Maximum Likelihood Estimation on Stochastic Blockmodels for Directed Graph Clustering

Mihai Cucuringu MIHAI.CUCURINGU@STATS.OX.AC.UK
Department of Statistics and Mathematical Institute
University of Oxford, Oxford, UK
The Alan Turing Institute, London, UK

Xiaowen Dong XIAOWEN.DONG@ENG.OX.AC.UK
Department of Engineering
University of Oxford, Oxford, UK
The Alan Turing Institute, London, UK

Ning Zhang* NING.ZHANG@STATS.OX.AC.UK
Department of Statistics
University of Oxford, Oxford, UK

Abstract

This paper studies the directed graph clustering problem through the lens of statistics, where we formulate clustering as estimating underlying communities in the directed stochastic block model (DSBM). We conduct the maximum likelihood estimation (MLE) on the DSBM and thereby ascertain the most probable community assignment given the observed graph structure. In addition to the statistical point of view, we further establish the equivalence between this MLE formulation and a novel flow optimization heuristic, which jointly considers two important directed graph statistics: edge density and edge orientation. Building on this new formulation of directed clustering, we introduce two efficient and interpretable directed clustering algorithms, a spectral clustering algorithm and a semidefinite programming based clustering algorithm. We provide a theoretical upper bound on the number of misclustered vertices of the spectral clustering algorithm using tools from matrix perturbation theory. We compare, both quantitatively and qualitatively, our proposed algorithms with existing directed clustering methods on both synthetic and real-world data, thus providing further ground to our theoretical contributions.

Keywords: graph clustering, directed graphs, maximum likelihood estimation, spectral methods, matrix perturbation analysis, semidefinite programming.

Authors are listed in alphabetical order.
∗ This is the corresponding author.

Contents

1. Introduction

Graph clustering, or community detection, aims to partition a graph into disjoint clusters (or communities) such that vertices within the same cluster are more "similar" to each other compared to vertices from different clusters. As one of the most fundamental graph data analysis methods, graph clustering is well motivated by the need to reveal hidden patterns in real-world networks, such as segmenting images Shi and Malik (2000), finding communities in social networks Oliveira and Gama (2012), and detecting pairwise lead-lag communities in financial networks Bennett et al. (2022).

There has been extensive research on clustering undirected graphs. In particular, a large volume of work Amini and Levina (2018); Hajek et al. (2016); Montanari and Sen (2016); Abbe et al. (2015); Amini et al. (2013); Decelle et al. (2011); Chen et al. (2014); Li et al. (2021) formulates graph clustering as estimating the planted communities of random graph models, typically the stochastic block model (SBM) Holland et al. (1983) or its variants Karrer and Newman (2011); Avrachenkov et al. (2020). These lines of work start with applying statistical estimation methods, including maximum likelihood estimation and maximum posterior marginal estimation, in order to estimate community labels. This estimation procedure subsequently yields a combinatorial optimization formulation of the graph clustering problem, such as minimizing the cross-community edges Abbe et al. (2015) (similar to *graph cut* Shi and Malik (2000)), and modularity maximization Newman (2016). These combinatorial optimization problems can further be relaxed or approximated, leading to computationally efficient clustering algorithms using spectral methods Von Luxburg (2007); Newman (2013), semidefinite programming(SDP) Amini and Levina (2018); Hajek et al. (2016); Montanari and Sen (2016); Abbe et al. (2015); Li et al. (2021), and belief propagation algorithms Decelle et al. (2011). Such a methodological framework enables the integration of statistics, optimization, and spectral methods, and leads to a multi-viewed understanding of the strengths and limitations of the clustering algorithms.

While most of the existing studies on graph clustering are only targeted for undirected graphs, in a number of important applications, the relationships between vertices are not symmetric, as seen in causal relationships Pearl and Verma (1987), interbank debt Acemoglu et al. (2015), and paper citations An et al. (2004). Employing undirected graph representation in the aforementioned scenarios results in a loss of valuable information pertaining to edge directionality. This underscores the necessity for studying the directed graph clustering problem and developing fast and robust algorithms customized to the specific task at hand. While it is not applicable to naively transfer undirected clustering analysis to directed graphs due to the lack of symmetry of the adjacency matrix, a number

of studies attempt to circumvent this challenge in different ways Malliaros and Vazirgiannis (2013); we survey below two closely related approaches.

One line of work suggests a two-step framework, where they first restore the graph symmetry and then apply classical undirected clustering algorithms. For example, some propose to cluster the bibliographic coupling matrix AA^T Kessler (1963), the co-citation matrix A^TA Small (1973), and bibliometric symmetrization AA^T+A^TA Satuluri and Parthasarathy (2011). Performing clustering on such product of graph adjacency matrices implicitly groups vertices based on some notion of higher-order structure. For instance, constructing AA^T involves counting the number of common "offspring" vertices. In a similar spirit, Rohe et al. (2016) proposes a spectral algorithm denoted as DI-SIM, which builds on a dual measure of similarity from both "parent" vertices and "offspring" vertices.

Another more recent line of research adopts complex-valued Hermitian matrices to represent directed graphs. These studies explore the graph structural information through the analysis of the spectrum of Hermitian matrices, which leads to the creation of a variety of directed clustering algorithms Cucuringu et al. (2020); Fanuel et al. (2017); Laenen and Sun (2020). In Cucuringu et al. (2020), the authors suggest using a skew-symmetric Hermitian matrix, employing $\pm i$ to represent directed edges, and subsequently clustering vertices based on their embedding in the associated eigenspace. Later follow-up works Hayashi et al. (2022); Laenen (2019) also point out a connection between the Hermitian clustering algorithm in Cucuringu et al. (2020) and an optimization heuristic that maximizes the net flow between two clusters. In Laenen and Sun (2020), the authors choose the k-th root of unity to indicate a directed edge. They define a flow ratio optimization scheme and show that the optimal partition is given at the bottom eigenspace of their proposed Hermitian matrix. Other than optimization heuristics, the authors in Fanuel et al. (2017) also relate directed clustering to quantum physics heuristics. They apply the Magnetic Laplacian for spectral clustering on directed graphs, where this combinatorial Laplacian was initially designed for studying the quantum dynamics of particles under the influence of magnetic fluxes Shubin (1994).

Despite the success of the aforementioned methods in a variety of directed clustering tasks, there remains a significant lack of insights from a statistical perspective. Such a point of view is helpful as it allows one to justify why a particular clustering algorithm design is deemed suitable based on the statistical properties of the data, i.e., the structures and randomness inherent in random graph models. The rationales behind existing directed clustering algorithms rely on prespecified favoured patterns or optimization heuristics. For example, Cucuringu et al. (2020); Hayashi et al. (2022) optimize the between-cluster net flow, while Laenen and Sun (2020); Rohe et al. (2016); Satuluri and Parthasarathy (2011); Small (1973); Kessler (1963) count higher-order patterns. These algorithms, customized for certain prespecified patterns, have weak connections to the properties of the real-world data being studied, and there is a lack of rigorous arguments as to why and when those

methods are desirable or effective. This motivates us to explore a design that comes with a theoretical justification based on the statistical properties of the data. In this paper, we study the maximum likelihood estimation on the Directed Stochastic Block Model (DSBM), and its implication for the development of efficient directed clustering algorithms. Our work is mainly inspired by the multi-viewed research on the statistical estimation of the SBM and the Hermitian matrix representations of directed graphs. We summarize our main contributions as follows.

1. **Problem formulation.** Under the DSBM setting, we propose to formulate a maximum likelihood estimation (MLE) on the community labelling, where we maximize the log-likelihood function and thereby ascertain the most probable community assignment given the observed graph structure. Building on the MLE derivation, we introduce a novel Hermitian matrix representation for clustering directed graphs. In contrast to Hermitian matrices proposed in earlier studies Fanuel et al. (2017); Cucuringu et al. (2020); Laenen and Sun (2020), where the underlying optimization interpretations are heuristic, our proposed Hermitian matrix is derived from a well-established statistical estimation method. The connection between our proposed Hermitian matrix and MLE is established through the observation that the quadratic form of the Hermitian matrix associated with the complex community indicator vector, corresponds to the optimization objective of the MLE (Theorem 1). Furthermore, we establish the equivalence between the MLE on DSBM and a new (regularized) flow optimization heuristic, as detailed in Section 3.2. This optimization heuristic jointly considers both edge density and edge orientation when clustering directed graphs. Compared with existing optimization heuristics that also take into account the cross-cluster edge density and orientation Fanuel et al. (2017, 2018), our formulation provides a theoretical justification for how we balance the weighting parameters of the two terms. This flow optimization interpretation also extends the flexibility of our formulation, as it allows one to go beyond the assumptions on the underlying statistical model, and customize the weighting parameter between the cost incurred from the edge density and the one incurred from the edge orientations, based on one's domain knowledge.

2 **Efficient algorithms.** Based on the above theoretical framework, we introduce two new directed clustering algorithms, a spectral clustering algorithm (Algorithm 1) and a semidefinite programming (SDP) clustering algorithm (Algorithm 2), both of which are derived through relaxing the combinatorial optimization problem induced by the MLE. As these MLE-driven algorithms require the DSBM parameters as input, we adapt the iterative approach from Newman (2016) to learn the model parameters from the observable graph data. The combination of our proposed algorithms and the interactive learning method not only enables efficient computation on clusters, but also ensures a self-adaptive process that operates without the need for any prior knowledge. We compare our algorithms, both quantitatively and qualitatively, with

existing directed clustering methods on synthetic data (Section 5.2) and real-world data (Section 5.3). From these experiments, we observe that our proposed algorithms consistently outperform existing methods in most of the cases we have experimented with, thus providing further ground to our theoretical contributions.

3. **Theoretical guarantee.** We present a theoretical upper bound on the recovery error of the spectral clustering algorithm (Algorithm 1). For graphs generated from the DSBM, we establish, in Theorem 2, a high-probability upper bound on the number of misclustered vertices using tools from matrix perturbation analysis. Prior work Cucuringu et al. (2020) also proves an upper bound on the misclustering error on DSBM. The main difference between our bound and prior work lies in the DSBM setting. While Cucuringu et al. (2020) considers multiple equal-sized clusters with uniform edge density, i.e., $p = q$, this study focuses on a more general two-cluster case, where our analysis also covers scenarios with inhomogeneous edge density ($p \neq q$) and unequal cluster size ($n_1 \neq n_2$). Furthermore, the analysis in Cucuringu et al. (2020) involves a directed application of the Davis-Kahan theorem, which imposes an extra condition to ensure the gap between eigenvalues of the population matrix $\mathbb{E}[H]$ and the sample matrix H to be large enough. We circumvent this issue by adapting a variant of the Davis-Kahan theorem from Vu et al. (2013); Yu et al. (2015) into our proof, which allows us to derive an error bound that only considers eigenvalues of the population matrix.

The rest of the paper is organized as follows. In Section 2 we provide notations and background, and introduce the clustering problem under consideration. In Section 3, we introduce our statistical estimation formulation and summarize the main results of this study. In Section 4, we sketch the proof for the error bound on the Algorithm 1, and defer some of the proof details to the appendix. In Section 5, we explain the implementation details of our proposed algorithms, and report experimental results on both synthetic and real-world data sets. In Section 6, we discuss the advantages and limitations of our work, and highlight potential future improvements and avenues of research.

2. Preliminaries and problem formulation

2.1 Basic notations

Let $G(\mathcal{V}, \mathcal{E})$ be a directed graph on vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. For a pair of vertices $u, v \in \mathcal{V}$, we denote $u \rightsquigarrow v$ if there is an edge pointing from u to v and we denote $u \not\sim v$ if there is no edge between u and v. The edge set is a collection of ordered vertices pairs, where $\forall (u, v) \in \mathcal{E}$, we have $u \rightsquigarrow v$. A directed graph can be represented by its adjacency matrix $A \in \{0, 1\}^{N \times N}$, where $A_{uv} = 1$ iff $u \rightsquigarrow v$. This research studies two-cluster directed graphs, and we denote the clusters as $\mathcal{C}_1$ and $\mathcal{C}_2$, where $\mathcal{C}_1, \mathcal{C}_2$ are disjoint subsets of $\mathcal{V}$.

This study involves both real-valued matrices in $\mathbb{R}^{n\times n}$ and complex-valued matrices in $\mathbb{C}^{n\times n}$. We use i to denote the imaginary unit $\sqrt{-1}$. For $x \in \mathbb{C}$, we denote the conjugate of x as $\overline{x}$, and use $|x| =$ to represent the norm of x. Let $\Re(\cdot)$ be the operation of taking the real part of a complex number, and $\Im(\cdot)$ denote taking the imaginary part. When the inputs of $\Re(\cdot)$ and $\Im(\cdot)$ are matrices, we consider them as elementwise operations on every entry of the matrix.

For a matrix H, we use H^T to denote its transpose and H^* to denote the conjugate transpose and $H^* = \overline{H}^T$. Let $\mathcal{H}^{n\times n}$ be the set of Hermitian matrices of size n, where $\forall H \in \mathcal{H}^{n\times n}$ we have $H = H^*$. For an arbitrary Hermitian matrix H, it has n real eigenvalues, and throughout this paper, the eigenvalues are consistently organized in descending order of magnitude, i.e., $|\lambda_1(H)| \geq |\lambda_2(H)| \geq \cdots |\lambda_n(H)|$. We also employ several commonly used matrices: we use I_n to denote the identity matrix of size n, and J_n to denote the square all-one matrix of size n, while $\mathbb{1}_n$ is used to represent the all-one vector of length n. When the matrix dimension can be inferred from the context, we may drop the subscript for the sake of conciseness. This paper makes use of several matrix norms: we use $\|H\|$ to denote the spectral norm, which is the largest magnitude of any eigenvalues $|\lambda_1(H)|$. We let $\|H\|_F$ denote the Frobenius norm, where $\|H\|_F = \sqrt{\sum_j \lambda_j^2(H)}$. For two matrices H_1, H_2 with the same number of rows, we use $[H_1, H_2]$ to denote a new matrix by concatenating the columns of H_1 and H_2. Let $\mathrm{diag}(H)$ denote the operator that creates a diagonal matrix by considering the main diagonal elements of M. We use $\mathrm{Tr}(H)$ to denote the trace of the matrix H.

This paper considers large graphs, where we provide both non-asymptotic and asymptotic analysis. When it comes to asymptotic analysis, i.e., considering the graph size n converges to infinity, we use the big-O notation as conventions: we use $g_n = o(f_n)$ to denote that g_n is asymptotically dominate by f_n, i.e., $\lim_{n\to\infty} \frac{g_n}{f_n} = 0$. We use $g_n = O(f_n)$ when g_n is asymptotically bounded above by f_n, i.e., $\limsup_{n\to\infty} \frac{g_n}{f_n} < \infty$. We denote $g_n = \Theta(f_n)$ as g_n is bounded both above and below by f_n asymptotically, in other words, $g_n = O(f_n)$ and $f_n = O(g_n)$. We denote $g_n = \Omega(f_n)$ as g_n is bounded below by f_n, i.e., $\limsup_{n\to\infty} \frac{g_n}{f_n} > 0$. We denote g_n dominate f_n asymptotically as $g_n = \omega(f_n)$, where $\limsup_{n\to\infty} \frac{g_n}{f_n} = \infty$. We refer to Table 2 in Appendix A for a summary of notions used in this paper.

2.2 Directed Stochastic Block Model

The DSBM is a generative model for random directed graphs, initially proposed in Cucuringu et al. (2020). In this paper, we specialize their setting to the two-community case, consisting of the source community $\mathcal{C}_1$ of size n_1 and the target community $\mathcal{C}_2$ of size n_2. Here, the community membership labels are considered as *fixed parameters* that are unknown,

instead of treated as *latent variables*. We introduce a vector σ to indicate the community membership in the DSBM where $\sigma_u = \sigma_v$ iff vertex u, v belongs to the same community. The DSBM encodes the community information into the graph topology through the edge density parameters p, q and the edge orientation parameter η. To be more specific, for a directed graph sampled from the DSBM (n_1, n_2, p, q, η), the edges are generated independently conditioning on the community labelling σ as follows

- if a pair of vertices u and v belongs to the same cluster, then with probability (w.p.) p there exists an edge between them. This within-community edge is unordered in the sense that the probability of this edge pointing from u to v is the same as the probability of it pointing from v to u. In other words, we have that

$$\begin{cases} A_{uv} = 1, A_{vu} = 0 & \text{w.p. } p/2, \\ A_{uv} = 0, A_{vu} = 1 & \text{w.p. } p/2, \\ A_{uv} = A_{vu} = 0 & \text{w.p. } 1-p. \end{cases}$$

- if $u \in \mathcal{C}_1$ and $v \in \mathcal{C}_2$, then we have that the edge exists with probability q. The cross-community edge is oriented in the sense that the probability of this edge pointing from the $\mathcal{C}_1$ vertex to the $\mathcal{C}_2$ vertex is $1-\eta$, i.e.,

$$\begin{cases} A_{uv} = 1, A_{vu} = 0 & \text{w.p. } (1-\eta)q \ , \\ A_{uv} = 0, A_{vu} = 1 & \text{w.p. } \eta q, \\ A_{uv} = A_{vu} = 0 & \text{w.p. } 1-q. \end{cases}$$

Therefore, the conditional probability of a directed graph with adjacency matrix A can be written as

$$\mathbb{P}(A|\sigma) = \prod_{(u<v)} \mathbb{P}(A_{uv}|\sigma_u, \sigma_v),$$

where the probability distribution of a vertex pair $\mathbb{P}(A_{uv}|\sigma_u, \sigma_v)$ is specified in the above discussed six cases.

We use $p_{\max} = \max\{p, q\}$ to denote the maximum edge probability in DSBM (n_1, n_2, p, q, η). Throughout the discussion in this paper, we assume that the maximum edge probability is above the connectivity threshold Frieze and Karoński (2016), i.e.,

$$p_{\max} = \Omega(\log N/N). \tag{A-1}$$

This assumption is a necessary condition for us to provide a theoretical upper bound on the number of misclustered vertices. Otherwise, if $p_{\max} = o(\log N/N)$, then, with high probability, the sampled graphs contain multiple components, and there is no proper way to determine a cluster membership assignment over different components.

