
SubSearch: Robust Estimation and Outlier Detection for Stochastic Block Models via Subgraph Search

Leonardo Martins Bianco¹

Christine Keribin¹

Zacharie Naulet²

Abstract

Community detection is a fundamental task in graph analysis, with methods often relying on fitting models like the Stochastic Block Model (SBM) to observed networks. While many algorithms can accurately estimate SBM parameters when the input graph is a perfect sample from the model, real-world graphs rarely conform to such idealized assumptions. Therefore, robust algorithms are crucial—ones that can recover model parameters even when the data deviates from the assumed distribution. In this work, we propose SUBSEARCH, an algorithm for robustly estimating SBM parameters by exploring the space of subgraphs in search of one that closely aligns with the model’s assumptions. Our approach also functions as an outlier detection method, properly identifying nodes responsible for the graph’s deviation from the model and going beyond simple techniques like pruning high-degree nodes. Extensive experiments on both synthetic and real-world datasets demonstrate the effectiveness of our method.

1 INTRODUCTION

Community detection on a graph is the task of partitioning its set of nodes in such a way that each partition, also called a community, represents a group of nodes sharing a common property, such as a similar pattern of connections. This problem has a

wide range of applications, including understanding of the spread of epidemics (Stegehuis et al., 2016), customer segmentation in advertising (Lalwani et al., 2015), and identification of criminals through online activity (Sangkaran et al., 2020). One approach to community detection consists of fitting a model to the observed data, with the Stochastic Block Model (SBM) (Holland et al., 1983) being a popular choice. In the SBM, each node i is assigned a latent variable Z_i such that the sets $\Omega_k = \{i : Z_i = k\}$, $1 \leq k \leq K$, partition the graph into K communities. Any two nodes i and j are connected with a probability $\Gamma_{Z_i Z_j}$, which depends only on their respective communities. Given a graph, the task of inferring the hidden partition $\{\Omega_k\}_{k=1,\dots,K}$ is called community recovery (or retrieval), while estimating the connectivity parameters Γ is called parameter estimation. Common methods for these tasks include semidefinite programming (SDPs) (Li et al., 2021), spectral (Lei and Rinaldo, 2015), and variational approaches (Tabouy et al., 2020).

However, many algorithms that have guarantees under the assumption of an SBM are highly sensitive to model misspecification—when the input data is drawn from a distribution that differs from the model’s assumptions (Cai and Li, 2015). This sensitivity poses a significant challenge in practice, as real-world graphs rarely exhibit the level of connection homogeneity imposed by the SBM. As a result, these algorithms can produce inaccurate partitions or biased parameter estimates (see Figure 1). One alternative is to use more complex models to describe the data (for example, a Degree-Corrected SBM (Karrer and Newman, 2011)). While this works in some cases, it also introduces additional challenges, both theoretically and practically, due to the increased complexity of the model. Another solution, the one that interests us here, is to keep the simpler model and focus on developing robust algorithms—those producing reliable results even when given (slightly) misspecified inputs.

We consider the problem of robustly estimating connectivity parameters Γ under adversarial perturba-

¹Université Paris-Saclay, CNRS, Inria, Laboratoire de Mathématiques d’Orsay, 91405 Orsay, France

²Université Paris-Saclay, INRAE, MaIAGE, 78350, Jouy-en-Josas, France

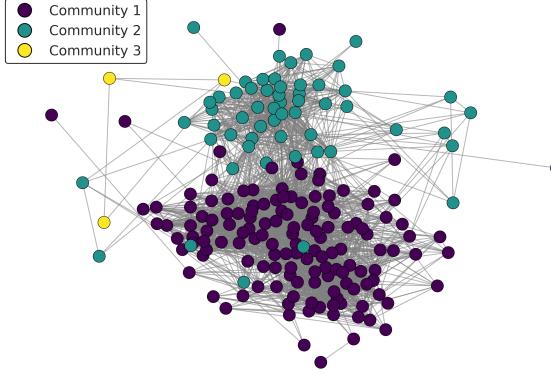


Figure 1: Spectral clustering applied to the jazz collaboration dataset ([Gleiser and Danon, 2003](#)). Nodes represent jazz musicians, with edges being collaborations during 1912 - 1940. This algorithm, which has guarantees under the SBM ([Lei and Rinaldo, 2015](#)), fails to separate the graph into its three main collaboration groups.

tions, a particular type of model misspecification in which an adversary modifies a sample from the (well-specified) model before passing it as input to the algorithm, with the goal of inducing as much error as possible. This problem is important for several reasons. First, knowing the community structure is insufficient for parameter estimation, as adversarial perturbations can significantly alter the observed connections within and between communities, distorting the true underlying probabilities. Second, identifying the outliers causing these perturbations is nontrivial. Third, several algorithms for robust community recovery require prior knowledge of the SBM parameters, creating a circular dependency between parameters and latent variables. Consequently, robustly estimating the parameters of an SBM is an important and complex problem in its own right.

To the best of our knowledge, the work of [Acharya et al. \(2022\)](#) is the first to address the problem of robust estimation on random graphs, focusing on the parameter estimation of a perturbed Erdős-Rényi model. This corresponds to the case of a single community ($K = 1$). Their algorithm employs a greedier iterative node removal scheme that often gets trapped in poor local optima when applied to graphs with multiple communities. Our work extends to the more relevant case of $K > 1$ where the real interest in community detection lies, and thoroughly explores the subgraph space, leading to improved solutions.

Additionally, our method provides an outlier-detection mechanism that identifies nodes most responsible for disturbing the model’s quality of fit, allowing for a

more nuanced analysis of outliers beyond simplistic strategies based solely on node degree distribution.

Finally, there is a notable gap in recent literature regarding experimental validation and implementation of algorithms. We provide experimental validation to our method, evaluating it across a range of synthetic data experiments to assess its performance as parameters vary, as well as performing experiments on real graphs.

Our Contributions

- We prove a bound for the estimation error in Theorem 3.1 that generalizes the bound appearing in [Acharya et al. \(2022\)](#) to the case of graphs with multiple communities.
- Based on this bound, we propose a cost function to be minimized for robustly estimating the connectivity parameters of an SBM.
- Our main contribution is to propose an algorithm, called SUBSEARCH, based on Simulated Annealing (S.A.) to minimize this cost function (Algorithm 1). The novelty of this algorithm is that it explores the space of solutions more thoroughly, managing to escape bad local optima that previous methods fall into.
- SUBSEARCH also serves as an outlier-detection method, identifying a set of nodes that are most deviant from the model, beyond the naïve pruning of nodes of extreme degree.
- We provide a variety of experiments to support the effectiveness of our approach, with both synthetic and real graphs. This addresses a gap in the literature of robust community detection, and to our knowledge our work is among the first to provide this experimental validation.

1.1 Related Work

The pioneer work of [Acharya et al. \(2022\)](#) estimates the parameter of an Erdős-Rényi random graph model under node corruptions. They establish a “certification” bound and propose an algorithm based on iterative node removal using eigenvector scores, which can be seen as a variant of the filtering algorithm used in [Diakonikolas et al. \(2019\)](#) to robustly fit a high-dimensional Gaussian. However, they do not consider graphs with multiple communities and their algorithm requires erasing a significant portion of the original graph. Recent work by [Chen et al. \(2024\)](#) extend robust estimation to inhomogeneous graphs, including SBMs. However, their approach is based on Sum-of-Squares relaxations to Semidefinite Programs (SDPs), which, despite their polynomial-time complexity in

theory, may be computationally expensive in practice. In [Jana et al. \(2024\)](#), an estimator similar to the one proposed here is presented, but their approach is more akin to a direct k -means with trimming rather than the exploration of subgraph space that we do here.

Much of the prior literature focused on robust community recovery, *i.e.*, accurately inferring community labels. Seminal works by [Cai and Li \(2015\)](#) and [Makarychev et al. \(2016\)](#) explored this problem using semidefinite programs (SDPs), while [Stephan and Massoulié \(2019\)](#) and [Abbe et al. \(2020\)](#) study robustness of algorithms based on the spectrum of the adjacency matrix (or matrices related to it). Previously [Ding et al. \(2023\)](#) studied robust recovery up to its fundamental limit on the SBM with sparse connections, using the Sum-of-Squares paradigm. The work of [Srivastava et al. \(2021\)](#) considers the problem of robust recovery in the more general case of sub-gaussian mixtures, and their approach is based on a linear programming relaxation of a robust SDP.

2 SETUP

Notation. In what follows, n and K are positive integers with $K \leq n$. For any finite set C , $|C|$ denotes the number of elements in it. All vector norms are the Euclidean norm $\|x\| = \sum_{i=1}^n x_i^2$, and matrix norms are the spectral norm, also called the operator norm, defined for $A \in \mathbb{R}^{m \times n}$ as $\|A\| := \sup_{x \in \mathbb{R}^n \setminus \{0\}} \|Ax\|/\|x\|$. It can be shown that $\|A\| = \sqrt{\lambda_{\max}(A^t A)}$, where $\lambda_{\max}(A^t A)$ denotes the maximal eigenvalue of $A^t A$. We write $g(n) = O(f(n))$ when there exist some positive real C and some positive integer n_0 such that $|g(n)| \leq C \cdot f(n)$ for all $n \geq n_0$.

Graphs. A graph is a pair of sets $G = (V, E)$, where $V = \{1, \dots, n\}$, is called the set of nodes and $E \subset V \times V$ is called the set of edges. We will deal with undirected simple graphs, meaning that $(i, j) \in E \Rightarrow (j, i) \in E$ and that $(i, i) \notin E$ for all i . Such a graph can be represented in matricial form by a symmetric matrix called an adjacency matrix, defined as

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

The degree of a vertex i is the number of edges connected to it, *i.e.*, $\deg(i) = \sum_{j=1}^n A_{ij}$.

