This is the supplementary material for the article “SubSearch: SubSearch: Robust Estimation and Outlier Detection for Stochastic Block Models via Subgraph Search”, submitted to AISTATS 2025. It is organized as follows. We start with a proof to Theorem 3.1 of the main paper. Then, we move on the additional experiments on real data, a study of the impact of the algorithms' intrinsic variability, and a study of the impact of other parameters on the results and convergence. We would like to emphasize that our code used for all our experiments is openly available at https://anonymous.4open.science/r/robust_estim_sbm-8210/README.md.

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 $Q \coloneqq Z \Gamma Z^t$. It holds that $\mathbb{E}[A] = Q - \operatorname{diag}(Q)$, where $\operatorname{diag}(Q)$ 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 $\hat{Q}(S) \coloneqq \mathbf{S} \hat{\Gamma} \mathbf{S}^t$, 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

$$\|A\| \geq \|A_{S \times S'}\|. \tag{1}$$

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

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

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,

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

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:

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

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.

2.1 Experiments on Real Graphs

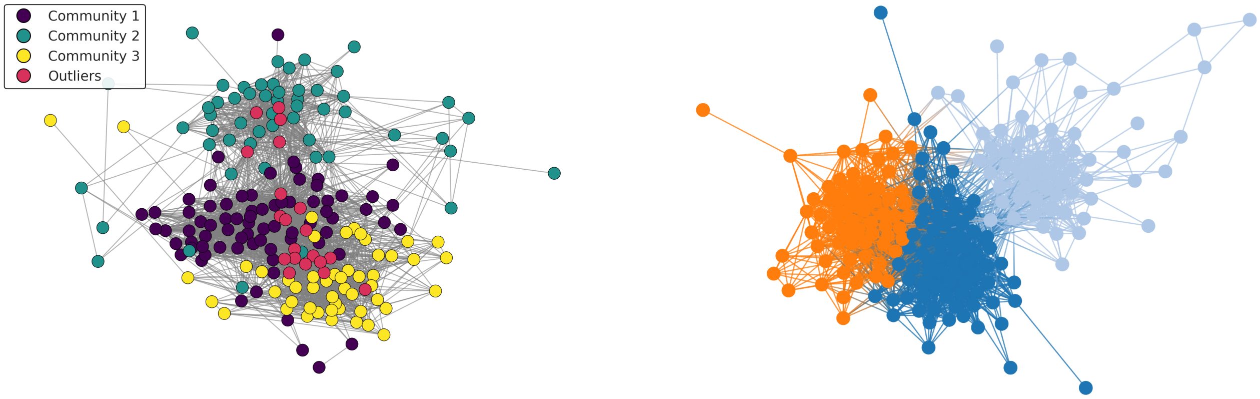
Degree-Corrected SBMs. In the experiments that follow, we consider the alternative presented in the Introduction, that of fitting a more complex model instead of using robust algorithms. In the case of networks, a popular generalization to the SBM is the Degree-Corrected SBM, or DC-SBM for short [Karrer and Newman, 2011]. 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 constraint $\sum_{i\in\Omega_k} q_i = |\Omega_k|$ for $k = 1, \ldots, K$ [Gao et al., 2018]. We use the `graph_tool` library [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.