2.3 Problem formulation

This study takes a statistical point of view on the clustering problem, considering a cluster as a collection of statistically equivalent vertices. Following this line of thought, we assume graphs are generated from the DSBM (n_1, n_2, p, q, η), where the edge generating process between a pair of vertices depends only on the respective communities to which they belong. Our goal of clustering is to estimate the underlying communities only based on the observed graph topology, without any additional side information on the underlying community structure. We use the indicator vector σ to denote the ground truth community labelling of the DSBM, and we use $\hat{\sigma}$ to denote the estimated community labelling vector. We evaluate the efficacy of the community recovery $\hat{\sigma}$ by counting the number of misclustered vertices $l(\sigma, \hat{\sigma})$, where

$$l(\sigma, \hat{\sigma}) = \sum_{j \in \mathcal{V}} \mathbb{1}(\sigma_j \neq \hat{\sigma}_j).$$

3. Main results

3.1 Maximum likelihood estimation on DSBM

Given a graph with adjacency matrix A generated from the DSBM, we infer the community labels in a way that renders the observed graph topology most probable. This intuition is formally achieved by applying the maximum likelihood estimation (MLE), where the desired community labelling maximizes the log-likelihood function. To be more concrete, we consider a directed graph with adjacency matrix A sampled from the DSBM, and we also assume that the DSBM parameters p, q, η are known. Then, the MLE on community assignments can be formally written as

$$\hat{\sigma}_{\text{MLE}} = \arg\max_{\sigma} \mathcal{L}(A; \sigma), \tag{1}$$

where $\mathcal{L}(A; \sigma)$ is the log-likelihood function and

$$\mathcal{L}(A; \sigma) = \sum_{u<v} \log(\mathbb{P}(A_{u,v} | \sigma_u, \sigma_v)). \tag{2}$$

We present in Theorem 1 the explicit optimization formulation derived from the MLE on DSBM (1). The key step in the derivation is to describe the optimization objective, the log-likelihood function $\mathcal{L}(A; \sigma)$. We manage this simply by grouping the terms in (2) according to the community assignment of vertex pairs in each term, which we summarize in Lemma 7. We then convert the real-valued optimization problem in Lemma 7 to its complex equivalence. The purpose of this conversion is to obtain a more compact expression of the optimization problem, where the optimization objective is a simple quadratic form. The detailed proof of Theorem 1 can be found in Appendix B.

Theorem 1 (MLE on DSBM) *Consider a directed graph with adjacency matrix A generated from the model DSBM(n_1, n_2, p, q, η). Let $\mathbf{x} \in \{i, 1\}^N$ be the indicator vector, where $\mathbf{x}_u = i$ if $u \in \mathcal{C}_1$, and $\mathbf{x}_u = 1$ if $u \in \mathcal{C}_2$. The maximum likelihood estimation on the community labels is equivalent to solving the following complex optimization problem*

$$\begin{aligned} \max \quad & \mathbf{x}^* \widetilde{H} \mathbf{x} \\ s.t. \quad & \mathbf{x} \in \{1, i\}^N, \end{aligned} \tag{Herm-MLE}$$

where $\widetilde{H}$ is a Hermitian matrix defined as

$$\begin{aligned} \widetilde{H} &= i \log \frac{1-\eta}{\eta}(A - A^T) + \log \frac{p^2(1-p)^2}{4\eta(1-\eta)q^2(1-q)^2}(A + A^T) + 2 \log \frac{1-p}{1-q}(J - I) \\ &\triangleq w_i i(A - A^T) + w_r(A + A^T) + w_c(J - I). \end{aligned} \tag{3}$$

3.2 A regularized flow optimization interpretation

In this section, we present a different view on the (Herm-MLE) problem by relating it to a flow optimization heuristic. In particular, we explain how the real and imaginary parts of the Hermitian matrix contribute to the edge density optimization and edge orientation optimization separately. This alternative view extends the flexibility of our proposed methodology, as it allows one to consider different matrix constructions that depend on their own judgment about what a "good" metric for modularity in directed graphs ought to be.

We start by defining graph statistics that are of interest in directed clustering tasks. Given two clusters $\mathcal{C}_1, \mathcal{C}_2$ in a directed graph, we use $\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)$ to denote the *total flow*, which is the number of cross-cluster edges given by

$$\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2) = \sum_{u \in \mathcal{C}_1 v \in \mathcal{C}_2} (A_{uv} + A_{vu}).$$

We use $\mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2)$ to denote the *net flow* from $\mathcal{C}_1$ to $\mathcal{C}_2$, which is the number of edges pointing from $\mathcal{C}_1$ to $\mathcal{C}_2$ subtracting the number of edges from $\mathcal{C}_2$ to $\mathcal{C}_1$

$$\mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2) = \sum_{u \in \mathcal{C}_1 v \in \mathcal{C}_2} (A_{uv} - A_{vu}).$$

Recall that the Hermitian matrix we derived is $\widetilde{H} = w_r(A + A^T) + i w_i(A - A^T) + w_c(J - I)$, and the objective in (Herm-MLE) is

$$\mathbf{x}^* \widetilde{H} \mathbf{x} = w_r \mathbf{x}^*(A + A^T)\mathbf{x} + i w_i \mathbf{x}^*(A - A^T)\mathbf{x} + w_c \mathbf{x}^*(J - I)\mathbf{x}, \tag{4}$$

where the indicator vector $\mathbf{x} \in \{1, i\}^N$ and $\mathbf{x}_u = i$ if $u \in \mathcal{C}_1$ and $\mathbf{x}_u = 1$ if $u \in \mathcal{C}_2$.

For the first term in (4), one can easily verify that $\mathbf{x}^*(A+A^T)\mathbf{x}$ counts the number of edges within $\mathcal{C}_1$ plus the number of edges within $\mathcal{C}_2$, which can be equivalently written as

$$\frac{1}{2}\mathbf{x}^*(A+A^T)\mathbf{x} = C - \mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2), \tag{5}$$

where the constant C is the total number of edges in the graph.

For the second term in (4), one can derive that $i\mathbf{x}^*(A-A^T)\mathbf{x}$ is the number of edges pointing from $\mathcal{C}_1$ to $\mathcal{C}_2$ subtracting the number of edges from $\mathcal{C}_2$ to $\mathcal{C}_1$, and thus

$$\frac{i}{2}\mathbf{x}^*(A-A^T)\mathbf{x} = \mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2). \tag{6}$$

The last term in (4) is weighted by $w_c = \log(1-p) - \log(1-q) = O(p-q)$, which is a very small constant. If we ignore this term for the moment, we then have that (Herm-MLE) optimizes a weighted combination of net flow and total flow: $-w_r\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2) + w_i\mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2)$. This optimization view agrees with the intuition that we aim to find a proper partition that considers both the edge density difference and the edge orientation between clusters. This optimization heuristic also shares a similar intuition with the cut imbalance ratio $\frac{|\mathbf{NF}(\mathcal{C}_1,\mathcal{C}_2)|}{2\mathbf{TF}(\mathcal{C}_1,\mathcal{C}_2)}$ from Cucuringu et al. (2020), where the authors proposed it as a measure of edge imbalance between clusters.

The last term in (4) $\mathbf{x}^*(J-I)\mathbf{x}$ simply calculates $\frac{1}{2}(n_1^2+n_2^2)$. By maximizing it, we implicitly penalize or encourage imbalanced clusters, depending on the sign of w_c. Adding an all-one matrix into the graph representation matrix also appears in previous studies, where this technique is known as *regularization*. Existing studies such as Amini et al. (2013); Joseph and Yu (2016); Le et al. (2017) have shown both theoretically and empirically that adding a regularization term will improve the performance of spectral clustering in the sparse regime, where the graph edge density is $p, q = o(\frac{\log N}{N})$, even as low as $p, q = \Theta(\frac{1}{n})$ which is the very sparse regime often arising in certain applications involving large-scale graphs. All the above studies only consider undirected graphs, with a real-valued matrix representation, and there is no analysis known for directed graphs.

Statistical estimation v.s. combinatorial optimization. Although in this section we have demonstrated the equivalence between MLE and the regularized flow optimization, it is still worth mentioning the difference between an optimization view and a statistical view. From a statistical perspective, clustering aims at recovering the probabilistic equivalent structure, while from an optimization standpoint, clustering is purely driven by finding structures that optimize the objectives, which can be irrelevant to the underlying data-generating model. In cases where the underlying probabilistic models are unknown or mathematically intractable, an optimization objective can explicitly guide the goal of clustering and serve as a metric for evaluating cluster results. In this vein, our flow optimization framework is ready to be extended to a more general weighted graph setting

without specifying a generating model. In addition to model independence, an optimization framework also naturally provides a way to integrate prior knowledge as constraints, leading to new constrained optimization formulations on directed clustering.

3.3 Proposed algorithms

In Section 3.1, we derive the optimization formulation for directed clustering (Herm-MLE) through applying MLE on the DSBM, and in Section 3.2, we discuss the heuristic interpretation for this problem. We now move on to presenting two efficient algorithms for solving the optimization problem (Herm-MLE). First note that exactly solving the above combinatorial optimization problem (Herm-MLE) involves enumerating all 2^N combinations in the worst case and is an NP-hard problem. For computational efficiency, we consider relaxing the problem (Herm-MLE) to some continuous convex domain and then projecting the relaxed solutions back to the indicator vectors. In this study, we introduce the following two relaxations: spectral relaxation and SDP relaxation, which lead to two new directed graph clustering algorithms, the spectral clustering algorithm (Algorithm 1) and SDP clustering algorithm (Algorithm 2).

The spectral relaxation. The general idea here is to relax the integer constraints in (Herm-MLE) to the continuous complex domain, and we arrive at the following new optimization problem

$$\begin{aligned} \max \quad & \mathbf{x}^* \widetilde{H} \mathbf{x} \\ s.t. \quad & \mathbf{x} \in \mathbb{C}^N \\ & \|\mathbf{x}\|_2^2 = N \end{aligned} \tag{SC-MLE}$$

Here, the continuous optimization problem (SC-MLE) is analytically solvable, and its maximum is attained when $\mathbf{x}$ is a rescaled (by $\sqrt{N}$) version of the leading eigenvector of $\widetilde{H}$. Note that the top eigenvector is not unique in the sense that its multiplication with $e^{i\theta}$ is also a top eigenvector for any $\theta \in [0, 2\pi]$. For this reason, we use the k-means algorithm to project the relaxed solution back to a complex indicator vector, which produces rotation-invariant results. We formally present the directed spectral clustering algorithm in Algorithm 1. To provide further intuition on this spectral algorithm, we visualize in Figure 1b the vertex embedding given by the top eigenvector, which is the key step of Algorithm 1.

The SDP relaxation. Another commonly used relaxation, which relies on semidefinite programming, considers lifting the vector variables $\mathbf{x} \in \mathbb{C}^N$ to a matrix. Because the objective in (Herm-MLE) has $\mathbf{x}^* \widetilde{H} \mathbf{x} = \mathrm{Tr}(\widetilde{H} \mathbf{x}\mathbf{x}^*)$, through defining $X = \mathbf{x}\mathbf{x}^*$ we obtain $\mathbf{x}^* \widetilde{H} \mathbf{x} = \mathrm{Tr}(\widetilde{H} X)$. Correspondingly, the integer constraints $\mathbf{x} \in \{1, i\}^N$ amount to con-

Input : directed graph $G(\mathcal{V}, \mathcal{E})$, DSBM parameters $\{p, q, \eta\}$
Output: community labels $\hat{\sigma}$
1 Compute the Hermitian matrix $\widetilde{H}$ according to (3) ;
2 Compute the top eigenvector $\hat{\mathbf{v}}$ of $\widetilde{H}$;
3 Apply k-means on the matrix $[\Re(\hat{\mathbf{v}}), \Im(\hat{\mathbf{v}})]$ to partition $\mathcal{V}$ into 2 clusters;

Algorithm 1: MLE Spectral Clustering Algorithm

straints on X, and we further relax them to obtain the following Hermitian SDP

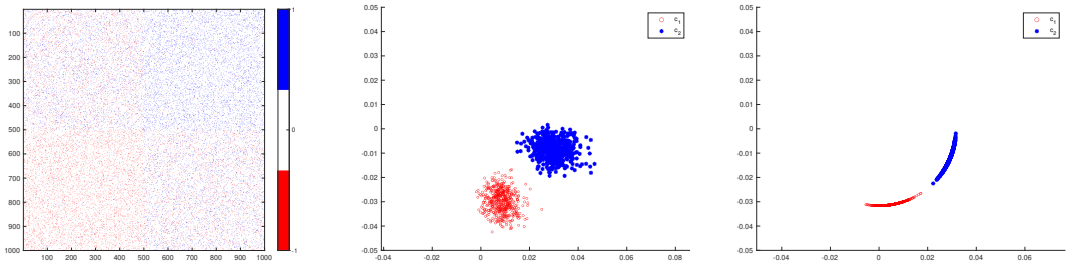
$$\begin{aligned} \max \quad & \mathrm{Tr}(\widetilde{H}X) \\ s.t. \quad & X \in \mathcal{H}^{N\times N} \\ & X \succeq 0 \\ & \mathrm{diag}(X) = I. \end{aligned} \tag{SDP-MLE}$$

Although more intricate than the spectral relaxation, this SDP can still be solved efficiently with standard optimization tools, which we discuss in more detail in Section 5. To project the SDP solution X back to an indicator vector, we first compute the leading eigenvector of X which provides the best rank-1 approximation of it. Then we apply the k-means algorithm on the embedding space given by this eigenvector, and obtain the two-cluster partition of the graph. We summarize the implementation steps in Algorithm 2. The embedding space given by the leading eigenvector of X is visualized in Figure 1c.

Input : directed graph $G(\mathcal{V}, \mathcal{E})$, DSBM parameters $\{p, q, \eta\}$
Output: community labels $\hat{\sigma}$
1 Compute the Hermitian matrix according to (3);
2 Solve (SDP-MLE) and compute the top eigenvector $\hat{\mathbf{v}}$ of the SDP solution;
3 Apply k-means on the matrix $[\Re(\hat{\mathbf{v}}), \Im(\hat{\mathbf{v}})]$ to partition $\mathcal{V}$ into 2 clusters.

Algorithm 2: MLE SDP Clustering Algorithm

The difference between the algorithms stems from their choice of relaxation. Compared with the spectral relaxation, the SDP relaxation approach preserves the constraint $\mathrm{diag}(X) = I$; therefore, we would expect that the SDP algorithm would yield clustering results closer to the solutions from the original combinatorial optimization problem (Herm-MLE) compared to the spectral clustering. This intuition is validated in experiments by comparing the two algorithms on the synthetic dataset sampled from DSBMs (see Figure 3, Figure 4 in Section 5.2 and Figure 8 in Appendix D.1), where we observe that Algorithm 2 consistently performs slightly better than Algorithm 1 over all tested datasets. In addition, this SDP optimization framework can accommodate various linear constraints, thereby offering flexibility to incorporate additional side information such as cluster sizes. Despite the aforementioned advantages of the SDP approach, we highlight that the spectral clustering



(a) Directed graph $A - A^T$ (b) Spectral embedding (c) SDP embedding

Figure 1: We sample a directed graph A from the DSBM with $n_1 = 500, n_2 = 500, p = q = 5\%, \eta = 5\%$. To highlight the direction of each edge, we visualize $A - A^T$ in (a), where a blue point represents a directed edge from its row index to the column index and a red pixel means the direction is from the column index to the row index. Given this directed graph, we compute $\widetilde{H}$ using (3) and plot in (b) the top eigenvector of $\widetilde{H}$ that gives the embedding of the vertex that separates the two clusters. The SDP embedding in (c) is obtained by first solving (SDP-MLE) and then plotting the top eigenvector of the SDP solution. Note that compared with spectral embedding, vertices in (c) reside on the unit circle, which is mainly due to the extra linear constraint $\mathrm{diag}(X) = I$ in the SDP relaxation.

algorithm is computationally more efficient than the SDP clustering algorithm. Therefore, the spectral algorithm is particularly more suitable for fast community discovery and clustering large-scale networks.

3.4 Error bound of Algorithm 1

In this section, we present an upper bound on the number of misclustered vertices using the spectral clustering algorithm. We use σ to denote the ground truth community indicating vectors, and $\hat{\sigma}$ denote the clustering result of Algorithm 1. Recall that we define $l(\sigma, \hat{\sigma})$ as the number of misclustered vertices, where $l(\sigma, \hat{\sigma}) = \sum_j \mathbb{1}(\sigma_j \neq \hat{\sigma}_j)$. While some pathological input graph instances may result in relatively large values of $l(\sigma, \hat{\sigma})$, we do not focus on providing guarantees for these worst-case scenarios. Instead, our focus lies on the majority of instances sampled from the DSBM, and we prove that with high probability, graphs sampled from the DSBM have bounded misclustering error, under Algorithm 1.

Theorem 2 (Error bound on Algorithm 1) *For a graph generated from the DSBM (n_1, n_2, p, q, η), let $\hat{\sigma}$ be the $(1+\epsilon)$-approximate solution of k-means from Algorithm 1. Then there exists $C = \Theta\left(\sqrt{w_r^2 + w_i^2}\right)$ (see (13)) and an absolute constant ϵ_0, such that with probability at least $1 - N^{-\epsilon_0}$, the error rate is such that*

$$\frac{l(\sigma, \hat{\sigma})}{N} \leq \frac{64(2+\epsilon)C^2 p_{\max} \log N}{d^2 \Delta^2}. \tag{7}$$

Here Δ lower bounds the eigengap $|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|$ and its expression is given in (11)*; d is the distance between the two cluster centroids of the population version $\mathbb{E}[\widetilde{H}]$, with its expression provided in* (28)*.*

The error bound that we obtain in Theorem 2 is inversely proportional to d^2 and Δ^2. Both d and Δ are determined by the population matrix $\mathbb{E}[\widetilde{H}]$. The entries in $\mathbb{E}[\widetilde{H}]$ are community-dependent and their values are determined by the model parameters p, q, and η, so as the centroid distance d and Δ. To establish a more explicit connection between the error bound (7) and the DSBM parameters, we explore a particular case that provides analytical forms on the error bound, and present in Corollary 3 interpretable error bounds on DSBM $(N/2, N/2, p, p, \eta)$. The proof of Corollary 3 can be found in Appendix C.6.

Corollary 3 *Consider directed graphs generated from the DSBM $(N/2, N/2, p, p, \eta)$. As $N \to \infty$, if $\eta \leq 0.5 - \epsilon$ with an absolute constant $\epsilon > 0$, then the misclustering error of Algorithm 1 is such that*

$$\frac{l(\sigma, \hat{\sigma})}{N} = O\left(\frac{\log N}{Np}\right) \tag{8}$$

If $\eta = 0.5 - o(1)$, we have that

$$\frac{l(\sigma, \hat{\sigma})}{N} = \omega\left(\frac{\log N}{Np}\right). \tag{9}$$

From (8) and (9), we infer that the error bound is inversely proportional to the average degree Np. This aligns with our intuition that DSBMs with a larger average degree provide more observations on edge orientations, rendering the generated graph more informative, and consequently lead to a smaller clustering error. Furthermore, for the noise level η, when $\eta \to 0.5$, the cross-cluster edges become nearly disordered and thus boost up the clustering error. This intuition is reflected in the different orders of the upper bounds provided in (8) and (9).

