Clustering with Queries under Semi-Random Noise

Alberto Del Pia DELPIA@WISC.EDU
Mingchen Ma MMA54@WISC.EDU
Christos Tzamos TZAMOS@WISC.EDU
University of Wisconsin-Madison

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

The seminal paper by Mazumdar and Saha (2017a) introduced an extensive line of work on clustering with noisy queries. Yet, despite significant progress on the problem, the proposed methods depend crucially on knowing the exact probabilities of errors of the underlying fully-random oracle. In this work, we develop robust learning methods that tolerate general semi-random noise obtaining qualitatively the same guarantees as the best possible methods in the fully-random model.

More specifically, given a set of n points with an unknown underlying partition, we are allowed to query pairs of points u, v to check if they are in the same cluster, but with probability p, the answer may be adversarially chosen. We show that information theoretically $O\left(\frac{nk\log n}{(1-2p)^2}\right)$ queries suffice to learn any cluster of sufficiently large size. Our main result is a computationally efficient algorithm that can identify large clusters with $O\left(\frac{nk\log n}{(1-2p)^2}\right)+\text{poly}\left(\log n, k, \frac{1}{1-2p}\right)$ queries, matching the guarantees of the best known algorithms in the fully-random model. As a corollary of our approach, we develop the first parameter-free algorithm for the fully-random model, answering an open question in Mazumdar and Saha (2017a).

Keywords: Clustering, Semi-random noise, Polynomial-time algorithm

1. Introduction

In a typical graph clustering problem, we are given a graph $G = (V, E)$ and we want to partition the vertices V into several clusters that satisfy certain properties. Clustering is ubiquitous in machine learning, theoretical computer science and statistics as this simple formulation has many applications in both theory and practice. Many theoretical problems can be formulated as clustering and it is a common NP-complete problem. Moreover, many practical problems where we want to put data or people that are similar together can be viewed as clustering, like record linkage and entity resolution (Fellegi and Sunter, 1969) in databases, or community detection in social networks.

However, in many applications, one may not have direct access to the full graph, and it may be costly to query the relationship between two nodes. For example, in entity resolution where the goal is to find records in a database that refer to the same underlying entity, it is common to use crowdsourcing to develop human in the loop systems for labeling the edges (Green Larsen et al., 2020; Karger et al., 2011; Wang et al., 2012; Dalvi et al., 2013; Gokhale et al., 2014; Vesdapunt et al., 2014; Mazumdar and Saha, 2017b). Asking humans requires effort, time, and money, so one would want to cluster the items efficiently without asking workers to compare every pair of items.

Motivated by these applications, the work of Mazumdar and Saha (2017a) introduced a theoretical model of clustering with queries. In this model, we don't have access to the edges of the graph in advance but may query a similarity oracle that may not always give the correct answer. The problem is defined as follows.

Definition 1 *(Clustering with a faulty oracle) We are given a set of points $V = [n]$, which contains k latent clusters $V_i^* \subseteq V$ for $i \in [k]$ such that $\bigcup_{i\in[k]} V_i^* = V$ and for every $1 \leq i < j \leq k$, $V_i^* \cap V_j^* = \emptyset$. For every pair of points $u, v \in V$, the edge (u, v) is labeled by 1, if u, v are in the same cluster, and is labeled by 0, if u, v are in different clusters. The number k and the clusters V_i^* for $i \in [k]$ are unknown. We are given an oracle $\mathcal{O} : V \times V \to \{0, 1\}$ to query point pairs of V. Every time the oracle $\mathcal{O}$ is invoked it takes a pair of points u, v as an input and outputs a label for the edge (u, v) which may be erroneous. Our goal is to recover the latent clustering with high probability, minimizing the queries to the oracle $\mathcal{O}$.*

While the above formulation of Definition 1 does not specify how the errors are introduced by the oracle, the work of Mazumdar and Saha (2017a) focused specifically on a fully-random faulty oracle that gives incorrect answers with a fixed probability of error $p \in [0, 1/2)$ known in advance.

Definition 2 *(Fully-random model of clustering with a faulty oracle) Under the fully-random model, the oracle $\mathcal{O}$ of Definition 1 behaves as follows. There is a known error parameter $p \in [0, 1/2)$ such that, for every point pair u, v, $\mathcal{O}(u, v)$ outputs the true label of (u, v) with probability $1 - p$ and outputs the wrong label of (u, v) with probability p.*

Mazumdar and Saha (2017a) showed that in this model $\Omega\left(\frac{nk}{(1-2p)^2}\right)$ queries are necessarily needed to recover the underlying clustering while $O\left(\frac{nk\log n}{(1-2p)^2}\right)$ queries suffice to learn any large enough cluster. They also designed a computationally-efficient algorithm with query complexity $O\left(\frac{nk^2\log n}{(1-2p)^4}\right)$ to recover large clusters. Since then, follow-up papers Green Larsen et al. (2020); Peng and Zhang (2021) extended their results and obtained efficient algorithms with lower query complexity $O\left(\frac{nk\log n}{(1-2p)^2}\right) + \text{poly}\left(\log n, k, \frac{1}{1-2p}\right)$.