A graph with communities is simply a graph along with a partition of its set of nodes V into K non-empty sets $(\Omega_k)_{k=1, \dots, K}$ called communities, *i.e.*, $V = \Omega_1 \cup \dots \cup \Omega_K$ with $\Omega_i \cap \Omega_j = \emptyset, \forall i \neq j$. Communities can be represented by a community assignment vector $z \in \{1, \dots, K\}^n$ or by a community assignment matrix

$Z \in \{0, 1\}^{n \times K}$ such that $\sum_j Z_{ij} = 1$ for every $i \in \{1, \dots, n\}$.

A popular generative model for graphs with communities is called the Stochastic Block Model (SBM) ([Holland et al., 1983](#)). It has a community size parameter $\Pi = (\pi_1, \dots, \pi_K)$ such that $\forall k, 0 < \pi_k < 1$ and $\sum_k \pi_k = 1$, and connectivity parameters $\Gamma \in [0, 1]^{K \times K}$. Given these parameters, the SBM with K communities is a probability \mathbb{P} over the space of graphs with communities determined by

$$\begin{aligned} \mathbb{P}(Z) &= \prod_{k=1}^K \pi_k^{|\Omega_k|}, \\ \mathbb{P}(A|Z) &= \prod_{i \neq j} \Gamma_{z_i z_j}^{A_{ij}} (1 - \Gamma_{z_i z_j})^{1 - A_{ij}}. \end{aligned}$$

Given community assignments Z , one can show that $\mathbb{E}[A] = Q - \text{diag}(Q)$, where $Q = Z\Gamma Z^t$ and $\text{diag}(Q)$ is the $n \times n$ containing only the diagonal of Q on the diagonal and zeroes elsewhere.

Submatrices. Let A be any $n \times n$ matrix, $S_1, S_2 \subset \{1, \dots, n\}$ be subsets of the row and column indices, respectively. Without loss of generality, we assume that S_1 is sorted and we denote $S_1(i)$ the i -th element of S_1 (respectively for S_2). The restriction of A to $S_1 \times S_2$ is the matrix $A_{S_1 \times S_2}$ given, for $i = 1, \dots, |S_1|$ and $j = 1, \dots, |S_2|$, by $(A_{S_1 \times S_2})_{ij} = A_{S_1(i)S_2(j)}$. When $S_1 = S_2 = S$, we will simply note $A_{S \times S}$ as A_S .

If A is an adjacency matrix and S_1, \dots, S_K are disjoint subsets of $\{1, \dots, n\}$, we can estimate the connectivity parameters associated to them, for $k, l \in \{1, \dots, K\}$, by

$$\hat{\Gamma}_{kl} = \frac{1}{|S_k||S_l|} \sum_{i=1}^{|S_k|} \sum_{j=1}^{|S_l|} (A_{S_k \times S_l})_{ij},$$

defining a $K \times K$ matrix $\hat{\Gamma}$. This can be extended to an $n \times n$ matrix $\hat{Q}(S) := \mathbf{S}\hat{\Gamma}\mathbf{S}^t$, where \mathbf{S} is the $|S| \times K$ matrix such that $\mathbf{S}_{ij} = 1$ if $S(i) \in S_j$ and 0 otherwise.

Clustering. Given an adjacency matrix A , the symmetric normalized Laplacian matrix associated to it is defined as ([Von Luxburg, 2007](#)) the matrix $L \in \mathbb{R}^{n \times n}$ with entries

$$L_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } \deg(i) \neq 0, \\ -\frac{1}{\sqrt{\deg(i)\deg(j)}} & \text{if } i \neq j \text{ and } A_{ij} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Spectral clustering ([Shi and Malik, 2000](#)) is a widely used clustering technique that consists on applying the K -means algorithm ([Macqueen, 1967](#)) to the rows of the matrix whose columns are the normalized eigenvectors corresponding to the K smallest non-zero eigenvalues of L . We assume that the number of clusters K

is known, and for the rest of the paper all clustering is performed using spectral clustering with this given K .

Problem Statement. We consider the node adversary perturbation model, where an adversary receives a sample (Z, A_0) of an SBM and is allowed to arbitrarily modify the adjacency of up to γn nodes, where $\gamma \in [0, 1/2]$ is a known parameter representing the amount of corruption. This leads to the observation of a corrupted adjacency matrix A . The nodes whose adjacencies were directly modified by the adversary are called outlier nodes, while the rest are called inlier nodes. We denote the set of inlier nodes as F . The goal is to accurately estimate the connectivity¹ parameters Γ of the original SBM from A , in the sense of minimizing the empirical estimation error $\sum_{kl} |\Gamma_{kl} - \hat{\Gamma}_{kl}|$.

3 MAIN RESULTS

Our method is based on directly optimizing a bound relating the error obtained by estimating the connectivity parameters on a subgraph S to the spectral norm $\|A_S - \hat{Q}(S)\|$ associated to it.

3.1 Error Bound

In the following result, we generalize a bound found in [Acharya et al. \(2022\)](#) to the $K > 1$ case. The proof is in the supplementary material.

Theorem 3.1. *Let A be an adjacency matrix sampled from a γ -corrupt SBM with K communities $(\Omega_k)_{k=1,\dots,K}$, connectivity parameters Γ , and inlier nodes F . Furthermore, let S_1, \dots, S_K be non-empty disjoint subsets of $\{1, \dots, n\}$, S be their union, and $\hat{Q}(S)$ be the estimation of the expected adjacency matrix restricted to S . Then,*

$$\begin{aligned} \sum_{k=1}^K \sum_{l=k}^K |\Gamma_{kl} - \hat{\Gamma}_{kl}| &\leq \frac{K^2}{\min_k |\Omega_k \cap S_k \cap F|} \\ &\times \left(\max_k \Gamma_{kk} + \|A_F - \mathbb{E}[A]_F\| + \|A_S - \hat{Q}(S)\| \right). \end{aligned}$$

We now use this bound to motivate our proposed objective function. On the right-hand side of the bound appearing in Theorem 3.1, the term $\max_k \Gamma_{kk}$ is bounded by 1 and becomes negligible as n increases. The term $\|A_F - \mathbb{E}[A]_F\|$ is $O(\sqrt{(1-\gamma)n})$, as it represents the spectral norm of the centered adjacency matrix of the inlier subgraph, which is distributed according to an SBM ([Lei and Rinaldo, 2015](#)). These

two terms are independent of our choice of S and its associated clustering. Now, suppose that we had a subgraph S such that $\|A_S - \hat{Q}(S)\| = O(\sqrt{n})$, mimicking the behavior of the inlier solution, and that $\min_k |S_k \cap \Omega_k \cap F| = O(n)$. Then, Theorem 3.1 implies an estimation error of $O(n^{-1/2})$. This motivates minimizing $\|A_S - \hat{Q}(S)\|$ as a criterion for selecting a subgraph S . We point out that the bound in Theorem 3.1 does depend on γ , but not in the same explicit way as in the case $K = 1$: making the dependence on γ explicit is harder for the $K > 1$ case.

3.2 SubSearch: Subgraph Search with Simulated Annealing

The Simulated Annealing (S.A.) algorithm, introduced by [Kirkpatrick et al. \(1983\)](#), is inspired by the metallurgical process of annealing, where the slow cooling of a heated solid brings it to a lower-energy state with fewer defects. Analogously, S.A. optimizes a given cost function c by exploring the state space with an initially high “temperature” parameter that is then slowly decreased, helping the algorithm to find global or near-global solutions.

In Algorithm 1, we propose SUBSEARCH, an algorithm using S.A. to explore the state space \mathcal{S} consisting of all subgraphs of size $(1 - \gamma)n$ of the input graph G , in search for a subgraph S minimizing the cost function $c(S) = \|A_S - \hat{Q}(S)\|$. The algorithm begins with a randomly selected connected subgraph $S_{\text{current}} \in \mathcal{S}$ and initial temperature T_0 . At each step t , the algorithm generates a Markov chain of length l_t at fixed temperature T_t in the following way. Define the neighborhood $N(S)$ of any subgraph $S \in \mathcal{S}$ as the subgraphs that can be obtained by swapping a node within S with an adjacent node outside of it. At each step l of the chain, a neighboring connected subgraph $S_{\text{candidate}} \in N(S_{\text{current}})$ is proposed as a candidate for the next state (call to the `neighbor` function in Algorithm 1). The cost difference between the current and the candidate state will be denoted $\Delta := c(S_{\text{current}}) - c(S_{\text{candidate}})$. The candidate is accepted with probability $\min(1, \exp(\Delta/T_t))$, in which case the state S_{current} is updated to it. After completing the Markov chain, the temperature is decreased following a geometric cooling schedule $T_{t+1} = cT_t$, where c is the cooling rate parameter, typically close to 1. This process is repeated until either a maximum number of iterations t_{max} is reached or the maximal absolute variation of the cost over t_{tol} chains falls below a tolerance threshold ε (call to `stopping_conditions` in Algorithm 1). The algorithm returns the subgraph with minimal cost. The initial temperature is determined adaptively: starting with $T_0 = 1$, a separate long Markov chain (e.g., with 100 states) is run,

¹An adversary can bias the size parameters while being undetectable: it suffices for it to resample the connections of nodes in one community as if they were in another. For this reason, we focus on the connectivity parameters.

and T_0 is multiplied by 1.5 until the rate of acceptance of neighboring states is close to one (call to `set_initial_temp` in Algorithm 1). This heuristic aims at finding an initial temperature high enough to allow for an effective initial exploration phase without being so high as to slow the convergence. Notice that we implicitly cluster S when calculating $c(S)$, due to $\hat{Q}(S)$.