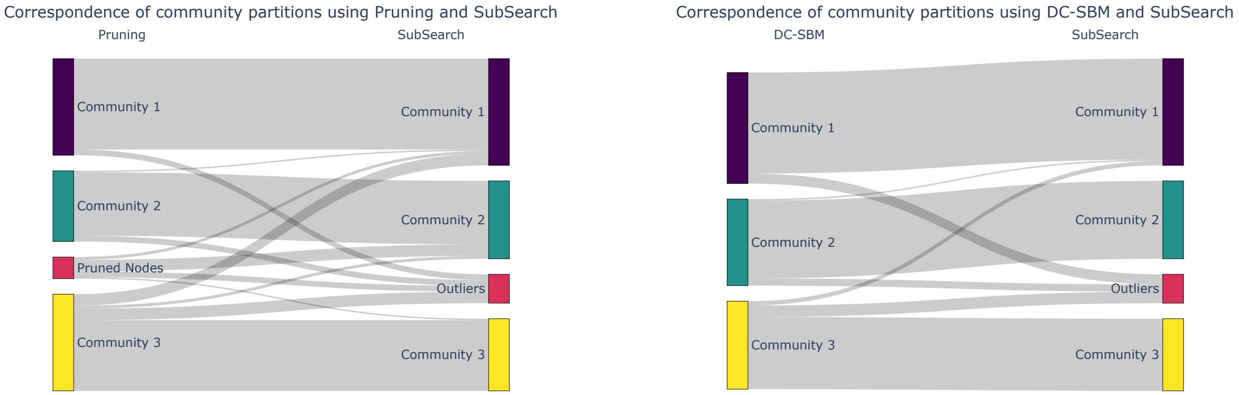
Jazz Collaboration Graph. We consider the jazz collaboration dataset [Gleiser and Danon, 2003], the one also presented in the main paper. Additionally to the pruning baseline presented in the main paper, we estimate the communities using a DC-SBM, then estimate the connectivity parameters. Figure 1b 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 Sankey diagram for the pruning baseline in Figure 2a, and recall its estimated parameters here. The obtained estimated parameters are

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



(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 `graph_tool` [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).

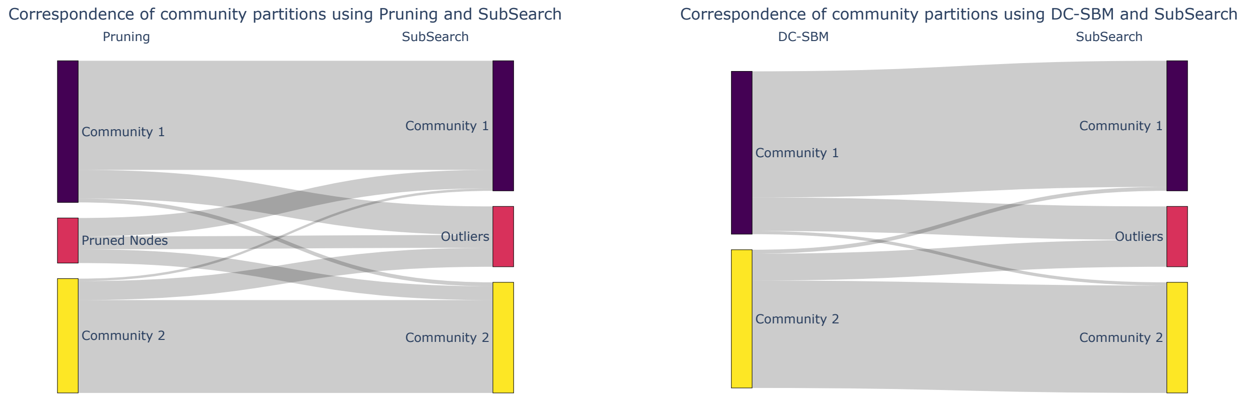
(b) Sankey diagram exhibiting the relation between the partition found for the jazz collaboration graph by fitting a DC-SBM (on the left) and our method (on the right).

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 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.

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.



(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).

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:

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

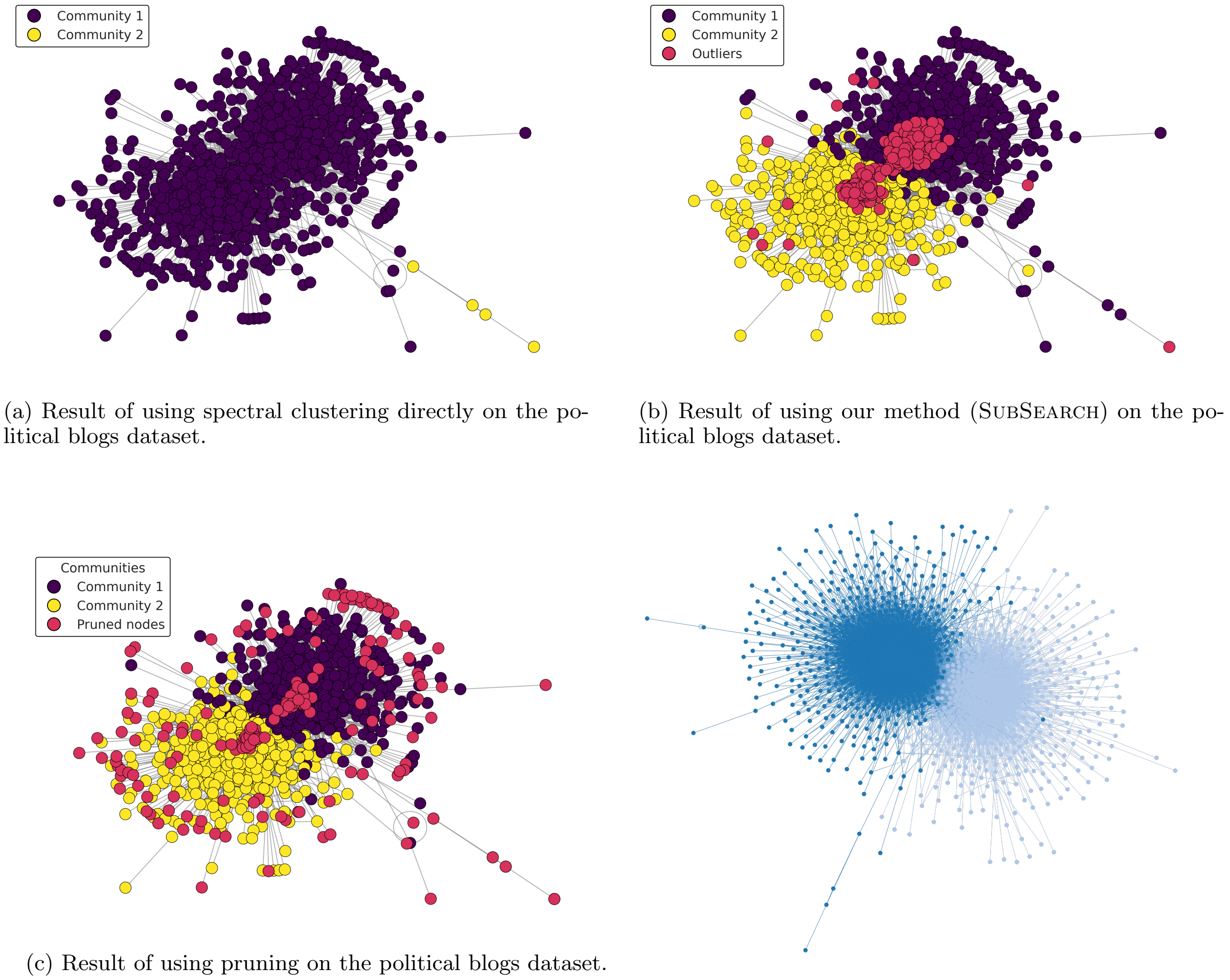
As with the jazz collaboration graph, this approach succeeds in clustering the graph correctly. But the fact that all nodes are included lead to considerably higher estimated connectivity parameter than those obtained with our method and pruning.

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 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.