While these works pin down the query complexity of the problem in the fully-random model, they crucially rely both on the fully-random model and the exact knowledge of the error probability parameter p. In fact, Mazumdar and Saha (2017a) posed as an open problem whether one can design a parameter-free algorithm with the same guarantees.

Motivated by these shortcomings of the fully-random model, our goal in this work is to obtain more robust algorithms that work beyond the fully-random model and do not rely on the knowledge of the error probabilities. Removing these crucial assumptions will enable broader applicability of the algorithms as in practice, the amount of noise may depend on the particular nodes being compared and may vary significantly from query to query making it impossible to know it or predict it in advance.

Our work focuses on a significantly more general semi-random model that allows the oracle answers to be given adversarially with some probability.

Definition 3 *(Semi-random model of clustering with a faulty oracle) Under Definition 1, the oracle $\mathcal{O}$ is defined in the following way. There is a known error parameter $p \in (0, 1/2)$ such that, for every point pair u, v, with probability $1 - p$, $\mathcal{O}(u, v)$ outputs the true label of (u, v) and with probability p, (u, v) is corrupted and $\mathcal{O}(u, v)$ outputs an arbitrary label given by an adversary, depending on the sequential output of $\mathcal{O}$ and the latent clustering.*

An important special case of the semi-random model corresponds to the case where every edge (u, v) has a fixed and unknown probability of error $p_{(u,v)}$ chosen in advance that is upper bounded

by the known bound p. We refer to this case as *non-adaptive semi-random noise* and note that the more general Definition 3 can handle more adaptive instantiations of noise where the answers of the oracle may depend on the answers given in advance.

The main contribution of our work is the design of novel algorithms that can tolerate semi-random noise essentially matching the guarantees obtained for the fully-random model. Before formally presenting our results, we give an overview of the existing methods and guarantees for the fully-random model.

1.1. Prior work on the fully-random model

In Table 1, we summarize previous results, as well as our main results, for the fully-random model and the semi-random model.

Model	Query Complexity	Reference	Remark
Fully-random	$\Omega\left(\frac{nk}{(1-2p)^2}\right)$	Mazumdar and Saha (2017a)	Lower bound
	$O\left(\frac{nk\log n}{(1-2p)^2}\right)$	Mazumdar and Saha (2017a)	Time inefficient
	$O\left(\frac{nk^2\log n}{(1-2p)^4}\right)$	Mazumdar and Saha (2017a)	
	$O\left(\frac{n\log n}{(1-2p)^2}+\frac{\log^2 n}{(1-2p)^6}\right)$	Green Larsen et al. (2020)	$k=2$
	$O\left(\frac{nk\log n}{(1-2p)^2}+\frac{k^4\log^2 n}{(1-2p)^4}\right)$	Peng and Zhang (2021)	Nearly balanced
	$O\left(\frac{nk\log n}{(1-2p)^2}+\frac{k^{10}\log^2 n}{(1-2p)^4}\right)$	Peng and Zhang (2021)	k is known
	$O\left(\frac{nk\log n}{(1-2p)^2}+\frac{k^9\log k\log^2 n}{(1-2p)^{12}}\right)$	Theorem 6	Parameter-free
Semi-random	$O\left(\frac{nk\log n}{(1-2p)^2}\right)$	Theorem 4	Time inefficient
	$O\left(\frac{nk\log n}{(1-2p)^2}+\frac{k^9\log k\log^2 n}{(1-2p)^{12}}\right)$	Theorem 5	
	$O\left(\frac{n\log n}{(1-2p)^2}+\frac{\log^2 n}{(1-2p)^6}\right)$	Theorem 15	$k=2$

Table 1: Query complexity of algorithms under the fully-random and the semi-random model

Previous work that studied the query complexity of the clustering problem focused on the fully-random model. Mazumdar and Saha (2017a) gave an information theoretical algorithm that can recover all clusters of size $\Omega\left(\frac{\log n}{(1-2p)^2}\right)$ with query complexity $O\left(\frac{nk\log n}{(1-2p)^2}\right)$, which matches the information theoretical lower bound of $\Omega\left(\frac{nk}{(1-2p)^2}\right)$ in the same paper within a $O(\log n)$ factor. They also designed an efficient algorithm that can recover all clusters of size at least $\Omega(\frac{k\log n}{(1-2p)^4})$ with query complexity $O(\frac{nk^2\log n}{(1-2p)^4})$. Followed by their work, Green Larsen et al. (2020) gave an efficient algorithm with an improved query complexity $O\left(\frac{n\log n}{(1-2p)^2}+\frac{\log^2 n}{(1-2p)^6}\right)$. More recently, Peng and Zhang (2021) designed an efficient algorithm that recovers all clusters of size $\Omega\left(\frac{k^4\log n}{(1-2p)^2}\right)$ with query complexity $O\left(\frac{nk\log n}{(1-2p)^2}+\frac{k^{10}\log^2 n}{(1-2p)^4}\right)$ for known k. For every constant k, their query complexity matches the information lower bound within a $O(\log n)$ factor. Their algorithm can