Algorithm 1 SUBSEARCH

Require: $A, K, \gamma, c, (l_t)_{t=0,\dots,t_{\max}}, t_{\max}, t_{\text{tol}}, \varepsilon.$

```

 $S_{\text{current}} \leftarrow$  connected subgraph with  $|S| = (1 - \gamma)n$ 
 $S_{\text{best}} \leftarrow S_{\text{current}}$ 
 $T_0 \leftarrow \text{set\_initial\_temp}(S_{\text{current}})$ 
for  $t = 1, \dots, t_{\max}$  do
    for  $l = 1, \dots, l_t$  do
         $S_{\text{candidate}} \leftarrow \text{neighbor}(S_{\text{current}})$ 
         $\Delta \leftarrow c(S_{\text{current}}) - c(S_{\text{candidate}})$ 
         $u \sim \mathcal{U}([0, 1])$ 
         $\text{accept\_prob} \leftarrow \min(1, \exp(\Delta/T_t))$ 
        if  $u < \text{accept\_prob}$  then
             $S_{\text{current}} \leftarrow S_{\text{candidate}}$ 
            if  $c(S_{\text{current}}) < c(S_{\text{best}})$  then
                 $S_{\text{best}} \leftarrow S_{\text{current}}$ 
         $T_{t+1} \leftarrow cT_t$ 
        if  $\text{stopping\_conditions}(\varepsilon, t_{\text{tol}})$  then
            break
    return  $S_{\text{best}}$ 

```

We emphasize that our approach to the $K > 1$ case goes beyond an incremental extension of existing methods for $K = 1$. In fact, when $K > 1$, both outliers and misclustered inliers can increase $\|A_S - \hat{Q}(S)\|$. The filtering approach proposed in Acharya et al. (2022), which works for $K = 1$, does not distinguish between outliers and misclustered inliers, resulting in an excessive removal of inliers. For this reason we propose subgraph exploration as an alternative.

Convergence Properties. Many works have studied the convergence properties of S.A. (Henderson et al., 2003). Intuitively, lowering the temperature slowly enough allows the observed sequence of states to form a “near-stationary” Markov chain, and the associated sequence of near-stationary distributions converges to a distribution supported on the set of global optima of the function being optimized.

To formally state this, let $P_{SS'}(T_t)$ be the probability of going from state S to state S' at temperature T_t , i.e., the probability of generating S' from S and then accepting it. This defines an $|\mathcal{S}| \times |\mathcal{S}|$ transition matrix $P(T_t)$ at each temperature value. Given an initial probability vector ν_0 over the state space, the

associated state probability vector at time t is defined as $\nu_t := P(T_{t-1}) \dots P(T_0) \nu_0$. Let \mathcal{S}^* denote the set of global minima of c . The vector $\mathbf{e}^* \in [0, 1]^{|\mathcal{S}|}$ is defined as the vector with entries $\mathbf{e}_S^* := 1/|\mathcal{S}^*|$ if $S \in \mathcal{S}^*$, 0 if $S \notin \mathcal{S}^*$. Notice \mathbf{e}^* is supported on the set of optimal solutions. Finally, we also denote $M := \max_{S \in \mathcal{S}} \max_{S' \in N(S)} |c(S) - c(S')|$ the maximal local variation of the cost function, $d(S, S')$ the minimal number of transitions needed to go from subgraph S to subgraph S' , and $r := \min_{S \in \mathcal{S} \setminus \mathcal{S}^*} \max_{S' \in \mathcal{S}} d(S, S')$ a quantity analogous to a measure of the “radius” of the state space according to d .

Proposition 3.2 (Mitra et al. (1986)). *Suppose that the temperature evolves according to*

$$T_t = \frac{C}{\log(t + t_0 + 1)}, \quad t = 0, 1, \dots,$$

for some $C > 0$ and arbitrary $1 \leq t_0 < \infty$. If $C \geq rM$, then, for any starting initial probability vector ν_0 ,

$$\lim_{t \rightarrow \infty} \|\nu_t - \mathbf{e}^*\| = 0.$$

The cooling schedule in Proposition 3.2, however, is often too slow for practical applications. The inverse-logarithmic decay not only progresses at a slow rate, requiring an exponential number of iterations to reach a desired temperature T , but C might also grow rapidly with n . In practice, faster cooling schedules are often successfully used despite their lack of theoretical guarantees, and in Algorithm 1 we use a geometric cooling rate.

We also note that since $|S| = (1 - \gamma)n$, only the subgraph of inliers is entirely free of outliers. However, many solutions include *a few* outliers, and for these subgraphs the method can still yield good parameter estimates. Intuitively, this is because outliers must “conspire together” to meaningfully affect the estimation error; when their number is small, they lose the ability to introduce significant bias to the estimation. See Diakonikolas et al. (2021) for a discussion of this intuition in the Gaussian case.

4 EXPERIMENTS

We conducted several experiments to demonstrate the applicability of Algorithm 1 and to compare it with other approaches. First, we apply our method to a single graph and analyze the relationship between the cost function, the estimation error, and the number of outliers in the subgraph. Next, we perform two “multi-run” experiments to understand how variations in specific parameters impact the algorithm. The first multi-run experiment explores the dependence of estimation error on the amount of perturbation γ , while

the second demonstrates that the cost-to-overlap ratio $\|A_S - \hat{Q}(S)\| / \min_k |S_k \cap \Omega_k \cap F|$ appearing in the bound of Theorem 3.1 decreases with $O(n^{-1/2})$. Finally, we test our method on a real-world graph. Additional experiments in the supplementary material include results on the political blogs network (with 1222 nodes), a comparison with the Degree-Corrected SBM, an analysis of exploration’s importance, and an examination of the algorithm’s intrinsic variability.

The code used is available at https://github.com/leobianco/robust_estim_sbm/. All experiments were executed on a 2.7 GHz Dual-Core Intel i5 processor and 8 GB of DDR3 RAM memory. All S.A. experiments were run with `seed=12345`, `cooling_rate=0.99`, `n_iters_outer=1000` outer iterations, Markov Chains of length $L_t = m = \gamma n$, and the stopping condition of `n_iters_tolerance=25` iterations with absolute cost variation below `tolerance=10^-4`. For a more detailed discussion on how to choose hyperparameters, see [Delahaye et al. \(2019\)](#).

Model for Corruptions. Since explicitly determining the worst-case perturbation for any given graph is challenging, we propose a perturbation model for our experiments on synthetic data as follows. Sample an uncorrupted graph from an SBM and $m = \lfloor \gamma n \rfloor$ nodes to be the outliers. For each outlier $i = 1, \dots, m$ and each $k = 1, \dots, K$, draw a new connection probability between that node and nodes in community k using a Beta distribution, *i.e.*, $\tilde{\Gamma}_{ik} \sim \mathcal{B}(\alpha, \beta)$. Here, α and β are chosen so that $\mathbb{E}[\tilde{\Gamma}_{ik}] = \Gamma_{z(i)k}$ and that the variance is the greatest possible (with the constraint that $\tilde{\Gamma}_{ik} \in [0, 1]$). This procedure deteriorates empirical edge density estimates while avoiding being obvious to detect due to variations of degree of the corrupted nodes.

Baselines. We compare our method to three baselines. The oracle baseline estimates the parameters using the subgraph of inlier nodes with their true community labels. The pruning baseline clusters the graph, removes `num_to_prune / K` nodes with the highest and lowest degrees from each community (also any isolated nodes that result from this pruning), then reclusters the graph. Finally, the “filtering” method of [Acharya et al. \(2022\)](#) also aims to minimize $c(S) = \|A_S - \hat{Q}(S)\|$, but in a manner different to ours. It starts with $S_0 = G$ and at each step removes a node i_t from S_t sampled according to $i_t \sim v_t^2$, where v_t an eigenvector associated with the top eigenvalue of $A_S - \hat{Q}(S)$. Filtering will be allowed to remove $n/2$ nodes, *i.e.* up to half of the graph.²

²For clarity of exposition, we assume knowledge of the true γ for all methods to achieve the best possible estima-

4.1 Single-run Experiments

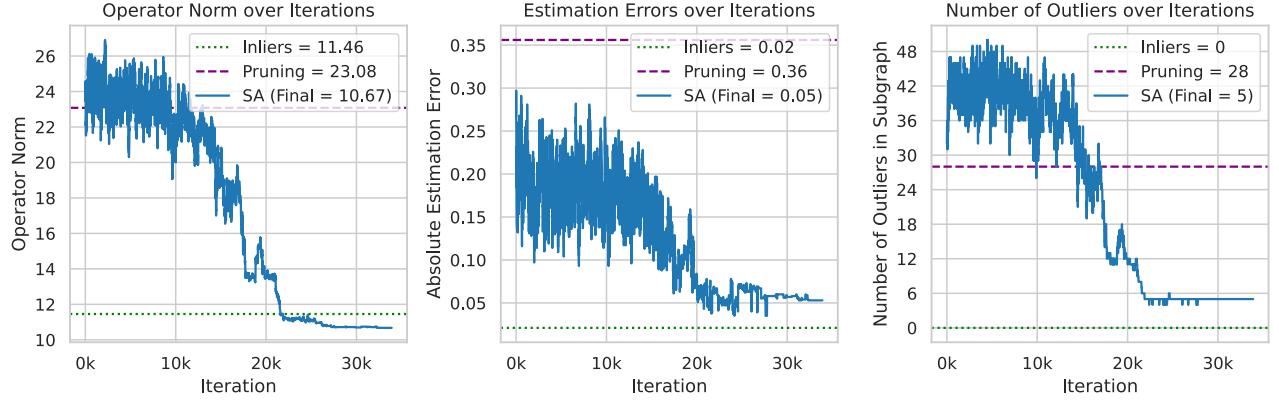
Let us analyze the behavior of a single run of our method on a perturbed graph. We consider a graph with $n = 200$ nodes, $K = 2$ communities, and connectivity parameters $\Gamma_{11} = \Gamma_{22} = 0.65$ and $\Gamma_{12} = 0.35$. We perturb a fraction $\gamma = 0.3$ of the nodes, *i.e.*, we have $m = 60$ outliers. We will search for a subgraph with $|S| = n - m = 140$ nodes with Markov chains of length $L_t = m = 60$ for all t ([Van Laarhoven et al., 1987](#)). Using the procedure explained in Section 3.2, the initial temperature leading to a high initial acceptance rate was determined to be $T_0 = 2.25$. The results are shown in Figure 2a.

