Journal of Machine Learning Research 26 (2025) 1-67 Submitted 5/24; Published 1/25

Curvature-based Clustering on Graphs

Yu Tian* YU.TIAN.RESEARCH@GMAIL.COM
Center for Systems Biology Dresden
Dresden, 01307, Germany

Zachary Lubberts* ZLUBBERTS@VIRGINIA.EDU
Department of Statistics
University of Virginia
Charlottesville, VA 22904, USA

Melanie Weber MWEBER@SEAS.HARVARD.EDU
Harvard University
Cambridge, MA 02138, USA

Editor: Michael Mahoney

Abstract

Unsupervised node clustering (or *community detection*) is a classical graph learning task. In this paper, we study algorithms that exploit the geometry of the graph to identify densely connected substructures, which form clusters or communities. Our method implements discrete Ricci curvatures and their associated geometric flows, under which the edge weights of the graph evolve to reveal its community structure. We consider several discrete curvature notions and analyze the utility of the resulting algorithms. In contrast to prior literature, we study not only single-membership community detection, where each node belongs to exactly one community, but also mixed-membership community detection, where communities may overlap. For the latter, we argue that it is beneficial to perform community detection on the line graph, i.e., the graph's dual. We provide both theoretical and empirical evidence for the utility of our curvature-based clustering algorithms. In addition, we give several results on the relationship between the curvature of a graph and that of its dual, which enable the efficient implementation of our proposed mixed-membership community detection approach and which may be of independent interest for curvature-based network analysis.

Keywords: Community Detection, Node Clustering, Discrete Curvature, Line Graph, Graph-based Learning

1. Introduction

Relational data, such as graphs or networks, is ubiquitous in machine learning and data science. Consequently, a large body of literature devoted to studying the structure of such data has been developed. Clustering on graphs, also known as *community detection* or *(unsupervised) node clustering*, is of central importance to the study of relational data. It seeks

. *Equal contribution

. A preliminary version of part of this work was presented at the NeurIPS Workshop on Symmetry and Geometry in Neural Representations 2022 (Tian et al., 2023).

(a) Single-Membership (b) Mixed-Membership

Figure 1: Clustering on Graphs.

to identify densely interconnected substructures (*clusters*) in a given graph. Such structure is ubiquitous in relational data: We may think of friend circles in social networks, metabolic pathways in biochemical networks or article categories in Wikipedia. A standard mathematical model for analyzing community structure in graphs is the *Stochastic Block Model (SBM)*, which has led to fundamental insights into the detectability of communities (Abbe, 2017). Classically, communities are identified by clustering the nodes of the graph. Popular methods include the Louvain algorithm (Blondel et al., 2008), the Girvan-Newman algorithm (Girvan and Newman, 2002) and Spectral clustering (Cheeger, 1969; Fiedler, 1973; Spielman and Teng, 1996). Recently, there has been growing interest in another class of algorithms, which applies a geometric lens to community detection.

Graph curvatures provide a means to understanding the structure of networks through a characterization of their geometry. More generally, curvature is a classical tool in differential geometry, which is used to characterize the local and global properties of geodesic spaces. While originally defined in continuous spaces, discrete notions of curvature have recently seen a surge of interest. Of particular interest are discretizations of *Ricci curvature*, which is a local notion of curvature that relates to the volume growth rate of the unit ball in the space of interest (geodesic dispersion). Curvature-based analysis of relational data (see, e.g., (Weber et al., 2017a,b)) has been applied in many domains, including to biological (Elumalai et al., 2022; Weber et al., 2017c; Tannenbaum et al., 2015), chemical (Leal et al., 2021; Saucan et al., 2018), social (Painter et al., 2019) and financial networks (Sandhu et al., 2016).

Community detection via graph curvatures is motivated by the observation that edges between communities (so called *bridges*, see Figure 1b(a)) have low Ricci curvature. By identifying and removing such bridges, one can learn a partition of the graph into its communities. This simple idea has given rise to a series of curvature-based approaches for community detection (Ni et al., 2019; Sia et al., 2019; Weber et al., 2018; Tian et al., 2023). However, the picture is far from complete. The proposed algorithms vary significantly in their implementation, in particular in the choice of the curvature notion that they utilize. In this paper, we will elucidate commonalities and differences among variants of the two most common curvature notions in terms of their utility for community detection. To this end, we propose a unifying framework for curvature-based community detection algorithms and perform a systematic analysis of the advantages and disadvantages of each curvature notion. The insights gained from this analysis give rise to several improvements, which mitigate the identified shortcomings in accuracy and scalability.

The present paper also addresses a second important gap in the literature. Existing work on partition-based community detection, specifically on curvature-based approaches, focuses almost exclusively on detecting communities with *unique* membership, i.e., each node can belong to only one community (*single-membership community detection*). However, in many complex systems that generate relational data, nodes may belong to more than one community: A member of a social network might belong to a circle of high school friends and a circle of college friends, a protein may have different functional roles in a biochemical network etc. This creates a need for algorithms that can cluster nodes with *mixed* membership (Airoldi et al., 2008; Yang and Leskovec, 2013; Zhang et al., 2020), so-called *mixed-membership community detection*. Observe that the mixed-membership structure precludes the existence of bridges between communities, because the communities overlap (see Figure 1b(b)). This renders community detection methods that rely on graph partitions, such as the curvature-based methods discussed above, inapplicable. In this paper, we propose a principled curvature-based approach for detecting mixed-membership community structure. We show that while partition-based approaches do not recover the underlying community structure when applied to the original graph, they perform well on its dual, the *line graph*. The line graph encodes the connectivity between the edges of the original graph (see Figure 2). This reformulation of the relational information in the graph allows for disentangling the overlapping community structure (see Figure 3). When applied to the line graph, our curvature-based community detection approach correctly identifies edges in the line graph, which can be cut to partition the graph into its communities and assign node labels that reflect mixed-membership community structure. Utilizing fundamental relationships between the curvature of a graph and its dual, we demonstrate that curvature-based mixed-membership community detection results in scalable and accurate methods.

1.1 Overview and Summary of Contributions

Our main contributions are as follows:

1. We propose a *unifying framework* for curvature-based community detection (unsupervised node clustering) algorithms (Section 2). Our framework encompasses several previously studied algorithms for curvature-based single-membership community detection (sec. 2.3, 6.1.1 and 6.2.1). Utilizing this framework, we systematically investigate the strengths and weaknesses of different discrete curvature notions (specifically, variants of Forman's and Ollivier's Ricci curvature, Sections 7 and 8).

2. We adapt our proposed framework to mixed-membership community detection (Section 6), providing the first curvature-based algorithms for this setting.

3. We elucidate significant scalability issues in community detection approaches that utilize Ollivier's notion of discrete curvature. To overcome these issues, we propose an effective approximation, which may be of independent interest in the broader area of curvature-based network analysis (Sections 4.1.3 and 5.2.2).

4. We demonstrate the utility of our framework through theoretical analysis (Section 6.1.2) and through a series of systematic experiments with synthetic and real network data. We further benchmark curvature-based methods against classical community detection approaches (Section 7).

5. As a byproduct, we derive several fundamental relations between discrete curvatures in a graph and its dual. Those relations may be of independent interest (Section 5).

2. Clustering on Graphs

Community structure is a hallmark feature of networks, characterized by clusters of nodes that have more internal connections than connections to nodes in other clusters. In networks with mixed-membership structure, which quantifies the idea of overlapping communities, nodes may belong to more than one cluster.

In the following, let $G = (V, E, W)$ denote a graph with vertex set V and edge set E. We may associate weights with both vertices and edges, defined as functions $\omega(v) : V \to \mathbb{R}_+$ and $\omega(e) : E \to \mathbb{R}_+$. The distance measure on G is the standard *path distance* d, i.e.,

$$d_G(u, v) := \inf_{\mathcal{P}} \sum_{i=1}^{l-1} w(\{z_i, z_{i+1}\}), \tag{2.1}$$

where the infimum is taken over all $u - v$ paths $\mathcal{P} = \{z_i\}_{i=1}^{l} \subseteq V$, with $\{z_i, z_{i+1}\} \in E$ for each i, and $z_1 = u, z_l = v$. We further define the *degree* d_u of a node u as the weighted sum of its adjacent edges, i.e.,

$$d_u := \sum_{u \sim e} w(e) \, . \tag{2.2}$$

We remark that the terms "network" and "graph" are often used interchangeably. Following the convention in the graph learning literature, we use "graph", if we refer to the mathematical object and "network" when referring to a data set or specific instance of a graph. We will use the terms "node" and "vertex" interchangeably.

2.1 Single-Membership Node Clustering

2.1.1 Single-Membership Community Structure

In classical community detection, each node can only belong to one community, which characterizes the community structure as being single-membership. Various algorithms exist by virtue of interdisciplinary expertise (Blondel et al., 2008; Girvan and Newman, 2002; von Luxburg, 2007), generally aiming to optimize a quality function with respect to different partitions of the network. We denote a quality function as Q which increases as the resulting community structure becomes clearer from some perspective, such as the modularity (Girvan and Newman, 2002), and denote a partition of the vertices as $\{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$ that are mutually exclusive and exhaustive, i.e., $\mathcal{B}_h \cap \mathcal{B}_l = \emptyset$ when $h \neq l$ and $\cup_{h=1}^{k} \mathcal{B}_h = V$, with k being the number of communities. The classical community detection problem can then be formulated as the combinatorial optimization problem

$$\max_{k} \max_{\{\mathcal{B}_1, \ldots, \mathcal{B}_k\}} Q(\{\mathcal{B}_1, \ldots, \mathcal{B}_k\}).$$

In this generality, the community detection problem is NP-hard, so heuristics-based algorithms are typically applied in order to provide a sufficiently good solution.

The *Stochastic Block Model* (short: *SBM*) (Holland et al., 1983) is a random graph model, which emulates the community structure found in many real networks. It is a

popular model for studying complex networks and, in particular, clustering on graphs. In the SBM, vertices are partitioned into subgroups called blocks, and the distribution of the edges between vertices is dependent on the blocks to which the vertices belong. Formally:

Definition 1 (Stochastic Block Model) Let $\{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$ be a partition of a graph's vertices into k *blocks*, and let $\mathbf{B} = (B_{hl})$ denote the characteristic matrix between the blocks, where B_{hl} shows the probability of an edge existing between a vertex in block $\mathcal{B}_h$ and one in $\mathcal{B}_l$. Then, if we denote the random adjacency matrix of the network by $\mathbf{A} = (A_{ij})$ where $A_{ij} = 1$ if there is an edge between vertices v_i and v_j,

$$\mathbb{P}\left(A_{ij} = 1\right) = B_{\sigma(i)\sigma(j)} \ ,$$

where $\sigma : V \to \{1, \ldots, k\}$ indicates the block membership. A *planted SBM* has the additional structure that $B_{hh} = p_{in}$, $\forall h$, and $B_{hl} = p_{out}$, $\forall h \neq l$, which we denote by $SBM(p_{in}, p_{out})$.

2.1.2 Partition-based clustering methods

Of particular importance for classical community detection are edges between clusters, often called *bridges*. Bridges connect two nodes in distinct communities (see Figure 1a). In graphs with single-membership community structure, i.e., where each node belongs to one community only, such edges always exist (otherwise each community is a connected component). This observation has motivated a range of *partition-based* community detection approaches (Blondel et al., 2008; Girvan and Newman, 2002; von Luxburg, 2007), which rely on identifying and cutting such bridges to partition the graph into communities. Among the most popular community detection methods is the Louvain algorithm (Blondel et al., 2008), which is a heuristic method proposed to maximize modularity hierarchically, starting from each node as a community and then successively merging with its neighboring communities whenever it improves modularity. Spectral clustering (von Luxburg, 2007), another widely used method, utilizes the spectrum of the graph Laplacian to find a partition of the graph with a minimal number of edge cuts (*mincut problem*). This optimization problem too is NP-hard, but it can be solved efficiently after relaxing some constraints. We give a brief overview of other classical partition-based community detection methods in Section 3. In this paper, we discuss methods that utilize discrete curvature to identify bridges: Below, we introduce a blueprint for partition-based clustering via discrete Ricci curvature.

2.2 Mixed-Membership Node Clustering

2.2.1 Mixed-membership Community Structure

In graphs with mixed-membership communities, nodes may belong to more than one community. Such community structure can be formalized via an extension of the SBM: The *Mixed-Membership Stochastic Block Model (short:* MMB*)* (Airoldi et al., 2008). In both models, each element of the adjacency matrix $\mathbf{A}$ above the diagonal is an independent Bernoulli random variable whose expectation only depends on the block memberships of the corresponding nodes. In the MMB, it is possible that nodes belong to more than one block, with various affiliation strengths. Formally:

Definition 2 (Mixed-Membership Block Model) We assume that the expectation $\mathbb{E}[\mathbf{A}]$ has the form $\mathbb{E}[\mathbf{A}] = \mathbf{XBX}^T$, where $\mathbf{X} \in [0,1]^{n\times k}$ is the community membership matrix with X_{il} indicating the affiliation of node i with community l and $\sum_l X_{il} = 1$, and n denotes the size of the network. The matrix $\mathbf{B} \in [0,1]^{k\times k}$ encodes the block connection probabilities. If $X_{il} = 1$ for some l, we call vertex i a *pure node*; otherwise i is a *mixed node*.

When all nodes are pure, we recover the ordinary SBM model. In our experiments, we consider a planted version where $\mathbf{B}_{hh} = p_{in},\ \forall h$, and $\mathbf{B}_{hl} = p_{out},\ \forall h \neq l$, which we denote by $MMB(p_{in}, p_{out})$.

2.2.2 Line Graphs

We denote the *dual* of a graph as its *line graph*:

Definition 3 *Given an unweighted, undirected graph $G = (V, E)$, its line graph is defined as $L(G) = (E, \mathcal{E})$, where each edge $\{\{u, v\}, \{r, s\}\} \in \mathcal{E}$ appears when $|\{u, v\} \cap \{r, s\}| = 1$. In other words, e, e' are adjacent in $L(G)$ when they are both incident on the same vertex.*

The line graph can be seen as a re-parametrization of the input graph, which encodes higher-order information about its edges' connectivity. Examples of graphs and their line graphs can be found in Figure 2. Importantly, cycles are preserved under this construction (e.g., Figure 2 (bottom)). Cliques and cycles in the line graph may also arise from nodes of degree three or higher (see, e.g., Figure 2 (top)).

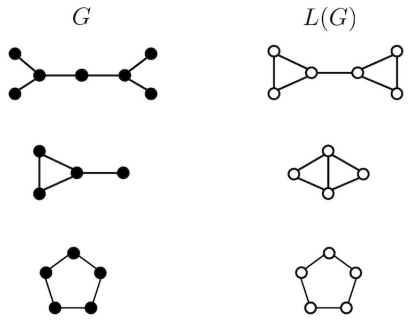


Figure 2: Graphs G with their corresponding line graphs $L(G)$. Note that vertices in G with degree at least 3 result in cliques in $L(G)$, and that cycles are preserved.

Notice that the line graph is typically larger in size than the original graph. In a graph with n vertices and average degree $n\alpha_n$, we expect to have $n^2\alpha_n$ edges. In the line graph, this means $n^2\alpha_n$ vertices and $n^3\alpha_n^2$ edges, though this number increases with greater variation of the vertex degrees in the original graph. For the applications to community detection that we will discuss below, this necessitates a careful consideration of the scalability of different community detection approaches.

The classical line graph construction gives a graph without edge weights, since edge weights in the original graph correspond to node weights in the line graph. We will discuss below (Section 5.2.2) how weights can be imposed on line graph edges that encode meaningful structural information in the original graph.

2.2.3 Partition-based Methods on the Line Graph.

We have seen above that edges between clusters, often called *bridges*, are of particular importance for single-membership community detection. However, overlaps between communities preclude the existence of these bridges (see Figure 1b), which limits the applicability of partition-based approaches. In this paper we argue that the line graph provides a natural input for partition-based mixed-membership community detection. While nodes may not have a unique label in this model, the adjacent edges may still be internal, connecting

two nodes that are in the same community (at least partially). In this case, each edge is associated with a single community (see Figure 3 (left)). Consequently, by representing the relationships among edges in a line graph, we disentangle the overlapping communities. Each edge in the line graph appears at a vertex in the original graph. When the vertex has mixed membership, a bridge between the communities arises in the line graph (Figure 3 (right)).

2.3 Curvature-based Clustering Algorithm

Algorithm 1 Curvature-based Clustering via Ricci Flow

Input: Graph $G = (V, E, W_0)$, hyperparameters ν (step size, flow), ϵ (tolerance), ϵ_d (drop threshold), T (number of iterations)
for $t = 1, \dots, T$ **do**
  for $\{u, v\} \in E$ **do**
    Compute curvature $\kappa(\{u, v\})$.
    Evolve weight under Ricci flow: $w^t(\{u, v\}) \leftarrow (1 - \nu \cdot \kappa(\{u, v\}))d_G(u, v)$.
  end for
  Renormalize edge weights W^t: $w^t(\{u, v\}) \leftarrow \frac{|E| d_G(u,v)}{\sum_{\{u',v'\}\in E} d_G(u',v')}$ for all $\{u, v\} \in E$.
end for
Construct cut-off points $\{x_0, x_1, \dots, x_{n_f}\}$.
Initialize $\mathbf{l} \in \mathbb{R}^{|V|}$ (list of node labels), $Q_{-1}, Q_{best} := \epsilon$.
for $i = 0, \dots n_f$ **do**
  Construct $\tilde{G}_i = (V, E_i, W_i^T)$ with $E_i = \{\{u, v\} : w^T(\{u, v\}) > x_i\}$ and $W_i^T = W^T\big|_{E_i}$.
  Determine connected components of $\tilde{G}_i$. Assign node labels by components.
  Compute the modularity Q_i of the resulting community assignment.
  if $Q_i > Q_{best}$, $\frac{Q_i - Q_{i-1}}{Q_i} > \epsilon_d$ **then**
    Store label assignments in $\mathbf{l}$. $Q_{best} = Q_i$.
  end if
end for
If $Q_{best} > \epsilon$, return $\mathbf{l}$.

In this section, we will describe the blueprint of our *curvature-based clustering algorithm*, which we will then specialize to the single- and mixed-membership problem described above. In the community detection literature, approaches based on discrete Ricci curvature and an associated Ricci flow have been studied for unique-membership communities (Ni et al., 2019; Sia et al., 2019; Weber et al., 2018). We discuss the details of those approaches in Sections 6.1 and 6.2. To the best of our knowledge, extensions of curvature-based methods to *mixed-membership community detection* have not been studied before.

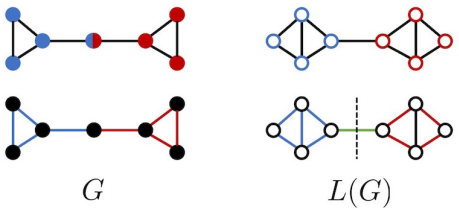


Figure 3: Community structure in a graph G and its line graph $L(G)$ with node labels (top) and edge labels (bottom).

Like previous approaches that utilize Ricci curvature (e.g., Ni et al. (2019)), we build on a notion of discrete Ricci flow first proposed by Ollivier (2009):

$$\frac{d}{dt}d_G(u,v)(t) = -\kappa(\{u,v\})(t) \cdot d_G(u,v)(t) \quad (\{u,v\} \in E) \,, \tag{2.3}$$

where $d_G(u,v)$ denotes the shortest path distance between adjacent nodes $u, v \in E$ and $\kappa(\{u,v\})$ the Ricci curvature along that edge. In this work, $\kappa(\{u,v\})$ may denote either Forman's or Ollivier's Ricci curvature. We will describe the two resulting methods in detail below.

Curvature-based clustering implements *discrete Ricci flow* via a combinatorial evolution equation, which evolves edge weights according to the local geometry of the graph. Consider a family of weighted graphs $G^t = \{V, E, W^t\}$, which is constructed from an input graph G by evolving its edge weights as

$$w^t(\{u,v\}) \leftarrow (1 - \kappa(\{u,v\}))d_G(u,v) \qquad (\{u,v\} \in E) \,, \tag{2.4}$$

where the curvature $\kappa(\{u,v\})$ and shortest-path distance $d_G(u,v)$ is computed on the graph G^{t-1}. Equation (2.4) can be viewed as a discrete analog of Ricci flow on Riemannian manifolds; it was first introduced by Ollivier (2010) with κ denoting ORC. Since then, several variants have been studied in the context of clustering, utilizing ORC (Ni et al., 2019; Sia et al., 2019; Tian et al., 2023), as well as FRC (Weber et al., 2018). Each of these algorithms is a special case of the general framework that we describe below.

The clustering procedure is initialized with the (possibly unweighted) input graph, i.e., $G^0 := G$. We then evolve the edge weights for T iterations under Ricci flow (Equation (2.4)). In each iteration, the edge weights are renormalized using

$$w^t(\{u,v\}) \leftarrow \frac{|E| d_G(u,v)}{\sum_{\{u',v'\} \in E} d_G(u',v')} \,. \tag{2.5}$$

Over time, the negativity of the curvature of edges that bridge communities intensifies, as edges with lower Ricci curvature contract more slowly under Ricci flow. On the other hand, edges with higher Ricci curvature contract faster. This results in a decrease of the weight of internal edges over time, while the weight of the bridges increases (see Figure 4). With that, the discrete Ricci flow reinforces the meso-scale structure of the network.

In order to recover single-membership labels, we cut the edges in the graph with high weights (or, equivalently, with very negative curvature) after evolving edge weights for T iterations. The resulting partition delivers a node clustering. In the mixed-membership case, we cut the edges in the line graph with the highest weight (or, equivalently, the most negative curvature) after evolving edge weights for T iterations. The resulting partition of the graph delivers an edge clustering from which we infer the community labels: We obtain a mixed-membership label vector y for each node v by computing

$$y_l(v) = \frac{1}{|E_v|} \sum_{e \in E_v} \chi_l(e) \,, \tag{2.6}$$

where χ_l is the indicator for the edge cluster C_l. Intuitively, each edge belonging to cluster l that is incident on a node v adds more evidence of affinity between the node v and the cluster C_l.

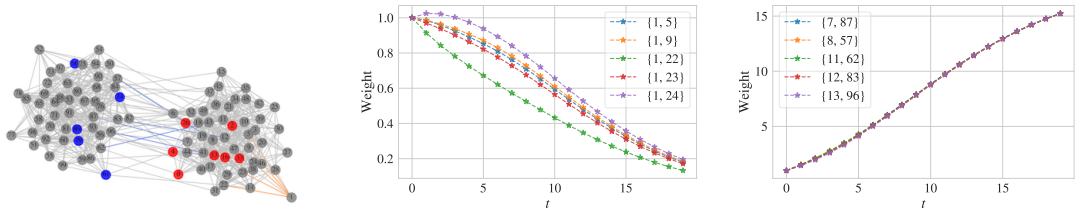


Figure 4: To illustrate the effect of the discrete Ricci flow in Algorithm 1, we consider a planted SBM of size $n = 100$, two equally-sized blocks, $p_{in} = 0.3$ and $p_{out} = 0.02$, visualized in the left panel. We select representative edges within a community, shown in orange on the graph, and plot the evolution of their weights in the middle panel. In the right panel, we show this evolution for selected edges between communities (i.e., bridges), shown in blue on the graph.

In both cases, the success of the method depends crucially on identifying a good weight threshold for cutting edges in the original graph (single-membership case) or the line graph (mixed-membership case). We propose to perform a hyperparameter search over *cut-off points* $\{x_0, x_1, \ldots, x_{n_f}\}$. The construction of the cut-off points is an important design choice and depends on the curvature notion used in the approach. We then compute the *modularity*, a classical quality metric in the community detection literature, for the label assignments corresponding to each cut-off point x_i. Specifically, modularity (Girvan and Newman, 2002; Gómez et al., 2009) is defined as

$$Q := \frac{1}{2w} \sum_{\{u,v\} \in E} \left(w(\{u,v\}) - \frac{d_u d_v}{2w} \right) \delta(\sigma(u), \sigma(v)) \,, \tag{2.7}$$

where $2w := \sum_{ij} w(\{i,j\})$. The larger the modularity, the better we expect the clustering to reflect the underlying community structure induced by the graph's connectivity. Our approach is schematically shown in Algorithm 1. Note that in Algorithm 1, the number of iterations are selected *a priori*, while the choice can also be integrated with the modularity check. This is because the choice of $T = 10$ generally provides sufficiently good results, which is verified with the comparison to the variant with intermediate modularity check and simple stopping criteria in Section 7.3.

3. Related Work

Community detection. Mixed-membership community detection is widely studied in the network science and data mining communities. Notable approaches include Bayesian methods (Airoldi et al., 2008; Hopkins and Steurer, 2017), matrix factorization (Yang and Leskovec, 2013), spectral clustering (Zhang et al., 2020), and vertex hunting (Jin et al., 2017), among others. In addition to the mixed-membership model that we consider here (Airoldi et al., 2008), there is a significant body of literature on closely related overlapping community models (Lancichinetti et al., 2009; Xie et al., 2013), which also study the problem of learning non-unique node labels. Curvature-based community detection methods for non-overlapping communities have recently received growing interest (Ni et al., 2019;

Sia et al., 2019; Gosztolai and Arnaudon, 2021; Weber et al., 2018). Such approaches utilize notions of discrete Ricci curvature (Sia et al., 2019; Gosztolai and Arnaudon, 2021; Weber et al., 2018), as well as an associated Ricci flow (Ni et al., 2019) to partition networks into communities, based on the observation that edges between communities have low curvature. The absence of such bridges in overlapping communities renders these approaches inapplicable to the setting studied in this paper. To the best of our knowledge, our algorithm is the first to study mixed-membership community structure with curvature-based methods.