even exactly recover the underlying clustering with query complexity $O\left(\frac{nk\log n}{(1-2p)^2}+\frac{k^4\log^2 n}{(1-2p)^4}\right)$ if each underlying cluster has size $\Omega\left(\frac{n}{k}\right)$.

1.2. Our contributions

We now present our contributions for the semi-random model in more detail.

An information-theoretically tight algorithm We first give an information theoretical algorithm for the problem presented in Section 3.

Theorem 4 *There is an algorithm* ESTIMATION(V,p) *such that under the semi-random model,* ESTIMATION(V,p) *has query complexity* $O\left(\frac{nk\log n}{(1-2p)^2}\right)$ *and recovers all clusters of size at least* $\Omega\left(\frac{\log n}{(1-2p)^2}\right)$ *with probability at least* $1-1/poly(n)$.

Theorem 4 shows even under the semi-random model, $O\left(\frac{nk\log n}{(1-2p)^2}\right)$ queries suffice to learn all clusters of size $\Omega\left(\frac{\log n}{(1-2p)^2}\right)$. This matches the performance of the information theoretical algorithm proposed in Mazumdar and Saha (2017a). Furthermore, since the fully-random model is a special case of our semi-random model and the information theoretical lower bound for the fully-random model is $\Omega\left(\frac{nk}{(1-2p)^2}\right)$, our query complexity matches the information theoretical lower bound within a $O(\log n)$ factor.

While Theorem 4 gives a nearly-tight information theoretical bound for the problem, the underlying algorithm is not computationally efficient. This is expected as there is a conjectured computational-statistical gap even in the case of fully-random noise (Peng and Zhang, 2021).

A computationally efficient algorithm We next turn to the question of what can be achieved using a computationally efficient algorithm. We obtain the following performance guarantee.

Theorem 5 *There is an algorithm* CLUSTERING(V,p)*, such that under the semi-random model, with probability at least* $1-1/poly(n)$*,* CLUSTERING(V,p) *recovers all* V_i^**, such that* $|V_i^*|=\Omega\left(\frac{k^4\log n}{(1-2p)^6}\right)$ *in polynomial time. Furthermore, the query complexity of* CLUSTERING(V,p) *is* $O\left(\frac{nk\log n}{(1-2p)^2}+\frac{k^9\log k\log^2 n}{(1-2p)^{12}}\right)$.

Our algorithm, presented in Section 4, can recover all large clusters under the semi-random model with a query complexity of $O\left(\frac{nk\log n}{(1-2p)^2}\right)+\text{poly}(k,1/(1-2p),\log n)$. This bound qualitatively matches the best known bound from Peng and Zhang (2021) for the fully-random model, and even achieves a slightly better dependence on k. We note that a bound of $\Omega\left(\frac{nk}{(1-2p)^2}\right)+\text{poly}(k,1/(1-2p))$ is conjectured by Peng and Zhang (2021) to be necessary for computationally efficient estimation even in the fully-random model.

A parameter-free algorithm for the fully-random model As a corollary of our approach, we design the first efficient parameter-free algorithm under the fully-random model whose performance is given by the following theorem and solves the open question given by Mazumdar and Saha (2017a).

Theorem 6 *Under the fully-random model, there is a parameter-free algorithm such that with probability at least* $1 - 1/poly(n)$*, recovers all clusters of size at least* $\Omega\left(\frac{k^4 \log n}{(1-2p)^6}\right)$*. Furthermore, the query complexity of the algorithm is* $O\left(\frac{nk \log n}{(1-2p)^2} + \frac{k^9 \log k \log^2 n}{(1-2p)^{12}}\right)$.

1.3. Technical overview

The main approach in developing algorithms with low query complexity is to first identify a small, but large enough, subset of vertices B that mostly come from the same cluster V_i^* and then compare all vertices in the graph to the vertices of B to fully identify the whole cluster V_i^* with high probability. Such a set B is called biased, and is computed by first subsampling a subgraph T of the whole graph and solving a clustering problem in the subgraph. Then, once we identify a cluster V_i^*, we can repeat the process to recover the remaining clusters as well. This is a common technique of the prior work (Mazumdar and Saha, 2017a; Green Larsen et al., 2020; Peng and Zhang, 2021) as well as our work. The main challenge which leads to the difference between the methods is how one can arrive at such a biased set.