4. Perturbation analysis on Algorithm 1

The key idea behind the spectral clustering algorithm (Algorithm 1) is that eigenvectors of the data matrices contain crucial information revealing the underlying community structure. Here, we rigorously articulate this using matrix perturbation analysis, and derive the error bound (7) as presented in Theorem 2. Our analysis builds on the following simple intuition: for directed graphs sampled from the DSBM, the expected value of our proposed Hermitian matrix $\mathbb{E}[\widetilde{H}]$, also known as the population version of the matrix $\widetilde{H}$, has a clear block-wise structure, and its top eigenvector $\mathbf{v} = \mathbf{v_1}(\mathbb{E}[\widetilde{H}])$ has exactly two distinct values

that perfectly indicate the true community labels. In practice, however, this well-behaved matrix $\mathbb{E}[\widetilde{H}]$ is unobservable , and we consider the observable Hermitian matrix $\widetilde{H}$ as a corrupted version of $\mathbb{E}[H]$. We know from classical matrix perturbation analysis that, if $\widetilde{H}$ deviates not far away from its population version $\mathbb{E}[\widetilde{H}]$, one would expect that the top eigenvector of $\hat{\mathbf{v}} = \mathbf{v_1}(\widetilde{H})$ might continue to be informative, leading to a relatively good recovery of the true community labels. Our proof of the error bounds follows a standard procedure, with the following main steps

(i) Using matrix perturbation theory, we characterize the eigenvector perturbation $\|\mathbf{v}\mathbf{v}^* - \hat{\mathbf{v}}\hat{\mathbf{v}}^*\|_F$ in Section 4.1;

(ii) Using the Matrix-Bernstein inequality from random matrix theory, we provide a high-probability upper bound on the random perturbation $\|\widetilde{H} - \mathbb{E}[\widetilde{H}]\|$ in Section 4.2;

(iii) Combining the results from the above two steps, we perform an error analysis on the k-means clustering step, and present the final spectral clustering error bound in Section 4.3.

4.1 Bounding perturbation on the top eigenvectors

We first characterize properties on the eigenspace of the population matrix and show that $\mathbf{v} = \mathbf{v_1}(\mathbb{E}[\widetilde{H}])$ perfectly recovers the community labels. Then we upper bound how far $\hat{\mathbf{v}}$ deviates from $\mathbf{v}$ using the Davis-Kahan Theorem on eigenspace perturbation.

Eigenspace of the population version – the ideal case.

For a directed graph generated from DSBM with two communities, entries of the matrix $\mathbb{E}[\widetilde{H}]$ have community-dependent values. To express this in a compact way, we use a cluster membership matrix $M \in \{0,1\}^{N\times 2}$ to represent the ground truth community labelling, where $M_{u1} = 1$ indicates that vertex u belongs to the community $\mathcal{C}_1$, and $M_{u2} = 1$ denotes $u \in \mathcal{C}_2$. We consider the two communities to have no shared vertices, and thus the two column vectors of M are orthogonal. Using the membership matrix, we can write out $\mathbb{E}[\widetilde{H}]$ as follows

$$\begin{aligned}\mathbb{E}(\widetilde{H}) &= MQM^T - (pw_r + w_c)I,\\ Q &\triangleq iw_i(1-2\eta)q\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix} + w_r\begin{bmatrix}p & q\\ q & p\end{bmatrix} + w_c\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}.\end{aligned}$$

By defining the 2×2 matrix Q, we have $(MQM^T)_{u,v} = Q_{c(u),c(v)}$ where $c(\cdot) : \mathcal{V} \to \{1,2\}$ is a function that maps a vertex to its true community. The matrix MQM^T is of rank two, and its top eigenvector has exactly two distinct values, indicating the community labels. Since $\widetilde{H}$ and MQM^T share the same top eigenvector, we can easily calculate the eigenpairs of Q and then combine them with M to obtain the eigenvectors of $\mathbb{E}[\widetilde{H}]$. We summarize the

eigenspace properties of $\mathbb{E}[\widetilde{H}]$ in the following Lemma 4, and leave the detailed computation to Appendix C.1

Lemma 4 *For the DSBM(n_1, n_2, p, q, η), the population version of the proposed Hermitian matrix $\mathbb{E}[\widetilde{H}]$ has a unique largest eigenvalue. The top eigenvector $\mathbf{v}$ has exactly two distinct values that indicate the community labels where the distance between them d can be easily computed using* (28)*. Moreover, the eigengap*

$$|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])| = \min\{2\Delta, |1/2N(w_r p + w_c)| + \Delta\} \geq \Delta. \tag{10}$$

where

$$\Delta = \frac{1}{2}\sqrt{N^2(w_r p + w_c)^2 - 4n_1 n_2\left((w_r p + w_c)^2 - |w_r + w_c + i w_i(1-2\eta)q|^2\right)} \tag{11}$$

Perturbation analysis on the top eigenvectors.

For the DSBM(n_1, n_2, p, q, η), the population matrix $\mathbb{E}[\widetilde{H}]$ is unobservable, but from the generated graphs we are able to construct the matrix $\widetilde{H}$. We consider $\widetilde{H}$ as a perturbed version of $\mathbb{E}[\widetilde{H}]$ and use $R = \widetilde{H} - \mathbb{E}[\widetilde{H}]$ to denote the random perturbation. From matrix perturbation analysis, the perturbation on the eigenspace and eigenvalues are characterized by two well-known theorems: Davis-Kahan's theorem (Theorem 9) and Wyle's inequality (Theorem 10). Applying the above two theorems directly to bound the eigenspace distance $\|\mathbf{v}\mathbf{v}^* - \hat{\mathbf{v}}\hat{\mathbf{v}}^*\|_F$ requires a sufficient gap between eigenvalues of $\mathbb{E}[\widetilde{H}]$ and $\widetilde{H}$, which may impose extra conditions for this bound to be valid. We circumvent this by adapting a variant of eigenspace perturbation bound from Vu et al. (2013) into our proof, which allows us to derive an upper bound of $\|\mathbf{v}\mathbf{v}^* - \hat{\mathbf{v}}\hat{\mathbf{v}}^*\|_F$ that involves only comparing eigenvalues of $\mathbb{E}[\widetilde{H}]$. We summarize our eigenspace perturbation result in Lemma 5 and defer the proof details to appendix C.3.

Lemma 5 *Given a directed graph from DSBM(n_1, n_2, p, q, η) and its Hermitian matrix representation $\widetilde{H}$, the projection matrix of the top eigenvector has*

$$\|\mathbf{v}\mathbf{v}^* - \hat{\mathbf{v}}\hat{\mathbf{v}}^*\|_F \leq 2\sqrt{2}\frac{\|R\|}{\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])}.$$

4.2 Bounding the random perturbation R

The goal of this section is to establish a high-probability upper bound on $\|R\|$, where R is the random perturbation with the following form

$$\begin{aligned} R &= \widetilde{H} - \mathbb{E}[\widetilde{H}] \\ &= i w_i(A - A^T) - (1-2\eta)qM\begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix} M^T + w_r(A + A^T) - w_r M \begin{bmatrix} p & q \\ q & p \end{bmatrix} M - w_r p I. \end{aligned}$$

From the expression above, we observe that each entry of R has a bounded absolute value as well as a bounded variance. Therefore, using the Matrix Bernstein inequality on Hermitian matrices (Lemma 12), we can directly obtain a high-probability upper bound on $\|R\|$, which we present in Lemma 6. The proof details of Lemma 6 can be found in Appendix C.4.

Lemma 6 (Bound on random perturbation R) *Consider a directed graph from DSBM (n_1, n_2, p, q, η) and its Hermitian matrix representation $\widetilde{H}$. We use $p_{\max} = \max\{p, q\}$ to denote the maximum edge probability. Assume that the maximum edge probability is above the connectivity threshold, i.e.,*

$$Np_{\max} = \Omega(\log N). \tag{12}$$

Then there exist an absolute constant ϵ and

$$C = (2+\epsilon)\sqrt{w_r^2 + w_i^2}\left(\frac{\log N}{Np_{\max}} + 1\right) = \Theta\left(\sqrt{w_r^2 + w_i^2}\right), \tag{13}$$

such that the random perturbation $R = \widetilde{H} - \mathbb{E}[\widetilde{H}]$ has

$$\mathbb{P}(\|R\| \geq C\sqrt{Np_{\max}\log N}) \leq N^{-\epsilon}.$$

4.3 Proof of Theorem 2

Theorem 2 (Error bound on Algorithm 1) *For a graph generated from the DSBM (n_1, n_2, p, q, η), let $\hat{\sigma}$ be the $(1+\epsilon)$-approximate solution of k-means from Algorithm 1. Then there exists $C = \Theta\left(\sqrt{w_r^2 + w_i^2}\right)$ (see (13)) and an absolute constant ϵ_0, such that with probability at least $1 - N^{-\epsilon_0}$, the error rate is such that*

$$\frac{l(\sigma, \hat{\sigma})}{N} \leq \frac{64(2+\epsilon)C^2 p_{\max}\log N}{d^2\Delta^2}. \tag{7}$$

Here Δ lower bounds the eigengap $|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|$ and its expression is given in (11); d is the distance between the two cluster centroids of the population version $\mathbb{E}[\widetilde{H}]$, with its expression provided in (28).

Proof Recall that the key steps of our spectral clustering algorithm (Algorithm 1) involves: first compute the top eigenvector of $\widetilde{H}$, and then cluster the vertices using k-means on the embedding space given by the real and imaginary part of the top eigenvector. We use $\hat{U} = [\Re(\hat{\mathbf{v}}), \Im(\hat{\mathbf{v}})]$ to denote the embedding space given by the top eigenvector of $\widetilde{H}$ and $U = [\Re(\mathbf{v}), \Im(\mathbf{v})]$ for that of $\mathbb{E}[\widetilde{H}]$, where both $U, \hat{U} \in \mathbb{R}^{N\times 2}$. For the clustering outcomes, we denote by $\hat{\sigma}$ the clustering result using $\widetilde{H}$, and use σ to represent the clustering given by $\mathbb{E}[\widetilde{H}]$. First, note that from Lemma 4, we have the leading eigenvector of $\mathbb{E}[\widetilde{H}]$ perfectly indicates the true community membership, therefore σ is the true community membership

vector. Next, given that the k-means clustering step achieves a $(1+\epsilon)$ approximation, using the error bound (39) from Lemma 13, we have that

$$l(\sigma,\hat{\sigma})d^2 \leq 4(4+2\epsilon)\|\hat{U}-U\|_F^2,$$

where d is the distance between the two cluster centroids of the population version $\mathbb{E}[\widetilde{H}]$, with its expression provided in (28). Given that a rotation of $\hat{U}$ does not change the k-means clustering result, the tightest upper bound we can obtain is

$$l(\sigma,\hat{\sigma})d^2 \leq 4(4+2\epsilon)\min_{O\in\mathcal{O}_2}\|\hat{U}-OU\|_F^2 = 4(4+2\epsilon)\min_{r\in\mathbb{C}_1}\|\mathbf{v}-r\hat{\mathbf{v}}\|_F^2 \leq 4(4+2\epsilon)\|\hat{\mathbf{v}}\hat{\mathbf{v}}^*-\mathbf{v}\mathbf{v}^*\|_F^2, \tag{14}$$

where the first equality follows from the fact that $\|U-\hat{U}\|_F = \|\mathbf{v}-\hat{\mathbf{v}}\|_F$ and the last inequality follows from Lemma 11.

Recall that combining Lemma 4, Lemma 5 and Lemma 6, we can derive that for a absolute constant ϵ_0, with probability at least $1-N^{-\epsilon_0}$

$$\|\hat{\mathbf{v}}\hat{\mathbf{v}}^*-\mathbf{v}\mathbf{v}^*\|_F \leq \frac{2\sqrt{2}C\sqrt{Np_{\max}\log N}}{\lambda_1(\mathbb{E}[\widetilde{H}])-\lambda_2(\mathbb{E}[\widetilde{H}])} \leq \frac{2\sqrt{2}C\sqrt{Np_{\max}\log N}}{\Delta}. \tag{15}$$

Combining (14) and (15), we eventually obtain that with probability at least $1-N^{-\epsilon_0}$

$$\frac{l(\sigma,\hat{\sigma})}{N} \leq \frac{64(2+\epsilon)C^2p_{\max}\log N}{d^2\Delta^2}.$$

■

5. Algorithmic implementation and experiments

5.1 Implementation details and complexity analysis

This section outlines the implementation details of our proposed algorithms. Our MATLAB code is available on Github[1].

Implementation details of the Spectral Clustering Algorithm. Algorithm 1 requires computing the top eigenvector of the Hermitian matrix, with a compute time of $O(N^2)$ via the power method. For the k-means step, we employ the k-means++ algorithm Arthur et al. (2007), which produces solutions $O(\log 2)$ competitive to the optimal k-means solution.

1 https://github.com/ningz97/MLE-DSBM

Implementation details of the SDP Algorithm. One way of solving the (SDP-MLE) in polynomial time is to use standard interior point methods based tools, such as SDPT3 Toh et al. (2012) and Mosek. However, those solvers are quite memory intensive for large graphs (in practice, around thousands of vertices). To avoid memory outage, in this paper, we used the Burer-Monteiro approach Burer and Monteiro (2003), which is a rank-restricted non-convex programming algorithm that allows a much smaller search space. The Burer-Monteiro method considers the following optimization problem

$$\begin{aligned} \max \quad & \mathrm{Tr}(Z^* \widetilde{H} Z) \\ s.t. \quad & Z \in \mathbb{C}^{N \times r} \\ & \mathrm{diag}(ZZ^*) = I_N. \end{aligned} \tag{MLE-BM}$$

This non-convex optimization problem (MLE-BM) is guaranteed to map to the global optimal of the SDP when the rank satisfies $r^2 > N$ Boumal et al. (2016). For solving (MLE-BM), one can use either the Augmented Lagrangian method Boyd et al. (2011) or manifold optimization tools Boumal (2023). With all these in mind, we summarize the steps of the clustering algorithm in Algorithm 3.

Input : directed graph $G(\mathcal{V}, \mathcal{E})$ of size N, DSBM parameters $\{p, q, \eta\}$
Output: community labels $\hat{\sigma}$
1 Compute the Hermitian matrix according to (3);
2 Solve (MLE-BM) with $r = \lceil \sqrt{N} \rceil$ and compute the top eigenvector of the solution $\hat{\mathbf{v}}$;
3 Apply k-means on the concatenated matrix $[\Re(\hat{\mathbf{v}}), \Im(\hat{\mathbf{v}})]$;

Algorithm 3: Burer-Monteiro for MLE clustering

Learning Model Parameters. Note that our proposed algorithms, Algorithm 1, Algorithm 2 and Algorithm 3, all require knowing the DSBM parameters as inputs so that one can compute the MLE Hermitian matrix using (3). Such a requirement is often practically infeasible because it is hard to compute or approximate p, q and η without knowing the true community label. To circumvent this limitation, we adopt an iterative approach from Newman (2016), and combine it with our proposed clustering algorithms to learn the model parameters. We summarize this iterative clustering approach in Algorithm 4.

We conduct empirical tests on this iterative approach on directed graphs generated from the DSBM ensemble. In Figure 2, we show how the learned model parameters vary as one repeats the updating process in Algorithm 4. Through our study on the synthetic data sets from the DSBM, we observe that in most cases, this iterative algorithm converges near the truth model parameters very fast (within 10 iterations). We also conduct experiments to test how different initialization strategies may influence the clustering outcomes. We present detailed comparisons of different initialization strategies in Appendix D.4. From

Input : directed graph $G(\mathcal{V}, \mathcal{E})$, threshold t

Output: estimated DSBM parameters $\{p, q, \eta\}$, community label $\hat{\sigma}$

1 Randomly initialize p, q, and η;

2 Apply MLE-driven algorithms clustering (Algorithm 1, Algorithm 2 or Algorithm 3) and get two clusters $\mathcal{C}_1, \mathcal{C}_2$;

3 Update the DSBM parameters as follows:

$p := \frac{|\mathcal{E}| - \mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)}{\binom{|\mathcal{C}_1|}{2} + \binom{|\mathcal{C}_2|}{2}}$, which is the within-community edge frequency;

$q := \frac{\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)}{|\mathcal{C}_1| \times |\mathcal{C}_2|}$, which is the between-community edge frequency;

$\eta := \min\left\{ \frac{|(\mathcal{C}_1 \times \mathcal{C}_2) \cap \mathcal{E}|}{\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)}, \frac{|(\mathcal{C}_2 \times \mathcal{C}_1) \cap \mathcal{E}|}{\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)} \right\}$, which computes the ratio of oriented cross-community edges;

4 Repeat step 2 and step 3 until the update is below the convergence threshold t;

5 Finalize the clusters using MLE-driven algorithms with converged DSBM parameters.

Algorithm 4: Iterative MLE clustering

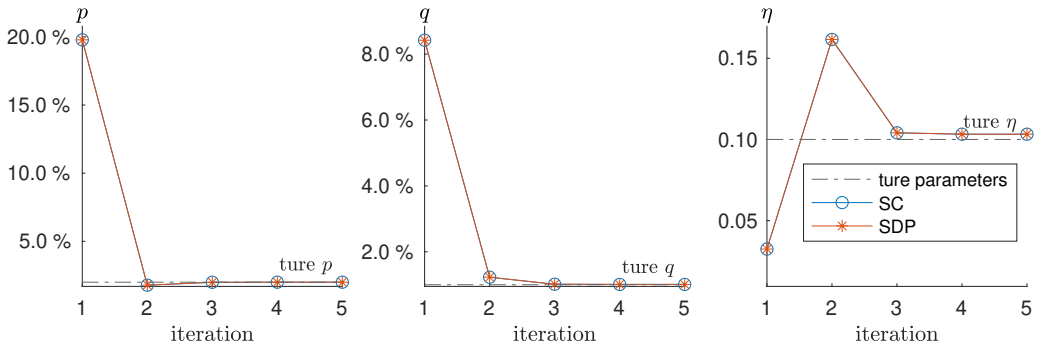


Figure 2: Illustration on the convergence of iterative learning algorithm. We first sample a directed graph from DSBM with $n_1 = n_2 = 1000, p = 2\%, q = 1\%, \eta = 0.1$. Then starting from a random guess on the model parameters, we apply Algorithm 4 to learn the DSBM parameters. The lines with circles represent model parameter learning using the spectral clustering algorithm (Algorithm 1) and those with stars represent learning with the SDP clustering algorithm (Algorithm 3).

our empirical studies, we summarize the following three types of strategies to provide a relatively good initialization, and facilitate the iterative learning of the parameters:

a. When the edge orientation is the main consideration for clustering, we suggest initializing the Hermitian matrix as $\widetilde{H}_0 = i(A - A^T)$, which, according to our discussion in Section 3.2, corresponds to the net flow optimization.
b. When edge density difference is the main consideration, we suggest initializing the Hermitian matrix as $\widetilde{H}_0 = A + A^T$ as it corresponds to the total flow optimization scheme;
c. When both edge density and orientation need to be taken into account, we suggest initializing with $\widetilde{H}_0 = i(A - A^T) + (A + A^T)$, which jointly considers the two factors without bias.