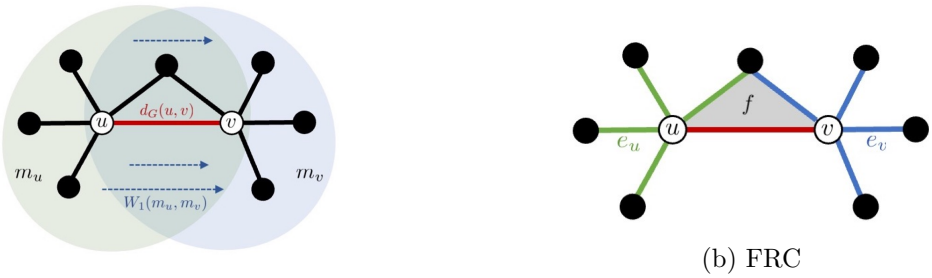
Higher-order structure in Networks. Historically, much of the network analysis literature has focused on the structural information encoded in nodes and edges. Recently, the analysis of higher-order structures has received increasing attention. A plethora of methods for analyzing higher-order structure in relational data has been developed (Benson et al., 2016; Battiston et al., 2020). Our work follows this line of thought, in that it focuses on the relations *between edges* and the structural information encoded therein. Curvature of higher-order structure has previously been studied in (Weber et al., 2017b; Saucan and Weber, 2018; Leal et al., 2021). Here, networks have been studied as polyhedral complexes (Weber et al., 2017b), or interactions between groups of nodes have been encoded in hypergraphs (Saucan and Weber, 2018; Leal et al., 2021). To the best of our knowledge, this is the first work that studies the curvature of line graphs.

Line Graphs. The notion of the *line graph* goes back at least to (Whitney, 1932), where it is shown that whenever $|V| \geq 5$, two graphs are isomorphic if and only if their line graphs are. An effective version of this result, which reports whether a given graph is a line graph, and if it is, returns the base graph, appears in Lehot (1974). Community detection and more generally network analysis via the line graph has been recently studied in (Chen et al., 2017; Krzakala et al., 2013; Lubberts et al., 2021; Evans and Lambiotte, 2010).

Discrete Curvature. There exists a large body of literature on discrete notions of curvature. Notable examples include Gromov's δ-hyperbolicity (Gromov, 1987), Bakry-Emre curvature (Erbar and Maas, 2012), as well as Forman's and Ollivier's Ricci curvatures (Forman, 2003; Ollivier, 2010). While there is some work on studying those curvature notions on higher-order networks (Bloch, 2014; Weber et al., 2017b) and hypernetworks (Saucan and Weber, 2018; Leal et al., 2021), they have not been studied on line graphs. Curvature-based analysis of relational data (see, e.g., (Weber et al., 2017a,b)) has been applied in many domains, including to biological (Elumalai et al., 2022; Weber et al., 2017c; Tannenbaum et al., 2015), chemical (Leal et al., 2021; Saucan et al., 2018), social (Painter et al., 2019), information (Ni et al., 2015) and financial networks (Sandhu et al., 2016). Discrete curvature has also found applications in Representation Learning, in particular for identifying representation spaces in graph embeddings (Lubold et al., 2023; Weber, 2020; Weber and Nickel, 2018).

4. Discrete Graph Curvature

Curvature is a classical tool in Differential Geometry, which is used to characterize the local and global properties of geodesic spaces. In this paper, we investigate *discrete* notions of curvature that are defined on graphs. Specifically, we focus on discretizations of *Ricci*



(a) ORC

(b) FRC

Figure 5: Computation of discrete Ricci curvature.

curvature, which is a local notion of curvature that relates to the volume growth rate of the unit ball in the space of interest (geodesic dispersion).

In the following, we only consider *undirected* networks. We introduce two classical notions of *discrete Ricci curvature*, which were originally introduced by Ollivier (2010) and Forman (2003), respectively.

4.1 Ollivier's Ricci Curvature

Our first notion of discrete curvature relates geodesic dispersion to optimal mass transport on the graph.

4.1.1 Formal Definition

Consider the transportation cost between two distance balls (i.e., vertex neighborhoods) along an edge in the network. In an unweighted graph, we endow the neighborhoods of vertices u, v adjacent to an edge $e = \{u, v\}$ with a uniform measure, i.e.,

$$m_u(z) := \frac{1}{d_u} \qquad \forall z,\ s.t.\ u \sim z, \tag{4.1}$$

and analogously for $m_v(z)$. Here, $u \sim z$ indicates that u, z are neighbors. In a weighted graph, for $\alpha \in [0, 1]$ and $p \geq 0$, we set

$$m_u^{\alpha,p}(z) := \begin{cases} \alpha & \text{if } z = u, \\ \frac{1-\alpha}{C_u} \exp(-d_G(u,z)^p) & \text{if } z \sim u, \\ 0 & \text{else.} \end{cases} \tag{4.2}$$

where the normalizing constant $C_u = \sum_{z \sim u} \exp(-d_G(u,z)^p)$. Notice that for neighboring vertices u, z we have $d_G(u,z) = w(\{u,z\}) := w_{u,z}$. When $\alpha = 0$, and $p = 0$ or we have an unweighted graph, this reduces to the uniform measure. We define *Ollivier's Ricci curvature* (Ollivier, 2010) (short: ORC) with respect to the Wasserstein-1 distance W_1 between those measures, i.e.,

$$\mathrm{Ric}_O(e) := 1 - \frac{W_1(m_u, m_v)}{d_G(u,v)}\ . \tag{4.3}$$

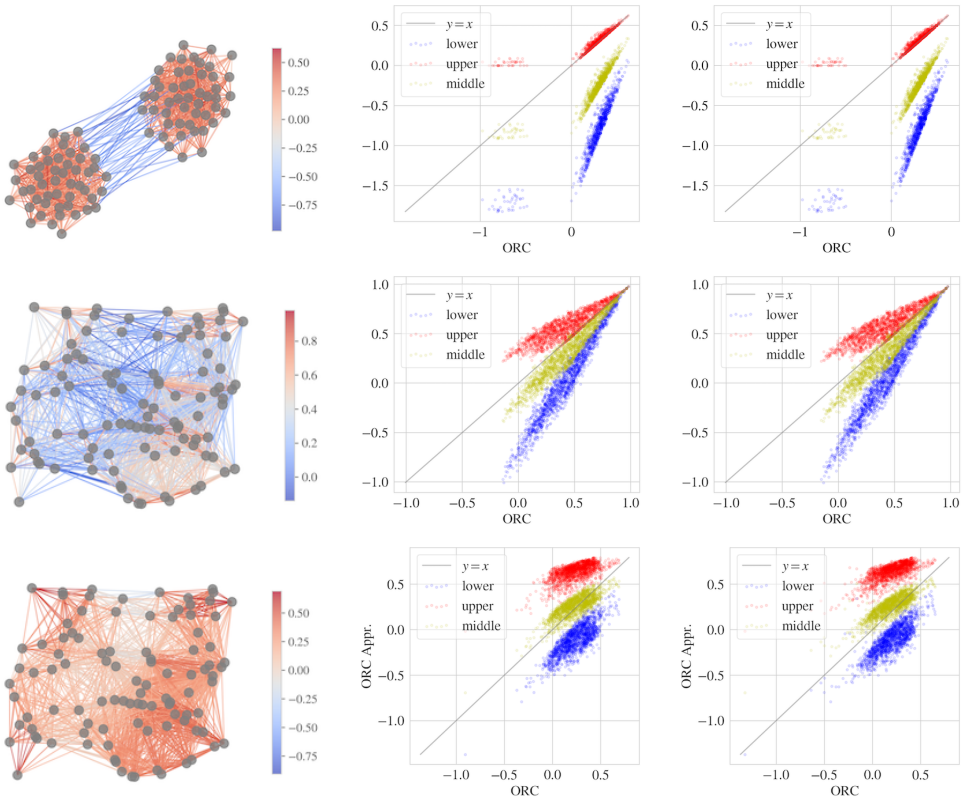


Figure 6: Scatter plot of the ORC on G obtained from the optimization versus its approximation, where the optimization is done by solving the earth mover's distance exactly (middle) and approximately with Sinkhorn (right), and G is obtained from the planted SBM of size $n = 100$, two equally-sized blocks, $p_{out} = 0.02$ and $p_{in} = 0.4$, with weights being 1 (top), and the 2-d RGG of size $n = 100$ and radius $r = 0.4$ with weights being 1 (middle), and proportional to the distance (bottom). G is visualised in the left column, with the color indicating the ORC obtained by solving the earth mover's distance exactly.

The computation of ORC is illustrated in Figure 5a. We can further define ORC for vertices with respect to the curvature of its adjacent edges. Formally, let $E_v := \{e \in E : v \in e\}$ denote the set of edges adjacent to a vertex v. Then its curvature is given by

$$\mathrm{Ric}_O(v) = \sum_{e \in E_v} \mathrm{Ric}_O(e) \, . \tag{4.4}$$

4.1.2 Computational Considerations

Computing ORC on a graph is quite expensive. The high cost arises mainly from the computation of the W_1-distance, which, in the discrete setting, is also known as *earth mover's distance*. Computing $W_1(m_u, m_v)$ between the neighborhoods of two vertices u, v

that are connected by an edge, corresponds to solving the linear program

$$\begin{aligned} \min \quad & W_1(m_u, m_v) = \inf_{\Lambda} \sum_{i\sim x} \sum_{j\sim y} \lambda_{ij} d_G(i,j) \\ \text{s.t.} \quad & \sum_{i\sim x} \lambda_{ij} = m_y^j \;\; \forall\, j \sim y; \qquad \sum_{j\sim y} \lambda_{ij} = m_x^i \;\forall\, i \sim x; \\ & \lambda_{ij} \geq 0 \;\; \forall\, i,j\ . \end{aligned} \tag{4.5}$$

Classically, $W_1(m_u, m_v)$ is computed using the *Hungarian algorithm* (Kuhn, 1955), the fastest variant of which has a cubic complexity. With growing neighborhood size, the computational cost of the Hungarian algorithm becomes prohibitively large. Alternatively, $W_1(m_u, m_v)$ can be approximated with the faster Sinkhorn algorithm (Sinkhorn and Knopp, 1967), whose complexity is only quadratic. However, even a quadratic cost introduces a significant computational bottleneck on the large-scale network data that we encounter in machine learning and data science applications. Here, we propose a different approach. Instead of approximating the W_1-distance, we propose an approximation of ORC. Our approximation can be written as a simple combinatorial quantity, which can be computed in linear time.

We begin by recalling a few classical bounds on ORC, first proven by Jost and Liu (2014). Let $\#(u,v)$ be the number of triangles that include the edge (x,y), $a \wedge b := \min\{a,b\}$ and $a \vee b := \max\{a,b\}$. With respect to these quantities, upper and lower bounds on ORC are given as follows:

Theorem 4 (Unweighted case (Jost and Liu, 2014))

1. *Lower bound on ORC:*

$$\begin{aligned} \mathrm{Ric}_O(\{u,v\}) \geq & -\left(1 - \frac{1}{d_v} - \frac{1}{d_u} - \frac{\#(u,v)}{d_u \wedge d_v}\right)_+ \\ & -\left(1 - \frac{1}{d_v} - \frac{1}{d_u} - \frac{\#(u,v)}{d_u \vee d_v}\right)_+ + \frac{\#(u,v)}{d_u \vee d_v}\ . \end{aligned} \tag{4.6}$$

   Note that a simpler, but less tight lower bound with respect to node degrees only is given by

$$\mathrm{Ric}_O(\{u,v\}) \geq -2\left(1 - \frac{1}{d_v} - \frac{1}{d_u}\right)\ .$$

2. *Upper bound on ORC:*

$$\mathrm{Ric}_O(\{u,v\}) \leq \frac{\#(u,v)}{d_u \vee d_v}\ . \tag{4.7}$$

In the following section, we will derive an extension of these results to weighted graphs.

Now, let Ric_O^{up} and Ric_O^{low} denote the upper and lower bounds, respectively. We propose to approximate $\mathrm{Ric}_O(\{u,v\})$ as the arithmetic mean of the upper and lower bounds, i.e.,

$$\widehat{\mathrm{Ric}_O}(\{u,v\}) := \frac{1}{2}\left(\mathrm{Ric}_O^{up}(\{u,v\}) + \mathrm{Ric}_O^{low}(\{u,v\})\right)\ . \tag{4.8}$$

Like the bounds themselves, $\widehat{\mathrm{Ric}_O}$ has a simple combinatorial form and can be computed in linear time. Since the computation relies only on local structural information, the procedure can be parallelized easily. Figure 6 demonstrates the utility of this approximation in practise (in both the SBM and Random Geometric Graphs (short: RGG); see Appendix B for more explanation).

4.1.3 Bounds on Ollivier's curvature in weighted graphs

Since our curvature-based algorithms work by modifying the edge weights in a graph according to their curvature values, we require an analogue of Theorem 4 for weighted graphs, in order to use a simple combinatorial approximation to the curvature as in Equation (4.8). To do so, we make use of the dual description of the Wasserstein distance between two measures m_1, m_2:

$$W_1(m_1, m_2) = \inf_{\pi \in \Pi(m_1, m_2)} \mathbb{E}_{(x,y)\sim\pi} d(x, y) = \sup_{\|h\|_L \leq 1} \left[\mathbb{E}_{x\sim m_1}[h(x)] - \mathbb{E}_{y\sim m_2}[h(y)]\right],$$

where $\Pi(m_1, m_2)$ is the set of couplings of m_1, m_2, and the supremum is over all functions $h : V \to \mathbb{R}$ which are Lipschitz with constant at most 1 (with respect to shortest path distance). We may obtain an upper bound on $W_1(m_1, m_2)$ by finding any coupling π of m_1, m_2, and a lower bound by finding any 1-Lipschitz function h.

Upper bound on $W_1(m_1, m_2)$. By constructing a transport plan taking m_x to m_y, we obtain the following upper bound:

Lemma 5 *Let m_x, m_y denote the measures on the node neighborhoods of x and y given by taking $\alpha = 0$ and $p = 1$ in Equation (4.2). Denote the vertices in $N(x) \setminus N(y)$ with ℓ, $N(y) \setminus N(x)$ with r, and $N(x) \cap N(y)$ with c. Introduce the shorthands $L_x := \sum_\ell m_x(\ell)$, $L_y := \sum_\ell m_y(\ell)$, as well as $X_x := m_x(x)$, $X_y := m_y(x)$. Then*

$$\begin{aligned} W_1(m_x, m_y) \leq &\sum_\ell w_{\ell,x} m_x(\ell) + \sum_r w_{r,y} m_y(r) \\ &+ \sum_c w_{c,y}(m_x(c) - m_y(c))_+ + w_{c,x}(m_y(c) - m_x(c))_+ \\ &+ \left| L_x + X_x - X_y - \sum_c (m_y(c) - m_x(c))_+ \right| w_{x,y} \,. \end{aligned}$$

Remark 6 *This upper bound is exact for weighted trees.*

As a computational tool, this bound is a step in the right direction: Consider the usual procedure for calculating ORC. We need to compute m_x, m_y, and $d_G(u, v)$ for each pair of vertices $u, v \in V_H$. Then we need to carry out the optimization, which can be realized as a linear program as in Equation (4.5). This lower bound still requires the first step, but avoids the need for the optimization over couplings, which is the major bottleneck in the curvature computation.

Lower bound on $W_1(m_1, m_2)$. We now provide a lower bound on $W_1(m_x, m_y)$:

Lemma 7 *Let m_x, m_y be as defined above, and let $\mathcal{P} = \{v \in N(x) \cup N(y) : m_x(v) - m_y(v) > 0\}$, $\mathcal{N} = \{v \in N(x) \cup N(y) : m_x(v) - m_y(v) < 0\}$. For $S \subseteq V$ and $v \in V$, let $d_G(v, S) = \min\{d_G(v, u) : u \in S\}$. Then*

$$W_1(m_x, m_y) \geq \max\left\{\sum_{v\in\mathcal{P}} d_G(v,\mathcal{N})(m_x(v) - m_y(v)), \sum_{v\in\mathcal{N}} d_G(v,\mathcal{P})(m_y(v) - m_x(v))\right\}.$$

Remark 8 A lower bound for $W_1(m_x, m_y)$ was previously given by Jost and Liu (2014). However, the ORC notion considered therein differed from ours in the choice of the mass distribution imposed on node neighborhoods. In addition, we believe that the result in Jost and Liu (2014) has a slight inaccuracy; for details, see the discussion in Remark 28.

Combinatorial bounds on ORC in the weighted case. We now utilize the combinatorial upper and lower bounds on $W_1(m_x, m_y)$ to obtain combinatorial bounds on ORC itself.

Theorem 9 (ORC bounds (weighted case)) *For $x, y \in V, x \sim y$, we have the following bounds:*

1. *Upper bound on ORC:*

$$\mathrm{Ric}_O(\{x,y\}) \leq 1 - \max\left\{\sum_v \frac{d_G(v,\mathcal{N})}{w_{xy}}(m_x(v) - m_y(v))_+, \sum_v \frac{d_G(v,\mathcal{P})}{w_{xy}}(m_y(v) - m_x(v))_+\right\}.$$

2. *Lower bound on ORC:*

$$\begin{aligned}\mathrm{Ric}_O(\{x,y\}) \geq 1 &- \sum_\ell \frac{w_{\ell,x}}{w_{x,y}} m_x(\ell) - \sum_r \frac{w_{r,y}}{w_{x,y}} m_y(r) \\ &- \sum_c \left[\frac{w_{c,y}}{w_{x,y}}(m_x(c) - m_y(c))_+ + \frac{w_{c,x}}{w_{x,y}}(m_y(c) - m_x(c))_+\right] \\ &- \left|L_x + X_x - X_y - \sum_c (m_y(c) - m_x(c))_+\right|.\end{aligned}$$

While technical, both bounds are purely combinatorial and can be computed in linear time without the need to solve a linear program.

4.2 Forman's Ricci Curvature

Our second notion of discrete curvature utilizes an analogy between spectral properties of manifolds and CW complexes to define a discrete Ricci curvature.

4.2.1 Formal Definition

Forman's discrete Ricci curvature (short: FRC) was originally defined as a measure of geodesic dispersion on CW complexes (Forman, 2003). In its most general form, FRC is defined via a discrete analogue of the Bochner-Weitzenböck identity

$$\square_d = B_d + F_d \ ,$$

which establishes a connection between the (discrete) Riemann-Laplace operator $\square_d$, the Bochner Laplacian F_d and Forman's curvature tensor F_d. Networks can be viewed as polyhedral complexes, which is one instance of a CW complex. In particular, by viewing a network's edges as 1-cells, we can define FRC for edges $e = (v_1, v_2)$ as

$$\mathrm{Ric}_F(e) = \omega_e \left(\frac{\omega_{v_1}}{\omega_e} + \frac{\omega_{v_2}}{\omega_e} - \sum_{e_{v_1} \sim e} \frac{\omega_{v_1}}{\sqrt{\omega_e \omega_{e_{v_1}}}} - \sum_{e_{v_2} \sim e} \frac{\omega_{v_2}}{\sqrt{\omega_e \omega_{e_{v_2}}}} \right) \tag{4.9}$$

To arrive at this expression, we have specialized Forman's general notion for F_d to the case of 1-cells (see also (Weber et al., 2017a,b)). Here, e_v denotes an edge that shares the vertex v with e. Note that if G is unweighted, then this reduces to

$$\mathrm{Ric}_F(e) = 4 - d_{v_1} - d_{v_2} \ .$$

We can define FRC for vertices (i.e., 0-cells) with respect to the curvature of its adjacent edges E_v (i.e., $E_v := \{e \in E : \ e = (\cdot, v) \ \text{ or } e = (v, \cdot)\}$):

$$\mathrm{Ric}_F(v) = \sum_{e \in E_v} \mathrm{Ric}_F(e) \ . \tag{4.10}$$

It is well-known that higher-order structures impact the community structure in a graph, as well as the ability of many well-known methods to detect community structure. For example, the clustering coefficient, a well-known graph characteristic, can be defined via triangle counts (Watts and Strogatz (1998), see also Section 8.2.1). Therefore, we additionally define an FRC notion, which takes contributions of higher-order structures into account. To characterize such structure, we introduce the following notion: We say that two edges $e, \hat{e}$ are *parallel*, denoted as $e \| \hat{e}$, if they are adjacent to either the same vertex ($v \sim e, \hat{e}$) or the same face ($f \sim e, \hat{e}$), but not both. In the following, we consider the 2-complex version of FRC (Weber et al., 2017b):

$$\mathrm{Ric}_F(e) := \omega_e \left(\Big(\sum_{f \sim e} \frac{\omega_e}{\omega_f} \Big) + \frac{\omega_{v_1}}{\omega_e} + \frac{\omega_{v_2}}{\omega_e} - \sum_{\hat{e} \| e} \left| \sum_{f \sim \hat{e}, e} \frac{\sqrt{\omega_e \cdot \omega_{\hat{e}}}}{\omega_f} - \sum_{v \sim e, \hat{e}} \frac{\omega_v}{\sqrt{\omega_e \cdot \omega_{\hat{e}}}} \right| \right) \ . \tag{4.11}$$

Note that for each edge $\hat{e}$ parallel to e, only one of the two sums inside the absolute value will have any terms. Here, f denotes a 2-face of order k, such as a triangle ($k = 3$), a quadrangle ($k = 4$), a pentagon ($k = 5$), etc.

4.2.2 Computational Considerations

Notice that Equation (4.9) and Equation (4.10) give a simple, combinatorial curvature notion, which can be efficiently computed even on large-scale graphs. To compute the 2-complex FRC of an edge e exactly, one needs to identify all higher order faces in the neighborhood of an edge e and compute their respective contributions in Equation (4.11). This limits the scalability of this notion significantly. However, notice that the likelihood of finding a face of order k in the neighborhood of an edge decreases rapidly as k increases. Figure 7 illustrates this observation with summary statistics for simulations on two popular graph models (the SBM and RGG). We propose to approximate the 2-complex FRC by only considering triangular faces and ignoring structural information involving k-faces with $k > 3$. This choice reduces the computational burden of the 2-complex FRC significantly. We demonstrate below that experimentally, the performance of our FRC-based clustering algorithm does not improve if higher-order faces are taken into account for the curvature contribution (see Table 5). Specifically, we will utilize the following *augmented FRC*:

$$\mathrm{Ric}_F(e) := \omega_e \left(\Big(\sum_{\triangle \sim e} \frac{\omega_e}{\omega_\triangle} \Big) + \frac{\omega_{v_1}}{\omega_e} + \frac{\omega_{v_2}}{\omega_e} - \left[\sum_{\substack{e_{v_1} \sim e \\ e_{v_1} \| e}} \frac{\omega_{v_1}}{\sqrt{\omega_e \omega_{e_{v_1}}}} + \sum_{\substack{e_{v_2} \sim e \\ e_{v_2} \| e}} \frac{\omega_{v_2}}{\sqrt{\omega_e \omega_{e_{v_2}}}} \right] \right) . \quad (4.12)$$

To be clear, the sum $e_{v_1} \sim e, e_{v_1} \parallel e$ considers all edges e_{v_1} which are incident on v_1, hence share a vertex in common with e, but do not form a triangle with e.

We consider the following weighting scheme for faces, which utilizes Heron's formula to determine face weights via edge weights: Let $f = (e_i, e_j, e_k)$ denote a triangle in G, i.e., $e_i \sim e_j$, $e_j \sim e_k$ and $e_i \sim e_k$. Then we set

$$\begin{aligned} \omega_f &:= \sqrt{s(s-\omega_{e_i})(s-\omega_{e_j})(s-\omega_{e_k})} \\ s &= \frac{\omega_{e_i} + \omega_{e_j} + \omega_{e_k}}{2} . \end{aligned} \quad (4.13)$$

This weighting scheme was previously used in (Weber et al., 2017b).

5. Discrete Curvature on the Line Graph

In this section, we investigate the computation of discrete Ricci curvature on the line graph. Again, we consider two notions of curvature, Ollivier's (ORC) and Forman's (FRC). Specifically, we want to describe the relationship between a graph's curvature and the curvature of its line graph.

We have seen in Section 4 that curvature can be defined on the node and edge levels. Since nodes in the line graph correspond to edges in the underlying graph, it is natural to study the relationship between node-level curvature in the line graph and edge-level curvature in the original graph. In the context of the clustering applications considered in this paper, we are further interested in the insight that curvature can provide on differences between node clustering (based on connectivity in the original graph) and edge clustering (based on connectivity in the line graph). In this context, it is instructive to study the relationship between edge-level curvature in the line graph and edge-level curvature in the underlying graph.