To get an information theoretical algorithm, Mazumdar and Saha (2017a) found the largest subcluster of T by computing the heaviest subgraph of T. However, as we show in Appendix C.3, this method fails under the semi-random model even if we have two clusters. To get an efficient algorithm, Peng and Zhang (2021) did this by filtering small subclusters of T via counting degree of each vertex and running an algorithm proposed by Vu (2018) for a community detection problem under the Stochastic Block Model, which highly depends on the fully-random noise. On the other hand, Mazumdar and Saha (2017a); Green Larsen et al. (2020) used a simple disagreement counting method to cluster the subgraph T. While this simple technique is again very tailored to the fully-random model, we can extend this to the semi-random model but only in a very special case. We obtain an algorithm for semi-random noise where there are $k = 2$ clusters and the noise is non-adaptive (see Theorem 15 in Appendix B.1). As we show, this technique breaks down completely once any of these two restricting assumptions are removed. In general, previous efficient algorithms on fully random models can fail easily under semi-random models, because they all use techniques such as counting disagreements or counting degrees locally to obtain information from a single vertex or a pair of vertices. These statistics highly depend on the exact knowledge of the noise rate and thus under the semi-random model, an adversary can easily make the algorithms fail. A detailed discussion can be found in Appendix B. To obtain more robust efficient algorithms under the semi-random model, a key challenge is to design a statistic that can obtain information from a larger neighborhood of vertices and can be computed efficiently.

Our Approach To obtain robust algorithms for clustering under more than 2 clusters and more general semi-random noise we require a more involved clustering procedure for the subsampled graph T which we carefully choose.

For our information theoretical algorithm, our method computes the largest subset of T that has no negative cut (assuming edges that are labeled 0 contribute as -1). As we show, such a set must correspond a set of vertices all coming from the same cluster in the underlying partition, provided that T is large enough.

As this step is computationally intractable, to obtain a computationally-efficient algorithm, our method relies on efficiently computing an (approximate) correlation clustering of T. Our key observation is that when T is large enough, every clustering that has a small cost must be close to the

underlying clustering and must have a special structure. To make this more specific, such a clustering function must contain some very large cluster and each of these large clusters must be biased to contain a majority of points from the same true cluster. This implies if we can compute a correlation clustering $\tilde{T}$ of T then we can use those large clusters in $\tilde{T}$ to recover the corresponding underlying clusters.

To obtain an approximation to the correlation clustering of the sample set T, we rely on an approximation algorithm developed by Mathieu and Schudy (2010); Ailon et al. (2008) that obtains an SDP relaxation of the clustering problem and then performs a rounding step. We show that the resulting clustering that the algorithm obtains has a sufficiently small an additive error $O\left(\frac{|T|^{3/2}}{(1-2p)}\right)$ that enables us to identify heavily biased clusters efficiently. By carefully choosing the size of T, we show that with high probability, the clustering we obtain must contain at least one big cluster, which is a biased set for a true cluster V_i^* for $i \in [k]$.

1.4. Further related work

There has been a lot of work in developing algorithms for clustering. A lot of research has focused specifically on clustering under random graphs. Typical problems include community detection under stochastic block models (SBM) (Abbe, 2017) and clique detection under planted clique models (Alon et al., 1998). In these problems, a hidden structure such as a clustering or a clique is planted in advance, a random graph is generated according to some distribution and we are asked to recover the hidden structure efficiently using the given random graph.

Another popular clustering problem is correlation clustering, which was proposed in Bansal et al. (2004). In this problem, we are given an undirected graph G and our goal is to partition the vertices into clusters so that we minimize the number of disagreements or maximize the number of agreements. As the correlation clustering problem is NP-hard and many works develop efficient approximation algorithms (Bansal et al., 2004; Demaine and Immorlica, 2003; Giotis and Guruswami, 2006; Swamy, 2004; Charikar et al., 2005; Ailon and Karnin, 2012; Makarychev et al., 2015; Mathieu and Schudy, 2010) for worst case instances, while others (Shamir and Tsur, 2007; Joachims and Hopcroft, 2005) focus on the average case complexity of clustering when the graph is generated according to some underlying distribution.

A popular application of clustering is the signed edge prediction problem (Leskovec et al., 2010; Burke and Kraut, 2008; Brzozowski et al., 2008; Chen et al., 2014). In this problem, we are given a social network, where each edge is labeled by '+' or '-' to indicate if two nodes have positive relations or negative relations. The goal here is to use a small amount of information to recover the sign of the edges, which implies we want to reconstruct the network by partial information.

Besides the large body of work on clustering problems with access to the full graph, recently other papers studied clustering problems with queries under different settings. (Ashtiani et al., 2016; Gamlath et al., 2018) study the k-means problem with same-cluster queries. (Saha and Subramanian, 2019; Ailon et al., 2018) study the correlation clustering problem with same-cluster queries. Some other recent works on clustering with queries include (Huleihel et al., 2019; Li et al., 2021; Bressan et al., 2020).