5.2 Experiments on DSBM synthetic data

In this section, we conduct experiments on directed graphs sampled from the DSBM with different model parameters p, q, η. To measure the performance of each algorithm, we calculate the Adjusted Rand Index (ARI) Gates and Ahn (2017), which quantifies how well the clustering output aligns with the ground truth community labelling. The ARI takes value in $[-1, 1]$, where an ARI value of 1 indicates a perfect recovery on the ground truth; a nearly 0 ARI implies that the recovery is almost like a random guess; and -1 indicates that the recovered clusters are completely different from the ground truth.

In the experiments, we cluster the directed graphs using our proposed spectral clustering algorithm (MLE-SC) and the SDP clustering algorithm (MLE-SDP), where we assume no prior knowledge about the model parameters and we employ the iterative learning approach to learn them from data. We compare our proposed algorithms with other existing approaches for clustering directed graphs: the DI-SIM algorithm in Rohe et al. (2012), the Hermitian clustering algorithms from Cucuringu et al. (2020) (Herm and HermRW), the bibliometric symmetrization from Satuluri and Parthasarathy (2011) (B-Sym), and spectral clustering on naïve symmetrization using $A + A^T$.

Experiments on DSBM with $p = q$. We first conduct experiments on DSBM graphs with homogeneous edge probability. With $p = q$, the generated directed graphs have roughly the same between-cluster and within-cluster edge densities, therefore community recovery can only rely on the information attached to the edge directions. In each of the experiments, we independently sample 10 directed graphs with a fixed parameter set, and we averaged the ARI values over these graph samples. We summarize the ARI comparisons in Figure 3. From the quantitative comparisons, we observed that our proposed MLE-SC (Algorithm 4 with Algorithm 1) and MLE-SDP (Algorithm 4 with Algorithm 2 or Algorithm 3) attain nearly the same performance, and they outperform all other algorithms when the noise value η is small. In the high-noise regime, i.e., η close to 0.5, DI-SIM is the best-performing algorithm. For a more intuitive illustration, we also visual-

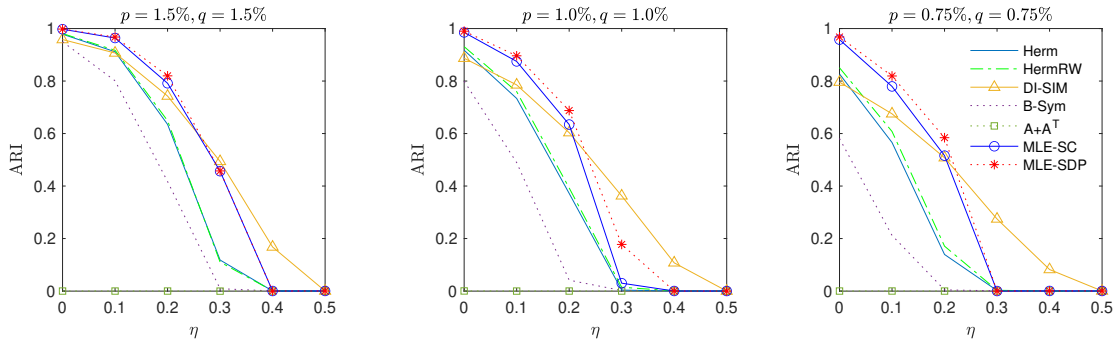


Figure 3: Performance comparison of all algorithms on DSBM with $n_1 = n_2 = 1000$, $p = q$ (for three different values of the edge density), and varying η.

ize in Appendix D.2 the graph adjacency relation before and after applying the clustering algorithms.

Experiments on DSBM with $p \neq q$. We consider experiments for which the edge densities are different, and test on graphs from DSBM with $p \neq q$. In Figure 4, we summarize the comparisons of the clustering results on directed graphs sampled with different values of p, q. As before, each reported ARI value is obtained by averaging over 10 independently sampled directed graphs with fixed DSBM parameters. We observed that overall our proposed algorithms MLE-SC and MLE-SDP have the highest ARIs with MLE-SDP slightly better than MLE-SC.

It is also worth mentioning that our MLE-based Hermitian clustering algorithms empirically outperform the other two closely related Hermitian clustering algorithms, namely Herm and HermRW from Cucuringu et al. (2020). The main difference between the latter two algorithms and our approaches is that our proposed Hermitian matrix contains both real and imaginary components with a derived weighting parameter. Recall that from the optimization interpretation in Section 3.2, we derived that $\Re(\widetilde{H})$ corresponds to the $\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)$ optimization, and $\Im(\widetilde{H})$ corresponds to the $\mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2)$ optimization. Therefore, this comparison between our proposed algorithms and those in Cucuringu et al. (2020) suggests the importance of having a joint analysis of the edge density and edge orientation when clustering directed graphs.

In addition to the above tests on DSBM, we also conduct experiments following the same setup but using true DSBM parameters p, q, η as inputs for Algorithm 1 and Algorithm 3 instead of employing the iterative approach to learn the parameters. We report test results with true DSBM parameters in Figure 8 in Appendix D.1. Comparing the performance between those with and without true model parameters (Figure 8 *v.s.* Figure 3 and 4), we observe that given true model parameters as inputs, Algorithm 1 and Algorithm 3 have nearly the same performance as MLE-SC and MLE-SDP. This comparison indicates that

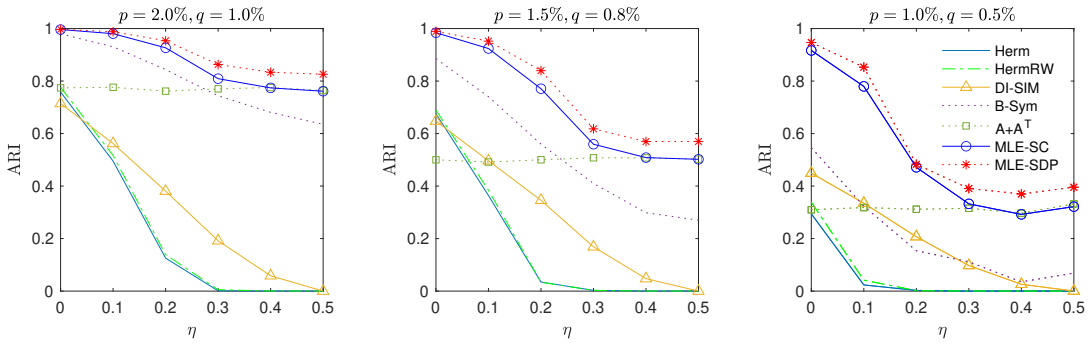


Figure 4: Performance comparison of all algorithms on DSBM with $n_1 = n_2 = 1000$, $p \neq q$ (for three different combinations of p, q), and varying η.

the iterative approach consistently converges to the true DSBM parameters on our tested dataset.

5.3 Experiments on real-world data sets

We first perform experiments on the real-world data set email-Eu-core Leskovec et al. (2007), which is a network containing email exchange data from European research institutions. In this network, there is an edge (u, v) if a person u sent person v at least one email. This dataset comes with "ground-truth" community membership labelling, where each community represent one of the 42 departments at the research institute. We perform experiments on two-community subgraphs of the email-Eu-core dataset by selecting pairs from the three largest communities in the data set [2].

We compare the cluster results with the "ground-truth" community assignment and report in Table 1 the ARIs of each algorithm, where each ARI is averaged from 10 repeating experiments. On the tested data set, we observe that our algorithms MLE-SC and MLE-SDP outperform all other baselines. In particular, we observe that MLE-SC is the best among all spectral clustering algorithms, and the MLE-SDP outperforms all other algorithms by a large margin.

Data set	Herm	HermRW	DI-SIM	B-Sym	$A + A^T$	MLE-SC	MLE-SDP
email-Eu-core12	0.048	0.003	-0.005	0.359	0.423	**0.631**	**0.957**
email-Eu-core23	-0.009	0.0059	-0.005	0.383	0.490	**0.578**	**0.978**

Table 1: ARIs from tests on email-Eu-core.

2 email-Eu-core12: pick the 1$^{\text{st}}$ and 2$^{\text{nd}}$ largest communities from email-Eu-core dataset
email-Eu-core23: pick the 2$^{\text{nd}}$ and 3$^{\text{rd}}$ largest communities from email-Eu-core dataset

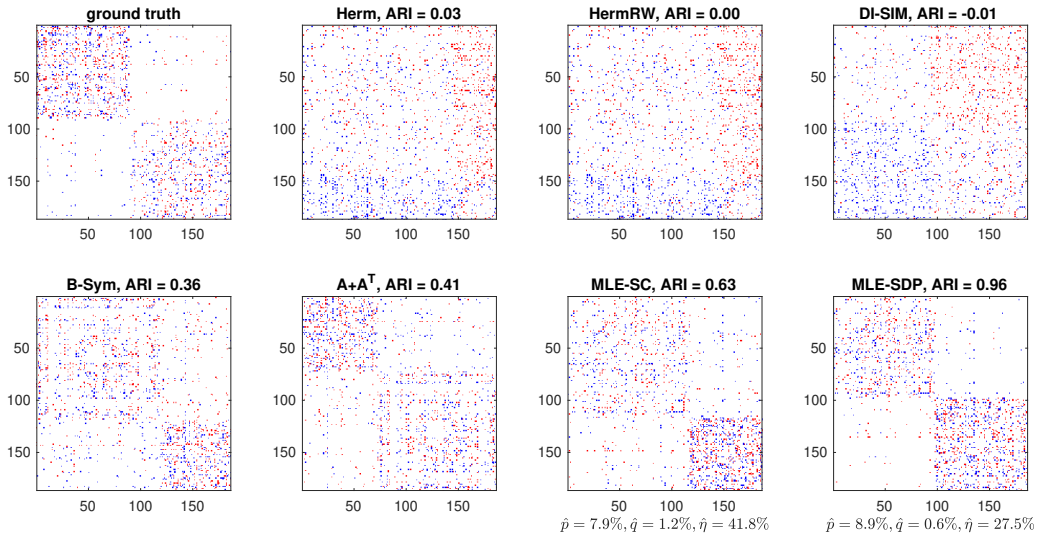


Figure 5: $A - A^T$ after clustering. In this experiment, we test on a directed graph consisting of the 1st and 2nd largest communities from the email-En-core dataset Leskovec et al. (2007). Below the figures of MLE-SC and MLE-SDP, we report the learned DSBM parameters from the iterative algorithm.

We present in Figure 5 and Figure 6 visual representations of the graph adjacency relation after clustering. From the visualization, we observe distinctive clustering patterns exhibited by the compared algorithms: Herm, HermRW and DI-SIM tend to cluster vertices in a way that the cross-cluster edges are oriented in the same direction while not accounting much for the edge density. In contrast, both the naïve symmetrization and B-Sym demonstrate an awareness of inhomogeneity between-cluster and within-cluster edge density. Our proposed approaches strike a balance between the cross-cluster edge orientation and the inhomogeneity edge densities, where the weights are learned from the iterative approach.

We also report and visualize in Appendix D.3 the case being studied where our proposed algorithms attain low ARI values. We comment that in those experiments, our proposed algorithms still lead to meaningful community discovery where the clusters exhibit blockwise patterns from the edge density or edge orientation, but the clusters recovered by our algorithms do not agree with the community labeling provided.

6. Concluding remarks and discussions

This paper studies the directed graph clustering problem through the lens of statistics. In particular, we formulate the task of directed clustering as a statistical estimation prob-

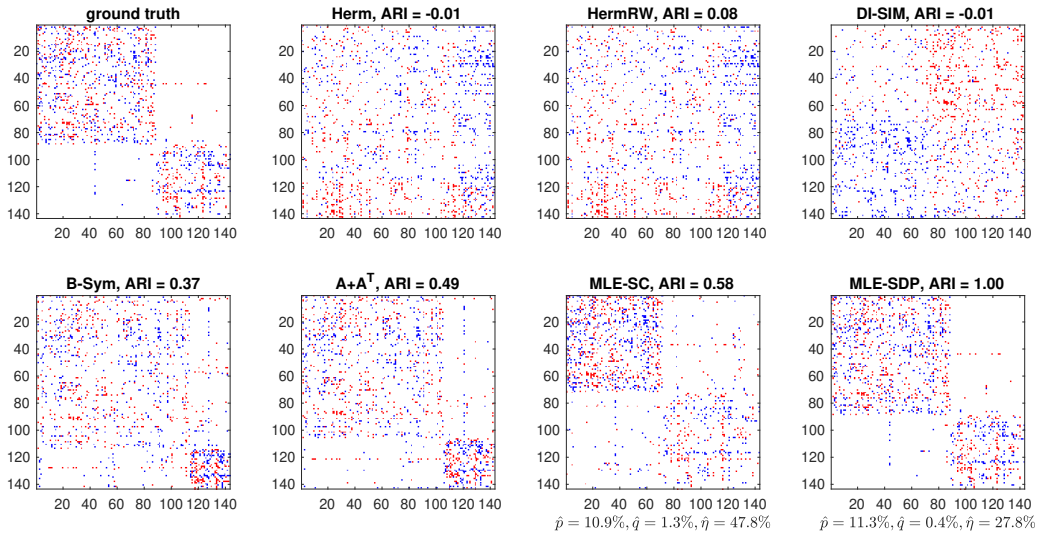


Figure 6: $A - A^T$ after clustering. In this experiment, we test a directed graph consisting of the 2^{nd} and 3^{rd} largest communities from email-En-core Leskovec et al. (2007). Below the figures of MLE-SC and MLE-SDP, we report the learned DSBM parameters from the iterative algorithm.

lem on the DSBM, and employ the MLE to infer the underlying community labels. This statistical formulation gives rise to a novel flow optimization heuristic, which jointly considers the edge density and edge orientation. Building on our formulation, we propose a new Hermitian matrix representation of directed graphs, establishing a connection between its spectrum and the MLE optimization problem. Based on this Hermitian matrix representation, we introduce two efficient and interpretable directed clustering algorithms, a spectral clustering algorithm (Algorithm 1) and an SDP clustering algorithm (Algorithm 2, along with its more scalable version, Algorithm 3). We compare, both quantitatively and qualitatively, our proposed algorithms with existing directed clustering methods on synthetic data and real-world data. In addition to our experimental evaluation, we conduct a perturbation analysis using Davis-Kahan's theorem and establish an upper bound on the misclustering error of Algorithm 1.

Our current work on directed graph clustering does not consider any prior knowledge about the community assignment. Therefore, it would be natural to extend this work by systematically integrating side information or prior knowledge. One way to achieve this is to consider the community labels as latent random variables and the side information can be modelled as the prior distribution of community labels. Correspondingly, community recovery can then be achieved by applying Bayesian methods, such as maximum-a-posteriori, and Gibbs sampling. In addition to the statistical view, we also present an equivalent (regularized)

flow optimization heuristic for directed clustering. From this optimization perspective, one can directly incorporate prior knowledge about community labels by imposing constraints on the flow optimization problem. Alternatively, the optimization interpretation also allows using a constrained clustering method from Cucuringu et al. (2016), where one can merge the data matrix with any available constraints by converting the constraints into a penalty term of the optimization objective.

This work focuses on studying the two-cluster directed stochastic block model. Extending the analysis to multi-cluster models is not as easy as the inductive analysis on undirected graphs. The challenge stems from the asymmetric nature of the problem. To elaborate on this point, if one additional cluster is added to the current two-cluster model, there would be multiple approaches to instantiate the problem: e.g., add a new source cluster to the existing source cluster, add a new sink cluster to the existing source cluster, etc. Therefore, for a multi-cluster model, one needs to specify a directed *meta-graph* structure to determine the relationship between different clusters, which may vary from case to case. Consequently, the task of inferring the number of clusters, as well as clustering multi-cluster directed graphs, may require a more elaborated analysis and potentially further assumptions. The main difficulty of the problem stems from the fact that this underlying meta-graph structure is not known to the user, a-priori. While we defer such work for future research, we comment that one may iteratively apply the two-cluster algorithms on the clustered subgraphs to obtain a hierarchical clustering structure.

Our study focuses on clustering unweighted simple directed graphs, and it would be practically meaningful and interesting to consider clustering more general and complex directed graphs. For instance, our flow optimization heuristic can readily be adapted for clustering weighted directed graphs. In addition, recent work Tian and Lambiotte (2023) adopts a Hermitian matrix representation for signed graphs. It would be interesting to explore the connections between signed graphs and directed graphs, which could possibly lead to a more general Hermitian matrix design that can unify these two types of graphs, as in the recent work of He et al. (2022) that introduced the *Magnetic Signed Laplacian*, a Hermitian PSD matrix. Moreover, from a statistical perspective, the DSBM we studied in this paper has its limitations in describing real-world networks, and thus it would be interesting to extend our analysis to random directed graph models that better capture some of the real-world network features, such as power-law distribution Michel et al. (2019) or heterogeneity of node degrees Rohe et al. (2016). Finally, extending our proposed methodology to the setting of time-evolving networks Matias and Miele (2017) is another timely avenue of future research, potentially operating under the assumption that the node cluster labels vary smoothly over time.

References

Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on information theory*, 62(1):471–487, 2015.

Daron Acemoglu, Asuman Ozdaglar, and Alireza Tahbaz-Salehi. Systemic risk and stability in financial networks. *American Economic Review*, 105(2):564–608, 2015.

Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

Arash A Amini and Elizaveta Levina. On semidefinite relaxations for the block model. 2018.

Arash A Amini, Aiyou Chen, Peter J Bickel, and Elizaveta Levina. Pseudo-likelihood methods for community detection in large sparse networks. 2013.

Yuan An, Jeannette Janssen, and Evangelos E Milios. Characterizing and mining the citation graph of the computer science literature. *Knowledge and Information Systems*, 6:664–678, 2004.

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035, 2007.