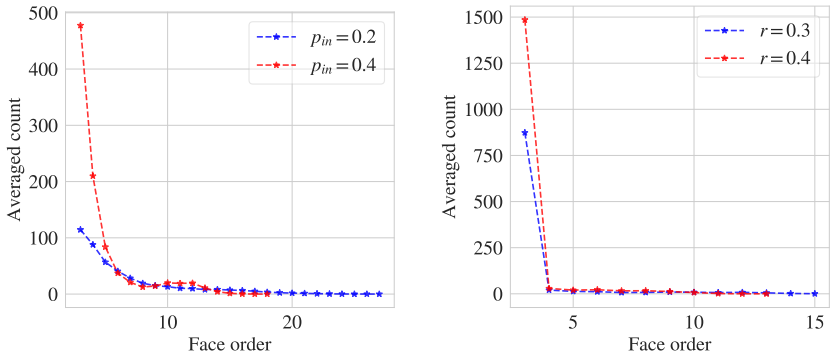


Figure 7: Number of k-faces in graphs obtained from the planted SBM of size $n = 100$, two equally-sized blocks, $p_{out} = 0.02$ (left), and the 2-d RGG of size $n = 100$ (right), where the results are averaged over $n_s = 50$ realisations of each set of parameters.

Recall that, given an unweighted, undirected graph $G = (V, E)$, its line graph is defined as $L(G) = (E, \mathcal{E})$, where an edge $\{\{u, v\}, \{r, s\}\} \in \mathcal{E}$ appears in the line graph whenever $|\{u, v\} \cap \{r, s\}| = 1$. We will make use of the following simple observation throughout the section:

Lemma 10 *Node degrees in the line graph are given as*

$$d_{\{u,v\}} = d_u + d_v - 2 \,,$$

where the left hand side denotes node degrees in the line graph $L(G)$, and the right hand side the degrees of the vertices u and v in G.

This formula follows from the fact that the edges connecting to $\{u, v\}$ in $L(G)$ are precisely those edges of the form $\{u, z\} \in E$ or $\{v, z\} \in E$, where $z \neq u, v$.

5.1 Numerical observations

We explore the relationship of (1) edge-level curvature in the original graph G and node-level curvature in the corresponding line graph L numerically, as well as (2) edge-level curvatures in the original graph G and its line graph L. Our empirical results consider the two notions of curvature introduced above: Forman's Ricci curvature (FRC) and Ollivier's Ricci curvature (ORC). Our data sets are described in Appendix B. To compute the classical FRC and ORC notions in the original graph G and its line graph L numerically, we use the built-in ORC computation both for edges and for nodes in GraphRicciCurvature (Ni et al., 2019). Our implementations for augmented FRC and our ORC approximation build on the code in this library. Results from each random graph model are obtained from $n_s = 50$ realizations.

5.2 Ollivier's Ricci Curvature

In this section, we establish theoretical evidence for the observed relationship between Ollivier's curvature (ORC) of a graph and its line graph.

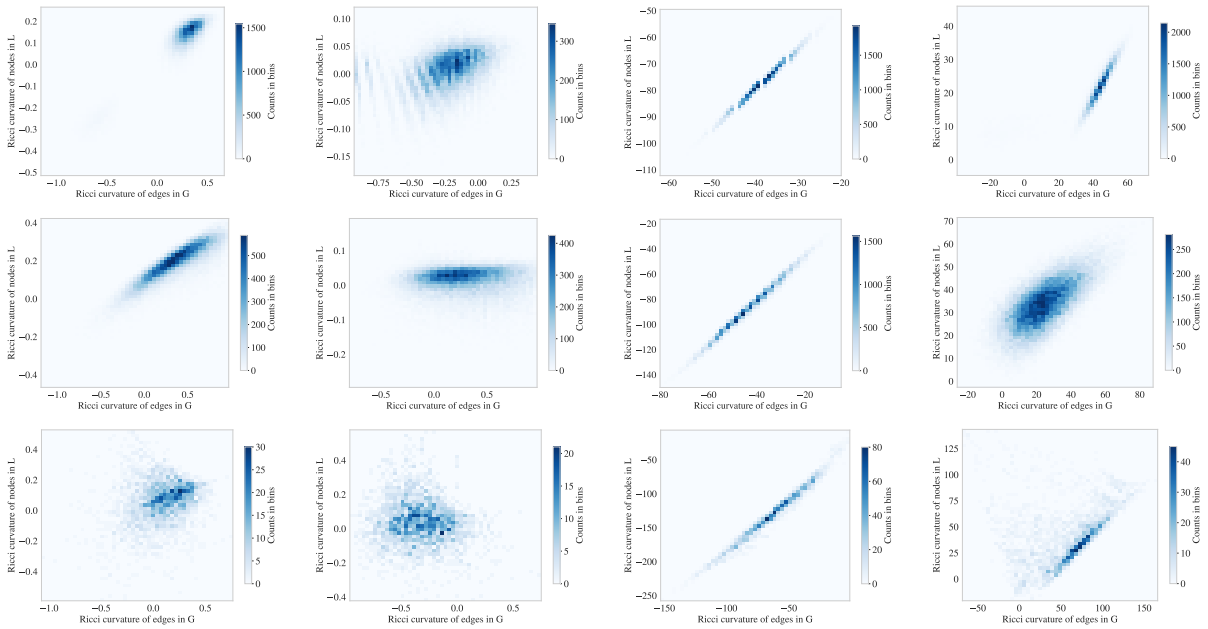


Figure 8: Correlation of the distributions of the classical ORC (left), the approximation of ORC from the average of the upper and lower bounds (middle left), the classical FRC (middle right), and the augmented FRC (right) **for edges in G versus that for vertices in L** where the networks are obtained from (top) the planted SBM of size $n = 100$, two equally-sized blocks, $p_{out} = 0.02$ and $p_{in} = 0.4$; (middle) the 2-d RGG of size $n = 100$, and radius $r = 0.3$; (bottom) the "Worm" data set. Notice that the range of curvature values varies between curvature notions, but the shape of the distributions has a close resemblence.

5.2.1 Unweighted case

Consider an unweighted graph G, and denote ORC in G and $L(G)$ by Ric_O^G, $\mathrm{Ric}_O^{L(G)}$, respectively. We prove combinatorial upper and lower bounds for ORC in $L(G)$, which can be computed solely from structural information in the underlying graph G. These results are the analogues of Equations (4.6), (4.7), but make use of the relationships between the line graph $L(G)$ and its base graph to avoid computing curvatures in the line graph.

Lemma 11 (Bounds on line graph ORC) *Let d_u denote the degree of the vertex $u \in V$ in the graph G, and $d_{\{u,v\}}$ the degree of the vertex $\{u, v\} \in E$ in the line graph $L(G)$. Then we have the following bounds on the ORC of edges in $L(G)$:*

1. *Upper bound on ORC:*

$$\mathrm{Ric}_O^{L(G)}(\{\{u,v\},\{v,w\}\}) \leq \frac{d_v - 2 + \mathbf{1}_E(\{u,w\})}{(d_u \vee d_w) + d_v - 2}\ ; \tag{5.1}$$

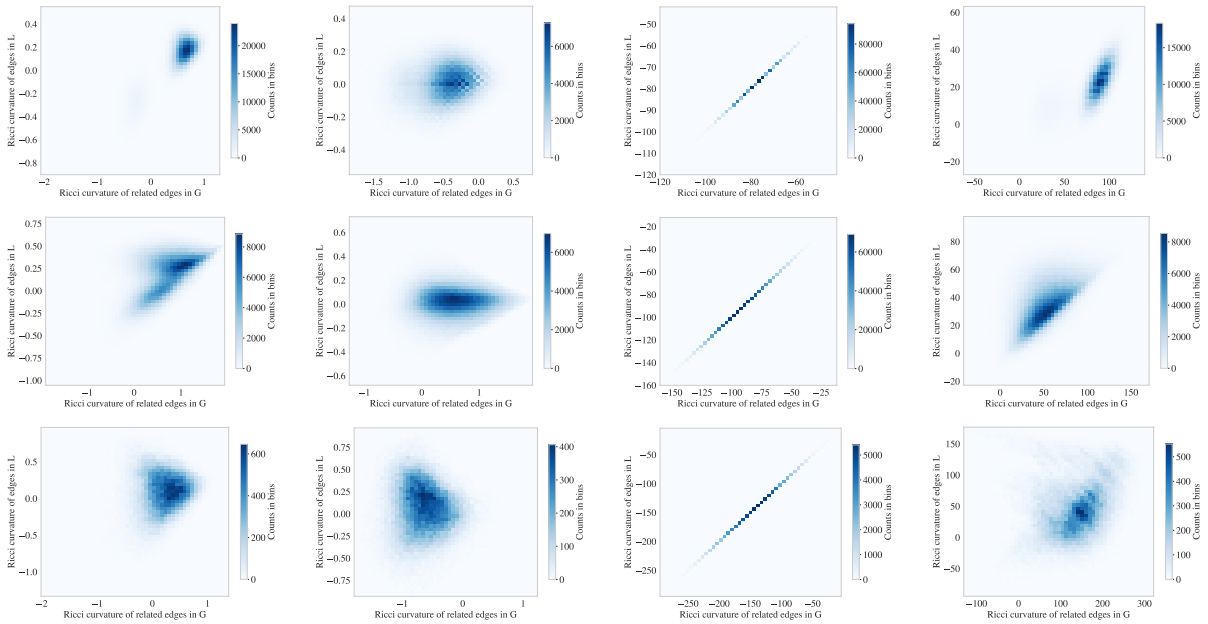


Figure 9: Correlation of the distributions of the classical ORC (left), the approximation of ORC from the average of the upper and lower bounds (middle left), the classical FRC (middle right), and the augmented FRC (right) **for edges in G versus that for edges in L** where the networks are obtained from (top) the planted SBM of size $n = 100$, two equally-sized blocks $p_{out} = 0.02$ and $p_{in} = 0.4$; (middle) the 2-d RGG of size $n = 100$, and radius $r = 0.3$; (bottom) the "Worm" data set. Notice that the range of curvature values varies between curvature notions, but the shape of the distributions has a close resemblance.

2. Lower bound on ORC:

$$\begin{aligned}\mathrm{Ric}_O^{L(G)}(\{\{u,v\},\{v,w\}\}) \geq &-\left(1-\frac{1}{d_u+d_v-2}-\frac{1}{d_v+d_w-2}-\frac{d_v-2+\mathbf{1}_E(\{u,w\})}{(d_u\wedge d_w)+d_v-2}\right)_+\\ &-\left(1-\frac{1}{d_u+d_v-2}-\frac{1}{d_v+d_w-2}-\frac{d_v-2+\mathbf{1}_E(\{u,w\})}{(d_u\vee d_w)+d_v-2}\right)_+\\ &+\frac{d_v-2+\mathbf{1}_E(\{u,w\})}{(d_u\vee d_w)+d_v-2}\,. \end{aligned} \tag{5.2}$$

Here, $\mathbf{1}_E(\{u,w\}) = 1$ if u, w are connected by an edge in G, and $\mathbf{1}_E(\{u,w\}) = 0$ otherwise.

A crucial feature of this lemma is that it does not require the computation of triangle counts in $L(G)$: It only requires evaluating the degrees of the constituent vertices, and the existence of the edge $\{u,w\} \in E$. The latter results in an additional triangle in the base graph G and, since $L(K_3)$ is isomorphic to K_3, an additional triangle in the line graph.

5.2.2 Weighted case

Importantly, we can show a version of these bounds in the weighted case, too. Let $\omega(\{u,v\},\{v,w\})$ denote the weight of the edge $\{\{u,v\},\{v,w\}\} \in \mathcal{E}$. Then for an edge $\{e_1,e_2\} \in \mathcal{E}$, Theorem 9 gives us the following result, which makes use of the notation from Section 4.1.3.

1. Upper bound on ORC:

$$\mathrm{Ric}_O(\{e_1,e_2\}) \leq 1 - \max\left\{ \sum_e \frac{d_{L(G)}(e,\mathcal{N})}{\omega(e_1,e_2)}(m_1(e)-m_2(e))_+, \sum_e \frac{d_{L(G)}(e,\mathcal{P})}{\omega(e_1,e_2)}(m_2(e)-m_1(e))_+ \right\}.$$

2. Lower bound on ORC:

$$\begin{aligned}\mathrm{Ric}_O(\{e_1,e_2\}) \geq 1 &- \sum_\ell \frac{\omega(\ell,e_1)}{\omega(e_1,e_2)} m_1(\ell) - \sum_r \frac{\omega(r,e_2)}{\omega(e_1,e_2)} m_2(r) \\ &- \sum_c \left[\frac{\omega(c,e_2)}{\omega(e_1,e_2)}(m_1(c)-m_2(c))_+ + \frac{\omega(c,e_1)}{\omega(e_1,e_2)}(m_2(c)-m_1(c))_+ \right] \\ &- \left| L_1 + X_1 - X_2 - \sum_c (m_2(c)-m_1(c))_+ \right| .\end{aligned}$$

Note that the upper bound for ORC in the line graph still requires one to compute the line graph. However, we can always use the trivial upper bound $\mathrm{Ric}_O(\{e_1,e_2\}) \leq 1$ if an approximation will suffice. See Section 7.4 for experimental results utilizing this approach.

Summary. In the applications discussed in the next section, we again approximate ORC as the arithmetic mean of the upper and lower bounds for the sake of computational efficiency:

Definition 12 (Approximate ORC in the Line Graph)

$$\widehat{\mathrm{Ric}_O}^{L(G)}(\{u,v\}) := \frac{1}{2}\Big(\big(\mathrm{Ric}_O^{L(G)}\big)^{up}(\{u,v\}) + \big(\mathrm{Ric}_O^{L(G)}\big)^{low}(\{u,v\}) \Big) . \tag{5.3}$$

In Figure 10, we compare this approximation to the original ORC for some simulated graphs. Note that the approximations in the right two columns can be computed only from information contained in the graph G, so that L does not need to be constructed.

5.3 Forman's Ricci Curvature

In this section, we establish a series of relationships between Forman's curvature of a graph and its line graph. We further discuss how higher order structures in the line graph may be incorporated in the curvature computation.

5.3.1 Unweighted case

Consider an unweighted graph G, and denote FRC in G and $L(G)$ by Ric_F^G, $\mathrm{Ric}_F^{L(G)}$, respectively. At first, we only consider the 1-complex curvature notion (Equations (4.9), (4.10)), which does not consider contributions of higher-order structures, such as triangles or quadrangles. We begin by investigating the relationship of edge-level Forman curvature in a graph G and the node and edge-level curvatures in its line graph $L(G)$.

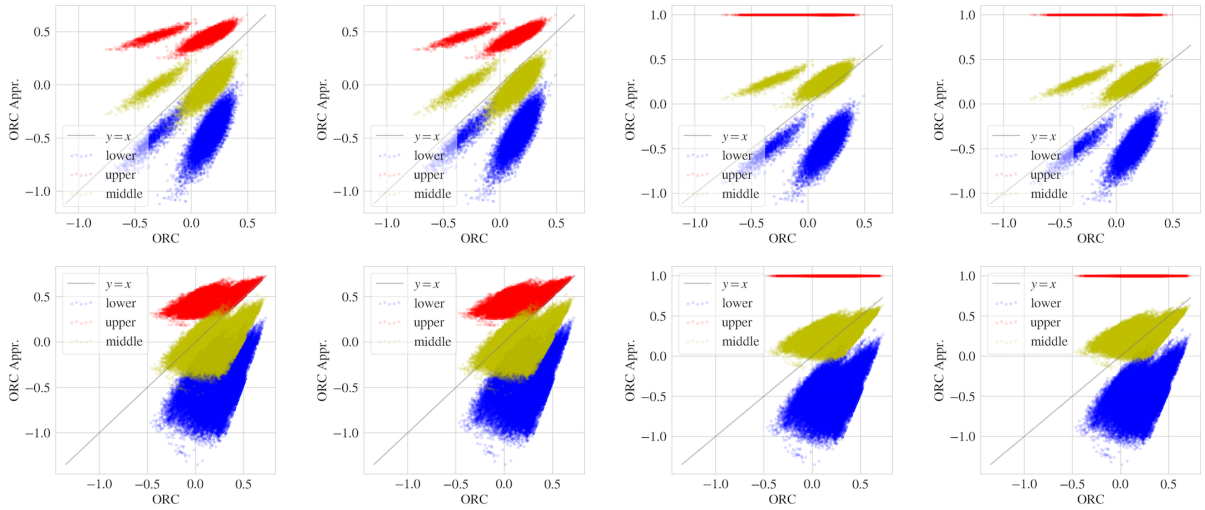


Figure 10: Scatter plot of the ORC on L obtained from optimization versus its approximation with L constructed (left two columns) and without (right two columns), where the optimization is done by solving the earth mover's distance exactly (left most and middle right) and the approximately with Sinkhorn (middle left and right most), and the original graph G is obtained from the planted SBM of size $n = 100$, two equally-sized blocks, $p_{out} = 0.02$ and $p_{in} = 0.4$ (top) and the 2-d RGG of size $n = 100$ and radius $r = 0.4$ (bottom).

Lemma 13 *Edge-level FRC in G relates to edge-level FRC in L as*

$$\mathrm{Ric}_F^{L(G)}(\{\{u,v\},\{v,w\}\}) = \mathrm{Ric}_F^G(\{u,v\}) + \mathrm{Ric}_F^G(\{v,w\})\,. \tag{5.4}$$

Edge-level FRC in G relates to node-level FRC in L as

$$\mathrm{Ric}_F^{L(G)}(\{u,v\}) = \mathrm{Ric}_F^G(u) + \mathrm{Ric}_F^G(v) - \mathrm{Ric}_F^G(\{u,v\})^2\,. \tag{5.5}$$

Importantly, this lemma allows us to compute node-level FRC in the line graph with respect to quantities in the original graph only. Specifically, the right hand side can be computed from edge-level FRC only, utilizing Equation (4.10). See the third columns in Figures 8 and 9 for an illustration of these relationships in a few simulated and real-data graphs.

5.3.2 Weighted case

When the vertex weights $\omega_u = 1$ for all $u \in V$, and using the line graph edge weights $\omega_{\{e_1,e_2\}} = \omega_{e_1}\omega_{e_2}$, we get the following formula for the relationship between the edge-level FRC in G versus the edge-level FRC in $L(G)$:

Lemma 14

$$\mathrm{Ric}_F^{L(G)}(\{\{u,v\},\{v,w\}\}) = \omega_{\{u,v\}}\left(2 - \sqrt{\frac{\omega_{\{v,w\}}}{\omega_{\{u,v\}}}}(2 - \mathrm{Ric}_F^G(\{u,v\}))\right) + \omega_{\{v,w\}}\left(2 - \sqrt{\frac{\omega_{\{u,v\}}}{\omega_{\{v,w\}}}}(2 - \mathrm{Ric}_F^G(\{v,w\}))\right).$$

5.3.3 Incorporating higher-order structures

Finally, we discuss how to account for contributions of higher-order structures, such as triangles or quadrangles, in the computation of curvature in the line graph. As discussed in Section 4, the curvature contributions of such structures are crucial in curvature-based clustering. We have demonstrated that the successful application of the FRC-based approach to single-membership community detection relies on the consideration of triangular faces, such as in our 2-complex FRC notion (Equation (4.11)). In the following, we will illustrate the necessity of including contributions of *quadrangular faces* in the curvature contribution on the line graph. We first define a 2-complex FRC notion on the line graph, which accounts for curvature contributions of triangular faces.

Let $\Gamma(v) = \{v' : \{v,v'\} \in E\}$ denote the set of neighbors for a vertex $v \in V$ in G. Suppose $\{u,v\},\{v,w\} \in E$ denote neighboring edges in G and $(\{u,v\},\{v,w\}) \in \mathcal{E}$ the corresponding edge in the line graph. We define the set of neighbors for an edge $\{u,v\}$ as follows:

$$\Gamma\big(\{u,v\}\big) := \Big\{\{u,u'\} : u' \in \Gamma(u)\backslash\{v\}\Big\} \cup \Big\{\{v,v'\} : v' \in \Gamma(v)\backslash\{u\}\Big\}$$

The set of triangles in the line graph containing $\{u,v\},\{v,w\}$ is then given by

$$\begin{aligned}\mathcal{F}_\Delta &:= \Big\{\big(\{u,v\},\{v,w\},e\big) :\ e \in \Gamma(\{u,v\}) \cap \Gamma(\{v,w\})\Big\} \\ &= \Big\{\big(\{u,v\},\{v,w\},e\big) :\ e \in \big\{\{v,v'\} : v' \in \Gamma(v)\backslash\{u,w\}\big\} \cup \big(\{\{u,w\}\} \cap E\big)\Big\}.\end{aligned}$$

Hence, there are only two possible structures in G, which give rise to triangles in its line graph L: (i) edges incident to the common node of the two edges under consideration, or (ii) the other endpoints of the two edges are connected with each other. The 2-complex FRC with triangle contributions can then be written as follows:

Definition 15 (2-complex FRC in the Line Graph ($k \leq 3$))

$$\begin{aligned}&\mathrm{Ric}_F\big(\{\{u,v\},\{v,w\}\}\big) \\ &= \omega_{\{u,v\},\{v,w\}}\Bigg(\sum_{f\in\mathcal{F}_\Delta}\frac{\omega_{\{u,v\},\{v,w\}}}{\omega_f} + \frac{\omega_{\{u,v\}}}{\omega_{\{u,v\},\{v,w\}}} + \frac{\omega_{\{v,w\}}}{\omega_{\{u,v\},\{v,w\}}} \\ &\quad - \Bigg[\sum_{x\in\Gamma(u)\backslash\{v,w\}}\frac{\omega_{\{u,v\}}}{\sqrt{\omega_{\{u,v\},\{v,w\}}\cdot\omega_{\{u,v\},\{u,x\}}}} + \sum_{x\in\Gamma(w)\backslash\{u,v\}}\frac{\omega_{\{v,w\}}}{\sqrt{\omega_{\{u,v\},\{v,w\}}\cdot\omega_{\{v,w\},\{w,x\}}}}\Bigg]\Bigg).\end{aligned}$$

In the special case of an unweighted network (i.e., $\omega_e = 1$ for all $e \in V$ and $\omega_{\{e,e'\}} = 1$ for all $\{e, e'\} \in \mathcal{E}$), we have

$$\mathrm{Ric}_F\Big(\{\{u,v\},\{v,w\}\}\Big) = 2 + \sum_{f \in \mathcal{F}_\Delta} \frac{1}{\omega_f} - \Big|\Big\{e \in \Gamma\big(\{u,v\}\big) : \big(\{u,v\},\{v,w\},e\big) \notin \mathcal{F}_\Delta\Big\}\Big| - \Big|\Big\{e \in \Gamma(\{v,w\}) : \{\{u,v\},\{v,w\},e\} \notin \mathcal{F}_\Delta\Big\}\Big| \ . \tag{5.6}$$

If we assign each triangle the same weight ω_Δ, this further simplifies the above equation as follows:

Lemma 16 (2-complex FRC in the Line Graph (triangles only))

$$\mathrm{Ric}_F(\{\{u,v\},\{v,w\}) = \frac{d_v}{\omega_\Delta} - d_u - d_w + 4 - \frac{2}{\omega_\Delta} + \left(\frac{1}{\omega_\Delta} + 2\right) \mathbf{1}_E(\{u,w\}) \ . \tag{5.7}$$

The lemma implies that the triangles-only 2-complex FRC in the line graph can still be largely determined by the degree of the nodes in the original graph G; the local triangle count enters via the terms $\mathbf{1}_E(\{u,w\})$ (which is 1 if $\{u,w\} \in E$ and 0 otherwise). A simple calculation shows that $\omega_\Delta = \sqrt{3}/4$, i.e., $1/\omega_\Delta + 2 = (4\sqrt{3}+6)/3 \approx 3.309$, while node degree could be as high as $O(|V|)$. As such, a triangle in G increases the curvature value in L by $1/\omega_\Delta + 2$. However, our experimental results (see Table 11) demonstrate that clustering methods based on the triangles-only 2-complex FRC can have low accuracy. While considering triangles was sufficient for FRC-based community detection on the original graph, we need to incorporate additional higher-order information to perform community detection on the line graph with high accuracy. Our experiments suggest that accounting for curvature contributions of quadrangles leads to a significant improvement in the performance of FRC-based mixed-membership community detection (see Table 11). We found that incorporating curvature contributions of k-faces with $k \geq 5$ does not improve performance, while significantly increasing the computational burden. This is expected, given the decreasing frequency of such structures (see Figure 7). These design choices lead to the following 2-complex FRC notion, which we will utilize in our FRC-based clustering method below: The set of quadrangles in the line graph, which contains $\{u,v\},\{v,w\}$ is given by

$$\mathcal{F}_\square := \Big\{\big(\{u,v\},\{v,w\},\{w,x\},\{x,u\}\big) : \ x \in (\Gamma(w) \cap \Gamma(u)) \backslash \{v\}\Big\} \ .$$

As such, there are two ways for quadrangles to form in the line graph: First, any quadrangle in the original graph G gives rise to one in the line graph L. Second, a quadrangle with a single chord $(\{u,v,w,x\},\{\{u,v\},\{v,w\},\{w,x\},\{x,u\},\{u,w\}\})$ still results in a quadrangle in L from $(\{u,v\},\{v,w\},\{w,x\},\{x,u\})$, since there are no edges from $\{u,v\}$ to $\{w,x\}$ or $\{v,w\}$ to $\{x,u\}$ in L.