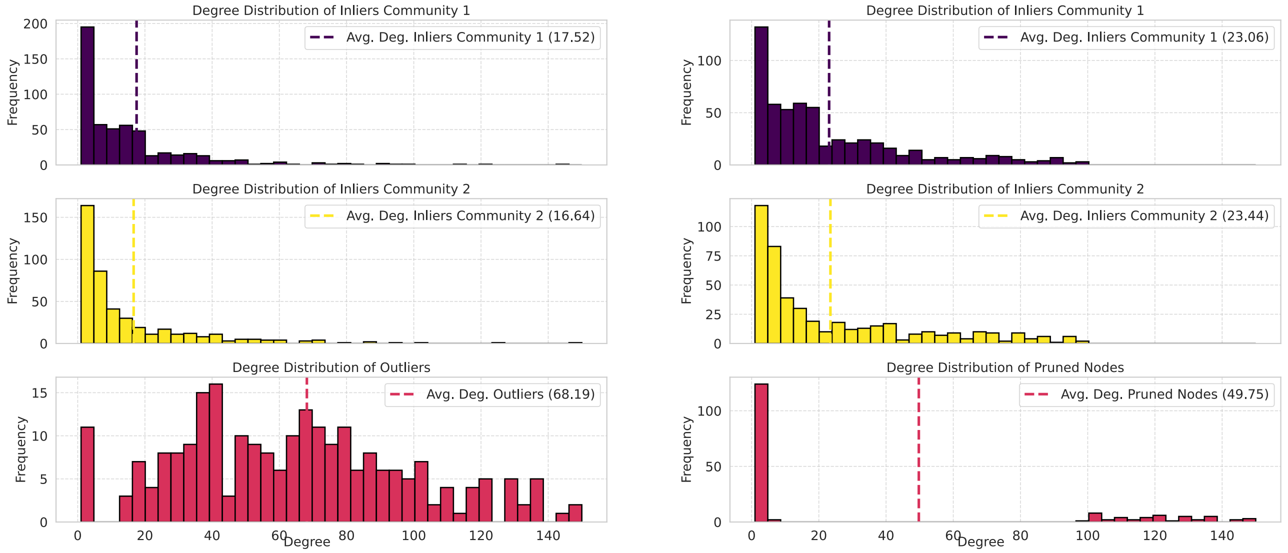
(a) Result of using spectral clustering directly on the political blogs dataset.

(b) Result of using our method (SUBSEARCH) on the political blogs dataset.

(c) Result of using pruning on the political blogs dataset.

(d) Result of fitting a DC-SBM to the political blogs collaboration dataset, using `graph_tool` [Peixoto, 2014].

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.

A second way of seing 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.

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

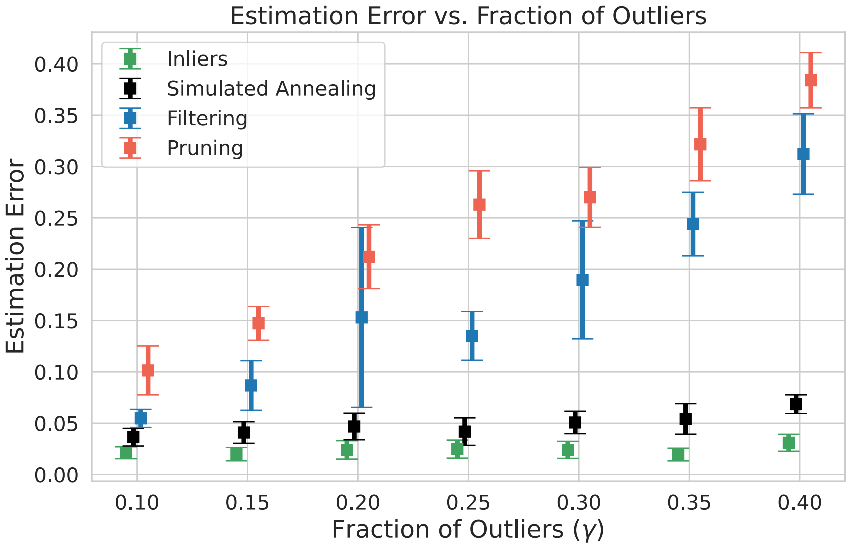


Figure 6: Error vs. perturbation amount multi-run experiment results when taking all runs into account.

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.

References

[Adamic and Glance, 2005] 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 et al., 2018] Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. (2018). Community detection in degree-corrected block models.