Konstantin Avrachenkov, Maximilien Dreveton, and Lasse Leskelä. Community recovery in non-binary and temporal stochastic block models. *arXiv preprint arXiv:2008.04790*, 2020.

Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. Lead–lag detection and network clustering for multivariate time series with an application to the us equity market. *Machine Learning*, 111(12):4497–4538, 2022.

Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. *Advances in Neural Information Processing Systems*, 29, 2016.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical programming*, 95(2): 329–357, 2003.

Yudong Chen, Sujay Sanghavi, and Huan Xu. Improved graph clustering. *IEEE Transactions on Information Theory*, 60(10):6440–6455, 2014.

Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, et al. Spectral methods for data science: A statistical perspective. *Foundations and Trends® in Machine Learning*, 14(5):566–806, 2021.

Mihai Cucuringu, Ioannis Koutis, Sanjay Chawla, Gary Miller, and Richard Peng. Simple and scalable constrained clustering: a generalized spectral method. In *Artificial Intelligence and Statistics*, pages 445–454. PMLR, 2016.

Mihai Cucuringu, Huan Li, He Sun, and Luca Zanetti. Hermitian matrices for clustering directed graphs: insights and applications. In *International Conference on Artificial Intelligence and Statistics*, pages 983–992. PMLR, 2020. URL http://128.84.4.34/pdf/1908.02096.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical review E*, 84(6):066106, 2011.

Michaël Fanuel, Carlos M Alaiz, and Johan AK Suykens. Magnetic eigenmaps for community detection in directed networks. *Physical Review E*, 95(2):022302, 2017.

Michaël Fanuel, Carlos M Alaíz, Ángela Fernández, and Johan AK Suykens. Magnetic eigenmaps for the visualization of directed networks. *Applied and Computational Harmonic Analysis*, 44(1):189–199, 2018.

Alan Frieze and Michał Karoński. *Introduction to random graphs*. Cambridge University Press, 2016.

Alexander J Gates and Yong-Yeol Ahn. The impact of random models on clustering similarity. *arXiv preprint arXiv:1701.06508*, 2017.

Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming: Extensions. *IEEE Transactions on Information Theory*, 62 (10):5918–5937, 2016.

Koby Hayashi, Sinan G Aksoy, and Haesun Park. Skew-symmetric adjacency matrices for clustering directed graphs. *arXiv preprint arXiv:2203.01388*, 2022. URL https://arxiv.org/pdf/2203.01388.pdf.

Yixuan He, Michael Perlmutter, Gesine Reinert, and Mihai Cucuringu. MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian. In Bastian Rieck and Razvan Pascanu, editors, *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pages 40:1–40:39. PMLR, 09–12 Dec 2022. URL https://proceedings.mlr.press/v198/he22c.html.

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

Wolfram Research, Inc. Mathematica, Version 14.0. URL https://www.wolfram.com/mathematica. Champaign, IL, 2024.

Antony Joseph and Bin Yu. Impact of regularization on spectral clustering. 2016.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

Maxwell Mirton Kessler. Bibliographic coupling between scientific papers. *American documentation*, 14(1):10–25, 1963.

Steinar Laenen. Directed graph clustering using hermitian laplacians. *Master's thesis*, 2019.

Steinar Laenen and He Sun. Higher-order spectral clustering of directed graphs. *Advances in neural information processing systems*, 33:941–951, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0a5052334511e344f15ae0bfafd47a67-Abstra

Can M Le, Elizaveta Levina, and Roman Vershynin. Concentration and regularization of random graphs. *Random Structures & Algorithms*, 51(3):538–561, 2017.

Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. 2015.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1 (1):2–es, 2007.

Xiaodong Li, Yudong Chen, and Jiaming Xu. Convex relaxation methods for community detection. 2021.

Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics reports*, 533(4):95–142, 2013.

Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society Series B*, 79(4):1119–1141, 2017. URL https://EconPapers.repec.org/RePEc:bla:jorssb:v:79:y:2017:i:4:p:1119-1141.

Jesse Michel, Sushruth Reddy, Rikhav Shah, Sandeep Silwal, and Ramis Movassagh. Directed random geometric graphs. *Journal of Complex Networks*, 7(5):792–816, 2019.

Andrea Montanari and Subhabrata Sen. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 814–827, 2016.

Mark EJ Newman. Community detection and graph partitioning. *Europhysics Letters*, 103 (2):28003, 2013.

Mark EJ Newman. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5):052315, 2016.

Marcia Oliveira and Joao Gama. An overview of social network analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(2):99–115, 2012.

Judea Pearl and Thomas Verma. The logic of representing dependencies by directed graphs. In *Proceedings of the sixth National conference on Artificial intelligence-Volume 1*, pages 374–379, 1987.

Karl Rohe, Tai Qin, and Bin Yu. Co-clustering for directed graphs: the stochastic co-blockmodel and spectral algorithm di-sim. *arXiv preprint arXiv:1204.2296*, 2012. URL https://arxiv.org/pdf/1204.2296.pdf.

Karl Rohe, Tai Qin, and Bin Yu. Co-clustering directed graphs to discover asymmetries and directional communities. *Proceedings of the National Academy of Sciences*, 113(45):12679–12684, 2016. URL https://www.pnas.org/doi/full/10.1073/pnas.1525793113.

Venu Satuluri and Srinivasan Parthasarathy. Symmetrizations for clustering directed graphs. In *Proceedings of the 14th international conference on extending database technology*, pages 343–354, 2011. URL https://dl.acm.org/doi/pdf/10.1145/1951365.1951407?casa_token=IOameE8wvjAAAAAA:s7sr2Ue_mPhbf

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

MA Shubin. Discrete magnetic laplacian. *Communications in mathematical physics*, 164 (2):259–275, 1994.

Henry Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for information Science*, 24 (4):265–269, 1973.

Yu Tian and Renaud Lambiotte. Structural balance and random walks on complex networks with complex weights. *arXiv preprint arXiv:2307.01813*, 2023.

Kim-Chuan Toh, Michael J Todd, and Reha H Tütüncü. On the implementation and usage of sdpt3–a matlab software package for semidefinite-quadratic-linear programming, version 4.0. *Handbook on semidefinite, conic and polynomial optimization*, pages 715–754, 2012.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

Vincent Q Vu, Juhee Cho, Jing Lei, and Karl Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. *Advances in neural information processing systems*, 26, 2013.

Hermann Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

Yi Yu, Tengyao Wang, and Richard J Samworth. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

Appendix A. Summary of notations

Notation	Definition
$G(\mathcal{V}, \mathcal{E})$	Graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$
$u \rightsquigarrow v$	There is an edge pointing from vertex u to vertex v
$u \not\rightsquigarrow v$	There is no edge between vertex u and vertex v
A	Graph adjacency matrix, $A \in \{0,1\}^{n\times n}$ and $A_{uv} = 1$ iff $u \rightsquigarrow v$
$\mathcal{C}_1$	Source community
$\mathcal{C}_2$	Target community
$\mathcal{V}$	Set of all vertices, $\mathcal{V} = \mathcal{C}_1 \cup \mathcal{C}_2$
$c(\cdot)$	Community labelling function, $c(\cdot) : \mathcal{V} \to \{1,2\}$ where $c(u) = 1$ iff $u \in \mathcal{C}_1$
$\widetilde{H}$	Hermitian matrix derived from MLE on DSBM (see (3))
σ	A general community indicator vector, $\sigma_u = \sigma_v$ iff $c(u) = c(v)$
$\mathcal{L}(A;\sigma)$	Log-likelihood function
$\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)$	Total flow between $\mathcal{C}_1$ and $\mathcal{C}_2$, $\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2) = \sum_{u\in\mathcal{C}_1 v\in\mathcal{C}_2} (A_{uv} + A_{vu})$
$\mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2)$	Net flow from $\mathcal{C}_1$ to $\mathcal{C}_2$, $\mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2) = \sum_{u\in\mathcal{C}_1 v\in\mathcal{C}_2} (A_{uv} - A_{vu})$
p	Within-community edge probability
q	Cross-community edge probability
$p_{\max}$	$\max\{p, q\}$
η	Probability of a cross-community edge pointing from $\mathcal{C}_2$ to $\mathcal{C}_1$
H^T	Transpose of H
H^*	Conjugate transpose of H
H_{j*}	The j-th row vector of H
$[H_1, H_2]$	Concatenating columns of H_1 and H_2
$\|H\|$	Spectral norm of H, $\|H\| = \|\lambda_1(H)\|$
$\|H\|_F$	Frobenius norm of H, $\|H\|_F = \sqrt{\sum_j \lambda_j^2(H)}$
$\langle H_1, H_2\rangle$	Frobenius inner product, $\langle H_1, H_2\rangle = \mathrm{Tr}(H_1^* H_2)$
$\mathrm{diag}(H)$	Create a diagonal matrix by taking the main diagonal elements of H
$\Re(H)$	Take the real part of the matrix H;
$\Im(H)$	Take the imaginary part of the matrix H
$\mathbf{v_j}(H)$	The j-th eigenvector of H
ARI	Adjusted Rand Index
$M \in \{0,1\}^{N\times 2}$	Membership matrix
$\mathbb{1}(\cdot)$	Indicator function, $\mathbb{1}(p) = 1$ if claim p is true, otherwise $\mathbb{1}(p) = 0$
$\mathbb{1}_{\mathcal{C}_1}$	Binary indicator vector for community $\mathcal{C}_1$, $\mathbb{1}_u = 1$ if $u \in \mathcal{C}_1$ otherwise $\mathbb{1}_u = 0$
$\mathbb{1}_{\mathcal{C}_2}$	Binary indicator vector for community $\mathcal{C}_2$, $\mathbb{1}_u = 1$ if $u \in \mathcal{C}_2$ otherwise $\mathbb{1}_u = 0$
$g_n = o(f_n)$	g_n is asymptotically dominated by f_n, i.e., $\lim_{n\to\infty} \frac{g_n}{f_n} = 0$

$g_n = O(f_n)$	g_n is asymptotically bounded above by f_n, i.e., $\limsup_{n\to\infty} \frac{g_n}{f_n} < \infty$
$g_n = \Theta(f_n)$	$\limsup_{n\to\infty} \frac{g_n}{f_n} < \infty$ and $\liminf_{n\to\infty} \frac{g_n}{f_n} > 0$
$g_n = \Omega(f_n)$	g_n bounded below by f_n asymptotically, i.e., $\limsup_{n\to\infty} \frac{g_n}{f_n} > 0$
$g_n = \omega(f_n)$	g_n dominate f_n asymptotically, i.e., $\limsup_{n\to\infty} \frac{g_n}{f_n} = \infty$

Table 2: Summary on notations

Appendix B. Proof of Theorem 1

We detail the derivation of the optimization problem (MLE) from the maximum likelihood estimator. To start with, we explicitly express the likelihood function as a matrix, which simply relies on subdividing the likelihood function according to which community an edge belongs to.

Lemma 7 *Consider a directed graph with adjacency matrix A sampled from the model DSBM(n_1, n_2, p, q, η). Then, applying the maximum likelihood estimation is equivalent to solving the following combinatorial optimization problem*

$$\begin{aligned} \max \quad & \frac{1}{2}\left\langle M_{intra}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle + \left\langle M_{inter}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle \\ s.t. \quad & \mathbb{1}_{\mathcal{C}_1} \in \{0,1\}^N \\ & \mathbb{1}_{\mathcal{C}_1} + \mathbb{1}_{\mathcal{C}_2} = \mathbb{1} \end{aligned} \tag{MLE}$$

where $\mathbb{1}_{\mathcal{C}_1}, \mathbb{1}_{\mathcal{C}_2} \in \{0,1\}^N$ are the indicator vectors for cluster $\mathcal{C}_1$ and $\mathcal{C}_2$ separately, and

$$\begin{aligned} M_{intra} &= \log(1/2p)(A + A^T) + \log(1-p)(J - I - A - A^T), \\ M_{inter} &= \log(q(1-\eta))A + \log(\eta q)A^T + \log(1-q)(J - I - A - A^T), \end{aligned}$$

are derived from the log-likelihood functions for intra-community and inter-community edges.

Proof Let A be the adjacency matrix of a directed graph generated from DSBM(n_1, n_2, p, q, η). For a particular clusterization of the graph, we use c to denote its community labeling function $c : \mathcal{V} \to \{\mathcal{C}_1, \mathcal{C}_2\}$, and we use the vectors $\mathbb{1}_{\mathcal{C}_1}, \mathbb{1}_{\mathcal{C}_2} \in \{0,1\}^N$ to indicate community $\mathcal{C}_1$ and $\mathcal{C}_2$ separately, where $\mathbb{1}_{\mathcal{C}_1} + \mathbb{1}_{\mathcal{C}_2} = \mathbb{1}$. The log likelihood function of A given $\mathbb{1}_{\mathcal{C}_1}$ and $\mathbb{1}_{\mathcal{C}_2}$ can be decomposed as follows

$$\begin{aligned} \mathcal{L}(A;\sigma) = \log \mathbb{P}(A|\mathbb{1}_{\mathcal{C}_1}, \mathbb{1}_{\mathcal{C}_2}) &= \sum_{u<v} \log \mathbb{P}(A_{uv}|c(u), c(v)) \\ &= \sum_{\substack{u<v \\ c(u)=c(v)}} \log \mathbb{P}(A_{uv}|c(u), c(v)) + \sum_{\substack{u<v \\ c(u)=1, c(v)=2}} \log \mathbb{P}(A_{uv}|c(u), c(v)), \end{aligned} \tag{16}$$

where the first term in (16) is only summing over intra-community pair, and the second term handles the inter-community pair.

For an intra-community vertex pair u, v, the log-likelihood function is

$$\log \mathbb{P}(A_{uv}|c(u) = c(v)) = \begin{cases} \log(1/2p) & \text{if } u \rightsquigarrow v, \\ \log(1/2p) & \text{if } v \rightsquigarrow u, \\ \log(1-p) & \text{if } u \not\sim v. \end{cases}$$

This intra-community log-likelihood function coincides with the matrix

$$M_{intra} \triangleq \log(1/2p)(A + A^T) + \log(1-p)(J - I - A - A^T), \tag{17}$$

on the entries that represent intra-community pairs, thus allowing us to convert the intra-community summation in (16) into the following matrix multiplication form

$$\sum_{\substack{u<v \\ c(u)=c(v)}} \log \mathbb{P}(A_{uv}|c(u), c(v)) = \frac{1}{2}\left\langle M_{intra}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle. \tag{18}$$

For an inter-community vertex $u \in \mathcal{C}_1, v \in \mathcal{C}_2$, the log-likelihood function is

$$\log \mathbb{P}(A_{uv}|c(u) = \mathcal{C}_1, c(v) = \mathcal{C}_2) = \begin{cases} \log((1-\eta)q) & \text{if } u \rightsquigarrow v, \\ \log(\eta q) & \text{if } v \rightsquigarrow u, \\ \log(1-q) & \text{if } u \not\sim v. \end{cases}$$

Similar to the approach followed for the intra-community case, we convert the inter-community summation in (16) into the matrix multiplication form

$$\sum_{\substack{u<v \\ c(u)=1, c(v)=2}} \log \mathbb{P}(A_{uv}|c(u), c(v)) = \left\langle M_{inter}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle. \tag{19}$$

where

$$M_{inter} \triangleq \log((1-\eta)q)A + \log(\eta q)A^T + \log(1-q)(J - I - A - A^T). \tag{20}$$

Combining (18), (19) and (16), we have

$$\log \mathbb{P}(A|\mathbb{1}_{\mathcal{C}_1}, \mathbb{1}_{\mathcal{C}_2}) = \frac{1}{2}\left\langle M_{intra}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle + \left\langle M_{inter}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle. \tag{21}$$

■

To arrive at a more compact expression for the optimization formulation, we introduce an equivalent Hermitian matrix optimization framework. The transformation from the real-valued matrix optimization to the Hermitian optimization builds on the following observation.

Lemma 8 *Consider an arbitrary Hermitian matrix $H = \Re(H) + i\Im(H)$, where $\Re(H) \in \mathbb{R}^{n\times n}$ with all 0 diagonal entries is symmetric, and $\Im(H) \in \mathbb{R}^{n\times n}$ is skew-symmetric. Let $\mathbf{x} \in \{i, 1\}^n$ be the complex community indicator vector, where $\mathbf{x}_u = i$ for $u \in \mathcal{C}_1$. Then,*

the quadratic form $\mathbf{x}^* H \mathbf{x}$ is the sum of entries in $\Re(H)$ that are in the same community, plus the sum of entries in $\Im(H)$ that belong to different communities, i.e.,

$$\mathbf{x}^* H \mathbf{x} = \sum_{\substack{u,v\in\mathcal{C}_1 \\ or\ u,v\in\mathcal{C}_2}} \Re(H)_{uv} + \sum_{\substack{u\in\mathcal{C}_1 v\in\mathcal{C}_2 \\ or\ u\in\mathcal{C}_2, v\in\mathcal{C}_1}} \Im(H)_{uv} = 2\sum_{\substack{u<v \\ u,v\in\mathcal{C}_1 \\ or\ u,v\in\mathcal{C}_2}} \Re(H)_{uv} + 2\sum_{\substack{u<v, \\ u\in\mathcal{C}_1 v\in\mathcal{C}_2 \\ or\ u\in\mathcal{C}_2, v\in\mathcal{C}_1}} \Im(H)_{uv}.$$

In other words,

$$\mathbf{x}^* H \mathbf{x} = \langle \Re(H), \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T \rangle + 2\langle \Im(H), \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T \rangle. \tag{22}$$

With the real-valued MLE optimization formula derived in Lemma 7 and the observation made in Lemma 8, we are ready to prove the Hermitian optimization formulation of the MLE on DSBM in Theorem 1.

Theorem 1 (MLE on DSBM) *Consider a directed graph with adjacency matrix A generated from the model DSBM(n_1, n_2, p, q, η). Let $\mathbf{x} \in \{i, 1\}^N$ be the indicator vector, where $\mathbf{x}_u = i$ if $u \in \mathcal{C}_1$, and $\mathbf{x}_u = 1$ if $u \in \mathcal{C}_2$. The maximum likelihood estimation on the community labels is equivalent to solving the following complex optimization problem*

$$\begin{aligned} \max \quad & \mathbf{x}^* \widetilde{H} \mathbf{x} \\ s.t. \quad & \mathbf{x} \in \{1, i\}^N, \end{aligned} \tag{Herm-MLE}$$

where $\widetilde{H}$ is a Hermitian matrix defined as

$$\begin{aligned} \widetilde{H} &= i \log \frac{1-\eta}{\eta}(A - A^T) + \log \frac{p^2(1-p)^2}{4\eta(1-\eta)q^2(1-q)^2}(A + A^T) + 2\log\frac{1-p}{1-q}(J - I) \\ &\triangleq w_i i(A - A^T) + w_r(A + A^T) + w_c(J - I). \end{aligned} \tag{3}$$

Proof From Lemma 7, we derive the log-likelihood

$$\log \mathbb{P}(A|\mathbb{1}_{\mathcal{C}_1}, \mathbb{1}_{\mathcal{C}_2}) = \frac{1}{2}\left\langle M_{intra}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle + \left\langle M_{inter}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle. \tag{23}$$

Compared this equation with the observation (22) in Lemma 8, we need the matrix corresponding to the intra-cluster log-likelihood to be symmetric and need the inter-cluster log-likelihood matrix to be skew-symmetric, so that we can directly apply (22) to convert the real-valued objective into a more compact complex-valued expression.