Definition 17 (2-complex FRC in the Line Graph ($k \leq 4$))

$$\begin{aligned}
&\mathrm{Ric}_F\Big(\{\{u,v\},\{v,w\}\}\Big) \\
&= \omega_{\{u,v\},\{v,w\}}\Bigg(\sum_{f\in\mathcal{F}_\Delta}\frac{\omega_{\{u,v\},\{v,w\}}}{\omega_f}+\sum_{f\in\mathcal{F}_\square}\frac{\omega_{\{u,v\},\{v,w\}}}{\omega_f}+\frac{\omega_{\{u,v\}}}{\omega_{\{u,v\},\{v,w\}}}+\frac{\omega_{\{v,w\}}}{\omega_{\{u,v\},\{v,w\}}} \\
&-\Big|\sum_{\big(\{u,v\},\{v,w\},e_1,e_2\big)\in\mathcal{F}_\square}\frac{\sqrt{\omega_{\{u,v\},\{v,w\}}\cdot\omega_{e_1,e_2}}}{w_f} \\
&\quad-\sum_{x\in\Gamma(u)\setminus\Gamma_0(w)}\frac{\omega_{\{u,v\}}}{\sqrt{\omega_{\{u,v\},\{v,w\}}\cdot\omega_{\{u,v\},\{u,x\}}}}-\sum_{x\in\Gamma(w)\setminus\Gamma_0(u)}\frac{\omega_{\{v,w\}}}{\sqrt{\omega_{\{u,v\},\{v,w\}}\cdot\omega_{\{v,w\},\{w,x\}}}}\Big|\Bigg),
\end{aligned}$$

where $\Gamma_0(x)=\Gamma(x)\cup\{x\}$.

Our choice of weights for quadrangles is also based on geometric intuition utilizing Heron's Formula. Full details for this are given in Appendix A.4.

6. Algorithms

6.1 ORC-based approach

The curvature-based community detection algorithms (Algorithm 1) are built on the fact that *bridges* between communities are more negatively curved than edges within communities. We have discussed earlier that this can be observed in both graphs (in the single-membership case) and their line graphs (in the case mixed-membership communities). In this section, we provide a detailed discussion of Algorithm 1 with κ denoting Ollivier's Ricci curvature (ORC). We begin by giving some intuition for the differences in curvature values of bridges and internal edges, before formalizing the argument below. By construction, ORC is closely linked to the behavior of two random walks starting at neighboring vertices (Ollivier, 2009; Jost and Liu, 2014). Informally, they are more likely to draw apart if the edge between them has negative ORC, and to draw closer together otherwise. Random walks that start at nodes adjacent to a bridge (*bridge nodes*), typically move into the communities to which the respective nodes belong. As a consequence, they draw apart quickly. Random walks that start at non-bridge nodes, i.e., nodes adjacent to internal edges, are more likely to stay close to each other, as they remain within the same community. Hence, we expect bridges to have much lower ORC than internal edges, a fact that is easily confirmed empirically (see, e.g., Figure 4).

Remark 18 We note that an alternative notion of Ollivier's curvature (Equation (4.3)) was given by Lin et al. (2011), where node neighborhoods are endowed with the measure

$$m_u^\alpha(z) := \begin{cases} \alpha, & u=z \\ \frac{1-\alpha}{|\mathcal{N}_u|}, & u\sim z \\ 0, & \text{else} \end{cases}. \tag{6.1}$$

instead of the uniform measure. A weighted version of this curvature was considered in (Ni et al., 2019). Notice that for $\alpha > 0$, this curvature notion connects to *lazy* random walks starting at neighboring nodes. Here, the parameter α could be seen as controlling whether the random walk is likely to revisit a node, which in turn relates to a distinction between exploration of node neighborhoods in the style of "breadth-first search" vs. "depths-first search". A suitable choice of α is highly dependent on the topology of the graph; hence, replacing ORC with Equation (6.1) provides an additional means for incorporating side information on the structure of the underlying graph. However, in this work we restrict ourselves to the classical ORC notion.

6.1.1 Algorithm

Single-Membership Community Detection. We implement Algorithm 1 via ORC by setting $\kappa(\cdot) = \widehat{\mathrm{Ric}_O}(\cdot)$, i.e., the curvature is chosen to be the combinatorial ORC approximation proposed in Section 4.1. Notice that as the edge weights evolve under the Ricci flow, the graph is weighted, even if the input graph was unweighted. Consequently, we use weighted ORC curvature, constructing the approximation from the bounds in Theorem 9. We further choose equally-spaced cut-off points $\{x_i\}_{i=0}^{n_f}$, where $x_0 = \max_{\{u,v\}\in E} w_{u,v}^T$, $x_1 = x_0 - \delta, \ldots, x_{n_f} = ((x_0 - 1) \mod \delta + 1)$, and the step size for the cut-off points is set to be $\delta = 0.025$. Other hyperparameter choices include a constant step size $\nu = 1$, $\epsilon = 10^{-4}$, drop threshold $\epsilon_d = 0.1$, and stopping time $T = 10$, i.e., we evolve edge weights under Ricci flow for 10 iterations. Instances of Algorithm 1 via ORC were previously considered in (Ni et al., 2019; Sia et al., 2019; Gosztolai and Arnaudon, 2021). Both approaches computed ORC either exactly (via the earth mover's distance) or approximately (via Sinkhorn's algorithm). However, due to the higher computational cost of either variant of the ORC computation, the approaches were limited in their scalability (see discussion Section 6.1.3 below).

Mixed-Membership Community Detection. As in the single-membership case, we implement Algorithm 1 via our combinatorial ORC approximation (Equation (5.3)). The input graph G is the line graph of the underlying graph, which is constructed before the clustering procedure is started. All edge weights are initially set to 1, i.e., the input graph is unweighted. Hyperparameter choices are analogous to the single-membership case. Instance of Algorithm 1 via ORC were first considered in (Tian et al., 2023). Again, due to the high computational cost of the ORC computation, the approach was limited in its scalability.

6.1.2 Theoretical results.

Single-Membership case. To give theoretical intuition for curvature-based community detection algorithms in the single-membership case, we considered the following model class of graphs with community structure:

Definition 19 *Let $G_{a,b}$ $(a > b \geq 2)$ be a graph constructed as follows:*

1. *Construct a complete graph with b vertices $\{v_1, \ldots, v_b\}$.*
2. *For each $i = 1, \ldots, b$, introduce vertices u_{ij}, $j = 1, \ldots, a$, and make all possible connections between vertices in $\{v_i\} \cup \{u_{ij}\}_{j=1}^{a}$, resulting in b copies of complete graphs K_{a+1}.*

We show examples of graphs of the form $G_{a,b}$ in Figure 11. Notice that in a graph $G_{a,b}$ we have three types of edges:

1. *bridges* between communities, we denote vertices adjacent to bridges as *bridge nodes* and all others as *internal nodes*;
2. *internal edges* that connect a bridge node and an internal node;
3. *internal edges* that connect two internal nodes.

As before, we consider the following mass distribution on the node neighborhoods:

$$m_x = \begin{cases} \frac{1}{C} e^{-w_{x,y}}, & x \sim y \\ 0, & \text{else} \end{cases} . \tag{6.2}$$

Here, C denotes the normalizing function. This choice follows from the more general scheme in Equation (6.1) by setting $\alpha = 0$, $p = 1$. We assume that at initialization, all edges are assigned weight 1.

Theorem 20 *Let the edge weights in $G_{a,b}$ evolve under the Ricci flow $w_i^{t+1} \leftarrow (1 - \kappa_i^t) w_i^t$ (where κ_i^t denotes ORC for the edge type i using the weights at time t).*

1. *The weights of internal edges (type (3)) contract asymptotically, i.e., $w_3^t \to 0$ as $t \to \infty$.*
2. *The weights of bridges are larger than those of internal edges of types (2) and (3), i.e., $w_1^t > w_2^t$ and $w_1^t > w_3^t$ for all $t > 0$.*

Remark 21 *We note that the class of graphs $G_{a,b}$ was previously considered in (Ni et al., 2019), which also give asymptotic guarantees for the evolution of edge weights under the Ricci flow. However, they assume a different mass distribution, hence, their asymptotic results are not directly applicable to our setting.*

We will present the full proof of this theorem in Appendix A.5, but we give a sketch of the proof here. The argument proceeds by induction, first calculating the exact ORC curvatures of the three edge types at iteration 1, and then calculating the exact ORC curvatures of the three edge types at iteration t, given the weights w_i^t. In each of these steps, we construct transport plans, as well as 1-Lipschitz functions with equal objective function values to show their optimality. This proves that we have the exact values of the Wasserstein distances. We use these exact formulas to show that the evolution of the weights has the properties given in the statement.

Mixed-Membership case. We modify the model class introduced in Definition 19 to account for mixed-membership community structure:

Definition 22 *Let $L_{a,b}$ ($a > b \geq 2$) be a graph constructed as follows:*

1. *Construct a star-shaped graph, consisting of a center node of degree b and b leaf nodes of degree 1.*

2. Replace each leaf node with a star-shaped graph with $a+1$ vertices.

Notice that $L_{a,b}$ consists of b blocks (communities) of size $a+2$ with one common node, i.e., the center node is a member of each of the b communities.

It is easy to see that the dual of a graph $L_{a,b}$ is a graph of the form $G_{a,b}$. Consequently, we recover guarantees for curvature-based clustering on the line graph from Theorem 20.

6.1.3 Computational Considerations

The two key bottlenecks in ORC instances of Algorithm 1 are (1) the computation of the W_1-distance in the ORC computation (both single- and mixed-membership case), and (2) the construction of the line graph (mixed-membership case). Following our discussion in Section 4.1, we can circumvent bottleneck (1) by approximating ORC with the arithmetic mean of its upper and lower bounds, which are given by Equation (4.8) in the original graph (use in single-membership case) and Equation (5.3) in the line graph (use in mixed-membership case). This reduces the complexity of the ORC computation to $O(|E|d_{\max})$, in comparison with previous approaches (Ni et al., 2019; Sia et al., 2019), which achieved at best $O(|E|d_{\max}^2)$ via Sinkhorn's approximation of the Wasserstein distance. The second bottleneck may also be circumvented in special cases. We discuss the details in Section 7.4 below.

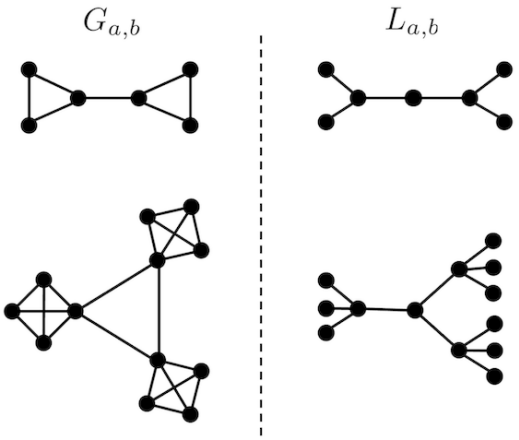


Figure 11: Examples of graphs of the model classes $G_{a,b}$ and $L_{a,b}$ (top: $a=b=2$, bottom: $a=b=3$). For both pairs, $G_{a,b}$ is the line graph of $L_{a,b}$.

6.2 FRC-based approach

Finally, we discuss the second instance of Algorithm 1, where κ denotes Forman's Ricci curvature (FRC). Again, we observe that, typically, bridges have lower FRC than internal edges in communities, which can be exploited for curvature-based community detection. As in the previous section, we first provide some informal intuition on this observation, before formally introducing the algorithm. Notice that bridge nodes have typically a high node degree, as they connect not only to other nodes within the same community, but also to nodes in other communities. It is easy to see from the definition (recall, $\mathrm{Ric}_F(e) = 4 - d_{v_1} - d_{v_2}$) that, as a result, bridges have usually lower curvature than internal edges in unweighted networks. This imbalance in curvature values is iteratively reinforced as edge weights evolve under the Ricci flow. In particular, we see that, in the case of bridges, the 2-complex FRC (Equation (4.11))

$$\mathrm{Ric}_F(e) := \omega_e \left(\left(\left(\sum_{f \sim e} \frac{\omega_e}{\omega_f} \right) + \frac{\omega_{v_1}}{\omega_e} + \frac{\omega_{v_2}}{\omega_e} \right) - \sum_{\hat{e} \| e} \left| \sum_{f \sim \hat{e}, e} \frac{\sqrt{\omega_e \cdot \omega_{\hat{e}}}}{\omega_f} - \sum_{v \sim e, \hat{e}} \frac{\omega_v}{\sqrt{\omega_e \cdot \omega_{\hat{e}}}} \right| \right)$$

is increasingly dominated by the second term, due to the high degree of the adjacent vertices and, consequently, the large number of parallel edges. This is reinforced as the edge weight increases under Ricci flow, which decreases the first term, resulting in negative curvature with high absolute value. In contrast, for internal edges, the second term has a smaller magnitude due to the smaller degrees of the adjacent vertices. This is again reinforced as the edge weight decreases under Ricci flow.

6.2.1 Algorithm

Single-Membership Community Detection. We implement Algorithm 1 via FRC by setting $\kappa(\cdot) = \mathrm{Ric}_F(\cdot)$, chosen to be the 2-complex FRC with triangle contributions (Equation (4.12)). In the absence of given edge weights, we initialize edge weights to 1 as before. Adaptive step sizes are chosen proportional to the inverse of the highest absolute curvature of any edge in a given iteration, i.e., $\nu_t = 1/(1.1 \times \max_{\{u,v\}\in E} |\kappa^t_{u,v}|)$. In our experiments below, we evolve edge-weights for $T = 10$ iterations under the Ricci flow. To infer communities, cut-off points are chosen as $\{x_i\}_{i=0}^{n_f}$, where $x_0 = \max_{\{u,v\}\in E} w^T_{u,v}$, $x_1 = \max_{\{u,v\}\in E, w^T_{u,v}<x_0} w^T_{u,v}$, $\ldots$, $x_{n'_f} = w^T_q$, $x_{n'_f+1} = x_{n'_f} - \delta$, $\ldots$, $x_{n_f} = ((x_{n'_f} - 1.1 w_{\min}) \bmod \delta + 1.1 w_{\min})$. Here, w^T_q is the qth quantile of all weights; we set $q = 0.999$, step size $\delta = 0.25$, and $w_{\min} = \min_{\{u,v\}\in E} w^T_{u,v}$. Note that this differs from the uniformly spaced cut-off points in the ORC case. Under FRC-based Ricci flow, the magnitude of some edge weights grows rapidly, resulting in a fat-tailed edge weight distribution. Hence, adjusting the spacing between cut-off points accordingly reduces the number of cut-off points needed to achieve high accuracy in cluster assignments. We remark that a special case of Algorithm 1 was previously proposed in (Weber et al., 2018), although considering the classical FRC notion (Equation (4.9)) instead of the 2-complex FRC and utilizing a simpler scheme to identify bridges between communities using FRC.

Mixed-Membership Community Detection. We implement Algorithm 1 via FRC on the line graph, by setting $\kappa(\cdot,\cdot) = \mathrm{Ric}_F(\cdot,\cdot)$, chosen to be the 2-complex FRC with triangle *and quadrangle* contributions (Definition 17). In the mixed-membership case, bridges between communities appear only in the line graph. Hence, we first construct the line graph before the actual clustering procedure is started. The input to Algorithm 1 is the pre-computed line graph. All edge weights are initially set to 1, i.e., the input graph is unweighted. The adaptive step sizes, stopping time and cut-off points are chosen analogously to the single-membership case. To the best of our knowledge, FRC-based approaches have not been considered previously in this mixed-membership setting.

6.2.2 Computational Considerations

Forman's Ricci curvature for edges is given by a simple combinatorial formula, which can be computed in linear time. The variants considered in this work (2-complex Forman curvature with triangle and quadrangle contributions) require an additional subroutine, which computes the number of triangles that include the respective edge. Counting the number of triangles in the 2-hop neighborhood is linear in the node degree, hence the overall complexity is $O(|E| d_{\max})$, where $d_{\max}$ denotes the maximum node degree. Similarly, computing the number of quadrangles in the same 2-hop neighborhood is quadratic in the

degree, hence the overall complexity is $O(|E|d_{\max}^2)$. Since this subroutine requires only local information, it is easily parallelizable.

7. Experiments

7.1 Network Data

Synthetic Data. In order to explore the interaction between discrete Ricci curvature and the structure of complex networks, we have considered both Stochastic Block Model and Random Geometric Graphs in our previous analysis. In this section, we systematically examine the performance of the algorithms on single- and mixed-membership Stochastic Block Models, where the ground-truth community membership is available. We have introduced the two variants of the Stochastic Block Model above in Section 2.

Real network data. To test our methods on real data, we consider two data sets, (i) collaboration networks from DBLP (Yang and Leskovec, 2015), and (ii) ego-networks from Facebook (Leskovec and Mcauley, 2012), for which ground truth labels are available through SNAP (Leskovec and Krevl, 2014). Specifically, the collaboration network is constructed by a comprehensive list of research papers in computer science provided by the DBLP computer science bibliography. Here, an edge from one author to another indicates that they have published at least one joint paper. There are intrinsic communities defined by the publication venue, e.g., journals or conferences. Here, we maintain the choice of two communities, and randomly select two venues which have at least one node of mixed membership. We report the results for two representative networks: "DBLP-1" from publication venues no. 12 and 1654, and "DBLP-2" from publication venues no. 12 and 2592. While in the second data set (ego-networks), all the nodes are friends of one central user, and the friendship circles set by this user can be used as ground truth communities. We carried out the preprocessing in a similar manner as (Zhang et al., 2020), and then select two networks for which the modularity of the ground-truth communities is greater than 0.5: "FB-1" for no. 414, and "FB-2" for no. 1684. To better understand the characteristics of the different real networks, we provide the following summary statistics for each network (see Table 1): (i) average node degree d, (ii) degree heterogeneity measured by the standard deviation of node degrees σ_d over d, (iii) the actual proportion of nodes with overlapping membership $\hat{\pi}_o$, and (iv) the modularity of the ground-truth community. We note that the Facebook networks tend to be denser, with more nodes of mixed membership, while the collaboration networks tend to have heterogeneous degrees and communities that are not well described by modularity.

	n	$\lvert E\rvert$	$\lvert\mathcal{E}\rvert$	k	d	σ_d/d	$\hat{\pi}_o$	Modularity
DBLP-1	82	173	867	2	4.22	0.65	0.012	0.15
DBLP-2	54	142	956	2	5.26	0.69	0.019	0.18
FB-1	128	1593	45635	3	24.89	0.44	0.055	0.53
FB-2	621	12399	718267	5	39.93	0.69	0.004	0.52

Table 1: Summary statistics of the real networks.

7.2 Evaluating clustering accuracy

To evaluate the accuracy of a clustering in the single- and mixed-membership settings, we introduce the following two notions of *Normalized Mutual Information* (NMI). When each node can only belong to one community, the classic NMI is defined by considering each clustering result as a random variable and then comparing the information contained in them (Strehl and Ghosh, 2002). Specifically, let X, Y be the random variables described by two different cluster labels. Let $I(X, Y)$ denote the mutual information between X and Y, and let $H(X), H(Y)$ denote the entropy of X, Y, respectively. Then NMI is defined as

$$NMI(X, Y) = \frac{I(X, Y)}{mean(H(X), H(Y))}, \tag{7.1}$$

where $mean(H(X), H(Y))$ denote the mean of the two, and we consider the arithmetic mean in our experiments, $mean(H(X), H(Y)) = (H(X) + H(Y))/2$. The NMI ranges from 0 to 1, where value 1 corresponds to perfect matching, or exact recovery if one of the clustering result is ground truth.

We note that the classic NMI is not well-defined for mixed-membership community detection, thus we later consider the following extended NMI (Lancichinetti et al., 2009). Specifically, for communities $C_1, C_2, \ldots, C_k$, we now express the community membership of each node i as a binary vector of length k. $(z_i)_l = 1$ if node i belongs to C_l; $(z_i)_l = 0$ otherwise. The l-th entry of this vector can then be viewed as a random variable Z_l, whose probability distribution is given by $P(Z_l = 1) = n_l/n$ and $P(Z_l = 0) = 1 - P(Z_l = 1)$, where $n_l = |C_l|$ and $n = |V|$. The same holds for the random variable Y_h associated with another set of communities $C'_1, C'_2, \ldots, C'_{k'}$. Both the empirical marginal probability distribution P_{Z_l} and the joint probability distribution $P(Z_l, Y_h)$ are used to further define entropy $H(\mathbf{Z})$ and $H(Z_l, Y_h)$. The conditional entropy of Z_l given Y_h is defined as $H(Z_l|Y_h) = H(Z_l, Y_h) - H(Y_h)$. The entropy of Z_l with respect to the entire vector $\mathbf{Y}$ is based on the best matching between Z_l and the components of $\mathbf{Y}$ given by

$$H(Z_l|\mathbf{Y}) = \min_{h \in \{1,2,\ldots,k'\}} H(Z_l|Y_h).$$

The normalized conditional entropy of $\mathbf{Z}$ with respect to $\mathbf{Y}$ is

$$H(\mathbf{Z}|\mathbf{Y}) = \frac{1}{k} \sum_l \frac{H(Z_l|\mathbf{Y})}{H(Z_l)}.$$

In the same way, we can define $H(\mathbf{Y}|\mathbf{Z})$. Finally, the extended NMI for two sets of communities $C_1, C_2, \ldots, C_k$ and $C'_1, C'_2, \ldots, C'_{k'}$ is given by

$$NMI(\mathbf{Z}|\mathbf{Y}) = 1 - [H(\mathbf{Z}|\mathbf{Y}) + H(\mathbf{Y}|\mathbf{Z})]/2 \, .$$

Hence to use the extended NMI, we convert the label vector y to a binary assignment by first normalizing it by its 2-norm and then thresholding each of its elements by $0.8/k$.

7.3 Single-Membership Community Detection

Benchmarking. For the first set of experiments, we explore the performance of curvature-based methods on planted SBMs with two blocks and various choices of parameters. We

start with graphs of two equally sized blocks, fixed size $n = 1000$ and probability $p_{out} = 0.01$, while changing the probability p_{in} from 0.05 to 0.2. We then explore the critical region when p_{in} is close to p_{out} and the two blocks have different sizes, n_1, n_2. For each set of parameters, we generate $n_s = 10$ graphs and run our algorithms for both ORC and FRC, together with other popular community detection methods, the Louvain algorithm ("Louvain") and variants of Spectral clustering ("Spectra"). Specifically, for ORC, the optimal transport can be obtained by solving the earth mover's distance exactly ("ORC-E") or approximately with Sinkhorn ("ORC-S"), while we can also approximate it by the mean of the upper and lower bounds we have developed ("ORC-A"). For FRC, we can consider the 1-d version as in Equation (4.9) where no face information is included ("FRC-1"), the augmented version with triangular faces as in Equation (4.11) ("FRC-2") and also the one with further quadrangular faces as in Definition 17 ("FRC-3"). For the Louvain algorithm, we run it for 100 times, construct the co-occurrence matrix of the clustering results, and then run the algorithm one more time on the graph constructed by the co-occurrence matrix, in order to obtain consistent communities. For Spectral Clustering, we present the results from the following variants: (i) the highly optimized version in `scikit-learn` (Pedregosa et al., 2011) together with a grid search for the number of clusters, from 2 up to 10 ("Spectra-S"), and (ii) automatic embedding dimension selection (Zhu and Ghodsi, 2006), followed by k-means clustering ("Spectra-A"). In Appendix C.1, we provide further details for the single-membership community detection, where we also vary the network size, compare different optimization methods for ORC, and explore the relationship between ORC and FRC. We also compare the curvature-based methods with other variants of Spectral Clustering in Appendix C.3, where instead of directly using the adjacency matrix, we can construct an affinity by (iii) radius basis functions or (iv) nearest-neighbor techniques.

We report the mean NMI, runtime, and their standard deviations (SD) for all methods. To maintain the detectability of the communities, we require the generated graphs to have modularity greater than 0.4 with the ground-truth communities if possible, or choose those with the highest modularity within 50 realizations. Our experimental results (see Table 2) demonstrate that the ORC approach outperforms most reference methods, and successfully recovers single-membership community structure, with an NMI greater than 0.8, as we vary the density via p_{in}. We note that, unlike the curvature-based methods, "Spectra-S" requires the number of communities as an input parameter. This makes the problem of detecting communities considerably easier. For a fairer comparison, we perform a grid search over this parameter, however, since this includes the true number of communities, this can still be considered an unfair advantage. In addition, the implementation that is used in our experiments is highly optimized. Despite these caveats, we observe that curvature-based methods achieve comparable performance on larger graphs with more heterogeneous community structures. On synthetic data (SBM), as p_{in}, p_{out} become closer to each other (i.e., close to the community detection threshold), we observe that the drop in performance is less significant in the ORC approach than in the reference methods (see Table 4). This indicates that curvature-based methods could be advantageous in regimes in which community detection is challenging with existing approaches. We only show the results from ORC-E, since ORC-S has almost the same performance as ORC-E.