[Gleiser and Danon, 2003] Gleiser, P. M. and Danon, L. (2003). Community structure in jazz. *Advances in complex systems*, 6(04):565–573.

[Karrer and Newman, 2011] 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, 2014] Peixoto, T. P. (2014). The graph-tool python library. *figshare*.

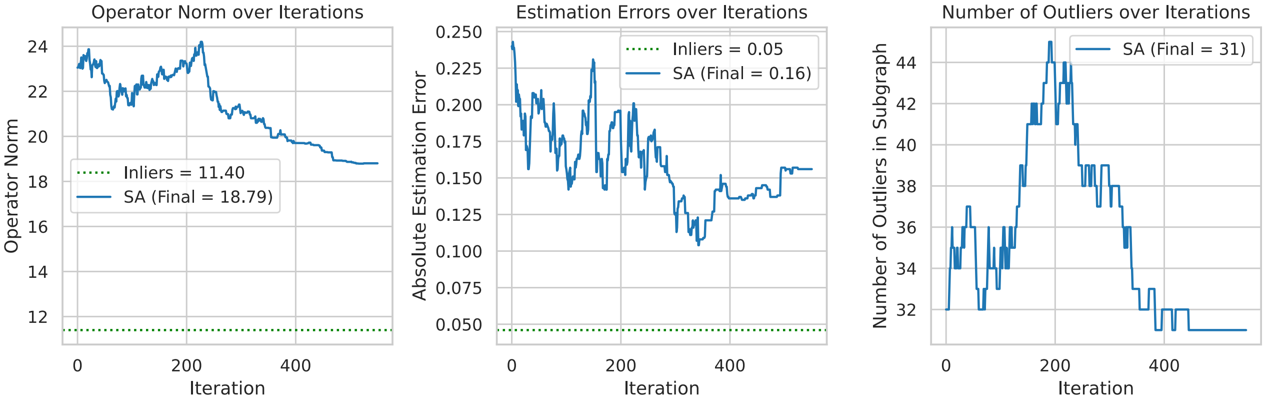


Figure 7: Results of our method using Markov chains of length one. The plot to the left shows the evolution of the cost function and compares it to the inlier baseline. The middle plot shows the evolution of the estimatior error and compares it to the inlier baseline. The plot to the right shows the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

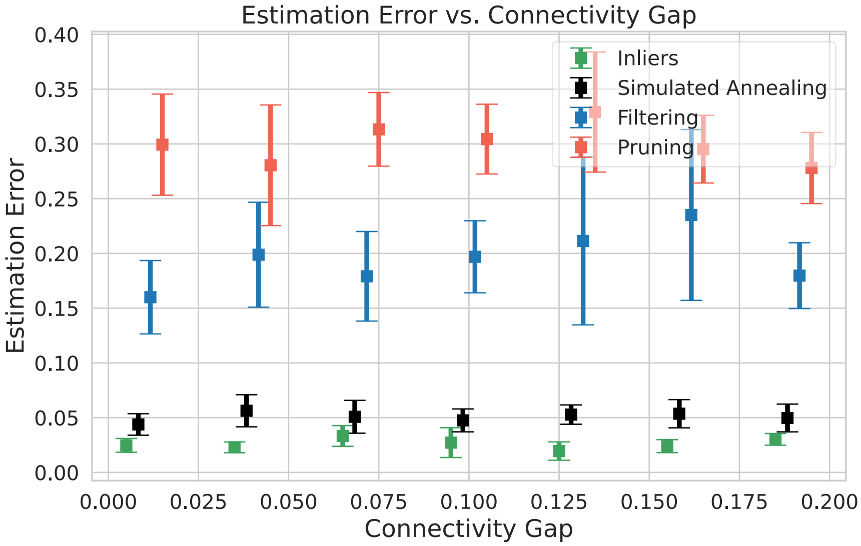


Figure 8: Multi-run experiment studying the dependence of the estimation error on the connectivity gap $p - q$.