From the definition in (17), we have that $M_{intra} = M_{intra}^T$. For the inter-cluster log-likelihood matrix, by definition (20), M_{inter} is not slew-symmetric. To circumvent this, we decomposed the inter-cluster log-likelihood matrix M_{inter} into a symmetric matrix plus a skew-symmetric matrix as follows

$$M_{inter} = \frac{1}{2}(M_{inter} + M_{inter}^T) + \frac{1}{2}(M_{inter} - M_{inter}^T),$$

where $M_{inter}+M_{inter}^T$ is symmetric and $M_{inter}-M_{inter}^T$ is skew-symmetric. Correspondingly, we have

$$\left\langle M_{inter}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T \right\rangle = \frac{1}{2}\langle M_{inter} + M_{inter}^T, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\rangle + \frac{1}{2}\langle M_{inter} - M_{inter}^T, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\rangle, \tag{24}$$

where the first term sums over the inter-cluster entries of a symmetric matrix and the second term sums over the inter-cluster entries of a skew-symmetric matrix. Due to the symmetry in the first term of (24), we can further write it as

$$\frac{1}{2}\langle M_{inter} + M_{inter}^T, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\rangle = \frac{1}{4}\langle M_{inter} + M_{inter}^T, J - \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T - \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T\rangle \tag{25}$$

Combining (24), (25) and (23), we can rewrite the log-likelihood objective as

$$\begin{aligned}
&\log \mathbb{P}(A|\mathbb{1}_{\mathcal{C}_1}, \mathbb{1}_{\mathcal{C}_2}) = \frac{1}{2}\left\langle M_{intra}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T\right\rangle + \left\langle M_{inter}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\right\rangle \\
&= \frac{1}{2}\left\langle M_{intra}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T\right\rangle + \frac{1}{2}\langle M_{inter} + M_{inter}^T, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\rangle + \frac{1}{2}\langle M_{inter} - M_{inter}^T, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\rangle \\
&= \frac{1}{2}\left\langle M_{intra}, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T\right\rangle + \frac{1}{4}\langle M_{inter} + M_{inter}^T, J - \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T - \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T\rangle \\
&\quad + \frac{1}{2}\langle M_{inter} - M_{inter}^T, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\rangle
\end{aligned}$$

Because the term $\langle M_{inter} + M_{inter}^T, J\rangle$ is always a constant and resealing the objective function by a constant factor 4 does not affect the optimal solution, therefore solving the (MLE) in Lemma 7 is equivalent to solve the following

$$\begin{aligned}
\max \quad & \left\langle 2M_{intra} - (M_{inter} + M_{inter}^T), \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_1}^T + \mathbb{1}_{\mathcal{C}_2}\mathbb{1}_{\mathcal{C}_2}^T\right\rangle + 2\left\langle M_{inter} - M_{inter}^T, \mathbb{1}_{\mathcal{C}_1}\mathbb{1}_{\mathcal{C}_2}^T\right\rangle \\
s.t. \quad & \mathbb{1}_{\mathcal{C}_1} \in \{0,1\}^N \\
& \mathbb{1}_{\mathcal{C}_1} + \mathbb{1}_{\mathcal{C}_2} = \mathbb{1}
\end{aligned}$$

Using (22) from Lemma 8, we convert the above real-valued optimization problem to the following complex-valued equivalence

$$\begin{aligned}
\max \quad & \mathbf{x}^* \widetilde{H} \mathbf{x} \\
s.t. \quad & \mathbf{x} \in \{i, 1\}^N
\end{aligned}$$

where the Hermitian matrix $\widetilde{H}$ has

$$\begin{aligned}
\Re(\widetilde{H}) &= 2M_{intra} - (M_{inter} + M_{inter}^T) \\
&= \log \frac{p^2}{4\eta(1-\eta)q^2}(A + A^T) + 2\log \frac{1-p}{1-q}(J - I - A - A^T) \\
&= \log \left(\frac{p^2(1-q)^2}{4\eta(1-\eta)q^2(1-q)^2} \right) (A + A^T) + 2\log \left(\frac{1-p}{1-q} \right) (J - I), \\
\Im(\widetilde{H}) &= M_{inter} - M_{inter}^T \\
&= \log \left(\frac{1-\eta}{\eta} \right) (A - A^T).
\end{aligned}$$

■

Appendix C. Proofs in perturbation analysis

C.1 Proof of Lemma 4

Lemma 4 *For the DSBM(n_1, n_2, p, q, η), the population version of the proposed Hermitian matrix $\mathbb{E}[\widetilde{H}]$ has a unique largest eigenvalue. The top eigenvector $\mathbf{v}$ has exactly two distinct values that indicate the community labels where the distance between them d can be easily computed using* (28)*. Moreover, the eigengap*

$$|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])| = \min\{2\Delta, |1/2N(w_r p + w_c)| + \Delta\} \geq \Delta. \tag{10}$$

where

$$\Delta = \frac{1}{2}\sqrt{N^2(w_r p + w_c)^2 - 4n_1 n_2\left((w_r p + w_c)^2 - |w_r + w_c + i w_i(1-2\eta)q|^2\right)} \tag{11}$$

Proof Recall that the population version of $\widetilde{H}$ has a block structure and can be written as

$$\begin{aligned}\mathbb{E}[\widetilde{H}] &= MQM^T - (pw_r + w_c)I \\ Q &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + w_r \begin{bmatrix} p & q \\ q & p \end{bmatrix} + w_c \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},\end{aligned}$$

where $M \in \{0,1\}^{N\times 2}$ is the community membership matrix, and $M_{uc} = 1$ denotes that vertex u belongs to community c. We further normalize the columns of M as follows

$$\begin{aligned}MQM^T &= MD^{-1}DQD(MD^{-1})^T \\ D &= \begin{bmatrix} \sqrt{n_1} & 0 \\ 0 & \sqrt{n_2}. \end{bmatrix}\end{aligned}$$

Here the normalized matrix MD^{-1} has orthonormal column vectors.

Let $DQD = U\Lambda U^*$ be the eigendecomposition on the 2×2 matrix, Then, the $N \times N$ matrix MQM^T can be diagonalized as

$$MQM^T = (MD^{-1}U)\Lambda(MD^{-1}U)^*,$$

where $\mathrm{diag}(\Lambda)$ contains the eigenvalues of MQM^T and the columns of $MD^{-1}U \in \mathbb{R}^{N\times 2}$ are the orthonormal eigenvectors. Therefore, the problem of computing the eigenpairs of $\mathbb{E}[\widetilde{H}]$ reduces to compute the eigenpairs of the 2×2 matrix DQD where

$$DQD = \begin{bmatrix} n_1(w_r p + w_c) & \sqrt{n_1 n_2}(w_r q + w_c + i(1-2\eta)q) \\ \sqrt{n_1 n_2}(w_r q + w_c - i(1-2\eta)q) & n_2(w_r p + w_c) \end{bmatrix}$$

For the eigenvalues, via a simple calculation we arrive at

$$\lambda_1(DQD) = \frac{1}{2}\left(N(w_r p + w_c) + 2\Delta\right),$$
$$\lambda_2(DQD) = \frac{1}{2}\left(N(w_r p + w_c) - 2\Delta\right),$$

where

$$2\Delta = \sqrt{N^2(w_r p + w_c)^2 - 4n_1 n_2((w_r p + w_c)^2 - |w_r q + w_c + i w_i q(1-2\eta)|^2)}.$$

Therefore, we obtain the eigenvalues of $\mathbb{E}[H] = MQM^T$

$$\lambda_1(\mathbb{E}[\widetilde{H}]) = \frac{1}{2}\left(N(w_r p + w_c) + 2\Delta\right) - (w_r p + w_c)$$
$$\lambda_2(\mathbb{E}[\widetilde{H}]) = \frac{1}{2}\left(N(w_r p + w_c) - 2\Delta\right) - (w_r p + w_c)$$
$$\lambda_3(\mathbb{E}[\widetilde{H}]) = \ldots = \lambda_N(\mathbb{E}[\widetilde{H}]) = -(w_r p + w_c).$$

The eigenvalue that obtains the largest magnitude is unique and it is $\lambda_1(\mathbb{E}[\widetilde{H}])$ when $N(w_r p + w_c) \geq 0$, or $\lambda_2(\mathbb{E}[\widetilde{H}])$ when $N(w_r p + w_c) < 0$. The gap between the largest and the second largest eigenvalue is

$$\min\{2\Delta, |1/2N(w_r p + w_c)| + \Delta\}. \tag{26}$$

One can easily verify that the eigengap lies in $[\Delta, 2\Delta]$. Therefore, the lower bound Δ is a good approximation to the spectral gap in the sense that they are of the same order.

Next, we move on to compute the top eigenvector of $\mathbb{E}[\widetilde{H}]$. We use $\mathbf{x} = (x_1, x_2) \in \mathbb{C}^2$ to denote the top eigenvector of DQD, and we have that $x_1 \neq x_2$. Then, the top eigenvector of $\mathbb{E}[\widetilde{H}]$ can be easily computed through $\mathbf{v} = MD^{-1}\mathbf{x}$, and it has two distinct values

$$\mathbf{v}(u) = \begin{cases} x_1/\sqrt{n_1} & \text{if } u \in \mathcal{C}_1, \\ x_2/\sqrt{n_2} & \text{if } u \in \mathcal{C}_2. \end{cases} \tag{27}$$

The distance between the two cluster centroids d is simply

$$d = \left| \frac{x_1}{\sqrt{n_1}} - \frac{x_2}{\sqrt{n_2}} \right|. \tag{28}$$

■

C.2 Useful theorems from matrix perturbation analysis

Theorem 9 (Davis-Kahan's perturbation bound Davis and Kahan (1970)) *Let $H, R \in \mathcal{H}$ be two Hermitian matrices. Then, for any $a \leq \beta$ and $\delta > 0$ it holds that*

$$\left\|P_{[\alpha,\beta]}(H) - P_{(\alpha-\delta,\beta+\delta)}(H+R)\right\| \leq \frac{\|R\|}{\delta}.$$

Here $P_{[\alpha,\beta]}(H)$ denotes the projection matrix on the subspace spanned by eigenvectors of H with corresponding eigenvalues lie between $[\alpha, \beta]$, and $P_{(\alpha-\delta,\beta+\delta)}(H+R)$ is the projection matrix on the subspace spanned by eigenvectors of $H+R$ with eigenvalues lie between $(\alpha-\delta, \beta+\delta)$.

Theorem 10 (Weyl's inequality Weyl (1912)) *Let $H, R \in \mathcal{H}$ be two Hermitian matrices. Then for every $1 \leq j \leq n$, the j-th largest eigenvalues of H and $H+R$ obey*

$$|\lambda_j(H) - \lambda_j(H+R)| \leq \|R\|.$$

In addition to the eigenspace perturbation bound in Themrem 9, we summarize in Lemma 11 comparisons over two different representations of the eigenspace distance, which will be useful in the error analysis of k-means in Section 4.3.

Lemma 11 *[Adapted From Lemma 2.1 in Chen et al. (2021)] For any $U, \tilde{U} \in \mathbb{C}^{\mathbb{N}\times k}$, we have*

$$\min_{O\in\mathcal{O}_{k\times k}} \|U - O\tilde{U}\|_F \leq \|UU^* - \tilde{U}\tilde{U}^*\|_F$$

C.3 Proof of Lemma 5

Lemma 5 *Given a directed graph from DSBM(n_1, n_2, p, q, η) and its Hermitian matrix representation $\widetilde{H}$, the projection matrix of the top eigenvector has*

$$\|\mathbf{v}\mathbf{v}^* - \hat{\mathbf{v}}\hat{\mathbf{v}}^*\|_F \leq 2\sqrt{2}\frac{\|R\|}{\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])}.$$

Proof From the Davis-Kahan's perturbation bound, we have

$$\|\hat{\mathbf{v}}\hat{\mathbf{v}}^* - \mathbf{v}\mathbf{v}^*\| \leq \frac{\|R\|}{|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\widetilde{H})|}. \tag{29}$$

Using Wyle's inequality, we have that

$$|\lambda_2(\mathbb{E}[\widetilde{H}]) - \lambda_2(\widetilde{H})| \leq \|R\|$$

Therefore, the denominator in (29) can be further lower bounded by $|\lambda_1(\mathbb{E}[\widetilde{H}])-\lambda_2(\mathbb{E}[\widetilde{H}])|-\|R\|$, and we obtain

$$\|\hat{\mathbf{v}}\hat{\mathbf{v}}^* - \mathbf{v}\mathbf{v}^*\| \leq \frac{\|R\|}{|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])| - \|R\|}. \tag{30}$$

The denominator in (30) involves comparing the spectral gap $|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|$ and $\|R\|$, which further requires an extra condition on the denominator being positive, to allow the inequality to hold. To circumvent this limitation, we divide the comparison into two cases

- if $\|R\| \geq \frac{1}{2}|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|$, then we have

$$\|\hat{\mathbf{v}}\hat{\mathbf{v}}^* - \mathbf{v}\mathbf{v}^*\| \leq 1 \leq \frac{2\|R\|}{|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|}.$$

- if $\|R\| \leq \frac{1}{2}|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|$, then we use the perturbation bound (30) and get

$$\|\hat{\mathbf{v}}\hat{\mathbf{v}}^* - \mathbf{v}\mathbf{v}^*\| \leq \frac{\|R\|}{|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])| - \|R\|} \leq \frac{2\|R\|}{|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|}.$$

Combining the two cases, we obtain that for any $\|R\|$, the following upper bound always holds

$$\|\hat{\mathbf{v}}\hat{\mathbf{v}}^* - \mathbf{v}\mathbf{v}^*\| \leq \frac{2\|R\|}{|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|}.$$

Since for any rank r Hermitian matrix H, $\|H\|_F \leq \sqrt{r}\|H\|$. Therefore, we have that

$$\|\mathbf{v}\mathbf{v}^* - \hat{\mathbf{v}}\hat{\mathbf{v}}^*\|_F \leq \sqrt{2}\|\mathbf{v}\mathbf{v}^* - \hat{\mathbf{v}}\hat{\mathbf{v}}^*\| \leq \frac{2\sqrt{2}\|R\|}{|\lambda_1(\mathbb{E}[\widetilde{H}]) - \lambda_2(\mathbb{E}[\widetilde{H}])|}.$$

■

C.4 Proof of Lemma 6

Lemma 12 (Matrix Bernstein Tropp et al. (2015)) *Consider a finite sequence $\{S_k\}$ of independent, random matrices with dimension d. Assume that*

$$\mathbb{E}S_k = 0 \text{ and } \|S_k\| \leq L \text{ for each index } k.$$

For the random matrix $Z = \sum_k S_k$, let $v(Z)$ be the matrix variance statistic of the sum:

$$v(Z) = \max\left\{ \left\| \sum_k \mathbb{E}\left(S_k S_k^*\right) \right\|, \left\| \sum_k \mathbb{E}\left(S_k^* S_k\right) \right\| \right\}.$$

Then, for all $t \geq 0$,

$$\mathbb{P}\{\|Z\| \geq t\} \leq 2d \exp\left(\frac{-t^2/2}{v(Z) + Lt/3}\right).$$

Lemma 6 (Bound on random perturbation R) *Consider a directed graph from DSBM (n_1, n_2, p, q, η) and its Hermitian matrix representation $\widetilde{H}$. We use $p_{\max} = \max\{p, q\}$ to denote the maximum edge probability. Assume that the maximum edge probability is above the connectivity threshold, i.e.,*

$$Np_{\max} = \Omega(\log N). \tag{12}$$

Then there exist an absolute constant ϵ and

$$C = (2+\epsilon)\sqrt{w_r^2 + w_i^2}\left(\frac{\log N}{Np_{\max}} + 1\right) = \Theta\left(\sqrt{w_r^2 + w_i^2}\right), \tag{13}$$

such that the random perturbation $R = \widetilde{H} - \mathbb{E}[\widetilde{H}]$ has

$$\mathbb{P}(\|R\| \geq C\sqrt{Np_{\max}\log N}) \leq N^{-\epsilon}.$$

Proof Recall that by definition random perturbation $R = \widetilde{H} - \mathbb{E}[\widetilde{H}]$ is Hermitian. We first decompose it into summation of perturbations on different entries $R = \sum_{j<l} R^{jl}$ where R^{jl} is also a random Hermitian and only has non-zero entries at (j,l) and (l,j). If j, l belongs to the same community $\sigma(j) = \sigma(l)$

$$R_{jl}^{jl} = \begin{cases} w_r(1-p) + iw_i & w.p.\ \ p/2 \\ w_r(1-p) - iw_i & w.p.\ \ p/2 \\ -w_r p & w.p.\ \ 1-p. \end{cases} \tag{31}$$

If j, l belongs to different communities $\sigma(j) \neq \sigma(l)$, and without loss of generality we assume $j \in \mathcal{C}_1, l \in \mathcal{C}_2$

$$R_{jl}^{jl} = \begin{cases} w_r(1-q) + iw_i(1-(1-2\eta)q) & w.p.\ \ q(1-\eta) \\ w_r(1-q) - iw_i(1+(1-2\eta)q) & w.p.\ \ q\eta \\ -w_r q - iw_i(1-2\eta)q & w.p.\ \ 1-q \end{cases} \tag{32}$$

From the Matrix Bernstein's inequality in Lemma 12, we have for any $t \geq 0$

$$\mathbb{P}(\|R\| \geq t) \leq 2N \exp\left(\frac{-t^2/2}{\mathrm{Var}(R) + Lt/3}\right),$$

where L is an upper upper of $\|R^{jl}\|$ and $\mathrm{Var}(R)$ is the variance.