Furthermore, our experiments show that leveraging our proposed combinatorial ORC approximation is a viable alternative that can achieve performance comparable to the Lou-

vain algorithm and the Spectral Clustering. This is especially pronounced in denser graphs, where computing exact ORC can be prohibitively expensive. Being computationally very efficient overall, the FRC approach can also retrieve the single-membership community structure when the graph is sufficiently dense. Here, only the results from FRC-2 are included for the reasons that we will discuss later. See Appendix C.1 for more results.

p_{in}	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-2
0.05	0.998 (0.004)	0.997 (0.007)	0.997 (0.007)	0.997 (0.008)	0.225 (0.053)	0.176 (0.018)
0.1	1 (0)	1 (0)	1 (0)	1 (0)	0.993 (0.005)	0.275 (0.182)
0.15	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	0.990 (0.031)
0.2	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)

Table 2: Mean (SD) of NMI from different methods on planted two-block SBMs with $n_1 = 500, n_2 = 500$, $p_{out} = 0.01$, and varying probabilities p_{in}.

p_{in}	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-2
0.05	38.16 (0.892)	1.203 (0.098)	0.415 (0.046)	7.537 (0.518)	6.084 (0.507)	4.100 (0.320)
0.1	33.85 (0.451)	1.115 (0.032)	0.385 (0.013)	13.57 (0.333)	7.496 (0.293)	4.932 (0.192)
0.15	37.70 (0.533)	1.111 (0.027)	0.387 (0.010)	24.85 (1.326)	11.35 (0.644)	7.526 (0.668)
0.2	42.97 (0.794)	1.124 (0.037)	0.395 (0.014)	41.78 (1.285)	19.32 (1.331)	14.85 (1.239)

Table 3: Mean (SD) of runtime (in seconds) from different methods on planted two-block SBMs with $n_1 = 500, n_2 = 500$, $p_{out} = 0.01$, and varying probabilities p_{in}.

p_{in}	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-2
0.02	0.019 (0.010)	0.160 (0.047)	0.056 (0.042)	0.213 (0.011)	0.126 (0.006)	0.128 (0.023)
0.04	0.008 (0.004)	0.018 (0.007)	0.007 (0.003)	0.170 (0.007)	0.132 (0.006)	0.114 (0.023)
0.06	0.003 (0.002)	0.010 (0.003)	0.003 (0.002)	0.151 (0.010)	0.128 (0.005)	0.108 (0.050)
0.08	0.004 (0.002)	0.012 (0.003)	0.004 (0.002)	0.092 (0.049)	0.130 (0.009)	0.078 (0.045)

Table 4: Mean (SD) of NMI from different methods on planted two-block SBMs with $n_1 = 250, n_2 = 750$, and varying probabilities p_{in}, p_{out} where $p_{in} - p_{out} = 0.01$.

Rationale for 2-complex FRC. For FRC, one could consider including contributions of higher-order faces in order to further improve the performance of community detection. Our experimental results (see Table 5) suggest that, while including triangle information is crucial, it is unclear whether encoding quadrangle leads to an improvement in accuracy. In dense graphs, including such information may even decrease performance. An exception is bipartite graphs, where quadrangle counts encode salient information.

In addition, the computation of quadrangle counts is expensive, resulting in a significant computational overhead, while the drop in efficiency is significant. We further observe that FRC-2 (which includes only triangle information) has the strongest correlation with the clustering coefficients, compared with FRC-1 and FRC-3 (see Figure 16 in Appendix C.1).

Towards an algorithm with integrated stopping criteria. In Algorithm 1, instead of choosing T *a priori*, we can integrate the choice with intermediate modularity check. One

p_{in}	FRC-1		FRC-2		FRC-3	
	NMI	runtime	NMI	runtime	NMI	runtime
0.05	0.107 (0.087)	10.95 (1.386)	0.177 (0.013)	11.14 (1.447)	0.220 (0.100)	18.28 (15.99)
0.1	0.106 (0.087)	16.73 (1.103)	0.229 (0.118)	17.48 (1.075)	0.077 (0.155)	202.7 (5.996)
0.15	0.073 (0.089)	23.35 (0.353)	0.990 (0.013)	27.39 (0.992)	0.019 (0.056)	101.4 (133.7)
0.2	0.054 (0.083)	29.91 (0.571)	1 (0)	39.79 (1.702)	0.092 (0.092)	111.9 (1.680)

Table 5: Mean (SD) of NMI and runtime (in seconds) from different FRC-based methods on planted two-block SBMs with $n_1 = 500, n_2 = 500$, $p_{out} = 0.01$ and varying probabilities p_{in}.

approach is to perform the modularity check at each iteration t, and record the best modularity value, Q^t_{best}. Then we stop the process when the best modularity value converges, where $(Q^t_{best} - Q^{t-1}_{best})/Q^{t-1}_{best} < \epsilon_d$. With such integrated stopping criteria with intermediate modularity check, our experiments indicate that the algorithm usually stops in 2 or 3 steps, with comparable performance with the original version of Algorithm 1; see Table 6. Hence, on the one hand, the choice of $T = 10$ in our experiments is reasonable to provide sufficiently good results, while not being far from the critical value for such good results. On the other hand, fine-tuning the stopping criteria in the integrated version is a promising direction to pursue in the future to further improve the scalability of the algorithm.

n	ORC-E	ORC-int	iterations
100	0.888 (0.106)	0.875 (0.108)	3.9 (1.136)
500	1 (0)	1 (0)	2.0 (1.244)
1000	1 (0)	1 (0)	2.0 (1.110)

Table 6: Mean (SD) of results from ORC-based methods on planted SBMs with two equally-sized blocks, $p_{in} = 0.1$, $p_{out} = 0.01$ and varying sizes n, where the first column corresponds to the NMI from Algorithm 1, the second column corresponds to the NMI from Algorithm 1 with intermediate modularity check and integrated stopping criteria, and the third column corresponds to the number of iterations in the integrated version.

7.4 Mixed-Membership Community Detection

Benchmarking. To demonstrate the performance of our algorithms, we generate graphs from planted MMBs with two equally-sized blocks $n_1 = n_2$ and different configurations. Here, we fix the network size of each block to be $n_1 = n_2 = 150$, $n_o = 1$ node of mixed membership and $p_{out} = 0$, while varying the probability p_{in} from 0.05 up to 0.2, and defer the results of varying the network size to Appendix C.2. For each set of parameters, we construct $n_s = 10$ graphs, and run our algorithms for both ORC and FRC, together with the Louvain algorithm and Spectral clustering, in the same way as in the single-membership setting, but on the line graphs $L(G)$. We report the mean NMI, runtime, and their standard deviations (SD) of the methods. For the purposes of detectability of communities, we consider graphs that have modularity greater than 0.4 with the ground-truth communities if possible or the ones with the highest modularity within 50 realizations. Our experimental results (see

Table 7 and Appendix C.2) demonstrate that the ORC-based approaches outperform most reference methods, where the ORC approach successfully recovers the mixed-membership community structure in almost all cases as we vary the probability p_{in}, while the FRC approach can also retrieve the mixed membership when the network is sufficiently dense. Since again the performance of ORC-S is almost the same as ORC-E, we only include the results from ORC-E. For the reasons that we have demonstrated in Section 5.3.3 and will discuss later, we only include the results from FRC-3.

p_{in}	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-3
0.05	0.957 (0.085)	0.997 (0.006)	0.442 (0.197)	0.996 (0.007)	0.590 (0.163)	0.657 (0.213)
0.1	0.968 (0.064)	0.999 (0.002)	0.940 (0.099)	0.998 (0.003)	0.683 (0.267)	0.730 (0.262)
0.15	0.984 (0.047)	0.999 (0.002)	0.971 (0.090)	0.999 (0.002)	0.709 (0.213)	0.800 (0.165)
0.2	1 (0)	1 (0)	0.982 (0.053)	1 (0)	0.714 (0.140)	0.814 (0.174)

Table 7: Mean (SD) of the extended NMI from different methods on planted two-block MMBs with $n_1 = 150$, $n_2 = 150$, $n_o = 1$ node of mixed membership, $p_{out} = 0$, and varying probabilities p_{in}.

p_{in}	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-3
0.05	73.83 (4.757)	1.267 (0.068)	0.578 (0.373)	10.96 (1.730)	9.484 (0.585)	18.56 (2.449)
0.1	381.9 (44.93)	5.309 (0.601)	2.244 (0.305)	44.82 (5.303)	32.88 (6.112)	49.43 (14.67)
0.15	909.1 (69.85)	13.17 (0.802)	5.847 (0.452)	126.4 (21.195)	73.18 (6.555)	82.58 (9.904)
0.2	1465 (63.71)	26.78 (1.587)	12.05 (0.621)	241.6 (36.52)	123.3 (7.045)	125.2 (18.35)

Table 8: Mean (SD) of runtime (in seconds) from different methods on planted two-block MMBs with $n_1 = 150$, $n_2 = 150$, $n_o = 1$ node of mixed membership, $p_{out} = 0$, and varying probabilities p_{in}.

ORC has comparable performance to spectral clustering on most datasets, and can outperform when the network has asymmetric communities, as shown in Table 4. Furthermore, we demonstrate the performance of our algorithms on real data via the two data sets described in Section 7.1. As in the synthetic networks, the ORC-based method outperforms the reference methods in most real networks, and the only case where the ORC-based method is ranked second is "FB-1", where the community structure is better captured by the Modularity, as implied by Table 1. Meanwhile, the FRC-based method can also outperform the reference methods, such as in "DBLP-2", indicating that the FRC can better characterize the underlying community structure in some cases. The results from the proposed approximation of ORC have more variance, which is expected since the real data has more variance in the structure of the networks and some may lie in the region where the approximation is relatively far from the exact value. However, the approximated version can also have comparable performance in most cases. We also observe an improvement of runtime from computing ORC by Sinkhorn optimization in "FB-2" dataset (ORC-E: 5523s; ORC-S: 4941s; ORC-A: 2530s), since we allocate much larger memory for the computation (250G here instead of 25G in Appendix C.1).

Towards avoiding line graph construction. We note that only the upper bound requires global information of the line graph, while local information is sufficient to compute

	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-3
DBLP-1	0.535	0.579	0.447	0.689	0.484	0.244
DBLP-2	0.652	0.708	0.358	0.969	0.463	0.737
FB-1	0.887	0.872	0.581	0.872	0.680	0.680
FB-2	0.516	0.652	0.510	0.652	0.242	0.287

Table 9: Extended NMI from different methods on the real data.

the lower bound. Hence, one step towards avoiding the construction of line graph is to consider an upper bound that can be obtained without the information of the whole graph while performing sufficiently well in the clustering tasks. Specifically, here we consider 1 as the upper bound, and denote the method using the average of 1 and the lower bound by "ORC-A1". Our numerical results indicate that ORC-A1 can further improve the efficiency from ORC-A while not significantly affecting the accuracy (see Table 10, where the NMI information is ignored because their values are close and all three methods can find the ground-truth communities when $p_{in} = 0.2$).

p_{in}	ORC-E	ORC-A	ORC-A1
0.1	426.8	182.2	144.6
0.2	2878	732.7	530.0

Table 10: Runtime (in seconds, including line graph construction) from different ORC-based methods on one sample of planted two-block MMBs with $n_1 = n_2 = 250$, $n_o = 1$ node of mixed membership, $p_{out} = 0$, and varying probabilities p_{in}.

Rationale for 2-complex FRC in line graph. We have shown in Section 5.3.3 that FRC-2 in line graph can largely depend on the degree of nodes in the original graph, while the performance of FRC-1 for the task of community detection is limited, hence the corresponding algorithm may not lead to satisfactory results. Our experimental results (see Table 11) verify the above statement, where FRC-2 can have comparable performance to FRC-3 when node degree is moderate, but the performance drops significantly and approaches FRC-1 as the network becomes denser.

p_{in}	FRC-1	FRC-2	FRC-3
0.1	0.315 (0.266)	0.816 (0.231)	0.854 (0.184)
0.2	0.269 (0.254)	0.856 (0.162)	0.895 (0.148)
0.3	0.328 (0.337)	0.859 (0.183)	0.982 (0.049)
0.4	0.560 (0.296)	0.673 (0.136)	0.978 (0.048)

Table 11: Mean (SD) of the extended NMI from different FRC-based methods on planted two-block MMBs with $n_1 = n_2 = 50$, $n_o = 1$ node of mixed membership, $p_{out} = 0$, and varying probabilities p_{in}.

Rationale for detecting mixed-membership communities on the line graph. In the mixed-membership setting, it is still possible to apply algorithms that are designed

for the single-membership case, but the performance varies. Our experimental results (see Table 12) indicate that the improvement from applying the algorithm on the line graph is more significant when the graph is sparser, even while there is only a few nodes of mixed membership, and also when there are more nodes of mixed membership. We should note that applying algorithms directly on the original graph cannot detect mixed-membership nodes, thus there is an upper bound on the accuracy that these methods can achieve and how far this upper bound is from 1 depends on the number of mixed-membership nodes.

p_{in}	n_o	ORC-E (G) / (L)	ORC-A (G) / (L)	FRC-2 (G) / FRC-3 (L)
0.05	1	0.993 / 0.999	0.470 / 0.648	0.314 / 0.679
0.15	1	0.993 / 1	0.993 / 0.714	0.784 / 0.865
0.15	30	0.773 / 0.969	0.493 / 0.497	0.176 / 0.385

Table 12: Mean of the extended NMI from different methods on the original graph G versus its line graph L, where the graphs are generated from planted two-block MMBs with $n_1 = n_2 = 150$, $p_{out} = 0.01$, and varying number of mixed-membership node(s) and probability p_{in}.

8. Comparison of Curvature Notions

In this final section, we discuss differences and commonality between the two curvature notions that we considered. Specifically, we address geometric considerations, i.e., which structural features are best captured by which curvature notion and how this is reflected in their performance in clustering tasks.

8.1 Relationship between 2-complex FRC and ORC.

There is already strong numerical (Pouryahya et al., 2017; Samal et al., 2018) and theoretical evidence (Jost and Münch, 2021) of a close relationship between the 2-complex FRC and ORC in the prior literature. Specifically, Jost and Münch (2021) proved the following relationship between the two notions:

Theorem 23 *(Jost and Münch (2021)) Ollivier's and Forman*'s curvature for the edges of a graph G are related as*

$$\mathrm{Ric}_O(e) = \max_K \mathrm{Ric}^*_F(e)\ , \tag{8.1}$$

where the maximum is taken over all complexes K that have G as a 1-skeleton.

Remark 24 The Forman* curvature considered by Jost and Münch (2021) deviates from Forman's orginal notion. However, one can recover the original notion as a special case via a specific choice of the cell weights. For details, see Jost and Münch (2021, sec. 7.4).

Theorem 23 implies that FRC and ORC coincide on the edges, when maximizing FRC over the choice of 2-cells whose contributions are included in curvature computation. The proof of this result proposes a means to "translate" between the contributions of 2-cells and the cost of transport plans.

Our numerical results (Figure 16) confirm that the correlation between FRC and ORC becomes stronger, if FRC is "augmented", i.e., if the contributions of higher-order structures (triangles, quadrangles) are taking into account for the curvature computation. In this paper, we considered a 2-complex notion of FRC in the original graph G (previously considered in (Weber et al., 2017b)) and "approximate" the contribution of k-faces, by including curvature contribution of faces up to order K (see Figures 17, 18 and 19 in Appendix C.4). Since the frequency of k-faces decreases sharply as k increases (illustrated in Figure 7), it is enough to consider small values of K. In our application to mixed-membership community detection, we considered $K = 4$. We found that this parameter choice balances fast computation (complexity increases with K) and performance (accuracy increases with K). Our numerical experiments confirmed that the performance of the FRC-based community detection algorithm closer resembles that of the ORC-based community detection algorithm as K increases.

We have previously noticed in Figures 8 and 9 that the range of curvature values varies between ORC and classical FRC, while the shapes of the distributions resemble each other. Notice that the range of curvature values between the 2-complex FRC and ORC is much better aligned, which is in agreement with their similar performance in downstream tasks.

8.2 Differences in structural features captured by FRC and ORC.

8.2.1 Relationship with clustering coefficient

The *clustering coefficient*, initially introduced by Watts and Strogatz (1998), is a network-theoretic measure, which is correlated with the ability of clustering algorithms to detect communities. Formally, it measures how close a node neighborhood is to a clique (i.e., a fully connected subgraph). It is defined as the ratio of the number of edges between neighbors of a node $u \in V$ and the number of edges in the clique, if the neighborhood of u were fully connected:

Definition 25 (Clustering coefficient) *Let $A = (a_{ij})_{1\leq i,j,\leq n}$ denote the adjacency of a graph G and $u \in V$ the node whose connectivity is characterized by the ith row in the adjacency. Then*

$$C(u) := \frac{\sum_{v\sim u} \#(u,v)}{d_u(d_u-1)} = \frac{\sum_{j,k} a_{ij}a_{jk}a_{ki}}{d_u(d_u-1)} \,. \tag{8.2}$$

It has been previously observed empirically (Pouryahya et al., 2017; Samal et al., 2018) that node-level ORC is highly correlated with the clustering coefficient. FRC is also correlated with the clustering coefficient, although the degree of correlation depends significantly on the type of higher-order structures that is included (see Figure 16). We see that FRC correlates best with the clustering coefficient, if triangle contributions are included in the curvature computation, i.e., for our notion of 2-complex FRC. This is also the FRC notion that resembles ORC most closely. Importantly, the close relationship between ORC and the clustering coefficient can also be established theoretically. In particular, Jost and Liu (2014) showed the following result:

Theorem 26 *(Jost and Liu (2014), Cor. 1)*

$$\frac{d_u-1}{d_u}C(u) \geq \mathrm{Ric}_O(u) \geq -2 + \frac{d_u-1}{d_u \vee \max_{v\sim u} d_v}C(u) \,.$$

To the best of our knowledge, no theoretical connection between FRC and the clustering coefficient is known.

8.2.2 Node clustering vs. edge clustering.

Notice that Algorithm 1 when applied to the underlying graph G performs a *node clustering*, whereas it performs an *edge clustering* when applied to the line graph $L(G)$. Specifically, if Algorithm 1 is applied to the line graph, then we obtain a mixed-membership label vector y for each vertex v by computing $y_l(v) = \frac{1}{|E_v|}\sum_{e\in E_v}\chi_{C_l}(e)$, where χ_{C_l} is the indicator for the cluster C_l. This assignment is based on an *edge clustering* in the line graph, i.e., in Equation (2.6) above we assume that edge clusters correspond to node clusters. Hence, our approach solves single-membership community detection via node clustering and mixed-membership community detection via edge clustering. Both clustering approaches utilize the implicit assumption that the underlying graph is *homophilic*, i.e., similar nodes are more densely connected than dissimilar nodes. Node clustering identifies clusters of nodes by finding densely connected subgraphs. We have seen above that dense connectivity increases curvature; consequently, our curvature-based algorithm finds clusters by identifying regions of low curvature. In contrast, edge clustering identifies groups of edges that connect a set of similar nodes. Due to the higher connectivity among sets of similar nodes, those edges have a higher number of neighboring edges (i.e., edges that are adjacent to the same node). Consequently, those edges form a densely connected subgraph in the line graph. Applying curvature-based clustering on the line graph allows us to identify these densely connected subgraphs, due to their higher curvature. Throughout the paper, we have provided empirical and theoretical evidence that all curvature notions considered here have higher value in densely connected subgraphs of any graph. However, our study has demonstrated that the difference of curvature values of edges between communities, which is directly linked to the performance of curvature-based clustering, is more pronounced for some notions. In particular, our analysis shows that ORC highlights this effect best. Among the different FRC variants studied here, we showed that incorporating higher-order faces (triangles in the original graph, triangles and quadrangles in the line graph) leads to the best results. The effect of Algorithm 1 may also be interpreted as a *graph coarsening* scheme. When applied to a graph G, our approach uncovers meso-scale structure, such as communities. The interpretation of Algorithm 1 as coarsening the line graph is perhaps less intuitive. Nevertheless, following again the homophily perspective above, we see that curvature-based edge clustering groups edges that connect similar nodes, uncovering crucial meso-scale structure in the original graph.

8.2.3 Computational Considerations.

An often cited argument in the previous literature is that variants of 2-complex FRC are preferable over ORC in applications, due to their faster computation. In particular, the known links between the spectrum of the graph Laplacian and clustering coefficients (Jost and Liu, 2014) mentioned above make ORC a natural choice for curvature-based clustering. Previous approaches have either computed ORC exactly via the Hungarian method ($O(|E|d_{\max}^3)$) or approximated it via Sinkhorn's method ($O(|E|d_{\max}^2)$), both of which are significantly more expensive than computing Forman's curvature with triangle augmentions

($O(|E|d_{\max})$). We propose above an approximation via the arithmetic mean of upper and lower curvature bounds and illustrate that this approximation is on par with, or in certain instances, even superior to the Sinkhorn approximation. Importantly, our proposed approximation can be computed in $O(|E|d_{\max})$ operations, so it has complexity that is comparable to that of FRC. Hence, computational considerations do not provide a clear preference for our approximate ORC approach over FRC. In addition, we propose above a variant of Algorithm 1 via ORC, which does not require the computationally expensive construction of the line graph.

9. Discussion

In this paper, we have proposed a unifying framework for curvature-based node clustering algorithms and systematically investigated the strength and weakness of different curvature notions, specifically variants of Forman's and Ollivier's Ricci curvature. In addition to single-membership community structure, which is the focus of most existing work, we have also considered the mixed-membership setting and extended curvature-based clustering approaches to this case. To this end, we have further studied discrete curvature on the line graph. Furthermore, we have proposed an effective approximation of Ollivier's notion of discrete curvature to overcome its scalability issues. This construction may be of independent interest. We emphasize that the results derived in this paper provide insights into the relationship of discrete curvatures on a graph and that of its dual, which may be of independent interest.

Throughout the experiments, we observe consistently convincing performance of the curvature-based methods. Specifically, the ORC-based approach can outperform classical community detection methods, the Louvain algorithm and Spectral Clustering, in dense synthetic and real networks, in both single- and mixed-memberships settings. In regimes where community detection is challenging (e.g., close to the detectability threshold in SBMs), curvature-based methods outperform reference methods. Although the experimental results indicate quadratic complexity of the ORC-based method, the proposed approximation's performance is comparable to classical methods and, in relatively large or dense graphs, even computationally superior. The results agree with the linear complexity of the approximated ORC-based approach, as demonstrated in Section 6.1.3. The FRC-based approach, which has a lower complexity, can outperform or has comparable performance to the Louvain algorithm, and to spectral clustering in dense graphs. Note that we consider different variants of Forman's Ricci curvature for single- and mixed-membership node clustering problems, where we only consider triangular faces in the former while incorporating the quadrangular faces in the latter. We also observe that FRC with triangular faces has a stronger relationship with the clustering coefficients in SBMs, while FRC with further quadrangular faces has a slightly stronger relationship with the ORC in MMBs. As byproduct, the results provide some experimental evidence to relate Forman's to Ollivier's Ricci curvature.

Several extensions of this work could be avenues for future investigation. In the variant of Algorithm 1 presented here, several crucial hyperparameters are chosen heuristically. Specifically, we believe that the choice of cut-off points could be improved, potentially utilizing geometric information, such as *curvature gaps* (Gosztolai and Arnaudon, 2021). The choice of the stopping criterion (T) could be guided by established heuristics in the community

detection literature. For instance, the Girvan-Newman algorithm constructs a dendrogram to inform the stopping criterion and edge threshold Δ, when the number of communities is not known a priori. Our mixed-membership approaches require the construction of the line graph, which adds significant computational cost when applied to large-scale graphs. Approaches that avoid the construction of the line graph, perhaps utilizing the relationship between the Ricci curvature of a graph and its dual, could remove this bottleneck. We have also seen that using 1 as the upper bound for ORC could already improve the efficiency in our experiments. These approximations may prove computationally beneficial in both the mixed-membership and single-membership settings. While Algorithm 1 allows for clustering in both unweighted and weighted networks, it cannot be applied to directed networks in its current form. An extension to directed networks would significantly widen the range of possible applications. Further study of the behavior of discrete curvatures in various random graph models could serve to improve both theoretical understanding of these graphs, as well as to inform the development of improved algorithms for certain graph distributions. Finally, while our experiments indicate that curvature-based methods compare favorably to community detection baselines, the question of theoretical optimality remains open and presents an important avenue for future work.

Acknowledgments

Part of the computations in this paper were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University. Part of the computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at the PDC Center for High Performance Computing, KTH Royal Institute of Technology, partially funded by the Swedish Research Council through grant agreement no. 2018-05973. We would also like to acknowledge the use of the University of Oxford Mathematical Institute compute facility in carrying out preliminary work.

Most of the work was done when YT was at Nordita, Stockholm University and KTH Royal Institute of Technology, where she was funded by the Wallenberg Initiative on Networks and Quantum Information (WINQ). Part of the work was done when YT was at Mathematical Institute, University of Oxford, where she was funded by the EPSRC Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with Tesco PLC. Part of this work was done while MW was at the University of Oxford, supported by a Hooke Research Fellowship. MW further acknowledges partial support from NSF CBET-2112085 and a Sloan Fellowship in Mathematics. Over the course of this work, ZL has been funded under NIH grant MH-19-147, NSF HDR TRIPODS grant 1934979, and NSF DMS grant 2113099. He would also like to acknowledge support from the Naval Engineering Education Consortium and Microsoft Research.

Appendix A. ORC on weighted graphs

A.1 Proofs of Results in Section 4.1

Proof of Lemma 5: Consider the subgraph of G given represented schematically in Figure 12.