The method succeeds in decreasing the spectral norm, and we see a correlation between this norm, the estimation error, and the amount of outliers inside the subgraph. The estimation error at the initial random state is $\sum_{kl} |\Gamma_{kl} - (\hat{\Gamma}_{\text{initial}})_{kl}| = 0.267$, and it strongly oscillates during the first iterations as the temperature is still high and we are prone to accept moving towards a state with a worse cost. As the temperature lowers we eventually stabilize at an absolute estimation error of $\sum_{kl} |\Gamma_{kl} - (\hat{\Gamma}_{\text{SA}})_{kl}| = 0.05$, and the final subgraph contains 5 outliers out of the 60.

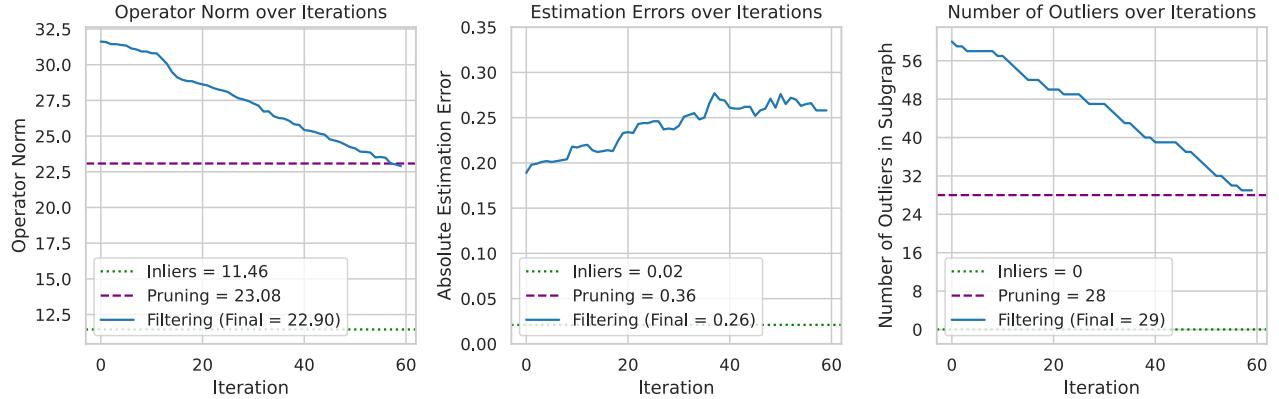
For comparison, the error of the oracle is $\sum_{kl} |\Gamma_{kl} - (\hat{\Gamma}_{\text{inliers}})_{kl}| = 0.02$. Figure 2b reveals that the filtering baseline applied on the same graph fails to decrease the estimation error, despite reducing the operator norm and number of outliers. This is because, unlike our method, filtering monotonically decreases the size of the subgraph considered, and it is this decrease in subgraph size that leads to smaller norm and less outliers. In contrast, our method keeps the subgraph size fixed and relatively large, thus the norm and outlier-count decrease due to our method finding better subgraphs rather than smaller ones. Additionally, the way filtering samples nodes to erase according to the top eigenvector is a greedier optimization technique that can lead to suboptimal solutions, whereas our S.A. exploration allows the finding of good optima. Finally, the pruning baseline with `num_to_prune = γn` commits an error of $\sum_{kl} |\Gamma_{kl} - (\hat{\Gamma}_{\text{pruning}})_{kl}| = 0.353$, much greater than the error of 0.05 committed by our method, and keeps 27 out of the 60 outliers, with a final cost of $c(S_{\text{pruning}}) = 22.15$.

4.2 Multi-run Experiments

Dependence on Amount of Perturbation. We perform an experiment to study the impact of γ on the estimation error. We fix a grid of increasing amounts of perturbation. However, experiments with varying subgraph sizes demonstrate that our method outperforms all others even without assuming knowledge of γ .

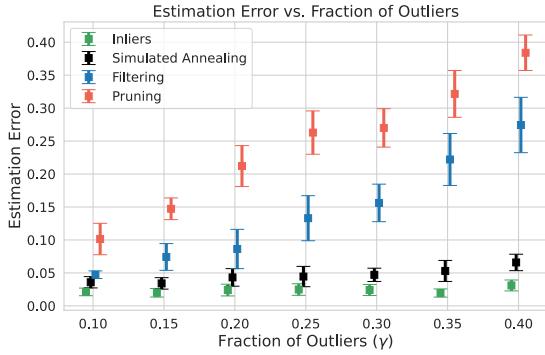


(a) Results of a single-run of our method (S.A.). It decreases the cost (operator norm) and the number of outliers by exploring subgraph space and finding good solutions, while keeping subgraph size constant.

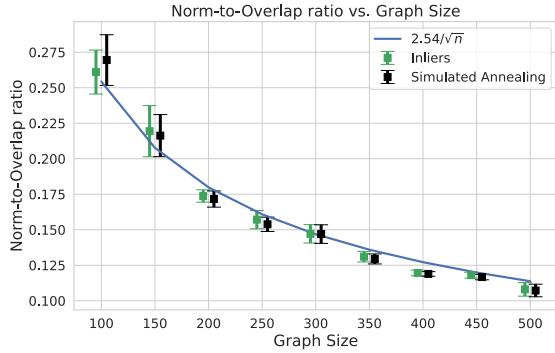


(b) Results of a single-run of the competing filtering baseline. It decreases the cost (operator norm) and the number of outliers by considering smaller subgraphs at each step, but fails to decrease the error due to the lack of exploration.

Figure 2: Results for single-run experiments.



(a) Estimation error of different methods as the amount of perturbation increases. Our method stays close to the inlier baseline (oracle).



(b) Cost-to-overlap ratio (defined as $\|A_S - \hat{Q}(S)\| / \min_k |S_k \cap \Omega_k \cap F|$) as n increases. As discussed after Theorem 3.1, it decreases with $O(n^{-1/2})$.

Figure 3: Results for multi-run experiments.

of perturbation $\gamma = [0.10, 0.15, \dots, 0.40]$, then generate `graphs_per_gamma` = 10 graphs for each amount of corruption. The estimation error is impacted by the variability of the graph generated and by the intrinsic randomness of the algorithm used. To isolate the impact due to the first of these effects from the second, we run each method `runs_per_graph` = 3 times per graph and keep only the one achieving the least norm.³ Other parameters remain the same as in the single-run experiment. The results are shown in Figure 3a, where the mean of the estimation error for each γ is represented along with a Student's 95% confidence interval. This experiment reveals the robustness of our method, as it is the one that remains closer to the error of the oracle as the amount of corruption increases.

Dependence on Graph Size. When discussing the terms in Theorem 3.1, we argued that if the cost of the solution found is $c(S) = O(\sqrt{n})$ and the com-

munity overlap within inliers in the denominator is $\min_k |S_k \cap \Omega_k \cap F| = O(n)$, then the error would decay as $O(n^{-1/2})$. We experimentally verified this, Figure 3b shows this behavior.

4.3 Application to Real Graphs

We consider⁴ the dataset of jazz collaborations introduced in Gleiser and Danon (2003). This graph contains 198 nodes, corresponding to jazz musicians, and 2742 edges, representing collaborations between them during the period of 1912 to 1940. Standard spectral clustering with $K = 3$ fails to properly distinguish its communities, so we turn to robust techniques.

We apply our method with a subgraph of size $|S| = 178$, corresponding to 90% of the size of the whole graph, and Markov chains of constant length `n_iters_inner` = $\gamma n = 19$ for each fixed temperature. The results are shown in Figures 4a and 4b. The outlier histogram indicates that our method prunes some nodes with extreme degree, but goes beyond pruning and removes nodes of moderate-degree whose connections are not well explained by the SBM. Low-degree nodes are not removed, which might be explained by the limited node-budget the algorithm has and the impact of other outliers to the quality-of-fit being greater. The obtained estimates are $\Gamma_{11} = 0.328$, $\hat{\Gamma}_{12} = 0.008$, $\hat{\Gamma}_{13} = 0.068$, $\hat{\Gamma}_{22} = 0.337$, $\hat{\Gamma}_{23} = 0.017$, and $\hat{\Gamma}_{33} = 0.351$, and the final cost is $c(S_{\text{S.A.}}) = 14.94$.

For comparison, the pruning baseline with `num_to_prune` = 30 yields the results in Figure 4c. Though the resulting degree distributions seem similar to those obtained before, their interpretation is different. Pruning does not detect nodes with moderate degree perturbing the estimation. The resulting estimates are $\Gamma_{11} = 0.345$, $\hat{\Gamma}_{12} = 0.007$, $\hat{\Gamma}_{13} = 0.063$, $\hat{\Gamma}_{22} = 0.342$, $\hat{\Gamma}_{23} = 0.017$, and $\hat{\Gamma}_{33} = 0.332$, and the final cost $c(S_{\text{pruning}}) = 21.11$.