Beyond clustering, there are also other settings in learning theory where semi-random noise makes the problem significantly more challenging and requires more sophisticated algorithms than the corresponding fully-random case. Semi-random noise corresponds to the popular Massart noise model (Massart and Nédélec, 2006) in the context of robust classification. While classification

under fully-random noise was known for many years (Blum et al., 1998), robust learning methods that can tolerate Massart noise were only recently discovered (Diakonikolas et al., 2019; Chen et al., 2020).

2. Preliminaries and notation

Let $V = [n]$ be a set of points, which contains k *underlying clusters* $V_i^* \subseteq V$, for $i \in [k]$, such that $\bigcup_{i\in[k]} V_i^* = V$ and $V_i^* \cap V_j^* = \emptyset$, for every $1 \leq i < j \leq k$. We say a set $S \subseteq V$ is a *subcluster* if $S \subseteq V_i^*$ for some $i \in [k]$. We say $\tilde{V} : V \times V \to \{0, 1\}$ is a *clustering function* over V based on $\{\tilde{V}_1, \ldots, \tilde{V}_t\}$, if $\{\tilde{V}_1, \ldots, \tilde{V}_t\}$ is a partition of V and, for every $(u, v) \in V \times V$,

$$\tilde{V}(u,v) = \begin{cases} 1 \text{ if } \exists i = j, \text{s.t.} u \in \tilde{V}_i, v \in \tilde{V}_j, \\ 0 \text{ if } \exists i \neq j, \text{s.t.} u \in \tilde{V}_i, v \in \tilde{V}_j. \end{cases}$$

In particular, throughout the paper, we denote by V^* the clustering function over V based on the underlying clusters and we denote by $\bar{V}$ the binary function over $V \times V$ corresponding to a realization of $\mathcal{O}$ over all point pairs of V. Given a binary function $F : V \times V \to \{0, 1\}$, the *adjacency matrix* of F is the matrix $M(F) \in \{0, 1\}^{|V|\times|V|}$, such that $M(F)_{uv} = F(u, v)$ for every $u, v \in V$. For convenience, when it does not create confusion, we use the same notation for a clustering function, the set of clusters it is based on, and its adjacency matrix.

Given $A, B \in \mathbb{R}^{n\times n}$, we define the *distance* between A, B to be $d(A, B) := \sum_{1\leq i\leq j\leq n} |A_{ij} - B_{ij}|$. Let F, H be two binary functions over $V \times V$. We define the *distance* between F, H to be

$$d(F, H) := d(M(F), M(H)) = \sum_{1\leq u\leq v\leq |V|} |F(u, v) - H(u, v)|.$$

Given a binary function E over $V \times V$, a *correlation clustering* $\tilde{V}$ of E is a clustering of V that minimizes $d(V', E)$ among all clustering V' of V.

Next, we introduce two definitions that will be heavily used throughout the paper.

Definition 7 *Let $\eta \in (0, 1/2]$ and $C \subseteq V$. A subset B of V is called an* (η, C)-biased set *if*

$$|B \cap C| \geq (\frac{1}{2} + \eta)|B|.$$

Intuitively, an (η, C)-biased set is a set whose majority of points come from C. On the other hand, if a set does not contain a significant fraction of points that come from an underlying cluster, we call it an η-bad set. Formally, we have the following definition.

Definition 8 *Let $\eta \in (0, 1/2]$. A subset B of V is called an* η-bad set *if for every $i \in [k]$, B is not an (η, V_i^*)-biased set.*

The importance of Definition 7 is that, under the semi-random model, we can recover an underlying cluster V_i^* from an (η, V_i^*)-biased set using the following simple procedure, which has been proposed in Ben-Dor et al. (1999); Mazumdar and Saha (2017a); Green Larsen et al. (2020); Peng and Zhang (2021).

Algorithm 1 DEGREETEST(v, B) (Test if $v \in V_i^*$ using an (η, V_i^*)-biased set B)

if $S = \sum_{u \in B} \mathcal{O}(u, v) \geq |B|/2$ **then return** "Yes" **else return** "No"

The intuition behind Algorithm 1 is that if more than half of the points of B come from V_i^*, then we can use B to distinguish if a point v is in V_i^* or not, by looking at the query results. According to Peng and Zhang (2021), for every constant $\eta \in (0, 1/2]$, we can use an (η, V_i^*)-biased set B of size $\Omega(\frac{\log n}{\eta^2(1-2p)^2})$ to recover V_i^* via DEGREETEST(v, B) with high probability under the fully-random model. However, under the semi-random model, to recover V_i^* using B, B needs to be sufficient large and biased so that it can be used to find the corresponding true cluster. To state this formally, we have the following Lemma 9. We leave the proof to Appendix A.