For computing L, recall that by definition

$$\|R^{jl}\| \leq L, \ \ \forall j \neq l.$$

Here the matrix spectral norm can be simplified to be upper bounded by $|R_{jl}|$ because the spectral norm is always upper bounded by the maximum absolute values of each entry. Therefore, it suffices to take L as an upper bound on $\max_{j\neq l} |R_{jl}|$. From (31), we have that, if $\sigma(j) = \sigma(l)$, then $|R_{jl}| \leq \sqrt{w_r^2(1-p)^2 + w_i^2}$; if $\sigma(j) \neq \sigma(l)$, then $|R_{jl}| \leq \sqrt{w_r^2(1-q)^2 + w_i^2(1+(1-2\eta)q)^2}$. Combining the two, it suffices for us to take

$$L = \sqrt{w_r^2(1-p_{min})^2 + w_i^2(1+(1-2\eta)q)^2} \leq 2\sqrt{w_r^2 + w_i^2}. \tag{33}$$

To compute the variance term $\mathrm{Var}(R)$, first recall by definition

$$\mathrm{Var}(R) = \max\left\{ \left\| \sum_{j<l} \mathbb{E}[R^{jl}(R^{jl})^*] \right\|, \left\| \sum_{j<l} \mathbb{E}[(R^{jl})^* R^{jl}] \right\| \right\}.$$

For each $j < l$, we have $R^{jl}(R^{jl})^* = (R^{jl})^* R^{jl}$ and we use M^{jl} to denote the product matrix. In M^{jl}, the only two non zero entries are M^{jl}_{jj} and M^{jl}_{ll} and $M^{jl}_{jj} = M^{jl}_{ll} = R_{jl}\overline{R_{jl}}$. Therefore, $\mathbb{E}[R^{jl}(R^{jl})^*]$ also only has two non-zero entries at (j,j) and (l,l) for every $j < l$ and thus, the spectral norm of the matrix summation is simply the largest diagonal element, i.e.,

$$\mathrm{Var}(R) = \max_{j\in[N]} \sum_{l\neq j} \mathbb{E}[M^{jl}_{jj}]. \tag{34}$$

In (34), M^{jl}_{jj} is a real random variable whose distribution can be derived from (31) and (32), and we have that, if $\sigma(j) = \sigma(l)$, then

$$M^{jl}_{jj} = \begin{cases} w_r^2(1-p)^2 + w_i^2 & w.p. \ \ p \\ w_r^2 p^2 & w.p. \ \ 1-p, \end{cases}$$

and $\mathbb{E}[M^{jl}_{jj}] = w_r^2 p(1-p) + p w_i^2$.

If $\sigma(j) \neq \sigma(l)$, then

$$M_{jj}^{jl} = \begin{cases} w_r^2(1-q)^2 + w_i^2(1-(1-2\eta)q)^2 & w.p.\ \ q(1-\eta) \\ w_r^2(1-q)^2 + w_i^2(1+(1-2\eta)q)^2 & w.p.\ \ q\eta \\ w_r^2 q^2 + w_i^2(1-2\eta)^2 q^2 & w.p. 1-q, \end{cases}$$

and $\mathbb{E}[M_{jj}^{jl}] = w_r^2 q(1-q) + w_i^2 q(1-(1-2\eta)^2 q)$. Since, without loss of generality, we assume $p, q \leq 0.5$, thus for all $j \neq l$ we have $\mathbb{E}[M_{jj}^{jl}] \leq w_r^2 p_{\max}(1-p_{\max}) + w_i^2 p_{\max}$. Therefore, from (34), we arrive at

$$\mathrm{Var}(R) \leq N(w_r^2 p_{\max}(1-p_{\max}) + w_i^2 p_{\max}) \leq N p_{\max}(w_r^2 + w_i^2). \tag{35}$$

Using the Matrix Bernstein's inequality, we have for $t = C\sqrt{N p_{\max} \log N}$,

$$\begin{aligned} \mathbb{P}(\|R\| \geq t) &\leq 2\exp\left(-\frac{C^2 N p_{\max} \log N}{2\mathrm{Var}(R) + 2LC\sqrt{N p_{\max}\log N}/3} + \log N\right) \\ &\leq 2\exp\left(-\frac{C^2 N p_{\max} \log N}{2N p_{\max}(w_r^2 + w_i^2) + 2L\sqrt{N p_{\max}\log N}/3} + \log N\right) \qquad (36) \\ &= 2\exp\left(-\frac{C^2}{2(w_r^2 + w_i^2) + 2LC/3\sqrt{\log N / N p_{\max}}} \log N + \log N\right). \end{aligned}$$

Here (36) follows from the analysis on $\mathrm{Var}(R)$ in (35). From (37), if there exist an absolute ϵ, such that

$$\frac{C^2}{(w_r^2 + w_i^2) + LC/3\sqrt{\log N / N p_{\max}}} \geq 2 + \epsilon, \tag{37}$$

then we have $\mathbb{P}(\|R\| \geq t) \leq N^{-\epsilon}$, which conclude the proof. It turns out that we can always find an absolute constant C such that (37) holds. To see this, first note that (37) is equivalent to

$$C \geq (1+\epsilon/2)L\sqrt{\frac{\log N}{N p_{\max}}} + \sqrt{(2+\epsilon)(w_r^2 + w_i^2) + (1+\epsilon/2)^2 L^2 \frac{\log N}{N p_{\max}}}.$$

Since $a^2 + b^2 \leq (a+b)^2$ for $a, b > 0$, it suffices to let

$$C = (2+\epsilon)L\frac{\log N}{N p_{\max}} + (2+\epsilon)\sqrt{w_r^2 + w_i^2}.$$

Since $L \leq 2\sqrt{w_r^2 + w_i^2}$, we have

$$C \leq (2+\epsilon)\sqrt{w_r^2 + w_i^2}\left(\frac{\log N}{N p_{\max}} + 1\right) = \Theta\left(\sqrt{w_r^2 + w_i^2}\right), \tag{38}$$

where the last equality is due to the connectivity assumption (12). ■

C.5 Useful theorem in k-means error analysis

Lemma 13 (k-means error adapted from Lemma 5.3 in Lei and Rinaldo (2015)) *For $\epsilon > 0$ and any two matrices $\hat{U}, U$, such that $U = MX$ with $M \in \{0,1\}^{N\times 2}$ be the indicator matrix and $X \in \mathbb{R}^{2\times 2}$ have its row vectors representing the centroids of two clusters, let $(\hat{M}, \hat{X})$ be a $(1+\epsilon)$ solution to the k-means problem and $\bar{U} = \hat{M}\hat{X}$. For $\delta = \|X_{1*} - X_{2*}\|$, define $S = \{j \in [N] : \|\bar{U}_{j*} - U_{j*}\|\} \geq \delta/2$ then*

$$|S|\delta^2 \leq 4(4+2\epsilon)\|\hat{U} - U\|_F^2. \tag{39}$$

C.6 Proof of Corollary 3

Corollary 14 *Consider directed graphs generated from the DSBM $(N/2, N/2, p, p, \eta)$. As $N \to \infty$, if $\eta \leq 0.5 - \epsilon$ with an absolute constant $\epsilon > 0$, then the misclustering error of Algorithm 1 is such that*

$$\frac{l(\sigma, \hat{\sigma})}{N} = O\left(\frac{\log N}{Np}\right) \tag{8}$$

If $\eta = 0.5 - o(1)$, we have that

$$\frac{l(\sigma, \hat{\sigma})}{N} = \omega\left(\frac{\log N}{Np}\right). \tag{9}$$

Proof From Theorem 2, we have the general upper bound on the error bound

$$\frac{l(\sigma, \hat{\sigma})}{N} \leq \frac{64(2+\epsilon)C^2 p_{\max}\log N}{d^2\Delta^2} = \Theta\left(\frac{C^2 p_{\max}\log N}{d^2\Delta^2}\right), \tag{40}$$

where d and Δ depends on $\mathbb{E}[\widetilde{H}]$. When $p = q$, we have that

$$w_r = \log\left(\frac{1}{4\eta(1-\eta)}\right), \quad w_i = \log\frac{1-\eta}{\eta}, \quad w_c = 0.$$

Moreover, notice that normalizing $\widetilde{H}$ does not affect the clustering error. For the rest of the discussion, we consider $1/w_i\widetilde{H}$ as the input Hermitian matrix for Algorithm 1, and correspondingly we denote the updated coefficient as

$$\tilde{w_r} = \log\left(\frac{1}{4\eta(1-\eta)}\right) / \log\left(\frac{1-\eta}{\eta}\right), \quad \tilde{w_i} = 1, \quad \tilde{w_c} = 0.$$

Because $\tilde{w_r} \leq 1$ and $\tilde{w_i} = 1$, the term C^2 in (40) has $C^2 = \Theta(\tilde{w_r}^2 + \tilde{w_i}^2) = \Theta(1)$. Therefore, we can further simplify (40) as follows

$$\frac{l(\sigma, \hat{\sigma})}{N} \leq \Theta\left(\frac{p\log N}{d^2\Delta^2}\right).$$

For analyzing the asymptotic behaviour of the error bound, we are only left with computing the centroid distance d and eigengap bound Δ.

Following from the definition of Δ in (11), we have

$$\Delta = \frac{Np}{2}\sqrt{\tilde{w_r}^2 + \tilde{w_i}^2(1-2\eta)^2}. \tag{41}$$

For computing the centroid distance d, recall that the population matrix can be written as

$$\mathbb{E}[\widetilde{H}] = MQM^T - \tilde{w_r}pI,$$

where M is the community indicator matrix and the 2×2 matrix Q has

$$Q = \begin{bmatrix} \tilde{w_r} & \tilde{w_r} + (1-2\eta)i \\ \tilde{w_r} - (1-2\eta)i & \tilde{w_r} \end{bmatrix} p.$$

Since $n_1 = n_2 = N/2$, we have that the two distinct values in $v_1(\mathbb{E}[\widetilde{H}])$ (see (27)) to be the values of $v_1(Q)$ divided by $\sqrt{N/2}$. We can easily compute that the top eigenvector has

$$v_1(\mathbb{E}[\widetilde{H}]) = \begin{cases} \frac{w_r + w_i(1-2\eta)i}{\sqrt{N}|-w_r + w_i(1-2\eta)i|} & \text{for } u \in \mathcal{C}_1 \\ 1/\sqrt{N} & \text{for } u \in \mathcal{C}_2. \end{cases} \tag{42}$$

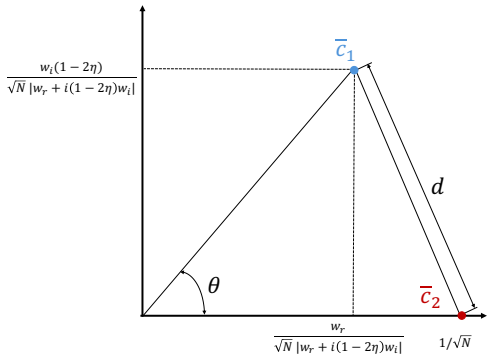
We denote $\bar{c_1}, \bar{c_2}$ the cluster centroids of $\mathcal{C}_1, \mathcal{C}_2$ in the embedding given by the top eigenvector of $\mathbb{E}[\widetilde{H}]$. The locations of two cluster centroids in the complex plane are exactly the two distinct values in (42). We visualize the two cluster centroids in Figure 7a. Let $\theta = \arccos\left(\frac{w_r}{|w_r + w_i(1-2\eta)i|}\right)$ be the angle between the two values in the complex plane. Therefore, we have

$$d^2 = \frac{1}{N}(1 - \cos\theta) = \frac{4\sin^2\theta/2}{N}. \tag{43}$$

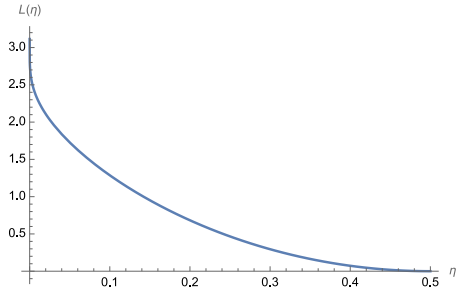
Combining (40), (41), and (43) and letting $L(\eta) = |\tilde{w_r} + i(1-2\eta)|\sin(\theta/2)$, we have

$$\frac{l(\sigma, \hat{\sigma})}{N} \leq \Theta\left(\frac{\log N}{NpL^2}\right). \tag{44}$$

From the above inequality (44), the upper bound of misclustering error is determined by two independent variables Np and L^2. The term Np is the average degree of the graph. The term L^2, by definition, is a function on η. To see how the value L changes as η varies from 0 to 0.5, we plot $L(\eta)$ in Figure 7b using Mathematica Inc.. We observe that



(a) Visualization on $\bar{c}_1, \bar{c}_2, d$ and θ.



(b) Plot of L as function of η.

Figure 7: Visualization of important parameters for representing the error bound.

$L(\eta) = \Theta(1)$ when η is bounded away from 0.5. Therefore, if $\eta \leq 0.5 - \epsilon$ with an absolute constant $\epsilon > 0$, then the misclustering error of Algorithm 1 is such that

$$\frac{l(\sigma, \hat{\sigma})}{N} = O\left(\frac{\log N}{Np}\right)$$

When η converges to 0.5 (when the imbalance structure disappears), $L(\eta)$ converges to 0. Therefore, if $\eta = 0.5 - o(1)$, we have that

$$\frac{l(\sigma, \hat{\sigma})}{N} = \omega\left(\frac{\log N}{Np}\right).$$

The above results on misclustering error bounds agree with the intuition that lower values of η denote a less noisy problem instance, and thus lead to a lower clustering error.

■

Appendix D. Additional experimental details

D.1 Experiments on DSBM with known model parameters

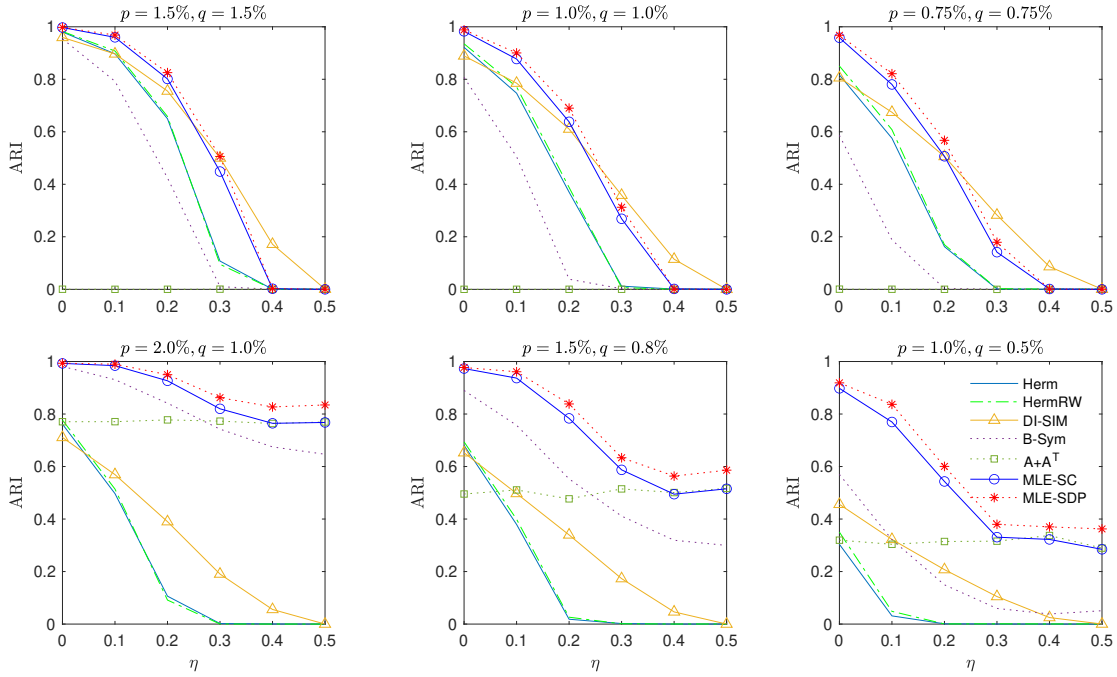


Figure 8: Cluster DSBM with $n_1 = n_2 = 1000$, different p, q and varying η. MLE-SC represents Algorithm 1 with true model parameters; MLE-SDP represents Algorithm 3 with true model parameters.

D.2 Visualization on directed adjacency matrices

In Figure 9, we test the clustering algorithms on DSBM and visualize the adjacency relation before and after clustering. To provide a clear visual demonstration, we generate directed graphs from DSBM with only 200 vertices.

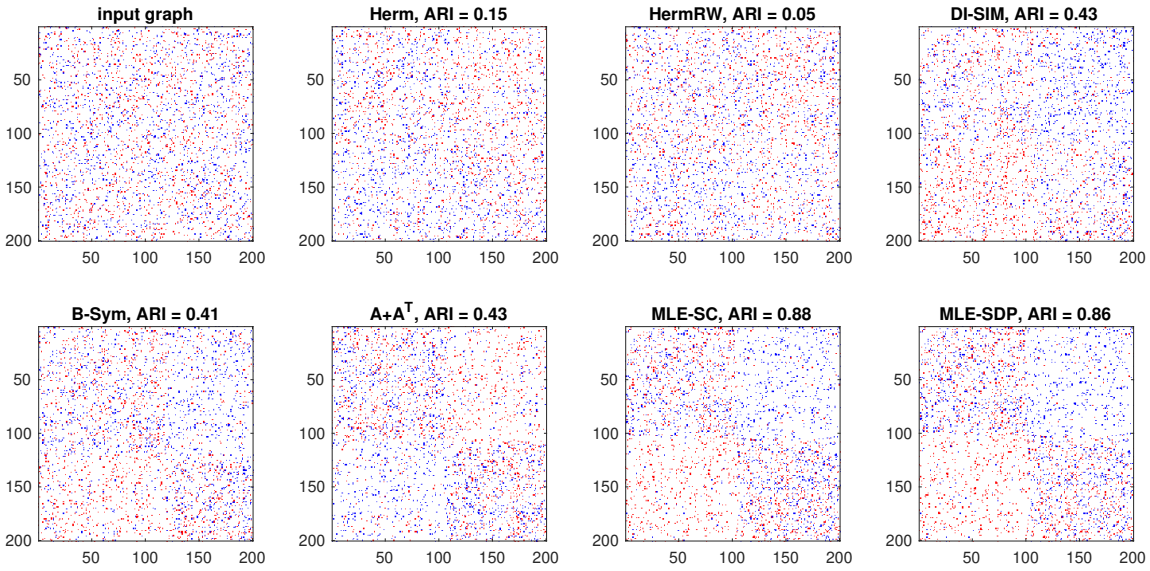


Figure 9: Visualization of $A - A^T$ before and after clustering where A is sampled from DSBM with $n_1 = n_2 = 100, p = 0.1, q = 0.05, \eta = 0.1$.