A comment on runtimes and scalability. Due to its exploration-based nature, SUBSEARCH demands a greater execution time when compared to other methods: this is a price to robustness. For instance, SUBSEARCH ran the single-run experiment in 14 minutes, while filtering ran in 10 seconds, and pruning in 1 second. We highlight, however, that what we call the filtering and pruning “baselines” do not, in fact, exhibit the robustness properties (*i.e.* low estimation error) that our method does. Thus comparing their speed with ours is not a fair comparison. Moreover,

⁴We chose the jazz collaboration dataset to illustrate the case $K = 3$. The supplementary material contains an additional experiment on a graph of political blogs, with over a thousand nodes. We also experiment fitting a DC-SBM.

³We study the variability due to the randomness of the algorithm separately, in the supplementary material.

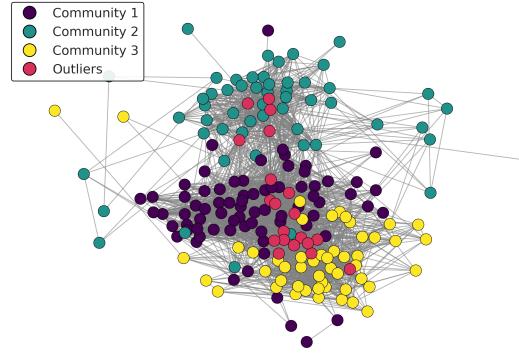
our code is not fully optimized in its current state. We believe our approach could scale to graphs with tens of thousands of nodes.

5 CONCLUSION

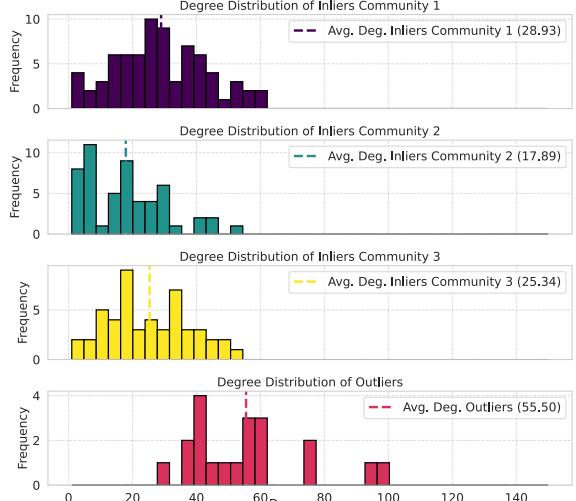
We introduce SUBSEARCH, a method for fitting SBMs to corrupted networks while identifying potential outliers. Unlike methods that get stuck in suboptimal solutions, SUBSEARCH explores the subgraph space to find high-quality fits. It identifies nodes whose connection patterns deviate from model expectations, going beyond basic outlier detection. Experiments on synthetic and real-world data show SUBSEARCH is effective in fitting SBMs and spotting outliers. Future work could explore metaheuristics beyond S.A., investigate the cost function’s theoretical properties for convergence and robustness, and apply the approach to other network models.

References

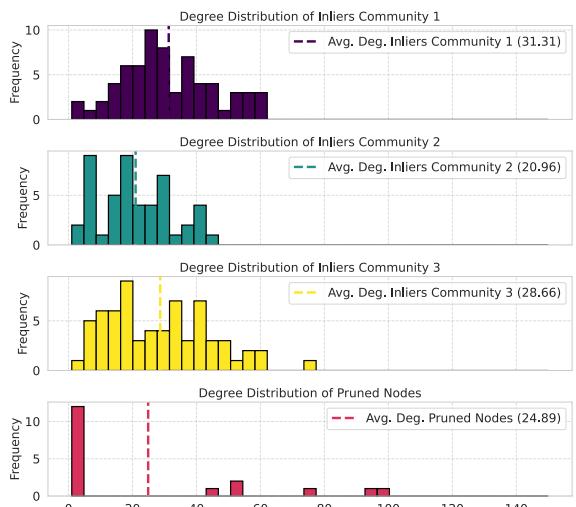
- Abbe, E., Boix-Adserà, E., Ralli, P., and Sandon, C. (2020). Graph powering and spectral robustness. *SIAM Journal on Mathematics of Data Science*, 2(1):132–157.
- Acharya, J., Jain, A., Kamath, G., Suresh, A. T., and Zhang, H. (2022). Robust estimation for random graphs. In Loh, P.-L. and Raginsky, M., editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 130–166. PMLR.
- Cai, T. T. and Li, X. (2015). Robust and computationally feasible community detection in the presence of arbitrary outlier nodes.
- Chen, H., Ding, J., Hua, Y., and Steurer, D. (2024). Private edge density estimation for random graphs: Optimal, efficient and robust. *arXiv preprint arXiv:2405.16663*.
- Delahaye, D., Chaimatanan, S., and Mongeau, M. (2019). Simulated annealing: From basics to applications. *Handbook of metaheuristics*, pages 1–35.
- Diakonikolas, I., Kamath, G., Kane, D., Li, J., Moitra, A., and Stewart, A. (2019). Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864.
- Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. (2021). Robustness meets algorithms. *Commun. ACM*, 64(5):107–115.
- Ding, J., d’Orsi, T., Hua, Y., and Steurer, D. (2023). Reaching Kesten-Stigum Threshold in the Stochastic Block Model under Node Corruptions. *arXiv:2305.10227 [cs, stat]*.



(a) Community partition of the jazz collaboration graph using our method (S.A.). It identifies outlier nodes to be ignored, allowing Spectral Clustering to find the three main collaboration groups (corresponding to bands in New York, Chicago, and other cities), something it had not been capable in Figure 1.



(b) Degree distribution of our method (S.A.).



(c) Degree distribution of pruning baseline.

Figure 4: Results for real data experiment.

- Gleiser, P. M. and Danon, L. (2003). Community structure in jazz. *Advances in complex systems*, 6(04):565–573.
- Henderson, D., Jacobson, S. H., and Johnson, A. W. (2003). The theory and practice of simulated annealing. *Handbook of metaheuristics*, pages 287–319.
- Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.
- Jana, S., Fan, J., and Kulkarni, S. (2024). A general theory for robust clustering via trimmed mean. *arXiv preprint arXiv:2401.05574*.
- Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 83(1):016107.
- Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.
- Lalwani, D., Somayajulu, D. V., and Krishna, P. R. (2015). A community driven social recommendation system. In *2015 IEEE International conference on big data (big data)*, pages 821–826. IEEE.
- Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215 – 237.
- Li, X., Chen, Y., and Xu, J. (2021). Convex relaxation methods for community detection.
- Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.
- Makarychev, K., Makarychev, Y., and Vijayaraghavan, A. (2016). Learning communities in the presence of errors. In Feldman, V., Rakhlin, A., and Shamir, O., editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1258–1291, Columbia University, New York, New York, USA. PMLR.
- Mitra, D., Romeo, F., and Sangiovanni-Vincentelli, A. (1986). Convergence and finite-time behavior of simulated annealing. *Advances in applied probability*, 18(3):747–771.
- Sangkaran, T., Abdullah, A., and Jhanjhi, N. (2020). Criminal community detection based on isomorphic subgraph analytics. *Open Computer Science*, 10(1):164–174.
- Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.
- Srivastava, P. R., Sarkar, P., and Hanasusanto, G. A. (2021). A Robust Spectral Clustering Algorithm for Sub-Gaussian Mixture Models with Outliers. *arXiv:1912.07546 [cs, math, stat]*.
- Stegehuis, C., Van Der Hofstad, R., and Van Leeuwaarden, J. S. (2016). Epidemic spreading on complex networks with community structures. *Scientific reports*, 6(1):29748.
- Stephan, L. and Massoulié, L. (2019). Robustness of spectral methods for community detection. In *Conference on Learning Theory*, pages 2831–2860. PMLR.
- Tabouy, T., Barbillon, P., and Chiquet, J. (2020). Variational inference for stochastic block models from sampled data. *Journal of the American Statistical Association*, 115(529):455–466.
- Van Laarhoven, P. J., Aarts, E. H., van Laarhoven, P. J., and Aarts, E. H. (1987). *Simulated annealing*. Springer.
- Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17:395–416.

Checklist

- For all models and algorithms presented, check if you include:
 - A clear description of the mathematical setting, assumptions, algorithm, and/or model. **[Yes]**
 - An analysis of the properties and complexity (time, space, sample size) of any algorithm. **[Yes]**
 - (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **[Yes]**
- For any theoretical claim, check if you include:
 - Statements of the full set of assumptions of all theoretical results. **[Yes]**
 - Complete proofs of all theoretical results. **[Yes]**
 - Clear explanations of any assumptions. **[Yes]**
- For all figures and tables that present empirical results, check if you include:
 - The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **[Yes]**
 - All the training details (e.g., data splits, hyperparameters, how they were chosen). **[Yes]**

- (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
- (a) Citations of the creator If your work uses existing assets. [Yes]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
- (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for SubSearch: Robust Estimation and Outlier Detection for Stochastic Block Models via Subgraph Search

1 PROOF OF THEOREM 3.1.

In this section we prove the bound appearing in Theorem 3.1 in the main paper. For the sake of completeness, let us briefly recall some of the notation (a detailed description can be found in Section 2 of the main paper). For any positive integer p , $\mathbf{1}_p$ denotes the vector of dimension p containing all ones. The matrix A is the adjacency matrix of a simple undirected graph with n nodes. For two subsets $S, S' \subset \{1, \dots, n\}$, $A_{S \times S'}$ is the $|S| \times |S'|$ matrix obtained by restricting A to the rows with indices in S and columns with indices in S' . In the case $S = S'$, we simply denote this restriction A_S . The connectivity parameters of an SBM with K communities $(\Omega_k)_{k=1, \dots, K}$ will be stored in a symmetric $K \times K$ matrix denoted Γ , and the latent community assignment of nodes are stored in a $n \times K$ matrix Z . We also need to introduce the $n \times n$ matrix $Q := Z\Gamma Z^t$. It holds that $\mathbb{E}[A] = Q - \text{diag}(Q)$, where $\text{diag}(Q)$ is the $n \times n$ matrix with the diagonal matching the diagonal of Q , and zero elsewhere. Given a subset of nodes S partitioned into K subsets S_1, \dots, S_K , we can estimate Γ as $\hat{\Gamma}_{kl} = |S_k|^{-1}|S_l|^{-1} \sum_{i \in S_k} \sum_{j \in S_l} A_{ij}$. This can be extended to an $n \times n$ matrix $\hat{Q}(S) := \mathbf{S}\hat{\Gamma}\mathbf{S}^t$, where \mathbf{S} is the $n \times K$ matrix such that $\mathbf{S}_{ik} = 1$ if $i \in S_k$ and zero otherwise. Finally, given a node adversary, we denote by F the set of inlier nodes, *i.e.* the nodes whose adjacencies were not arbitrarily modified by the adversary.

