrected modularity surpasses the DC-OSBM across all values of the exponents of the power-law edge length distributions. Notably, the DC-OSBM fails to correctly infer the time layers for particularly heavy-tail edge length distributions, falling below the dashed gray line.

- [1] M. Girvan and M. E. Newman, Community structure in social and biological networks, *Proc. Natl. Acad. Sci. USA* **99**, 7821 (2002).
- [2] J. Ji, A. Zhang, C. Liu, X. Quan, and Z. Liu, Survey: Functional module detection from protein-protein interaction networks, *IEEE Trans. Knowledge Data Eng.* **26**, 261 (2012).

- [3] A. L. Traud, P. J. Mucha, and M. A. Porter, Social structure of Facebook networks, *Physica A* **391**, 4165 (2012).
- [4] P. Guerra, W. Meira Jr, C. Cardie, and R. Kleinberg, A measure of polarization on social media networks based on community boundaries, in *Proceedings of the International AAAI Conference on Web and Social Media* (The Association for

- the Advancement of Artificial Intelligence, Cambridge, Massachusetts, 2013), Vol. 7, pp. 215–224.
- [5] D. Hric, K. Kaski, and M. Kivelä, Stochastic block model reveals maps of citation patterns and their evolution in time, *J. Info.* **12**, 757 (2018).
 - [6] A. Salloum, T. H. Y. Chen, and M. Kivelä, Separating polarization from noise: Comparison and normalization of structural polarization measures, *Proc. ACM Hum.-Comput. Interact.* **6**, 1 (2022).
 - [7] M. E. Newman, Finding community structure in networks using the eigenvectors of matrices, *Phys. Rev. E* **74**, 036104 (2006).
 - [8] V. A. Traag and J. Bruggeman, Community detection in networks with positive and negative links, *Phys. Rev. E* **80**, 036115 (2009).
 - [9] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, Fast unfolding of communities in large networks, *J. Stat. Mech.* (2008) P10008.
 - [10] T. Richardson, P. J. Mucha, and M. A. Porter, Spectral tripartitioning of networks, *Phys. Rev. E* **80**, 036111 (2009).
 - [11] S. White and P. Smyth, A spectral clustering approach to finding communities in graphs, in *Proceedings of the 2005 SIAM International Conference on Data Mining* (SIAM, Philadelphia, 2005), pp. 274–285.
 - [12] T. P. Peixoto, *Descriptive vs. inferential community detection in networks Pitfalls, myths and half-truths* (Cambridge University Press, Cambridge, 2023).
 - [13] B. H. Good, Y.-A. De Montjoye, and A. Clauset, Performance of modularity maximization in practical contexts, *Phys. Rev. E* **81**, 046106 (2010).
 - [14] L. Speidel, T. Takaguchi, and N. Masuda, Community detection in directed acyclic graphs, *Eur. Phys. J. B* **88**, 203 (2015).
 - [15] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, and A. Arenas, Mathematical formulation of multilayer networks, *Phys. Rev. X* **3**, 041022 (2013).
 - [16] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, Community detection in temporal multilayer networks, with an application to correlation networks, *Multiscale Model. Sim.* **14**, 1 (2016).
 - [17] P. Expert, T. S. Evans, V. D. Blondel, and R. Lambiotte, Uncovers space-independent communities in spatial networks, *Proc. Natl. Acad. Sci. USA* **108**, 7663 (2011).
 - [18] X. Liu, T. Murata, and K. Wakita, Extending modularity by capturing the similarity attraction feature in the null model, [arXiv:1210.4007](https://arxiv.org/abs/1210.4007).
 - [19] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez, Motif-based communities in complex networks, *J. Phys. A: Math. Theor.* **41**, 224001 (2008).
 - [20] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova, Clustering attributed graphs: Models, measures and methods, *Netw. Sci.* **3**, 408 (2015).
 - [21] P. Chunaev, Community detection in node-attributed social networks: A survey, *Comput. Sci. Rev.* **37**, 100286 (2020).
 - [22] V. Red, E. D. Kelsic, P. J. Mucha, and M. A. Porter, Comparing community structure to characteristics in online collegiate social networks, *SIAM Rev.* **53**, 526 (2011).
 - [23] C. Jia, Y. Li, M. B. Carson, X. Wang, and J. Yu, Node attribute-enhanced community detection in complex networks, *Sci. Rep.* **7**, 2626 (2017).
 - [24] Y. Asim, A. Majeed, R. Ghazal, B. Raza, W. Naeem, and A. K. Malik, Community detection in networks using node attributes and modularity, *Int. J. Adv. Comput. Sci. Appl.* **8**, 382 (2017).
 - [25] Y. Zhou, H. Cheng, and J. X. Yu, Graph clustering based on structural/attribute similarities, *Proc. VLDB Endow.* **2**, 718 (2009).
 - [26] Z. Huang, Y. Ye, X. Li, F. Liu, and H. Chen, Joint weighted nonnegative matrix factorization for mining attributed graphs, in *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23–26, 2017, Proceedings, Part I 21* (Springer, Berlin, 2017), pp. 368–380.
 - [27] C. Zhe, A. Sun, and X. Xiao, Community detection on large complex attribute network, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Association for Computing Machinery (ACM), Anchorage, Alaska, 2019), pp. 2041–2049.
 - [28] P. D. B. Parolo, R. K. Pan, R. Ghosh, B. A. Huberman, K. Kaski, and S. Fortunato, Attention decay in science, *J. Info.* **9**, 734 (2015).
 - [29] T. P. Peixoto, Ordered community detection in directed networks, *Phys. Rev. E* **106**, 024305 (2022).
 - [30] P. Erdős and A. Rényi, On random graphs I, *Publ. Math. Debrecen* **6**, 18 (1959).
 - [31] P. Erdős and A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17 (1960).
 - [32] M. Molloy and B. Reed, A critical point for random graphs with a given degree sequence, *Random Struct. Algorithms* **6**, 161 (1995).
 - [33] B. Bollobás, A probabilistic proof of an asymptotic formula for the number of labelled regular graphs, *Eur. J. Combinatorics* **1**, 311 (1980).
 - [34] P. W. Holland, K. B. Laskey, and S. Leinhardt, Stochastic block-models: First steps, *Social Netw.* **5**, 109 (1983).
 - [35] M. J. Barber, Modularity and community detection in bipartite networks, *Phys. Rev. E* **76**, 066102 (2007).
 - [36] B. Karrer and M. E. Newman, Random graph models for directed acyclic networks, *Phys. Rev. E* **80**, 046110 (2009).
 - [37] J. N. Pougé-Biyong and R. Lambiotte, Signedlouvain: Louvain for signed networks, [arXiv:2407.19288](https://arxiv.org/abs/2407.19288).
 - [38] G. Zeno, T. La Fond, and J. Neville, Dymond: Dynamic motif-nodes network generative model, in *Proceedings of the Web Conference 2021* (International World Wide Web Conference Committee (IW3C2), Ljubljana, Slovenia, 2021), pp. 718–729.
 - [39] S. Purohit, L. B. Holder, and G. Chin, Temporal graph generation based on a distribution of temporal motifs, in *Proceedings of the 14th International Workshop on Mining and Learning with Graphs* (Association for Computing Machinery (ACM), London, UK 2018), Vol. 7.
 - [40] B. Karrer and M. E. Newman, Stochastic blockmodels and community structure in networks, *Phys. Rev. E* **83**, 016107 (2011).
 - [41] E. A. Leicht and M. E. Newman, Community structure in directed networks, *Phys. Rev. Lett.* **100**, 118703 (2008).
 - [42] D. Krackhardt and R. N. Stern, Informal networks and organizational crises: An experimental simulation, *Social Psychol. Q.* **51**, 123 (1988).
 - [43] W. M. Rand, Objective criteria for the evaluation of clustering methods, *J. Am. Stat. Assoc.* **66**, 846 (1971).
 - [44] L. Hubert and P. Arabie, Comparing partitions, *J. Class.* **2**, 193 (1985).