Lemma 9 *Under the semi-random model, let $\eta \in (p, 1/2]$ and let B be an (η, V_i^*)-biased set for some $i \in [k]$. If $|B| \geq \max\{\frac{80 \log n}{\eta^2(1-2p)^2}, \frac{5 \log n}{(\eta - p)^2}\}$, then with probability $1 - 1/poly(n)$, for every $v \in V$,* DEGREETEST(v, B) *returns "Yes" if $v \in V_i^*$, and it returns "No" if $v \notin V_i^*$.*

3. Information theoretical algorithm

Before designing efficient algorithms, we first need to figure out how many queries are needed in order to recover the underlying clusters under the semi-random model. In this section, we answer this question formally and we propose an information theoretical algorithm. Our algorithm has a structure similar to the information theoretical algorithm in Mazumdar and Saha (2017a), but we use a different statistic to overcome the semi-random noise. In particular, our algorithm can achieve query complexity $O\left(\frac{nk \log n}{(1-2p)^2}\right)$ under the semi-random model, which matches the information theoretical lower bound $\Omega\left(\frac{nk}{(1-2p)^2}\right)$ within a $O(\log n)$ factor. Our main algorithm is Algorithm 2. The theoretical guarantee of Algorithm 2 is stated in Theorem 4 presented in the introduction. The proof of Theorem 4 is in Appendix C.1.

Algorithm 2 ESTIMATION(V, p) (Recover all large clusters of V)

Let $C = \emptyset$
Randomly select $T \subseteq V$ with $|T| = \frac{c \log n}{(1-2p)^2}$, $V \leftarrow V \setminus T$ ▷ c is a large enough constant
while $V \neq \emptyset$ **do**
  while FINDBIGCLUSTERS$(T) = \emptyset$ **do** ▷ Find subsets $T \cap V_i^*$ of size $\Omega\left(\frac{\log n}{(1-2p)^2}\right)$ exactly.
    Randomly select v from V,
    $T \leftarrow T \cup \{v\}, V \leftarrow V \setminus T$ ▷ Enlarge T until T has a subcluster of size $\Omega\left(\frac{\log n}{(1-2p)^2}\right)$.
  for $A \in$ FINDBIGCLUSTERS(T) **do** ▷ Each $A \in$ FINDBIGCLUSTERS(T) is a subcluster.
    Randomly select $B \subseteq A$, such that $|B| = \frac{320 \log n}{(1-2p)^2}$
    $A \leftarrow A \cup \{v \in V \mid$ DEGREETEST$(v, b) =$ "Yes"$\}$ ▷ Grow up A into a full cluster.
    $C \leftarrow C \cup \{A\}, V \leftarrow V \setminus A$
return C

ESTIMATION(V, p) outputs a set of clusters C. Each element in C is an underlying cluster. Each point $v \in V$ is a point that we cannot assign to a cluster in C. In the algorithm, we maintain

a set of points T as a sample set. If we can find all sets of the form $T_i^* = T \cap V_i^*$ such that $|T_i^*| = \Omega\left(\frac{\log n}{(1-2p)^2}\right)$, then we can use T_i^* to recover V_i^* with high probability, according to Lemma 9. If such T_i does not exist, we enlarge T until there is such a set. In this way, we can recover all large underlying clusters. To find these sets T_i^*, we an use the following Algorithm 3 with unlimited computational power.

Algorithm 3 FINDBIGCLUSTERS(T) (Extract all subsets $T \cap V_i^*, i \in [k]$, of large size)

Query every point pair in T and assign weight $w_{uv} = 2\mathcal{O}(u,v) - 1$ to each point pair
Let $C = \emptyset$
while $|T| \geq \frac{320 \log n}{(1-2p)^2}$ **do**
  Find the largest subset $S \subseteq T$ such that $val_S := \min_{A \subseteq S} \sum_{u \in A} \sum_{v \in S \setminus A} w_{uv} > 0$
  if $|S| < \frac{320 \log n}{(1-2p)^2}$ **then return** C
  $T \leftarrow T \setminus S, C \leftarrow C \cup \{S\}$
return C

FINDBIGCLUSTERS(T) assigns a weight $w_{uv} = 2\mathcal{O}(u,v) - 1$ to each point pair and extracts the largest subset $S \subseteq T$ such that S has no negative cut. We summarize the theoretical guarantee of Algorithm 3 via the following Theorem 10, which plays a key role in the proof of Theorem 4.

Theorem 10 *Let $T \subseteq V$ be a set of points. Under the semi-random model, with probability at least $1 - 1/poly(n)$,* FINDBIGCLUSTERS$(T) = \{T \cap V_i^* \mid |T \cap V_i^*| \geq \frac{320 \log n}{(1-2p)^2}\}$.