First note that for all $A \in \mathbb{R}^{m \times n}$, $S \subset \{1, \dots, m\}$, and $S' \subset \{1, \dots, n\}$, it holds that

$$\|A\| \geq \|A_{S \times S'}\|. \quad (1)$$

Now let us prove Theorem 3.1 By using the triangle inequality for the spectral norm and Equation (1), we have

$$\begin{aligned} & \|Q_{S \cap F} - \hat{Q}(S)_{S \cap F}\| \\ &= \|Q_{S \cap F} - A_{S \cap F} + A_{S \cap F} - \hat{Q}(S)_{S \cap F}\| \\ &\leq \|Q_{S \cap F} - A_{S \cap F}\| + \|\hat{Q}(S)_{S \cap F} - A_{S \cap F}\| \\ &\leq \|Q_F - A_F\| + \|\hat{Q}(S) - A_S\| \\ &\leq \|\text{diag}(Q)_F\| + \|\mathbb{E}[A]_F - A_F\| + \|\hat{Q}(S) - A_S\| \\ &\leq \|\text{diag}(Q)\| + \|\mathbb{E}[A]_F - A_F\| + \|\hat{Q}(S) - A_S\| \\ &= \max_{1 \leq k \leq K} \Gamma_{kk} + \|\mathbb{E}[A]_F - A_F\| + \|\hat{Q}(S) - A_S\|. \end{aligned}$$

On the other hand, Equation (1) implies, for all $k, l \in \{1, \dots, K\}$,

$$\|Q_{S \cap F} - \hat{Q}(S)_{S \cap F}\| \geq \|Q_{S_k \cap F \cap \Omega_k \times S_l \cap F \cap \Omega_l} - \hat{Q}(S)_{S_k \cap F \cap \Omega_k \times S_l \cap F \cap \Omega_l}\|.$$

Summing over k, l ,

$$\|Q_{S \cap F} - \hat{Q}(S)_{S \cap F}\| \geq \frac{1}{K^2} \sum_{k=1}^K \sum_{l=1}^K \|Q_{S_k \cap F \cap \Omega_k \times S_l \cap F \cap \Omega_l} - \hat{Q}(S)_{S_k \cap F \cap \Omega_k \times S_l \cap F \cap \Omega_l}\| \quad (2)$$

Notice that being in $S_k \cap F \cap \Omega_k$ (respectively $S_l \cap F \cap \Omega_l$) implies being in Ω_k (respectively Ω_l). This implies that for all $i \in \{1, \dots, |S_k \cap F \cap \Omega_k|\}, j \in \{1, \dots, |S_l \cap F \cap \Omega_l|\}$ we have

$$(Q_{S_k \cap F \cap \Omega_k \times S_l \cap F \cap \Omega_l})_{ij} = \Gamma_{kl}.$$

Similarly, being in $S_k \cap F \cap \Omega_k$ (respectively $S_l \cap F \cap \Omega_l$) implies being in S_k (respectively S_l). This implies that for all $i \in \{1, \dots, |S_k \cap F \cap \Omega_k|\}, j \in \{1, \dots, |S_l \cap F \cap \Omega_l|\}$ we have

$$(\hat{Q}_{S_k \cap F \cap \Omega_k \times S_l \cap F \cap \Omega_l})_{ij} = \hat{\Gamma}_{kl}.$$

This allows us to further simplify Equation 2:

$$\begin{aligned} \|Q_{S \cap F} - \hat{Q}(S)_{S \cap F}\| &\geq \frac{1}{K^2} \sum_{k=1}^K \sum_{l=1}^K \|(\Gamma_{kl} - \hat{\Gamma}_{kl}) \mathbf{1}_{|S_k \cap F \cap \Omega_k|} \mathbf{1}_{|S_l \cap F \cap \Omega_l|}^t\| \\ &= \frac{1}{K^2} \sum_{k=1}^K \sum_{l=1}^K |\Gamma_{kl} - \hat{\Gamma}_{kl}| \sqrt{|S_k \cap F \cap \Omega_k|} \sqrt{|S_l \cap F \cap \Omega_l|} \\ &\geq \frac{\min_{1 \leq k \leq K} |S_k \cap F \cap \Omega_k|}{K^2} \sum_{k=1}^K \sum_{l=1}^K |\Gamma_{kl} - \hat{\Gamma}_{kl}| \end{aligned}$$

Putting the lower and upper bound together, we arrive at

$$\sum_{k=1}^K \sum_{l=1}^K |\Gamma_{kl} - \hat{\Gamma}_{kl}| \leq \frac{K^2}{\min_{1 \leq k \leq K} |\Omega_k \cap S_k \cap F|} \left(\max_{1 \leq k \leq K} \Gamma_{kk} + \|\mathbb{E}[A]_F - A_F\| + \|\hat{Q}(S) - A_S\| \right),$$

which is what we wanted to show.

2 ADDITIONAL EXPERIMENTS

In this section, we provide additional experiments to enlighten the behavior of our method and compare it with other approaches. We start by providing new experiments on real data, then we move on to study the intrinsic variability of the algorithms used in the main paper, as well as the impact of other parameters on our method. In what follows, parameters not explicitly specified remain the same as they were in the main paper. We recall that the code used for all our experiments is available at https://github.com/leobianco/robust_estim_sbm/.

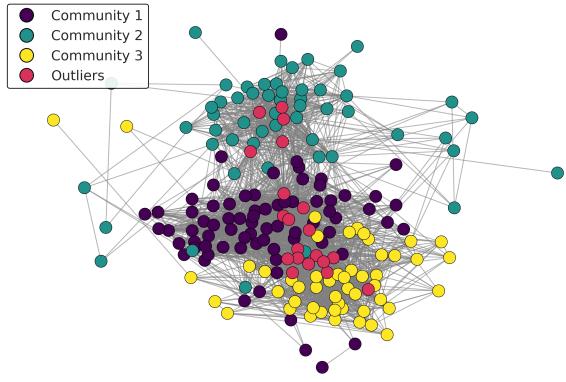
2.1 Experiments on Real Graphs

Degree-Corrected SBMs. In the experiments that follow, we consider the alternative presented in the Introduction, that of fitting a more complex model instead of using robust algorithms. In the case of networks, a popular generalization to the SBM introduced by [Karrer and Newman \(2011\)](#) is the Degree-Corrected SBM, or DC-SBM for short. It works by adding a parameter q_i to each node i such that the probability of nodes i and j connecting is given by $q_i q_j \Gamma_{z_i z_j}$. That is, besides the community effect, each node possesses an individual “tendency” to connect with others. For the model to be identifiable, a constraint on the q_i is required. We consider the same constraint used in [Gao et al. \(2018\)](#), namely $\sum_{i \in \Omega_k} q_i = |\Omega_k|$ for $k = 1, \dots, K$. We use the `graph_tool` library, developed by [Peixoto \(2014\)](#), to infer the community partition under a DC-SBM, where the number of communities K is assumed to be given and fixed. Notice that due to the constraint imposed on the q_i , it holds that

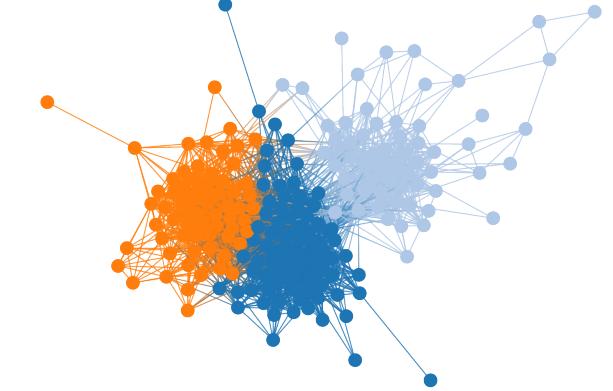
$$\mathbb{E} \left[\frac{\sum_{i \in \Omega_k} \sum_{j \in \Omega_l} A_{ij}}{|\Omega_k| |\Omega_l|} \right] = \frac{1}{|\Omega_k| |\Omega_l|} \Gamma_{kl} \sum_{i \in \Omega_k} \sum_{j \in \Omega_l} q_i q_j = \Gamma_{kl}.$$

Thus, we continue to use the empirical edge density to estimate the connectivity parameters given the estimated community partition, as before.