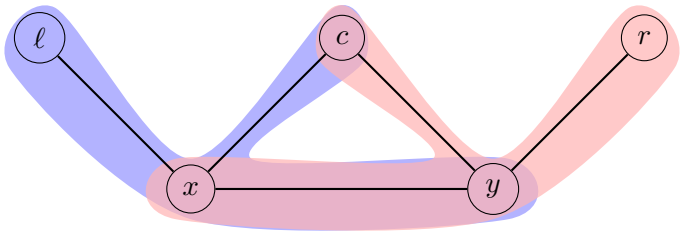


Figure 12: Schematic of the subgraph H of G, focusing on x, y and their neighbors. Note that typically there are several nodes $\ell_1, \ldots, \ell_{j_1}$, $c_1, \ldots, c_{j_2}$, and $r_1, \ldots, r_{j_3}$, satisfying $\deg(x) = j_1 + j_2 + 1, \deg(y) = j_2 + j_3 + 1$. The support of the m_x measure is shown in blue, m_y in red.

Now define the following quantities:

$$
\begin{aligned}
L_x &= \sum_{\ell} m_x(\ell) & L_y &= \sum_{\ell} m_y(\ell)\ (=0)\\
X_x &= m_x(x) & X_y &= m_y(x)\\
C_x &= \sum_{c} m_x(c) & C_y &= \sum_{c} m_y(c)\\
Y_x &= m_x(y) & Y_y &= m_y(y)\\
R_x &= \sum_{r} m_x(r)\ (=0) & R_y &= \sum_{r} m_y(r).
\end{aligned}
$$

If we consider transport plans restricted to H, then all of the L_x mass must be transported to x, and all of the R_y mass must be transported from y. As such, we may consider a modified diagram as in Figure 13.

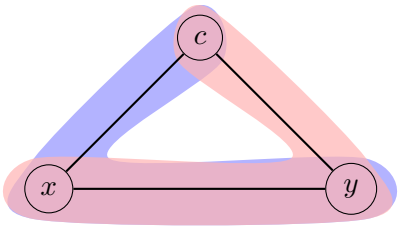


Figure 13: Schematic of the subgraph H' of H, focusing on x, y and their common neighbors. The support of the $\tilde{m}_x$ measure is shown in blue, $\tilde{m}_y$ in red.

Here we define $\tilde{m}_x, \tilde{m}_y$ as follows:

$$\tilde{m}_x(v) = \begin{cases} L_x + X_x & \text{if } v = x \\ m_x(c) & \text{if } v = c \\ Y_x + R_x \ (= Y_x) & \text{if } v = y, \end{cases} \quad \tilde{m}_y(v) = \begin{cases} L_y + X_y \ (= X_y) & \text{if } v = x \\ m_y(c) & \text{if } v = c \\ Y_y + R_y & \text{if } v = y. \end{cases}$$

Then we have

$$W_1(m_x, m_y) \leq \sum_{\ell} \omega(\ell, x) m_x(\ell) + \sum_{r} \omega(r, y) m_y(r) + W_1(\tilde{m}_x, \tilde{m}_y),$$

and this is tight when x, y have no unshared neighbors. Indeed, if we only consider the subgraph H of G, this upper bound is the exact value of $W_1(m_x, m_y)$. Define $\tilde{X}_x, \tilde{C}_x, \tilde{Y}_x, \tilde{X}_y, \tilde{C}_y, \tilde{Y}_y$ analogously for these new measures.

Now for each vertex of type c, if $\tilde{m}_x(c) - \tilde{m}_y(c) > 0$, we will send this amount to y along (c, y). Otherwise, we will send $\tilde{m}_y(c) - \tilde{m}_x(c)$ from x along (c, x). This results in the updated (signed) measure

$$\hat{m}_x(v) = \begin{cases} \tilde{X}_x - \sum_c (\tilde{m}_y(c) - \tilde{m}_x(c))_+ & \text{if } v = x \\ \tilde{m}_y(c) & \text{if } v = c \\ \tilde{Y}_x + \sum_c (\tilde{m}_x(c) - \tilde{m}_y(c))_+ & \text{if } v = y. \end{cases}$$

Finally, if $\hat{m}_x(x) - \tilde{X}_y > 0$, send this amount to y along the edge (x, y). Otherwise, send $\hat{m}_x(y) - \tilde{Y}_y = -(\hat{m}_x(x) - \tilde{X}_y)$ from y to x along the (x, y) edge.

This gives the upper bound

$$\begin{aligned} W_1(\tilde{m}_x, \tilde{m}_y) \leq & \sum_c (\tilde{m}_x(c) - \tilde{m}_y(c))_+ \omega(c, y) + (\tilde{m}_y(c) - \tilde{m}_x(c))_+ \omega(c, x) \\ & + \left| \tilde{X}_x - \tilde{X}_y - \sum_c (\tilde{m}_y(c) - \tilde{m}_x(c))_+ \right| \omega(x, y). \end{aligned}$$

Finally, we have

$$\begin{aligned} W_1(m_x, m_y) \leq & \sum_{\ell} \omega(\ell, x) m_x(\ell) + \sum_{r} \omega(r, y) m_y(r) \\ & + \sum_c \omega(c, y)(m_x(c) - m_y(c))_+ + \omega(c, x)(m_y(c) - m_x(c))_+ \\ & + \left| L_x + X_x - X_y - \sum_c (m_y(c) - m_x(c))_+ \right| \omega(x, y) = U_{x,y}. \end{aligned}$$

□

For completeness, we state and prove the following lower bound on the Wasserstein distance:

Lemma 27 *For any 1-Lipschitz function $h(v)$ on G, we have*

$$W_1(m_x, m_y) \geq \mathbb{E}_{v \sim m_x}[h(v)] - \mathbb{E}_{v \sim m_y}[h(v)] = \sum_{v \in N(x) \cup N(y)} h(v)(m_x(v) - m_y(v)) \ .$$

Proof Given a coupling π for m_x and m_y, and a 1-Lipschitz function $h : V \to \mathbb{R}$, we have

$$\begin{aligned}\mathbb{E}_{(u,w)\sim\pi}[d_G(u,w)] &= \sum_{\substack{u\sim N(x)\\ w\sim N(y)}} d_G(u,w)\pi(u,w)\\ &\geq \sum_{\substack{u\sim N(x)\\ w\sim N(y)}} (h(u)-h(w))\pi(u,w)\\ &= \sum_{v\in N(x)\cup N(y)} h(v)(m_x(v)-m_y(v)).\end{aligned}$$

Since the choice of coupling was arbitrary, we conclude that $W_1(m_x, m_y)$ satisfies the same lower bound. ■

Now we may prove Lemma 7.

Proof of Lemma 7: Let $V_H = N(x) \cup N(y)$ be the set of vertices from the subgraph H of G that we considered in the upper bound for $W_1(m_x, m_y)$. A difficulty that arises in defining a 1-Lipschitz function h on G is that there are more edges and paths in G than just those in H. These "short-circuits" in the larger graph mean that a 1-Lipschitz function defined on H does not necessarily extend to a 1-Lipschitz function on G. We define a partition of V_H into three subsets $\mathcal{P}, \mathcal{Z}, \mathcal{N}$ based on whether $m_x(v) - m_y(v)$ is Positive, Zero, or Negative. Observe that for a vertex $v \in \mathcal{P}$, the excess mass $m_x(v) - m_y(v)$ must be transported away from v in order to have a valid transport plan. In fact, it must be transported at least enough to reach $\mathcal{N}$, since

$$0 = \sum_{v\in V} m_x(v) - m_y(v) = \sum_{v\in\mathcal{P}} (m_x(v) - m_y(v)) + \sum_{v\in\mathcal{N}} (m_x(v) - m_y(v))$$

implies that $m_x(\mathcal{P}) - m_y(\mathcal{P}) = m_y(\mathcal{N}) - m_x(\mathcal{N})$. In other words, all excess mass from m_x in $\mathcal{P}$ must be transported to vertices in $\mathcal{N}$, or we cannot have a valid transport plan.

Motivated by this observation, we define the distance from a vertex $v \in V$ to a set of vertices $S \subseteq V$ as $d_G(v, S) = \min_{u\in S} d_G(v, u)$, where d_G is the shortest path distance in G. Thus, $d_G(v, S)$ gives the distance in G from the vertex v to the nearest element in the set S.

Consider the following 1-Lipschitz functions parameterized by $\lambda \in [0, 1]$:

$$h_\lambda(v) = \begin{cases} \lambda d_G(v,\mathcal{N}) & \text{if } v \in \mathcal{P}\\ -(1-\lambda) d_G(v,\mathcal{P}) & \text{if } v \in \mathcal{N}\\ \lambda d_G(v,\mathcal{N}) - (1-\lambda) d_G(v,\mathcal{P}) & \text{otherwise.}\end{cases} \tag{A.1}$$

Note that the third equation works for all $v \in V$, since $d_G(v, \mathcal{P}) = 0$ for $v \in \mathcal{P}$ and similarly for $\mathcal{N}$, but we split this into cases to show the behavior on the different sets more clearly.

These functions are all 1-Lipschitz since they are convex combinations of the two 1-Lipschitz functions $d_G(\cdot, \mathcal{N})$ and $-d_G(\cdot, \mathcal{P})$. Let's show that these are 1-Lipschitz: Let

$S \subseteq V$, and let $u, v \in V$. Let $w_u, w_v \in S$ be the nearest vertices in S to u, v respectively, so that $d_G(u, w_u) = d_G(u, S)$ and similarly for w_v. Then

$$\begin{aligned} d_G(v,S) &\leq d_G(v,w_u) \leq d_G(v,u) + d_G(u,w_u) \\ &= d_G(u,v) + d_G(u,S) \\ d_G(u,S) &\leq d_G(u,w_v) \leq d_G(u,v) + d_G(v,w_v) \\ &= d_G(u,v) + d_G(v,S) \\ \implies & |d_G(v,S) - d_G(u,S)| \leq d_G(u,v). \end{aligned}$$

But since the objective function $\mathbb{E}_{v\sim m_x}[h(v)] - \mathbb{E}_{v\sim m_y}[h(v)]$ is linear in h, we see that the maximizer of this quantity over $\{h_\lambda : \lambda \in [0,1]\}$ must be achieved at h_0 or h_1. This gives the following lower bound for $W_1(m_x, m_y)$:

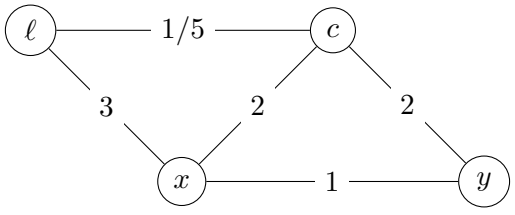
$$\begin{aligned} W_1(m_x,m_y) &\geq \max_{\lambda\in\{0,1\}} \sum_{v\in V_H} h_\lambda(v)(m_x(v) - m_y(v)) \\ &= \sum_{v\in\mathcal{P}} h_\lambda(v) \underbrace{(m_x(v) - m_y(v))}_{>0} + \sum_{v\in\mathcal{Z}} h_\lambda(v) \underbrace{(m_x(v) - m_y(v))}_{=0} \\ &\quad + \sum_{v\in\mathcal{N}} h_\lambda(v) \underbrace{(m_x(v) - m_y(v))}_{<0} \\ &= \max\left\{ \sum_{v\in\mathcal{P}} d_G(v,\mathcal{N})(m_x(v) - m_y(v)), \sum_{v\in\mathcal{N}} d_G(v,\mathcal{P})(m_y(v) - m_x(v)) \right\}. \end{aligned}$$

□

We note that vertices of type ℓ will always belong to $\mathcal{P}$, and vertices of type r will always belong to $\mathcal{N}$. Vertices of type c may fall into either set, and there will typically be some of these vertices in both sets. When we set $\alpha = 0$, then we always have $x \in \mathcal{N}$ and $y \in \mathcal{P}$, though if we choose $\alpha \geq 1/2$, x cannot be in $\mathcal{N}$, and y cannot be in $\mathcal{P}$.

A.2 Note on combinatorial ORC bound by Jost-Liu

Remark 28 *We note that the lower bound for $W_1(m_x, m_y)$ given in Jost and Liu (2014) for weighted graphs using the measure $m_x(v) = w_{vx}/d_x$ is actually incorrect. Consider the weighted graph G shown in Figure 14, which also summarizes the measures m_x, m_y.*



	x	y	ℓ	c
m_x	0	1/6	1/2	1/3
m_y	1/3	0	0	2/3
h	-6/5	-1/5	1	4/5

Figure 14: The weighted graph G, the measures m_x, m_y from Jost and Liu (2014), and our 1-Lipschitz function h.

An optimal transport plan sends 1/6 mass from y to x and from c to x, and 1/2 mass from ℓ to c, for a cost of 3/5. We can verify the optimality of this transport plan with the 1-Lipschitz function h given in the table in Figure 14, since this also achieves the value 3/5. As such, the exact ORC of the edge $\{x, y\}$ in G is $1 - 3/5 = 2/5$. However, the upper bound on this ORC from Jost and Liu (2014) is

$$\frac{w_{cx}}{d_x} \wedge \frac{w_{cy}}{d_y} = \frac{1}{3} \wedge \frac{2}{3} = \frac{1}{3} < \frac{2}{5}.$$

A.3 Proof of Results in Section 5.2

Here is the proof of Lemma 11.

Proof Let $\Delta(u, v)$ denote the number of triangles in G which include the edge $\{u, v\}$, and $\Delta(\{u, v\}, \{v, w\})$ the number of triangles in $L(G)$, which include $\{\{u, v\}, \{v, w\}\}$. Both bounds follow from a simple combinatorial computation: For the upper bound, we have

$$\mathrm{Ric}_O^{L(G)}(\{\{u, v\}, \{v, w\}\}) \leq \frac{\Delta(\{u, v\}, \{v, w\})}{d_{\{u,v\}} \vee d_{\{v,w\}}} = \frac{d_v - 2 + \mathbf{1}_E(\{u, w\})}{(d_u \vee d_w) + d_v - 2}\,.$$

The lower bound follows from

$$\begin{aligned}
\mathrm{Ric}_O^{L(G)}(\{\{u, v\}, \{v, w\}\}) \geq & -\left(1 - \frac{1}{d_{\{u,v\}}} - \frac{1}{d_{\{v,w\}}} - \frac{\Delta(\{u, v\}, \{v, w\})}{d_{\{u,v\}} \wedge d_{\{v,w\}}}\right)_+ \\
& - \left(1 - \frac{1}{d_{\{u,v\}}} - \frac{1}{d_{\{v,w\}}} - \frac{\Delta(\{u, v\}, \{v, w\})}{d_{\{u,v\}} \vee d_{\{v,w\}}}\right)_+ + \frac{\Delta(\{u, v\}, \{v, w\})}{d_{\{u,v\}} \vee d_{\{v,w\}}} \\
= & -\left(1 - \frac{1}{d_u + d_v - 2} - \frac{1}{d_v + d_w - 2} - \frac{d_v - 2 + \mathbf{1}_E(\{u, w\})}{(d_u \wedge d_w) + d_v - 2}\right)_+ \\
& - \left(1 - \frac{1}{d_u + d_v - 2} - \frac{1}{d_v + d_w - 2} - \frac{d_v - 2 + \mathbf{1}_E(\{u, w\})}{(d_u \vee d_w) + d_v - 2}\right)_+ \\
& + \frac{d_v - 2 + \mathbf{1}_E(\{u, w\})}{(d_u \vee d_w) + d_v - 2}\,.
\end{aligned}$$

■

A.4 Proofs and Remarks from Section 5.3.1

We prove Lemma 13.

Proof The relationship arises from simple combinatorial arguments. Recalling Lemma 10, we have

$$\begin{aligned}
\mathrm{Ric}_F^{L(G)}(\{\{u, v\}, \{v, w\}\}) &= 4 - d_{\{u,v\}} - d_{\{v,w\}} \\
&= 4 - (d_u - d_v - 2) - (d_v + d_w - 2) \\
&= 8 - d_u - d_v - d_v - d_w \\
&= \mathrm{Ric}_F^G(\{u, v\}) + \mathrm{Ric}_F^G(\{v, w\}),
\end{aligned}$$

which gives the claim.

Similarly,

$$\begin{aligned}
\mathrm{Ric}_F^{L(G)}(\{u,v\}) &= \sum_{\{\{u,v\},\{v,w\}\}\in\mathcal{E}} \mathrm{Ric}_F^{L(G)}(\{\{u,v\},\{v,w\}\}) + \sum_{\{\{u,v\},\{u,w\}\}\in\mathcal{E}} \mathrm{Ric}_F^{L(G)}(\{\{u,v\},\{u,w\}\}) \\
&= \sum_{\substack{\{v,w\}\in E \\ w\neq u}} \mathrm{Ric}_F^G(\{u,v\}) + \mathrm{Ric}_F^G(\{v,w\}) + \sum_{\substack{\{u,w\}\in E \\ w\neq v}} \mathrm{Ric}_F^G(\{u,v\}) + \mathrm{Ric}_F^G(\{u,w\}) \\
&= d_{\{u,v\}}\mathrm{Ric}_F^G(\{u,v\}) + \sum_{\substack{\{v,w\}\in E \\ w\neq u}} \mathrm{Ric}_F^G(\{v,w\}) + \sum_{\substack{\{u,w\}\in E \\ w\neq v}} \mathrm{Ric}_F^G(\{u,w\}) \\
&= (d_u + d_v - 2)\mathrm{Ric}_F^G(\{u,v\}) + \mathrm{Ric}_F^G(v) - \mathrm{Ric}_F^G(\{u,v\}) + \mathrm{Ric}_F^G(u) - \mathrm{Ric}_F^G(\{u,v\}) \\
&= \mathrm{Ric}_F^G(u) + \mathrm{Ric}_F^G(v) - \mathrm{Ric}_F^G(\{u,v\})^2.
\end{aligned}$$

■

We now prove Lemma 16.

Proof

$$\begin{aligned}
&\mathrm{Ric}_F(\{\{u,v\},\{v,w\}\}) \\
&\quad= \frac{|\mathcal{F}|}{\omega_\Delta} + 2 - \left|\left\{e \in \Gamma\big(\{u,v\}\big) : \big(\{u,v\},\{v,w\},e\big) \notin \mathcal{F}_\Delta\right\}\right| \\
&\qquad - \left|\left\{e \in \Gamma(\{v,w\}) : \{\{u,v\},\{v,w\},e\} \notin \mathcal{F}_\Delta\right\}\right| \\
&\quad= \frac{d_v - 2 + \mathbf{1}_E(\{u,w\})}{\omega_\Delta} + 2 - \big(d_u - 1 - \mathbf{1}_E(\{u,w\}) + d_w - 1 - \mathbf{1}_E(\{u,w\})\big) \\
&\quad= \frac{d_v}{\omega_\Delta} - d_u - d_w + 4 - \frac{2}{\omega_\Delta} + \left(\frac{1}{\omega_\Delta} + 2\right)\mathbf{1}_E(\{u,w\})\,.
\end{aligned}$$

■

Weights for quadrangles in augmented FRC To determine the weights of quadrangles, we again utilize Heron's formula: If we treat the two edges with the largest weights to be parallel, we can compute the weight of a quadrangular face via the area of the two triangles that form the trapezoid. Specifically, let $f = (e_i, e_j, e_k, e_l)$ denote a quadrangle in G, i.e., $e_i \sim e_j$, $e_j \sim e_k$, $e_k \sim e_l$, $e_l \sim e_i$ while $e_i \not\sim e_k$ and $e_j \not\sim e_l$, where $\omega_{e_i} \geq \omega_{e_j} \geq \omega_{e_k} \geq \omega_{e_l}$, and we consider three different cases here: (i) $\omega_{e_i} > \omega_{e_j}$, i.e., when the two largest weights (or lengths) are not the same, (ii) $\omega_{e_i} = \omega_{e_j}$ and $\omega_{e_k} > \omega_{e_l}$, i.e., when the two largest weights are the same while the other two are different, and (iii) $\omega_{e_i} = \omega_{e_j}$ and $\omega_{e_k} = \omega_{e_l}$, i.e., when the two largest are the same while the others are also the same. In case (i), we assume that e_i, e_j are parallel to each other, and then the area of the quadrangle can be obtained via

Heron's formula:

$$\omega_f := \sqrt{s(s-(\omega_{e_i}-\omega_{e_j}))(s-\omega_{e_k})(s-\omega_{e_l})}\left(1+2\frac{\omega_{e_j}}{\omega_{e_i}-\omega_{e_j}}\right) \tag{A.2}$$

$$s = \frac{(\omega_{e_i}-\omega_{e_j})+\omega_{e_k}+\omega_{e_l}}{2}\ . \tag{A.3}$$

In case (ii), we assume that e_k, e_l are parallel to each other, and similarly

$$\omega_f := \sqrt{s(s-(\omega_{e_k}-\omega_{e_l}))(s-\omega_{e_i})(s-\omega_{e_j})}\left(1+2\frac{\omega_{e_l}}{\omega_{e_k}-\omega_{e_l}}\right) \tag{A.4}$$

$$s = \frac{(\omega_{e_k}-\omega_{e_l})+\omega_{e_i}+\omega_{e_j}}{2}\ . \tag{A.5}$$

Finally, in case (iii), we assume that f is a rectangle, thus

$$\omega_f := \omega_{e_i}\omega_{e_k}\ . \tag{A.6}$$

A.5 Proof of Theorem 20

Proof Due to the homogeneity of the graphs $G_{a,b}$, which is preserved under the Ricci flow, it is sufficient to analyze the evolution of one edge of types (1), (2), (3). In the following, $d_i, D_i, w_i^t, \kappa_i^t$ denote the shortest-path distance, Wasserstein-1 distance, weight in iteration t, and ORC curvature of an edge of type (i) in iteration t, respectively. Notice that here

$$w_i^{t+1} = (1-\kappa_i^t)w_i^t = w_i^t - \Big(1-\frac{D_i}{d_i}\Big)w_i^t = D_i\ , \tag{A.7}$$

since $d_i = w_i^t$. Throughout the proof, x, y denote the vertices adjacent to the edge under consideration.

Iteration 1. In the first iteration, all edges have the same weight (i.e., 1), hence, Equation (6.2) reduces to the uniform distribution.

- Edges of type (3) have symmetric neighborhoods, with uniform mass distributions of $m_v = \frac{1}{a}$ on all neighbors of x, y. Thus, the total transportation cost of moving mass from x's neighbors to y's neighbors is $D_3 = \frac{1}{a}d_3$ and therefore $w_3^1 = D_3 = \frac{1}{a}$, since $d_3 = 1$.
- For edges of type (1), note that $d_x = d_y = a + (b-1)$, hence $m_v = \frac{1}{a+b-1}$ for all neighbors of x, y. By construction, x, y have $b-2$ common neighbors; due to the symmetry of the neighborhoods, no mass has to be moved for those neighbors. The optimal transportation plan for the remaining mass is as follows: Move the remaining mass $\frac{a}{a+b-1}$ from the neighbors of x (which are not neighbors of y) to x. This incurs a cost of $\frac{a}{a+b-1}d_2$, since this mass is transported along edges of type (2). Leave a mass of $\frac{1}{a+b-1}$ on x and transport the remainder to y, which incurs a cost of $\frac{a-1}{a+b-1}d_1$, since it is transported along an edge of type (1). Lastly, we distribute mass $\frac{a}{a+b-1}$ from y uniformly to its neighbors, which incurs a cost of $\frac{a}{a+b-1}d_2$, since it is transported along edges of type (2). In total, the transportation cost amounts to

$$D_1 = \frac{2a}{a+b-1}d_2 + \frac{a-1}{a+b-1}d_1\ ,$$

To prove that this is an optimal flow, consider the following 1-Lipschitz function h:

$$h(v) = \begin{cases} 2 & \text{if } v \sim x, v \not\sim y \\ 1 & \text{if } v = x \\ 0 & \text{if } v = y \text{ or } v \sim x, y \\ -1 & \text{if } v \sim y, v \not\sim x. \end{cases}$$

Then

$$\mathbb{E}_{v\sim m_x}[h(v)] - \mathbb{E}_{v\sim m_y}[h(v)] = 2\frac{a}{a+b-1} + 1\frac{(-1)}{a+b-1} - 1\frac{(-a)}{a+b-1} = \frac{3a-1}{a+b-1}.$$

This implies $w_3^1 = D_3 = \frac{3a-1}{a+b-1}$, since $d_1 = d_2 = 1$.

- For edges of type (2), let x denote the internal node and y the bridge node. By construction, x, y have $a-1$ common neighbors (internal nodes). We want to leave a mass of $\frac{1}{a+b-1}$ on each of those nodes (corresponding to the mass distribution in the neighborhood of y), and transport the excess mass to x and the unshared neighbors of y in an optimal way. Observe that $\frac{1}{a} > \frac{1}{a+b-1}$, so we have a total of $(a-1)\left(\frac{1}{a} - \frac{1}{a+b-1}\right)$ excess mass to move. If this exceeds $\frac{1}{a+b-1}$, we send that amount to x, and the remaining amount to y; otherwise we send all of this mass to x. We can simplify this condition as follows:

$$(a-1)\left(\frac{1}{a} - \frac{1}{a+b-1}\right) - \frac{1}{a+b-1} = \frac{(a-1)(b-1)-a}{a(a+b-1)} \geq 0$$

if and only if $(a-1)(b-1) - a = a(b-2) - (b-1) \geq 0$. When $b = 2$, this never holds, and when $b \geq 3$, this always holds, since $a(b-2) - (b-1) \geq a - b + 1 \geq 0$, given that $a \geq b$. Now let us consider these cases in turn.