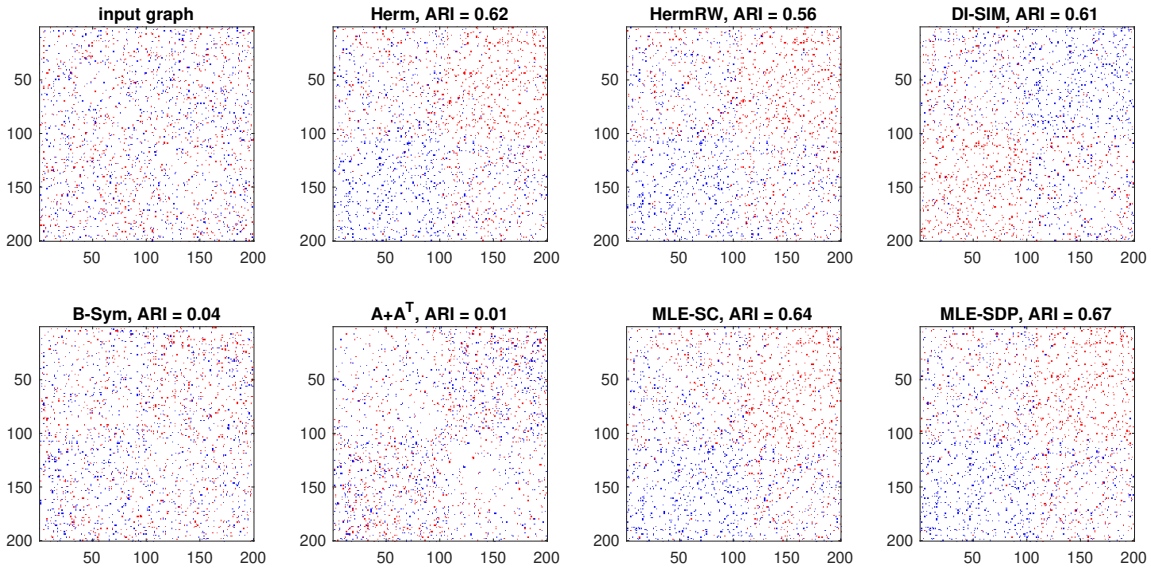


Figure 10: Visualization of $A - A^T$ before and after clustering where A is sampled from DSBM with $n_1 = n_2 = 100, p = 0.05, q = 0.05, \eta = 0.1$.

D.3 Additional real-world dataset

Data set	Herm	HermRW	DI-SIM	B-Sym	$A + A^T$	MLE-SC	MLE-SDP
PolBlog [3]	0.008	-0.000	-0.001	**0.148**	**0.177**	0.014	0.105

Table 3: ARIs from experiments on the real-world PolBlog data set.

4 Adamic and Glance (2005)

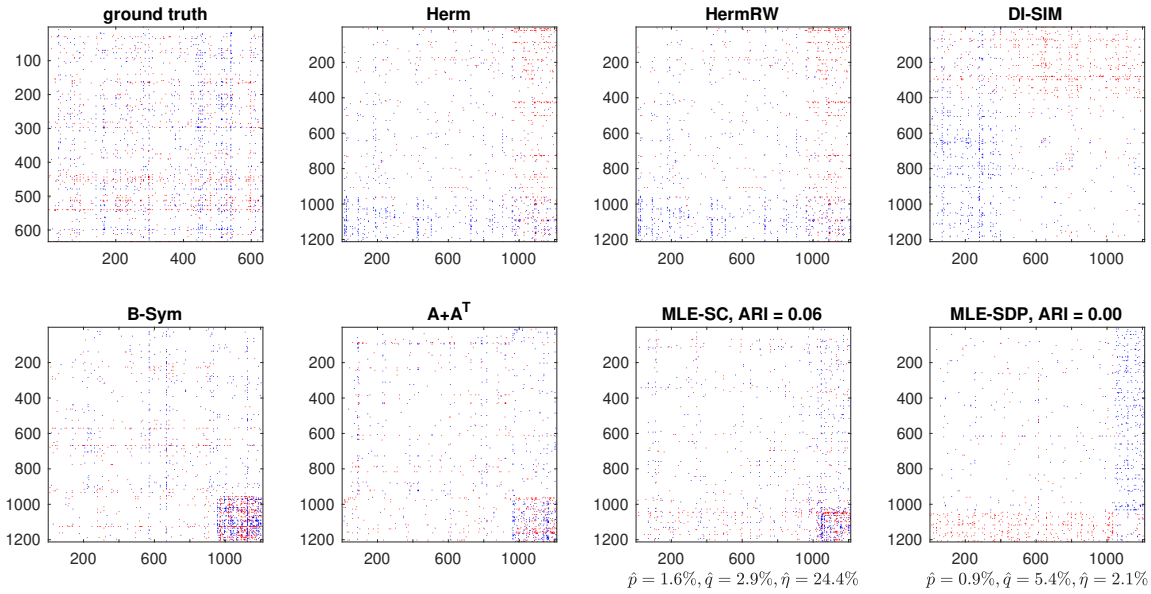


Figure 11: Visualization of $A - A^T$ before and after clustering where A is the adjacency matrix of the PloBlog graph.

D.4 Experiments with different initialization strategies

In this section, we demonstrate by example how different initialization strategies of Algorithm 4 may lead to different learning outcomes. We conduct experiments on the following three strategies:

Strategy 1. Total flow preferred. We initialize with $H_0 = A + A^T$. We call this total flow preferred initialization as $\mathbf{x}^* H_0 \mathbf{x} = C - \mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)$ according to our optimization interpretation in Section 3.2.

Strategy 2. Net flow preferred. We initialize with $H_0 = i(A - A^T)$. We call this net flow preferred initialization as $\mathbf{x}^* H_0 \mathbf{x} = \mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2)$ according to our discussion in Section 3.2.

Strategy 3. Balanced between net flow and total flow. We initialize with $H_0 = i(A - A^T) + (A + A^T)$, where we assign equal weights on $\mathbf{TF}(\mathcal{C}_1, \mathcal{C}_2)$ and $\mathbf{NF}(\mathcal{C}_1, \mathcal{C}_2)$.

We visualize the clustering results using the above strategies on the DSBM dataset (Figure 12) and email-Eu-core dataset (Figure 13, Figure 14). From the conducted experiments, we observe that directed graphs may have different types of clusters: one values more on edge density difference and leads to large $\frac{\max\{\hat{p}, \hat{q}\}}{\min\{\hat{p}, \hat{q}\}}$; the other cares more about between-cluster edge orientation, which produces small $\hat{\eta}$. When an input graph may exhibit different clustering possibilities, we observe that initializing with $H_0 = i(A - A^T)$ tends to converge to edge orientation favored clustering, while $H_0 = A + A^T$ are more likely to converge to clustering based on edge density.

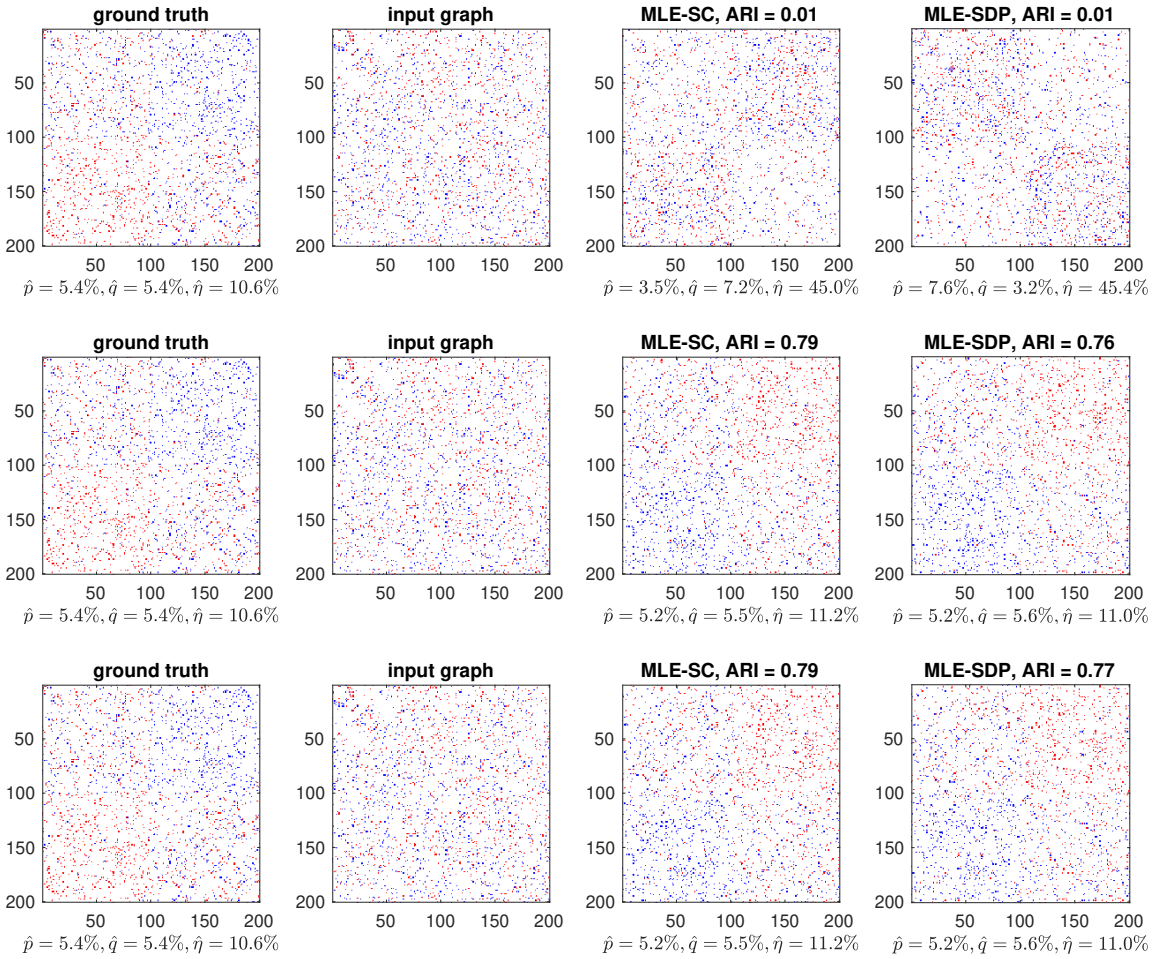


Figure 12: $A - A^T$ before and after clustering. We test on a directed graph sampled from DSBM with $n_1 = 100, n_2 = 100, p = 0.05, q = 0.05, \eta = 0.1$.
Results presented on the first row correspond to total flow preferred initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = A + A^T$. Results presented on the second row correspond to net flow preferred initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = i(A - A^T)$. Results presented on the third row correspond to balanced initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = i(A - A^T) + A + A^T$.

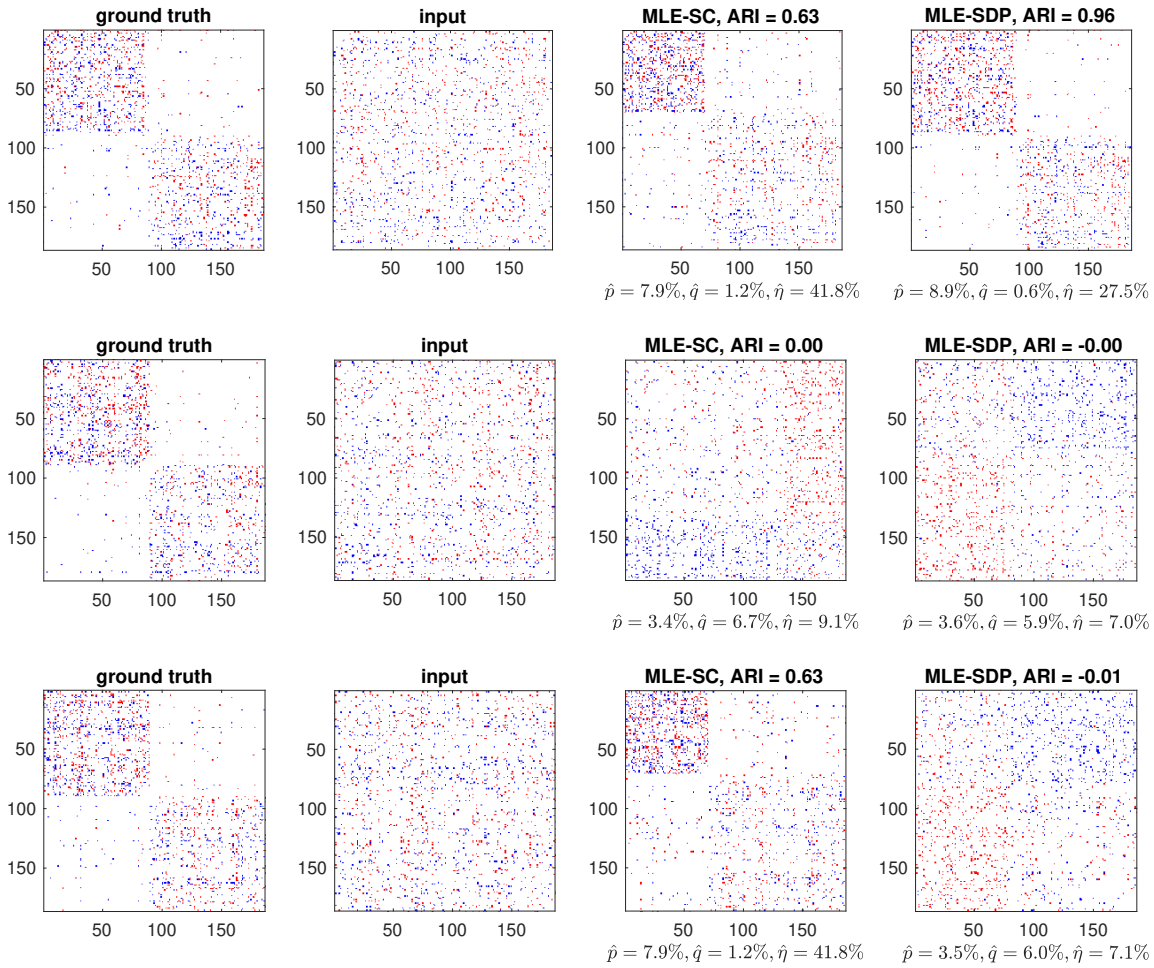


Figure 13: $A - A^T$ before and after clustering. We test on email-Eu-core12. Results presented on the first row correspond to total flow preferred initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = A + A^T$. Results presented on the second row correspond to net flow preferred initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = i(A - A^T)$. Results presented on the third row correspond to balanced initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = i(A - A^T) + A + A^T$.

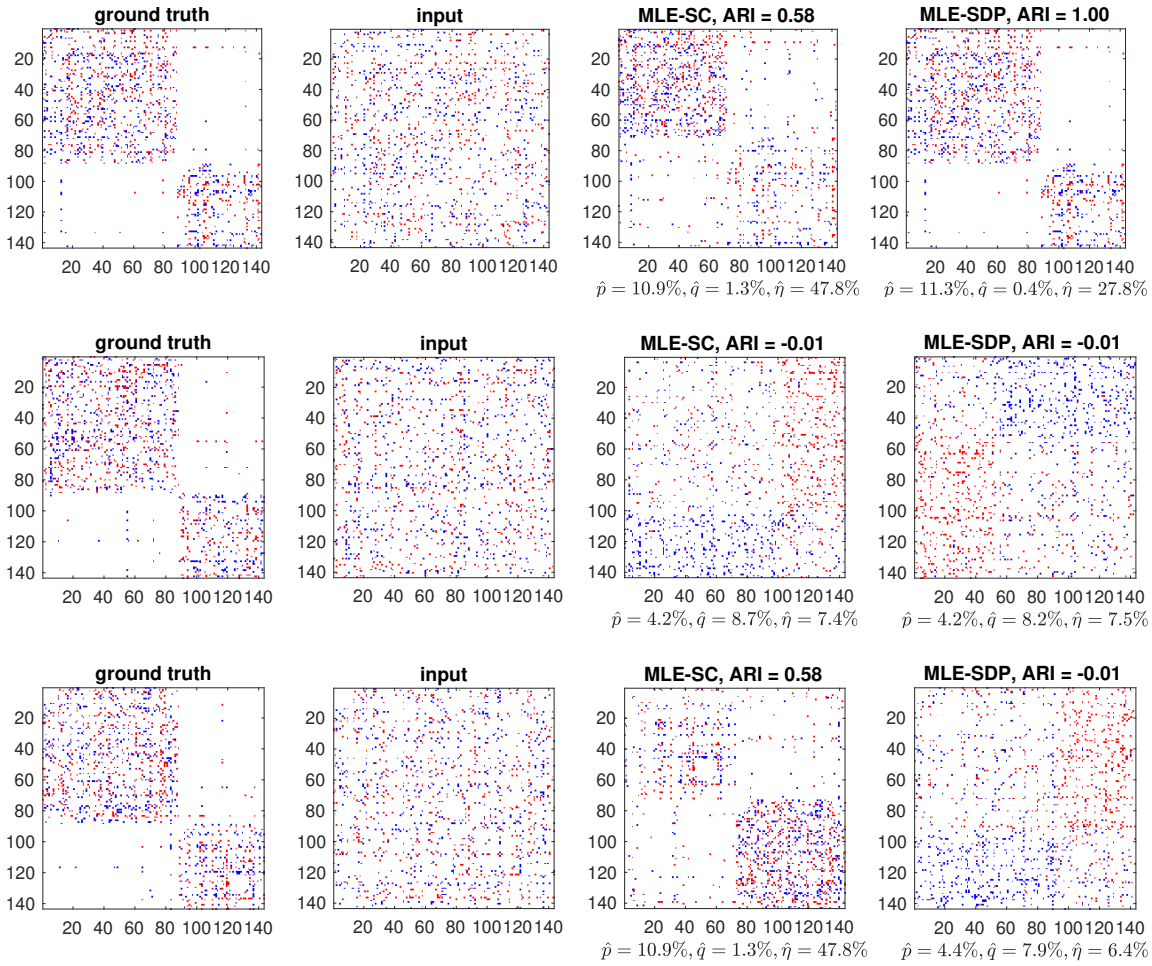


Figure 14: $A - A^T$ after clustering. We test on email-Eu-core12. Results presented on the first row correspond to total flow preferred initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = A + A^T$. Results presented on the second row correspond to net flow preferred initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = i(A - A^T)$. Results presented on the third row correspond to balanced initialization, where MLE-SC and MLE-SDP are both initialized with $H_0 = i(A - A^T) + A + A^T$.