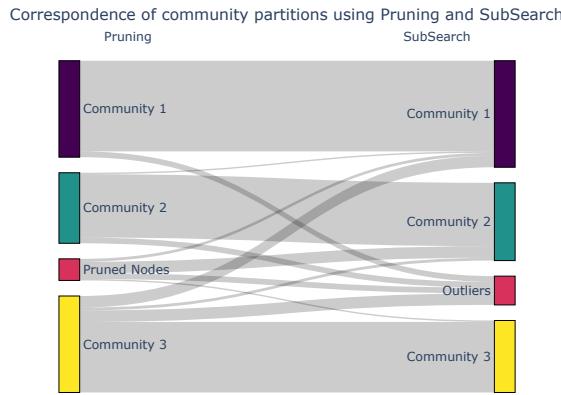
Jazz Collaboration Graph. We consider the jazz collaboration dataset introduced in [Gleiser and Danon \(2003\)](#), the one also presented in the main paper. Additionally to the pruning baseline presented in the main paper, we estimate the communities using a DC-SBM, then estimate the connectivity parameters. Figure 1b shows the graph partitions obtained, but perhaps a more useful representation is that in Figure 2b, which is a Sankey diagram. It works by showing the proportion of nodes in a community returned by the DC-SBM that stay in the same community, switch communities, or are identified as outliers by our method. We also present a



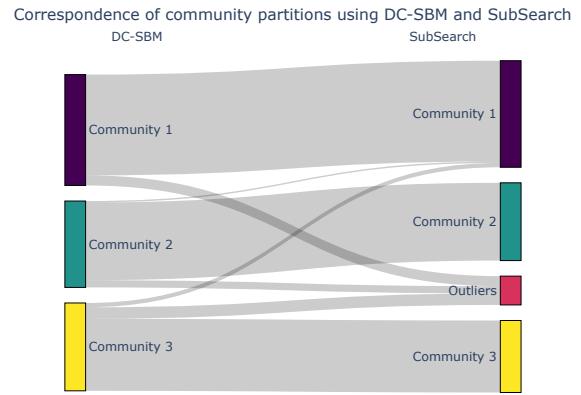
(a) Result of fitting our method (SUBSEARCH) to the Jazz collaboration dataset.



(b) Result of fitting a DC-SBM to the Jazz collaboration dataset using the `graph_tool` library of Peixoto (2014).



(a) Sankey diagram exhibiting the relation between the partition found for the jazz collaboration graph by pruning baseline (on the left) and our method (on the right).



(b) Sankey diagram exhibiting the relation between the partition found for the jazz collaboration graph by fitting a DC-SBM (on the left) and our method (on the right).

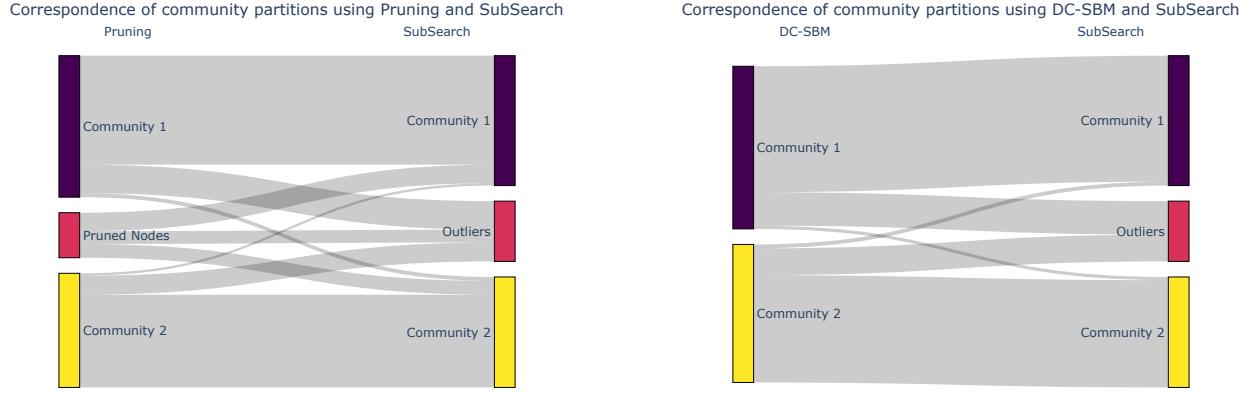
Sankey diagram for the pruning baseline in Figure 2a, and recall its estimated parameters here. The obtained estimated parameters are

$$\hat{\Gamma}_{\text{SubSearch}} = \begin{pmatrix} .328 & .008 & .068 \\ .008 & .337 & .017 \\ .068 & .017 & .351 \end{pmatrix}, \quad \hat{\Gamma}_{\text{Pruning}} = \begin{pmatrix} .345 & .007 & .063 \\ .007 & .342 & .017 \\ .063 & .017 & .332 \end{pmatrix}, \quad \hat{\Gamma}_{\text{DC-SBM}} = \begin{pmatrix} .361 & .006 & .086 \\ .006 & .293 & .028 \\ .086 & .028 & .337 \end{pmatrix}$$

We observe that the degree-correction introduced by the DC-SBM allows it to correctly partition the graph into communities, but the estimated parameters can be substantially different from ours due to all nodes being taken into account.

Political Blogs Graph. We consider the graph of political blogs introduced by Adamic and Glance (2005). Nodes correspond to blogs, labeled either “liberal” or “conservative”, and edges represent hyperlinks between them. We removed 268 isolated nodes, resulting in a graph of size $|G| = 1222$. Applying scikit’s standard spectral clustering algorithm leads to the result shown in Figure 4a. We see that it fails to correctly separate the two communities, despite using the normalized Laplacian to try to compensate for variations in degree.

Using our method, we search for a subgraph of size $|S| = 978$, corresponding to 80% of the size of the whole graph, and we use Markov chains of constant length `n_iters_inner=10` for each fixed temperature. The results



(a) Sankey diagram exhibiting the relation between the partition of the political blogs graph found by the pruning baseline (on the left) and our method (on the right).

(b) Sankey diagram exhibiting the relation between the partition of the political blogs graph found by fitting a DC-SBM (on the left) and our method (on the right).

are shown in Figures 4b and 5a. By observing the histogram of degree distributions in Figures 5a, we see that our method removes nodes across all ranges of degrees. In terms of runtime, this experiment took about 2 hours to run (in a 2015 laptop).

For comparison, the pruning baseline, which removes 244 nodes (20% of the graph), correctly clusters the graph into two communities. However, it disproportionately prunes low degree nodes—ten times more than our method—resulting in higher estimated parameters and more heterogeneous communities, as seen in the histogram in Figure 5b.

As argued in the main paper, the impact of high degree nodes to the quality of fit of an SBM is much higher than that due to nodes of lower degree, since they are farther from the mean degree of the inliers. This is taken into account by our method, and not by pruning. Therefore, pruning is possibly removing nodes from the graph that should not be erased.

Moreover, we also fit a DC-SBM to this graph. Figure 3b represents the correspondence between the partitions found by the DC-SBM and those found by our method. The estimated parameters for all methods were:

$$\hat{\Gamma}_{\text{SubSearch}} = \begin{pmatrix} .016 & .001 \\ .001 & .015 \end{pmatrix}, \quad \hat{\Gamma}_{\text{Pruning}} = \begin{pmatrix} .023 & .001 \\ .001 & .027 \end{pmatrix}, \quad \hat{\Gamma}_{\text{DC-SBM}} = \begin{pmatrix} .037 & .003 \\ .003 & .047 \end{pmatrix}$$

As with the jazz collaboration graph, this approach succeeds in clustering the graph correctly. But the fact that all nodes are included lead to considerably higher estimated connectivity parameter than those obtained with our method and pruning.

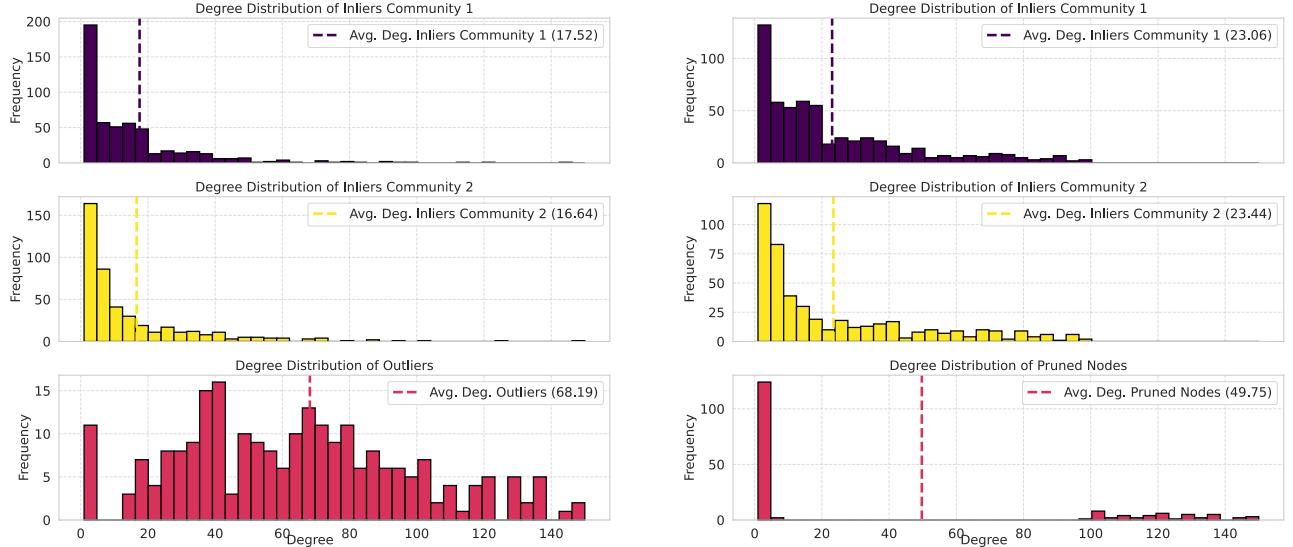
2.2 Impact of Algorithmic Variability

In the main paper, we conduct a multi-run experiment to study how the estimation error depends on the amount of perturbation. We pointed out the existence of two sources of variability on the estimation error. The first is due to the fact that distinct graphs have different difficulties for the clustering task, even when they are sampled from the same model. Another source of variability in the error is that which we call “algorithmic variability”. For instance, in the case of SUBSEARCH, we run it for a finite number of iterations and stop before its (asymptotic) convergence. Therefore, the fact that we randomly pick neighbors as we explore the state space leads to distinct solutions, even when running the algorithm on the same fixed graph. Notice that in theory (Proposition 3.2), once the algorithm converges to the global solution this would not make a difference. In the case of filtering, the algorithm possesses inherent randomness due to its sampling of nodes remove at each step.