Case $b = 2$: We send all of the excess mass from the common vertices to x, amounting to $\frac{a-1}{a(a+1)} < \frac{1}{a+1}$, meaning that we must send an additional $\frac{1}{a(a+1)}$ mass from y to x, and the remaining $\frac{1}{a+1}$ is sent to y's single unshared neighbor. The total cost is

$$D_2 = \frac{a-1}{a(a+1)}d_3 + \frac{1}{a(a+1)}d_2 + \frac{1}{a+1}d_1.$$

To see that this is an optimal flow, consider the following 1-Lipschitz function h:

$$h(v) = \begin{cases} 1 & \text{if } v \sim x, y \text{ or } v = y \\ 0 & \text{otherwise.} \end{cases}$$

This achieves a dual objective function value of $\frac{2}{a+1}$, so $w_2^1 = D_2 = \frac{2}{a+1}$, since $d_1 = d_2 = d_3 = 1$.

Case $b \geq 3$: We send $\frac{1}{a+b-1}$ of the excess mass from the common vertices to x, and the remainder $\frac{(a-1)(b-1)-a}{a(a+b-1)}$ to y. Since there is already a mass of $\frac{1}{a}$ at y, this gives a

total mass of $\frac{(a-1)(b-1)-a+(a+b-1)}{a(a+b-1)} = \frac{b-1}{a+b-1}$ to be sent to y's unshared neighbors, as required. The total cost is

$$D_2 = \frac{1}{a+b-1}d_3 + \frac{(a-1)(b-1)-a}{a(a+b-1)}d_2 + \frac{b-1}{a+b-1}d_1.$$

To see that this is an optimal flow, consider the following 1-Lipschitz function h:

$$h(v) = \begin{cases} 1 & \text{if } v \sim x, y \\ 0 & \text{if } v = x, y \\ -1 & \text{if } v \sim y, v \not\sim x. \end{cases}$$

This achieves a dual objective function value

$$\mathbb{E}_{v\sim m_x}[h(v)] - \mathbb{E}_{v\sim m_y}[h(v)] = 1\frac{(a-1)(b-1)}{a(a+b-1)} - 1\frac{-(b-1)}{a+b-1} = \frac{(2a-1)(b-1)}{a(a+b-1)},$$

which gives $w_2^1 = D_2 = \frac{(2a-1)(b-1)}{a(a+b-1)} = \frac{2a-1}{a+b-1}\left(\frac{b-1}{a}\right)$, since $d_1 = d_2 = d_3 = 1$.

We summarize these calculations in the following table:

Type	1	2	3
$b = 2$	$\frac{3a-1}{a+1}$	$\frac{2}{a+1}$	$\frac{1}{a}$
$b \geq 3$	$\frac{3a-1}{a+b-1}$	$\frac{(2a-1)(b-1)}{a(a+b-1)}$	$\frac{1}{a}$

In both cases, it is easy to check that $w_1^1 > w_2^1 > w_3^1$, using $a \geq b \geq 2$.

Iteration t+1. In round $t \geq 1$, the mass distributions on the neighborhood of a bridge node are given by

$$m_x(y) = \begin{cases} \frac{1}{C_b}e^{-w_1^t}, & y \text{ is bridge node} \\ \frac{1}{C_b}e^{-w_2^t}, & \text{else} \end{cases},$$

and for an internal node by

$$m_x(y) = \begin{cases} \frac{1}{C_{in}}e^{-w_2^t}, & y \text{ is bridge node} \\ \frac{1}{C_{in}}e^{-w_3^t}, & \text{else} \end{cases}.$$

Here the normalizing constants for bridge nodes is given by

$$C_b = ae^{-w_2^t} + (b-1)e^{-w_1^t},$$

and for internal nodes as

$$C_{in} = e^{-w_2^t} + (a-1)e^{-w_3^t}.$$

By way of induction, we suppose that $w_1^t > w_2^t$ and $w_1^t > w_3^t$, where we proved this in the case $t = 1$ above.

We again consider the three edge types separately. Given w_i^t, edge weights are updated as follows in iteration t:

- For edges of type (3), due to the symmetry of the neighborhoods, we again only move mass from x to y at a total cost of $D_3 = \frac{w_3^t}{C_{in}} e^{-w_3^t}$. Hence, the updated weights are $w_3^{t+1} = \frac{w_3^t}{C_{in}} e^{-w_3^t}$ by Equation (A.7).
- For edges of type (1), the transport plan is analogous to the one in iteration 1: We move a mass of $\frac{a}{C_b} e^{-w_2^t}$ from the neighbors of x to x, leave a mass of $\frac{1}{C_b} e^{-w_1^t}$ there, and transport the remaining mass $\frac{a}{C_b} e^{-w_2^t} - \frac{1}{C_b} e^{-w_1^t}$ to y. Picking up a mass of $\frac{1}{C_b} e^{-w_1^t}$ previously on y, we distribute the mass among y's neighbors. The total transportation cost is
$$D_1 = \frac{a}{C_b} e^{-w_2^t} d_2 + \Big(\frac{a}{C_b} e^{-w_2^t} - \frac{1}{C_b} e^{-w_1^t} \Big) d_1 + \frac{a}{C_b} e^{-w_2^t} d_2 \, .$$
To prove that this is an optimal flow, consider the following 1-Lipschitz function h:
$$h(v) = \begin{cases} w_2^t + w_1^t & \text{if } v \sim x, v \not\sim y \\ w_1^t & \text{if } v = x \\ 0 & \text{if } v = y \text{ or } v \sim x, y \\ -w_2^t & \text{if } v \sim y, v \not\sim x. \end{cases}$$
Then
$$\mathbb{E}_{v \sim m_x}[h(v)] - \mathbb{E}_{v \sim m_y}[h(v)] = (w_2^t + w_1^t) \frac{a}{C_b} e^{-w_2^t} + w_1^t \frac{(-1)}{C_b} e^{-w_1^t} - w_2^t \frac{(-a)}{C_b} e^{-w_2^t}.$$
Consequently, the updated weights are $w_1^{t+1} = \frac{2aw_2^t + aw_1^t}{C_b} e^{-w_2^t} - \frac{w_1^t}{C_b} e^{-w_1^t}$.
- For edges of type (2), let x denote the internal node and y the bridge node. At any of the $a-1$ common neighbors of x, y, we have
$$m_x(v) = \frac{e^{-w_3^t}}{C_{in}}, \quad m_y(v) = \frac{e^{-w_2^t}}{C_b}.$$
We define the following quantity
$$\begin{aligned} c_t = m_x(v) - m_y(v) &= \frac{e^{-w_3^t} C_b - e^{-w_2^t} C_{in}}{C_b C_{in}} \\ &= \frac{a e^{-w_2^t - w_3^t} + (b-1) e^{-w_1^t - w_3^t} - e^{-2w_2^t} - (a-1) e^{-w_2^t - w_3^t}}{C_b C_{in}} \\ &= \frac{(b-1) e^{-w_1^t - w_3^t} + e^{-w_2^t} (e^{-w_3^t} - e^{-w_2^t})}{C_b C_{in}}. \end{aligned}$$
When $w_2^t > w_3^t$, this is always positive, but if $w_2^t < w_3^t$, this could be negative.

If $c_t \geq 0$, then as before, we want to transport the excess mass from these common neighbors to x and the unshared neighbors of y in an optimal way. If the total excess mass exceeds $m_y(x)$, we send that amount to x, and the remaining amount to y; otherwise we send all of this mass to x. This condition may be written as
$$r_t := (a-1) \frac{e^{-w_3^t}}{C_{in}} - a \frac{e^{-w_2^t}}{C_b} \geq 0.$$

Clearly, $r_t = (a-1)c_t - \frac{e^{-w_2^t}}{C_b} < (a-1)c_t$.

Case $c_t, r_t \geq 0$: We send $m_y(x)$ mass from the common neighbors to x, and the remaining portion to y. Then we send all of the mass at y to its unshared neighbors, for a cost of

$$D_2 = \frac{e^{-w_2^t}}{C_b} d_3 + r_t d_2 + (b-1)\frac{e^{-w_1^t}}{C_b} d_1.$$

Consider the 1-Lipschitz function h given by

$$h(v) = \begin{cases} w_2^t & \text{if } v \sim x, y \\ w_2^t - w_3^t & \text{if } v = x \\ 0 & \text{if } v = y \\ -w_1^t & \text{if } v \sim y, v \not\sim x. \end{cases}$$

This gives a dual objective function value

$$w_2^t(a-1)c_t + (w_2^t - w_3^t)\frac{-e^{-w_2^t}}{C_b} - w_1^t \frac{-(b-1)e^{-w_1^t}}{C_b},$$

which matches the formula for D_2 above when we take into consideration that $d_j = w_j^t$ at iteration t.

Case $c_t \geq 0, r_t \leq 0$: We send all of the excess mass from the common neighbors to x, and get the remaining portion from y. We then send the rest of the mass at y to its unshared neighbors, for a total cost of

$$D_2 = (a-1)c_t d_3 + (-r_t)d_2 + (b-1)\frac{e^{-w_1^t}}{C_b} d_1.$$

Consider the 1-Lipschitz function h given by

$$h(v) = \begin{cases} w_3^t & \text{if } v \sim x, y \\ 0 & \text{if } v = x \\ w_2^t & \text{if } v = y \\ w_2^t - w_1^t & \text{if } v \sim y, v \not\sim x. \end{cases}$$

To show that this function is 1-Lipschitz, note that if v is a common neighbor of x, y, then $|h(v) - h(y)| = |w_2^t - w_3^t| \leq w_2^t$: either $w_2^t > w_3^t$, in which case $|w_2^t - w_3^t| = w_2^t - w_3^t \leq w_2^t$, or else $w_2^t < w_3^t$, and $w_3^t - w_2^t \leq w_2^t$ follows by the triangle inequality since $w_3^t = W_1(m_v, m_x) \leq W_1(m_v, m_y) + W_1(m_y, m_x) = 2w_2^t$, where W_1 is the Wasserstein-1 distance, and m_x, m_v, m_y are the measures at the vertices x, v, y with the weights from iteration $t-1$ (or when $t = 1$, $w_3^0 = 1 < 2 = w_2^0$).

This gives a dual objective function value

$$w_3^t(a-1)c_t + w_2^t \frac{e^{-w_2^t}}{C_{in}} + (w_2^t - w_1^t)\frac{-(b-1)e^{-w_1^t}}{C_b}.$$

This gives the equality when we note that

$$-r_t = a\frac{e^{-w_2^t}}{C_b} - (a-1)\frac{e^{-w_3^t}}{C_{in}} = 1 - \frac{(b-1)e^{-w_1^t}}{C_b} - \left(1 - \frac{e^{-w_2^t}}{C_{in}}\right) = \frac{e^{-w_2^t}}{C_{in}} - \frac{(b-1)e^{-w_1^t}}{C_b},$$

where the second equality follows from the definitions of C_b and C_{in}.

Case $c_t, r_t \leq 0$: If $c_t \leq 0$, then $r_t < (a-1)c_t \leq 0$. Since m_y places more mass at the common vertices than m_x, the only vertex on which m_x places more mass than m_y must be y. Thus, we send all of the needed mass from y to x, the common neighbors, and the unshared neighbors of y, for a cost of

$$D_2 = \frac{e^{-w_2^t}}{C_b}d_2 + (a-1)(-c_t)d_2 + (b-1)\frac{e^{-w_1^t}}{C_b}d_1.$$

Consider the 1-Lipschitz function h given by

$$h(v) = \begin{cases} -w_2^t & \text{if } v \sim x, y \\ -w_2^t & \text{if } v = x \\ 0 & \text{if } v = y \\ -w_1^t & \text{if } v \sim y, v \not\sim x. \end{cases}$$

This gives a dual objective function value

$$-w_2^t(a-1)c_t + (-w_2^t)\left(\frac{-e^{-w_2^t}}{C_b}\right) + (b-1)(-w_1^t)\left(\frac{-e^{-w_1^t}}{C_b}\right),$$

which matches the formula for D_2 above.

We summarize these calculations in the following table:

Type i	w_i^{t+1}
1	$w_1^t\frac{ae^{-w_2^t}-e^{-w_1^t}}{C_b} + 2w_2^t\frac{ae^{-w_2^t}}{C_b}$
2 $(c_t, r_t \geq 0)$	$w_1^t\frac{(b-1)e^{-w_1^t}}{C_b} + w_2^t r_t + w_3^t\frac{e^{-w_2^t}}{C_b}$
2 $(c_t \geq 0, r_t \leq 0)$	$w_1^t\frac{(b-1)e^{-w_1^t}}{C_b} + w_2^t(-r_t) + w_3^t(a-1)c_t$
2 $(c_t, r_t \leq 0)$	$w_1^t\frac{(b-1)e^{-w_1^t}}{C_b} + w_2^t(-r_t)$
3	$w_3^t\frac{e^{-w_3^t}}{C_{in}}$

We are now ready to prove the main two theorem statements. First, we show that w_3^t is decaying to 0. Since $a > b \geq 2$, we have that $a - 1 \geq 2$. Then since $C_{in} \geq (a-1)e^{-w_3^t}$, we get

$$w_3^{t+1} = w_3^t\frac{e^{-w_3^t}}{C_{in}} \leq w_3^t\frac{1}{a-1}.$$

This gives $w_3^{t+1} \leq [a(a-1)^t]^{-1}$, which tends to 0 exponentially quickly as $t \to \infty$.

Now we wish to show that $w_1^t > w_2^t$ and $w_1^t > w_3^t$ for all $t \geq 1$. From the proof above, we know that $w_3^t < 2w_2^t$. We will also need the inequalities

$$ae^{-w_2^t} - be^{-w_1^t} > 0 \tag{A.8}$$

which holds from $a > b$ and the induction hypothesis $w_1^t > w_2^t$; and

$$\begin{aligned} & \frac{ae^{-w_2^t}}{C_b} > r_t \\ \Leftrightarrow \quad & C_{in}ae^{-w_2^t} - C_{in}C_b r_t > 0 \\ \Leftrightarrow \quad & 2ae^{-2w_2^t} + a(a-1)e^{-w_2^t - w_3^t} - (a-1)(b-1)e^{-w_1^t - w_3^t} > 0, \end{aligned} \tag{A.9}$$

which holds since $e^{-w_2^t} > e^{-w_1^t}$ and $a > b - 1$. Note that this implies

$$\frac{ae^{-w_2^t}}{C_b} \geq \frac{a-1}{2}\frac{e^{-w_3^t}}{C_{in}} \geq \frac{e^{-w_3^t}}{C_{in}}.$$

We first show $w_1^{t+1} > w_3^{t+1}$.

$$\begin{aligned} w_1^{t+1} - w_3^{t+1} &= w_1^t \frac{ae^{-w_2^t} - e^{-w_1^t}}{C_b} + 2w_2^t \frac{ae^{-w_2^t}}{C_b} - w_3^t \frac{e^{-w_3^t}}{C_{in}} \\ &\geq w_1^t \frac{(b-1)e^{-w_1^t}}{C_b} > 0, \end{aligned}$$

using Equation (A.8), $w_3^t < 2w_2^t$, and the implication of Equation (A.9) just above.

Since Equation (A.8) clearly shows that the coefficient of w_1^t in $w_1^{t+1} - w_2^{t+1}$ will be nonnegative in any of the three cases (it is the same in all of them), we argue as follows:

$$w_1^{t+1} - w_2^{t+1} \geq \begin{cases} 2w_2^t \frac{ae^{-w_2^t}}{C_b} - w_2^t r_t - w_3^t \frac{e^{-w_2^t}}{C_b} & c_t, r_t \geq 0 \\ 2w_2^t \frac{ae^{-w_2^t}}{C_b} + w_2^t r_t - w_3^t (a-1)c_t & c_t \geq 0, r_t \leq 0 \\ 2w_2^t \frac{ae^{-w_2^t}}{C_b} + w_2^t r_t & c_t, r_t \leq 0. \end{cases}$$

The quantity in the first subcase may be bounded below as follows, using $w_3^t < 2w_2^t$:

$$w_2^t \left(2\frac{ae^{-w_2^t}}{C_b} - r_t \right) - w_3^t \frac{e^{-w_2^t}}{C_b} \geq w_2^t \left((a-2)\frac{e^{-w_2^t}}{C_b} + \frac{ae^{-w_2^t}}{C_b} - r_t \right).$$

This lower bound is always positive by Equation (A.9).

For the second subcase, observe that $(a-1)c_t = r_t + \frac{e^{-w_2^t}}{C_b} \leq \frac{e^{-w_2^t}}{C_b}$, so we may lower bound the quantity there (again using $w_3^t < 2w_2^t$) by

$$w_2^t \left(\frac{ae^{-w_2^t}}{C_b} + \frac{(a-1)e^{-w_3^t}}{C_{in}} \right) - w_3^t \frac{e^{-w_2^t}}{C_b} \geq w_2^t \left(\frac{(a-2)e^{-w_2^t}}{C_b} + \frac{(a-1)e^{-w_3^t}}{C_{in}} \right) > 0.$$

The third subcase is simplest of all, since the quantity above simplifies to

$$w_2^t \left(\frac{ae^{-w_2^t}}{C_b} + \frac{(a-1)e^{-w_3^t}}{C_{in}} \right) \geq 0.$$

This completes the proof.

■

Appendix B. Data sets

Real-world data. We consider both synthetic networks generated from the random graph models (see Section 5), and real networks[1], which characterize the connectomes of a set of different animal species. A connectome is a specific, cell-to-cell mapping of axonal tracts between neurons, created from cellular data obtained, for instance, via electron microscopy. Here specfically, we consider networks constructed from mixed species of cats ((de Reus and van den Heuvel, 2013), "Cat"), male C. elegans ((Jarrell et al., 2012), "Worm"), and rhesus ((Harriger et al., 2012), "Macaque"). Table 13 provides summary statistics for the real networks in our study.

	n	$\lvert E\rvert$	$\lvert\mathcal{E}\rvert$	d	σ_d/d	Modularity
Cat	65	730	18859	22.46	0.44	0.27
Worm	269	2902	93893	21.58	0.74	0.36
Macaque	242	3054	114991	25.24	0.73	0.36

Table 13: Summary statistics of the real networks.

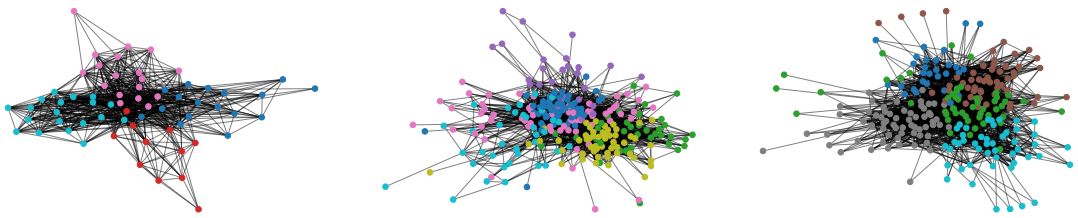


Figure 15: The real networks, "Cat" (left), "Worm" (middle) and "Macaque" (right), where we apply the Louvain algorithm to detect communities and vertices in different communities are shown in different colors.

Random Geometric Graphs. In our experiments, we consider Random Geometric Graphs (RGGs) which incorporate the spatial perspective (Penrose, 2003). For completeness, we define the RGG below.

1. Obtained from NeuroData's open source data base `https://neurodata.io/project/connectomes/`, accessed on 2 May 2022. The database hosts animal connectomes produced using data from a multitude of labs, across model species, using different modalities.

Definition 29 (Random Geometric Graphs) *In this model, vertices are placed in the unit cube uniformly at random, and two vertices will be connected by an edge if the distance between the vertices is less than a previously specified value called radius. Specifically, let the unit cube be* $[0,1)^{n_d}$ *where* n_d *defines its dimension, the radius be* $r \in (0,1)$*, and the distance function be* $d(x,y)$*,* $\forall x, y \in [0,1)^{n_d}$*. Then there will be an edge between vertices* v_i *and* v_j *if*

$$d(x_i, x_j) < r \ ,$$

where x_i, x_j *are the positions of the vertices* v_i, v_j*, respectively, in the unit cube* $[0,1)^{n_d}$*.*

Specifically, in our experiments, we start from the RGGs corresponding to 2-d unit cubes.

Appendix C. Experiments

C.1 Single-Membership Community Detection

Benchmarking. We report the mean NMI, runtime and their standard deviations (SD) of the methods on SBMs with two equally-sized blocks and varying network sizes. Consistent with the results in Section 7.3, the ORC-based approach outperforms most reference methods, and successfully recovers single-membership community structure, with an NMI greater than 0.8 in almost all cases, as we vary the network size n.

n	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-2
100	0.705 (0.121)	0.897 (0.081)	0.599 (0.078)	0.860 (0.117)	0.363 (0.070)	0.355 (0.112)
200	0.981 (0.025)	0.980 (0.020)	0.805 (0.285)	0.986 (0.018)	0.395 (0.107)	0.263 (0.035)
1000	1 (0)	1 (0)	1 (0)	1 (0)	0.990 (0.015)	0.437 (0.263)
3000	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	0.740 (0.392)

Table 14: Mean (SD) of NMI from different methods on planted SBMs with two equally-sized blocks, $p_{in} = 0.1$, $p_{out} = 0.01$, and varying sizes n.

n	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-2
100	0.828 (0.021)	0.045 (0.004)	0.132 (0.323)	0.482 (0.018)	0.419 (0.013)	0.373 (0.009)
200	2.767 (0.098)	0.076 (0.004)	0.037 (0.004)	1.194 (0.076)	0.991 (0.016)	0.889 (0.015)
1000	33.47 (0.584)	1.087 (0.043)	0.382 (0.016)	12.66 (0.168)	6.721 (0.103)	4.244 (0.086)
3000	265.4 (3.120)	13.44 (0.477)	4.802 (0.110)	165.8 (5.805)	85.63 (2.751)	44.27 (1.564)

Table 15: Mean (SD) of runtime (in seconds) from different methods on planted SBMs with two equally-sized blocks, $p_{in} = 0.1$, $p_{out} = 0.01$, and varying sizes n.

Rationale for ORC approximation. ORC-S has been considered in the literature in order to improve the efficiency of ORC computation, but we notice that the improvement is not clear when the graph is not sufficiently large or dense (see Table 16). However, the approximation we have proposed almost always significantly improve the computational efficiency (see also Tables 3 and 15). We also mention here that the current computations are only allocated with moderate level of memory, while we also observed that ORC-S can improve the efficiency when large amount of memory is available, as in the analysis of real data in Section 7.4.

p_{in}	ORC-E	ORC-S	ORC-A
0.1	2785	5859	1013
0.2	16876	68658	2933

Table 16: Runtime (in seconds) from different ORC-based methods on one sample of planted SBMs with two equally-sized blocks, size $n = 5000$, $p_{out} = 0.01$, and varying probabilities p_{in}.

Rationale for 2-complex FRC. FRC-2 is the one of the strongest correlation with the clustering coefficients, among the three versions of FRC (see Figure 16).

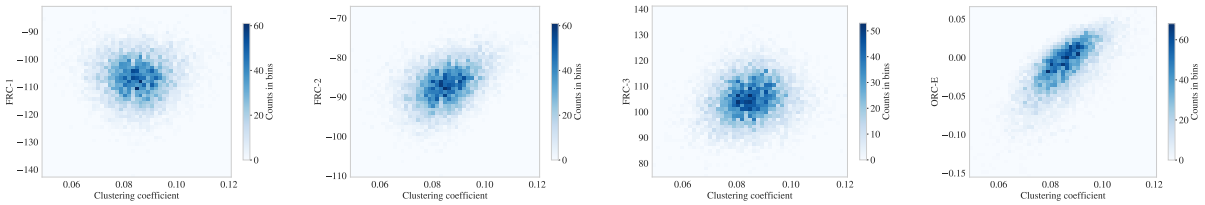


Figure 16: 2-d histogram of FRC-1 (left), FRC-2 (middle left), FRC-3 (middle right) and ORC-E (right) versus the clustering coefficients of nodes in G, where $n_s = 10$ networks are generated from the planted SBM with two equally-sized blocks, size $n = 1000$, $p_{in} = 0.1$ and $p_{out} = 0.01$.

C.2 Mixed-Membership Community Detection

Benchmarking. We report the mean NMI,, runtime and their standard deviations (SD) of the methods on MMBs with two equally-sized blocks and varying network sizes. Similar to the results in Section 7.4, the ORC-based method outperforms most reference methods, and successfully recovers mixed-membership community structure in almost all cases, with an NMI greater than 0.8, as we vary the network size n.

n	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-3
50	0.401 (0.068)	0.965 (0.038)	0.463 (0.151)	0.925 (0.081)	0.791 (0.155)	0.712 (0.264)
100	0.469 (0.107)	0.996 (0.011)	0.453 (0.164)	0.993 (0.015)	0.664 (0.167)	0.750 (0.233)
300	0.999 (0.002)	1 (0)	0.834 (0.264)	1 (0)	0.724 (0.123)	0.815 (0.173)
500	1 (0)	1 (0)	1 (0)	1 (0)	0.769 (0.213)	0.838 (0.205)

Table 17: Mean (SD) of the extended NMI from different methods on planted two-block MMBs with two equally-sized blocks, $n_o = 1$ node of mixed membership, $p_{in} = 0.1$, $p_{out} = 0$, and varying sizes n.