We sketch the proof of Theorem 10 here. We will show that if T contains a large subcluster, then with high probability, the largest subcluster will not contain a negative cut. On the other hand, with high probability, any large subset of T that is not a subcluster must contain a negative cut. Therefore, every time we find a large subset that contains no negative cut, we must find the largest subcluster contained in T. We summarize the above argument in Lemma 16 and Lemma 17 in Appendix C.2. A complete proof of Theorem 10 can also be found in Appendix C.2.

We remark that this information theoretical result is nontrivial. In our algorithm we process the sampled set T by finding the largest subset that has no negative cut, while in Mazumdar and Saha (2017a), the authors did this by computing the heaviest subgraph. A simple example with $k = 2$ can be used to show that their algorithm fails to recover the underlying clusters under the semi-random model. Suppose we have two underlying clusters with the same size. We run the algorithm in Mazumdar and Saha (2017a) to recover the two clusters. Every time we sample a set T of $\Omega(\log n)$ size, the adversary always outputs the true label for (u,v) if u, v are in the same underlying cluster, but outputs a wrong label if u, v are in different underlying clusters. When the noise level is high, in expectation, the heaviest subgraph of T is T itself and we have failed to recover the underlying clusters. We present this example in detail in Appendix C.3.

4. Computationally efficient algorithm

In this section, we develop a computationally efficient algorithm for our clustering problem under semi-random noise, presented in Algorithm 4. We analyze the performance of Algorithm 4 in Theorem 5 presented in the introduction. The full proof of Theorem 5 is in Appendix D.4.

Algorithm 4 CLUSTERING(V, p) (Recover all large clusters in V efficiently)

Let $C = \emptyset$, $s_t = \frac{c' t^3 \log n}{(1-2p)^6}$ ▷ c' is a large enough constant
while $V \neq \emptyset$ **do**
  $t = 1, h = 0$
  while $h = 0$ and $|V| \geq t s_t$ **do**
    Randomly select $T \subseteq V$ of size $t s_t$
    Let $\tilde{T} =$ APPROXCORRELATIONCLUSTER$(T, 1/\text{poly}(n))$ ▷ Compute an approximation of the correlation clustering of T, w.p. $1 - 1/\text{poly}(n)$
    Let $\{\hat{T}_1, \ldots, \hat{T}_h\} := \{\tilde{T}_i \in$ APPROXCORRELATIONCLUSTER$(T) \mid |\tilde{T}_i| > s_t/2\}$, $t \leftarrow 2t$
  if $|V| < t s_t$ and $h = 0$ **then return** C ▷ Stop when V only contains small clusters
  for $i \in [h]$ **do** ▷ Recover underlying clusters via η-biased sets
    Randomly select $B_i \subseteq \hat{T}_i$ of size $\frac{720 \log n}{(1-2p)^2}$
    Let $\tilde{V}_i = \{v \in V \mid$ DEGREETEST$(v, B_i) =$ "Yes"$\}$, $C \leftarrow C \cup \{\tilde{V}_i\}$, $V \leftarrow V \setminus \tilde{V}_i$
return C

The output of CLUSTERING(V, p) is a set of underlying clusters C. The set V contains points that we have not assigned to a cluster in C. In the algorithm, we maintain a variable t to estimate the number of underlying clusters in V. In each round, we sample a set of points T, whose size depends on t, and we compute a clustering $\tilde{T}$ of T to approximate the correlation clustering of T via APPROXCORRELATIONCLUSTER$(T, 1/\text{poly}(n))$. As we will see, when $|T|$ is large enough, with high probability, we can find (η, V_i^*)-biased sets from this approximate correlation clustering. Thus, we can use these biased sets to recover the corresponding underlying clusters. In this way, we can recover all large underlying clusters until V contains a small number of points.

Next, we present the outline of the remainder of this section. In Section 4.1, we give APPROXCORRELATIONCLUSTER and show how well it can approximate the correlation clustering of T. In Section 4.2, we present the structure of the approximate correlation clustering. Finally, we sketch the proof of Theorem 5 in Section 4.3.

4.1. Approximate correlation clustering

Let T be a set of points and let F be a binary function over $T \times T$. We consider the following natural SDP relaxation of the correlation clustering problem, which has been used for designing the approximate algorithm in Mathieu and Schudy (2010).

$$\begin{aligned} \min d(X, F) = & \sum_{1 \leq u \leq v \leq |T|} |X_{uv} - F(u, v)| \\ \text{s.t. } & X_{uv} + X_{vw} - X_{uw} \leq 1 \ \forall u, v, w \in T \\ & X_{uu} = 1 \ \forall u \in T, \ X_{uv} \geq 0 \ \forall u, v \in T \\ & X \succeq 0. \end{aligned} \tag{SDP(F)}$$

Algorithm 5 APPROXCORRELATIONCLUSTER(T, δ) (Approximate correlation clustering of T)