In the main paper, to minimize the impact of algorithmic randomness and focus on the increasing difficulty of the



Figure 4: Results of different methods on the political blogs dataset.



(a) Edge density histogram obtained by using our method (S.A.) on the political blogs dataset.

(b) Edge density histogram obtained by using the pruning baseline on the political blogs dataset.

Figure 5: Histograms for the political blogs dataset.

graphs as perturbation grows, we ran each method on each graph three times (`runs_per_graph = 3`) and used only the result with the lowest cost in the average of the estimation error over all graphs. The choice of three runs was made due to the significant computational cost introduced as the number of runs per graph increases.

We now investigate the algorithmic randomness through two experiments. First, we fix a single graph and run our method 10 times to assess the variability of the estimation error. We obtain an average error of 0.0465 with a standard deviation of 0.0097. The 95% Student's t-confidence interval for this error is 0.0465 ± 0.007 . For comparison, for filtering, the same procedure on the same graph leads to a mean estimation error of 0.1773 and standard deviation of 0.0389, corresponding to a Student's t-confidence interval of 0.1773 ± 0.0278 .

A second way of seeing this variability appear is to replicate Figure 3(a) from the main paper, but now averaging the estimation error across all three runs. The results, shown in Figure 6, reveal significantly larger error bars for the filtering baseline for $\gamma = 0.2$ and $\gamma = 0.3$, while our method remains mostly unaffected throughout all values of γ . This suggests that our algorithm consistently finds solutions of similar quality and the impact of algorithmic variability on it is minimal, despite running for a limited number of iterations. The filtering baseline can be largely influenced by this algorithmic variability. We'd like to point out that the pruning baseline does not change, as it is deterministic on the input graph and thus all runs provide the same result.

2.3 Importance of other parameters

In this section, we analyze the impact of other parameters on the quality of the solutions found by our method. Specifically, we examine the effect of the Markov chain length at each fixed temperature and the size of the connectivity gap $p - q$.

Ideally, the Markov chains should be long enough to explore a significant portion of the neighborhood around the current state before the temperature is lowered. However, in practice, this is often infeasible because the size of the neighborhood can grow rapidly with the graph's size. In the experiments presented in the main paper, we used a Markov chain length equal to the number of outliers in the graph, which proved sufficient to find good solutions.

It's important to emphasize how crucial this parameter is to the solution quality. For instance, if the Markov chain is too short—such as setting the length to one iteration—the method behaves as shown in Figure 7, where it fails to converge to a good solution. In other words, short Markov chains make the algorithm greedy too soon, leading it into poor local solutions.

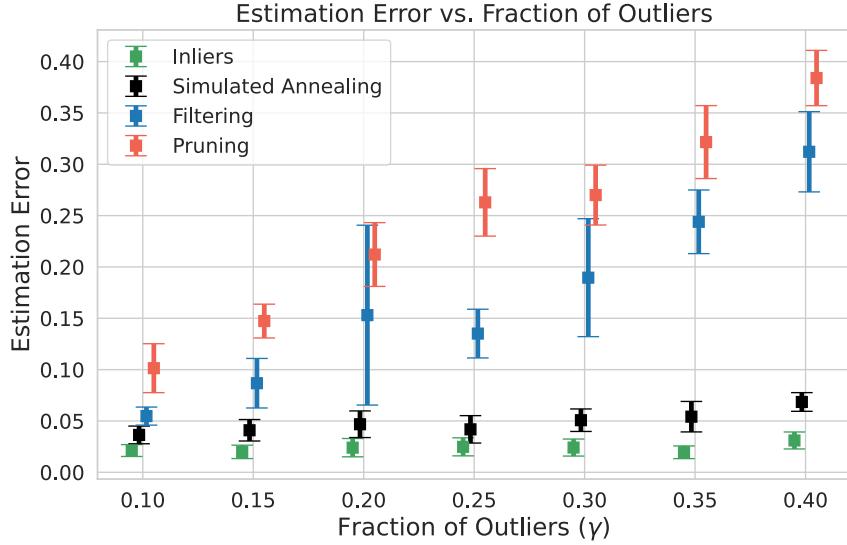


Figure 6: Error vs. perturbation amount multi-run experiment results when taking all runs into account.

Next, we examine the impact of the connectivity gap $p - q$ on estimation error. Specifically, we set $p = 0.5 + \varepsilon$ and $q = 0.5 - \varepsilon$, varying ε from 0.01 to 0.19 in increments of 0.03. For each ε , we simulate ten graphs and perform three runs per graph, selecting the run with the lowest norm for the final average estimation error. The perturbation level is fixed at $\gamma = 0.3$. The results are shown in Figure 8.

We observed no significant variation in the performance of any algorithm as a function of ε . This can be explained as follows: when ε is small, the clustering quality might be poor, but the similarity of the parameters still allows for a good estimation. Conversely, when ε is large, the clustering is likely to improve, and the estimation quality remains high. The challenge arises in the intermediate range of ε , where poor clustering could, in theory, lead to a higher estimation error due to the larger gap between true parameters. However, in our experiments, we did not observe a significant change in estimation error within this intermediate range.

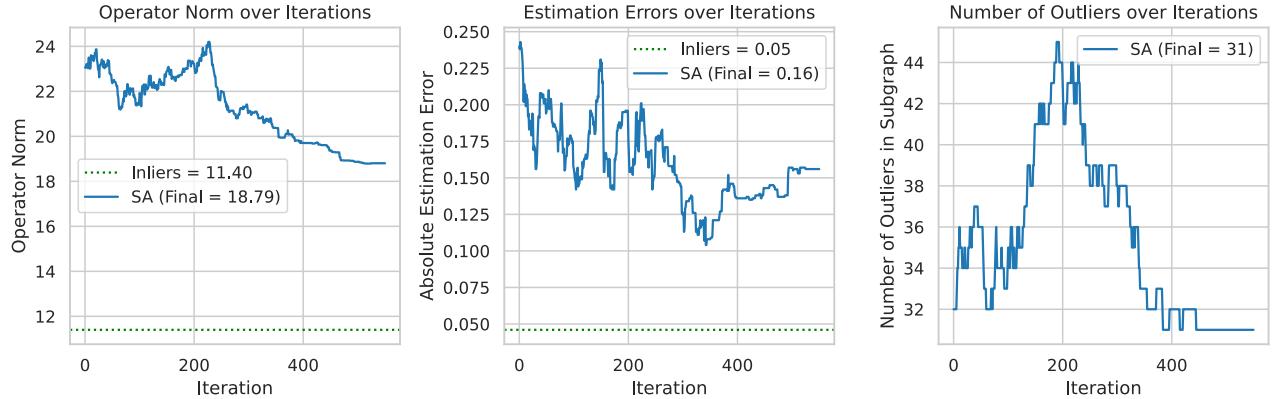


Figure 7: Results of our method using Markov chains of length one. The plot to the left shows the evolution of the cost function and compares it to the inlier baseline. The middle plot shows the evolution of the estimation error and compares it to the inlier baseline. The plot to the right shows the number of outliers inside the subgraph considered over the iterations. We see that the method fails to converge to a good solution.

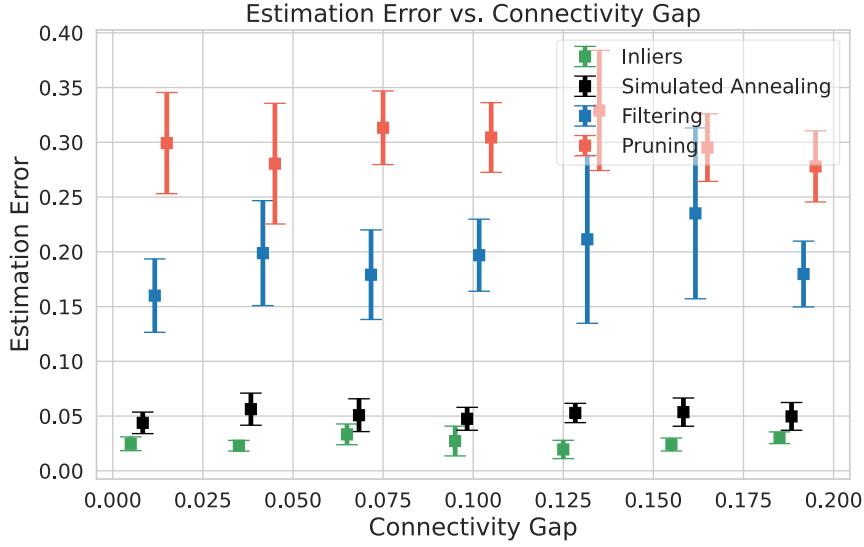


Figure 8: Multi-run experiment studying the dependence of the estimation error on the connectivity gap $p - q$.

References

- Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43.
- Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. (2018). Community detection in degree-corrected block models.
- Gleiser, P. M. and Danon, L. (2003). Community structure in jazz. *Advances in complex systems*, 6(04):565–573.
- Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 83(1):016107.
- Peixoto, T. P. (2014). The graph-tool python library. *figshare*.