C.3 Comparison with spectral clustering methods

Apart from the two versions of Spectral Clustering algorithm as presented in Section 7, we have also considered other versions of the Spectral Clustering, where instead of using

n	Louvain	Spectra-S	Spectra-A	ORC-E	ORC-A	FRC-3
50	0.699 (0.135)	0.067 (0.016)	0.118 (0.295)	0.741 (0.121)	0.626 (0.106)	0.730 (0.121)
100	6.405 (0.922)	0.139 (0.005)	0.051 (0.005)	2.330 (0.378)	2.051 (0.336)	3.346 (0.408)
300	391.5 (28.44)	5.388 (0.391)	2.242 (0.168)	48.88 (4.877)	33.35 (3.868)	46.03 (8.584)
500	2386 (107.4)	60.76 (3.543)	28.05 (1.454)	269.6 (25.12)	183.0 (11.95)	200.2 (44.79)

Table 18: Mean (SD) of runtime (in seconds) from different methods on planted two-block MMBs with two equally-sized blocks, $n_o = 1$ node of mixed membership, $p_{in} = 0.1$, $p_{out} = 0$, and varying sizes n.

the adjacency matrix directly, we can construct an affinity matrix by (i) radius basis functions ("SC-R") or (iv) n_k nearest neighbors ("SC-N") where n_k is chosen to be 10 in the experiments. Our experimental results (see Tables 19 and 20) demonstrate that the ORC approach can also outperform other variants of Spectral Clustering, in both single- and mixed-membership community detection.

n	ORC-E		SC-R		SC-N	
	NMI	runtime	NMI	runtime	NMI	runtime
100	0.860 (0.117)	0.482 (0.018)	0.417 (0.103)	0.042 (0.006)	0.619 (0.143)	0.063 (0.002)
500	0.986 (0.018)	1.194 (0.076)	0.615 (0.082)	0.061 (0.004)	0.889 (0.082)	0.099 (0.003)
1000	1 (0)	12.66 (0.168)	0.834 (0.276)	0.833 (0.063)	1 (0)	1.203 (0.102)
3000	1 (0)	165.8 (5.805)	1 (0)	17.28 (6.174)	1 (0)	20.13 (0.692)

Table 19: Mean (SD) of NMI and runtime (in seconds) from the ORC method and two different Spectral Clustering methods on planted SBMs with two equally-sized blocks, $p_{in} = 0.1$, $p_{out} = 0.01$, and varying sizes n.

n	ORC-E		SC-R		SC-N	
	NMI	runtime	NMI	runtime	NMI	runtime
50	0.925 (0.081)	0.741 (0.121)	0.304 (0.050)	0.077 (0.011)	0.623 (0.192)	0.086 (0.003)
100	0.993 (0.015)	2.330 (0.378)	0.128 (0.021)	0.183 (0.030)	0.683 (0.133)	0.159 (0.012)
300	1 (0)	48.88 (4.877)	0.526 (0.333)	1984 (886.8)	0.951 (0.078)	7.360 (0.387)
500	1 (0)	269.6 (25.12)	0.033 (0.029)	25332 (697.7)*	0.920 (0.079)	109.6 (4.948)

Table 20: Mean (SD) of NMI and runtime (in seconds) from the ORC method and two different Spectral Clustering methods on planted MMBs with two equally-sized blocks, $p_{in} = 0.1$, $p_{out} = 0$, $n_o = 1$ node of mixed membership, and varying sizes n. *The number of clusters is chosen to be the true one (2), instead of a grid search, because the runtime is expected to exceed a day for each graph.

Furthermore, we have also performed a detailed comparison with other spectral methods, e.g., (Zhang, 2023). However, since the appearance of nodes of significantly larger degree than the others is relatively rare in our datasets, the results do not change significantly in our experiments. Therefore, we have omitted the detailed results.

C.4 Synthetic data

In Section 7.4, we observe that the performance of the algorithm via augmented FRC is not comparable with the performance of the algorithm via ORC, which raises the important problem of relating FRC to ORC. We find that FRC-3 has a stronger linear relationship with ORC as the graphs become denser; see Figures 17, 18 and 19. This is potentially because there are more and more triangular and quadrangular faces in the graphs when more edges are present, and these characteristics become more dominant for the ORC. Meanwhile, we find that, when p_{in} is relatively small, all three variants of FRC cannot really separate the edges in L between the nodes in the same community versus the others initially, which might be the reason of a relatively worse performance of the algorithm via FRC in the experiments. However, FRC-3 can separate these two types of edges when p_{in} is relatively large, which indicates that the performance is expected to be better in the dense regime, where ORC might have computational issues.

C.5 Node neighborhood measures

We give some brief computational evidence that the choice of α in the node neighborhood measure as in Equation (4.2) has little impact on the performance of our curvature-based algorithms. Over a range of different choices of α, and for two real-data networks, we see that the results are highly conserved.

Data\α	0	0.1	0.2	0.3	0.4	0.5	0.6
DBLP-1	0.689	0.689	0.689	0.689	0.689	0.689	0.689
DBLP-2	0.969	0.969	0.969	0.969	0.969	0.969	0.969

Table 21: Extended NMI of ORC-E via line graph curvature on the real data, where we choose different values of α in Equation (4.2).

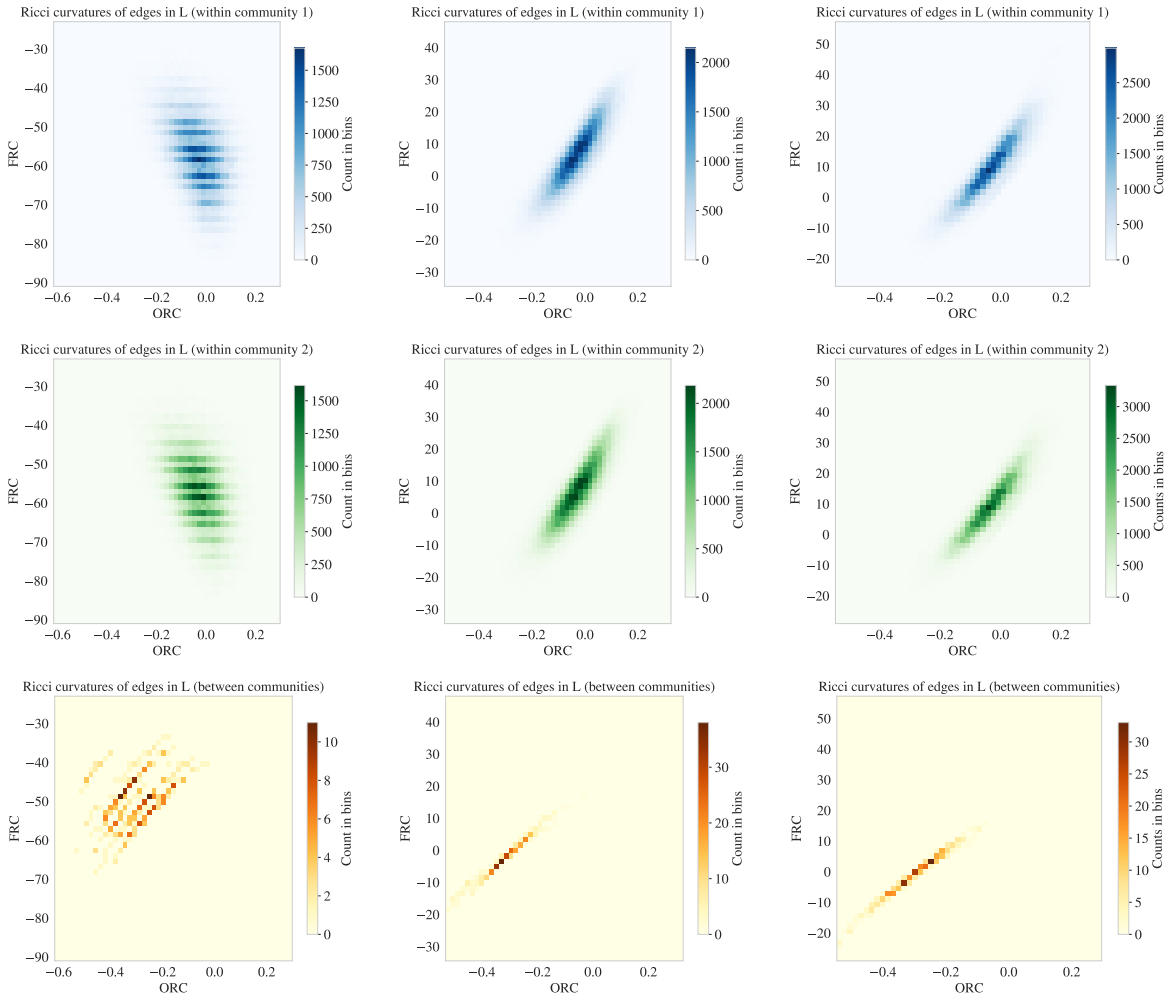


Figure 17: 2-d histograms of the ORC-E versus the FRC-1 (left), FRC-2 (middle) and FRC-3 (right) for edges in L, where the networks are obtained from the planted two-block MMB of size $n = 300$, $p_{in} = 0.1$, $p_{out} = 0$, $n_o = 1$ node of mixed membership, and we generate $n_s = 10$ networks for the results. (Row 1 and 2: edges between nodes within communities 1, in blue and 2, in green, respectively, in L; Row 3: edges between nodes in the different communities, in red).

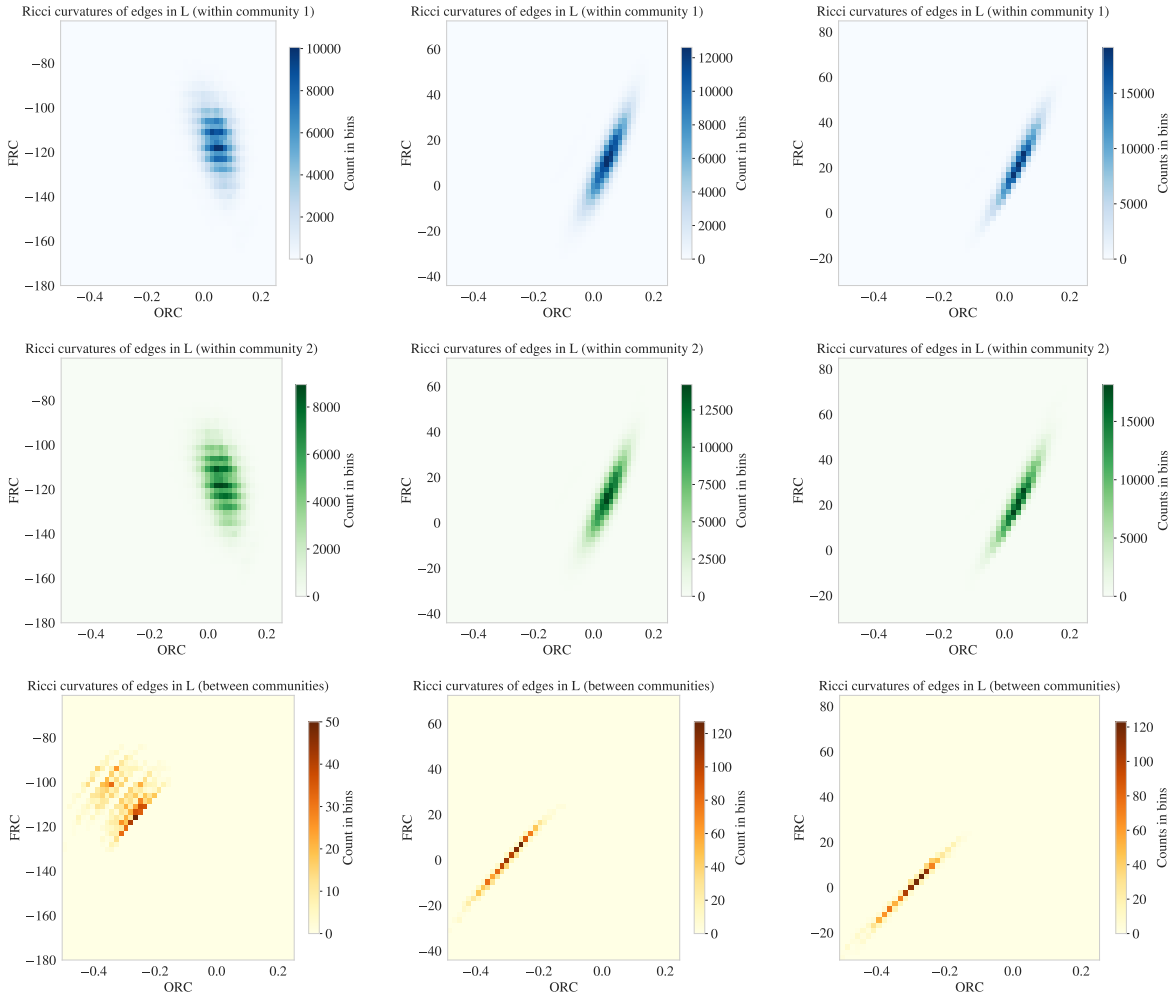


Figure 18: 2-d histograms of the ORC-E versus the FRC-1 (left), FRC-2 (middle) and FRC-3 (right) for edges in L, where the networks are obtained from the planted two-block MMB of size $n = 300$, $p_{in} = 0.2$, $p_{out} = 0$, $n_o = 1$ node of mixed membership, and $n_s = 10$.

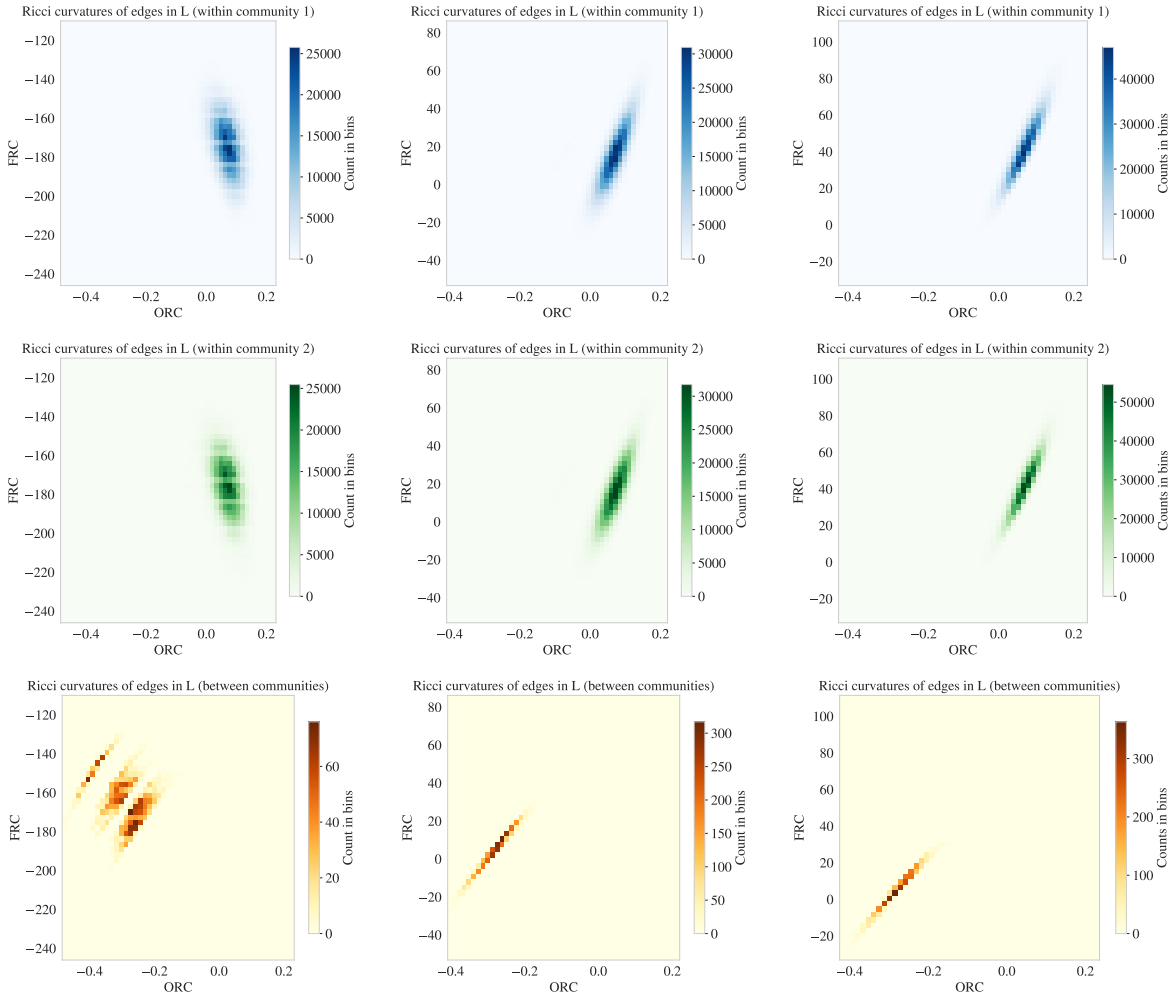


Figure 19: 2-d histograms of the ORC-E versus the FRC-1 (left), FRC-2 (middle) and FRC-3 (right) for edges in L, where the networks are obtained from the planted two-block MMB of size $n = 300$, $p_{in} = 0.3$, $p_{out} = 0$, $n_o = 1$ node of mixed membership, and $n_s = 10$.

References

E. Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *Advances in neural information processing systems*, 21, 2008.

F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J. Young, and G. Petri. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 874:1–92, 2020.

A. Benson, D. Gleich, and J. Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

E. Bloch. Combinatorial Ricci curvature for polyhedral surfaces and posets. *arXiv:1406.4598*, 2014.

V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), oct 2008.

J. Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In *Proceedings of the Princeton conference in honor of Professor S. Bochner*, pages 195–199, 1969.

Z. Chen, X. Li, and J. Bruna. Supervised community detection with line graph neural networks. *arXiv:1705.08415*, 2017.

M. de Reus and M. van den Heuvel. Rich club organization and intermodule communication in the cat connectome. *Journal of Neuroscience*, 33(32):12929–12939, 2013.

P. Elumalai, Y. Yadav, N. Williams, E. Saucan, J. Jost, and A. Samal. Graph Ricci curvatures reveal atypical functional connectivity in autism spectrum disorder. *Scientific reports*, 12(1):1–19, 2022.

M. Erbar and J. Maas. Ricci curvature of finite markov chains via convexity of the entropy. *Archive for Rational Mechanics and Analysis*, 206:997–1038, 2012.

T. Evans and R. Lambiotte. Line graphs of weighted networks for overlapping communities. *The European Physical Journal B*, 77(2):265–272, September 2010.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2): 298–305, 1973.

R. Forman. Bochner's method for cell complexes and combinatorial Ricci curvature. *Discrete and Computational Geometry*, 29(3):323–374, 2003.

M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

S. Gómez, P. Jensen, and A. Arenas. Analysis of community structure in networks of correlated data. *Physical Review E*, 80(1):016114, 2009.

A. Gosztolai and A. Arnaudon. Unfolding the multiscale structure of networks with dynamical Ollivier-Ricci curvature. *Nature Communications*, 12(1):4561, 2021.

M. Gromov. Hyperbolic groups. In *Essays in Group Theory*, volume 8. Mathematical Sciences Research Institute Publications. Springer, NY, 1987.

L. Harriger, M. van den Heuvel, and O. Sporns. Rich club organization of macaque cerebral cortex and its role in network communication. *PLOS ONE*, 7(9):1–13, 09 2012.

P. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

S. Hopkins and D. Steurer. Bayesian estimation from few samples: community detection and related problems, September 2017.

T. Jarrell, Y. Wang, A. Bloniarz, C. Brittin, M. Xu, J. Thomson, D. Albertson, D. Hall, and S. Emmons. The connectome of a decision-making neural network. *Science*, 337 (6093):437–444, 2012.

J. Jin, Z. Ke, and S. Luo. Estimating network memberships by simplex vertex hunting. *arXiv:1708.07852*, 2017.

J. Jost and S. Liu. Ollivier's Ricci curvature, local clustering and curvature-dimension inequalities on graphs. *Discrete & Computational Geometry*, 51(2):300–322, 2014.

J. Jost and F. Münch. Characterizations of forman curvature. *arXiv:2110.04554*, 2021.

F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborovmá, and P. Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

H. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

W. Leal, G. Restrepo, P. F. Stadler, and J. Jost. Forman–Ricci curvature for hypergraphs. *Advances in Complex Systems*, 24(01):2150003, 2021.

P. Lehot. An optimal algorithm to detect a line graph and output its root graph. *Journal of the ACM (JACM)*, 21(4):569–575, 1974.

J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

J. Leskovec and J. Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Y. Lin, L. Lu, and S. Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal*, 63(4): 605 – 627, 2011.

Z. Lubberts, A. Athreya, Y. Park, and C. E. Priebe. Beyond the adjacency matrix: random line graphs and inference for networks with edge attributes. *arXiv:2103.14726*, 2021.

S. Lubold, A. Chandrasekhar, and T. McCormick. Identifying the latent space geometry of network models through analysis of curvature. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(2):240–292, 2023.

C. Ni, Y. Lin, J. Gao, X. Gu, and E. Saucan. Ricci curvature of the internet topology. In *2015 IEEE conference on computer communications (INFOCOM)*, pages 2758–2766. IEEE, 2015.

C. Ni, Y. Lin, F. Luo, and J. Gao. Community detection on networks with ricci flow. *Scientific reports*, 9(1):1–12, 2019.

Y. Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

Y. Ollivier. A survey of Ricci curvature for metric spaces and Markov chains. In *Probabilistic Approach to Geometry*, pages 343–381. Mathematical Society of Japan, 2010.

D. T. Painter, B. C. Daniels, and J. Jost. Network analysis for the digital humanities: Principles, problems, extensions. *ISIS*, 110(3):538–554, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

M. Penrose. *Random Geometric Graphs*. Oxford studies in probability. Oxford University Press, 2003.

M. Pouryahya, J. Mathews, and A. Tannenbaum. Comparing three notions of discrete Ricci curvature on biological networks. *arXiv:1712.02943*, 2017.

A. Samal, R. Sreejith, J. Gu, S. Liu, E. Saucan, and J. Jost. Comparative analysis of two discretizations of Ricci curvature for complex networks. *Scientific reports*, 8(1):1–16, 2018.

R. Sandhu, T. Georgiou, and A. Tannenbaum. Ricci curvature: An economic indicator for market fragility and systemic risk. *Science advances*, 2(5):e1501495, 2016.

E. Saucan and M. Weber. Forman's Ricci curvature-from networks to hypernetworks. In *International conference on complex networks and their applications*, pages 706–717. Springer, 2018.

E. Saucan, A. Samal, M. Weber, and J. Jost. Discrete curvatures and network analysis. *MATCH*, 80(3):605–622, 2018.

J. Sia, E. Jonckheere, and P. Bogdan. Ollivier-Ricci curvature-based method to community detection in complex networks. *Scientific reports*, 9(1):1–12, 2019.

R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

D. Spielman and S. Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proceedings of 37th conference on foundations of computer science*, pages 96–105. IEEE, 1996.

A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

A. Tannenbaum, C. Sander, L. Zhu, R. Sandhu, I. Kolesov, E. Reznik, Y. Senbabaoglu, and T. Georgiou. Ricci curvature and robustness of cancer networks. *arXiv:1502.04512*, 2015.

Y. Tian, Z. Lubberts, and M. Weber. Mixed-membership community detection via line graph curvature. In *Proceedings of the 1st NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, volume 197 of *Proceedings of Machine Learning Research*, pages 219–233. PMLR, 03 Dec 2023.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(1):395–416, 2007.

D. Watts and S. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684): 440–442, 1998.

M. Weber. Neighborhood growth determines geometric priors for relational representation learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 266–276. PMLR, 26–28 Aug 2020.

M. Weber and M. Nickel. Curvature and representation learning: Identifying embedding spaces for relational data. In *NeurIPS Relational Representation Learning*, 2018.

M. Weber, E. Saucan, and J. Jost. Characterizing complex networks with Forman-Ricci curvature and associated geometric flows. *Journal of Complex Networks*, 5(4):527–550, 2017a.

M. Weber, E. Saucan, and J. Jost. Coarse geometry of evolving networks. *Journal of Complex Networks*, 6(5):706–732, 2017b.

M. Weber, J. Stelzer, E. Saucan, A. Naitsat, G. Lohmann, and J. Jost. Curvature-based methods for brain network analysis. *arXiv:1707.00180*, 2017c.

M. Weber, J. Jost, and E. Saucan. Detecting the coarse geometry of networks. In *NeurIPS Relational Representation Learning*, 2018.

H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932.

J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (csur)*, 45(4): 1–35, 2013.

J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596, 2013.

J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42:181–213, 2015.

Anderson Ye Zhang. Fundamental limits of spectral clustering in stochastic block models. *arXiv:2301.09289*, 2023.

Y. Zhang, E. Levina, and J. Zhu. Detecting Overlapping Communities in Networks Using Spectral Methods. *SIAM Journal on Mathematics of Data Science*, 2(2):265–283, January 2020. Publisher: Society for Industrial and Applied Mathematics.

M. Zhu and A. Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Comput. Stat. Data Anal.*, (2):918–930, 2006.