- [45] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, On modularity clustering, *IEEE Trans. Knowledge Data Eng.* **20**, 172 (2007).
- [46] A. Clauset, M. E. Newman, and C. Moore, Finding community structure in very large networks, *Phys. Rev. E* **70**, 066111 (2004).
- [47] M. E. Newman, Modularity and community structure in networks, *Proc. Natl. Acad. Sci. USA* **103**, 8577 (2006).
- [48] M. Newman, Spectral community detection in sparse networks, [arXiv:1308.6494](https://arxiv.org/abs/1308.6494).
- [49] J. Priem, H. Piwowar, and R. Orr, Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, [arXiv:2205.01833](https://arxiv.org/abs/2205.01833).
- [50] L. Peel, D. B. Larremore, and A. Clauset, The ground truth about metadata and community detection in networks, *Sci. Adv.* **3**, e1602548 (2017).
- [51] M. E. Newman and A. Clauset, Structure and inference in annotated networks, *Nat. Commun.* **7**, 11863 (2016).
- [52] D. Hric, T. P. Peixoto, and S. Fortunato, Network structure, metadata, and the prediction of missing nodes and annotations, *Phys. Rev. X* **6**, 031038 (2016).
- [53] D. Hric, R. K. Darst, and S. Fortunato, Community detection in networks: Structural communities versus ground truth, *Phys. Rev. E* **90**, 062805 (2014).
- [54] L. Mangold and C. Roth, Quantifying metadata relevance to network block structure using description length, *Commun. Phys.* **7**, 331 (2024).
- [55] <https://help.openalex.org/hc/en-us/articles/24397285563671-About-the-data>.
- [56] [10.5281/zenodo.14867442](https://zenodo.org/record/14867442)
- [57] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral, Modularity from fluctuations in random graphs and complex networks, *Phys. Rev. E* **70**, 025101 (2004).
- [58] J. Reichardt and S. Bornholdt, When are networks truly modular? *Physica D* **224**, 20 (2006).
- [59] T. P. Peixoto, Bayesian stochastic blockmodeling, *Advances in Network Clustering and Blockmodeling*, edited by P. Doreian, V. Batagelj, and A. Ferligoj, Chapter 11 (John Wiley & Sons, Ltd, 2019), pp. 289–332.
- [60] M. Jerdee, A. Kirkley, and M. Newman, Normalized mutual information is a biased measure for classification and community detection, [arXiv:2307.01282](https://arxiv.org/abs/2307.01282).