Query all point pairs of $T \subseteq V$ and construct the corresponding binary function $\bar{T}$
Compute X^*, a near optimal solution to SDP($\bar{T}$), with additive error at most $1/\text{poly}(n)$
Use X^* to do rounding $O\left(\log \frac{1}{\delta}\right)$ times and return the clustering $\tilde{T}$ that minimizes $d(\tilde{T}, X^*)$
(Compute a clustering $\tilde{T}$ by rounding around X^*): Start with $\tilde{T} = \emptyset$
while $T \neq \emptyset$ **do**
  Randomly select a point v from T and let $U = \{v\}$
  for $u \in T \setminus \{v\}$ **do** add u to U with probability X^*_{uv}
  $\tilde{T} \leftarrow \tilde{T} \cup \{U\}, T \leftarrow T \setminus \{U\}$
return $\tilde{T}$

Theorem 11 *Let $T \subseteq V$ and $\delta \in (0, 1)$. Then $\tilde{T} =$ APPROXCORRELATIONCLUSTER(T, δ) can be computed in poly $\left(|T|, \log \frac{1}{\delta}\right)$ time. Furthermore, under the semi-random model, there is a constant $c_1 > 0$ such that with probability at least $1 - O(\delta + \exp(-|T|)$,*

$$d(\tilde{T}, \bar{T}) \leq d(T^*, \bar{T}) + \frac{c_1 |T|^{\frac{3}{2}}}{1 - 2p},$$

where T^* is the underlying clustering of T and $\bar{T}$ is the query result over $T \times T$.

We remark that Theorem 11 is implicit in the proof of Theorem 1 in Mathieu and Schudy (2010). Here, we list the differences between the two results. First, the goal of Mathieu and Schudy (2010) is to design a $(1 + o_n(1))$-approximate algorithm, while here we focus on the additive error. Second, in Mathieu and Schudy (2010), Mathieu and Schudy used the optimal solution to the SDP to do rounding. However, to the best of our knowledge, it is unknown if such solution can be obtained in polynomial time. This is why, in this paper, we consider a near optimal solution and we show that it is sufficient to achieve the same theoretical guarantee. Finally, in Mathieu and Schudy (2010), the authors studied the performance of the algorithm in expectation, while here we give an exact bound for the probability that Algorithm 5 succeeds. The proof of Theorem 11 is given in to Appendix D.1.

4.2. Structure of the approximate correlation clustering

In Section 4.1, we have seen that, from Theorem 11, with high probability, the output of Algorithm 5 is close to the underlying clustering function. In this section, we study the structure of the output of Algorithm 5. We will see that, if we run Algorithm 5 over T with a large enough size, then the clusters in the output must contain some big clusters and all such big clusters are distinct (η, V_i^*)-biased sets. We summarize the main result of this section in the following theorem.

Theorem 12 *Let $T \subseteq V$ be a set of points such that $|T| = ts_t$, where $s_t = \frac{c' t^3 \log n}{(1-2p)^6}$, c' is a large enough constant and $t \in \mathbb{N}^+$. Let $\tilde{T} =$ APPROXCORRELATIONCLUSTER(T) and $\{\hat{T}_1, \ldots, \hat{T}_h\} := \{\tilde{T}_i \in \tilde{T} \mid |\tilde{T}_i| > s_t/2\}$. Let $\eta = \frac{1}{4} + \frac{p}{2}$. Under the semi-random model, with probability at least $1 - 1/\text{poly}(n)$ the following events happen.*

- *If there is some $i \in [k]$ such that $|T_i^*| > s_t$, where $T_i^* = T \cap V_i^*$, then $h > 0$.*
- *For every $i \in [h]$, $\hat{T}_i$ is an $\left(\eta, V_j^*\right)$-biased set for some $j \in [k]$.*

- *For every $i, j \in [h], i \neq j$, there is no $\ell \in [k]$ such that $\hat{T}_i, \hat{T}_j$ are both (η, V_ℓ^*)-biased sets.*

We sketch the proof of Theorem 12 here and leave the full proof to Appendix D.3. The key point of the proof of Theorem 12 is to show that, with high probability, any clustering function that is far from the underlying clustering function will have a large additive error. We summarize this result in Theorem 19 in Appendix D.2. Based on this technical theorem, we will show that if any event in the statement of Theorem 12 does not happen, then $\tilde{T}$ must be significantly far from T^*. By Theorem 11, we know that, with high probability, $\tilde{T}$ has a small additive error and cannot be too far from T^*. Thus, the three events in the statement of Theorem 12 must happen together with high probability.

4.3. Sketch of the proof of Theorem 5

According to Theorem 12, with probability at least $1 - 1/\text{poly}(n)$, once the sampled set T contains a subcluster of size s_t, APPROXCORRELATIONCLUSTER$(T, 1/\text{poly}(n))$ will contain some large clusters $\hat{T}_i$. Each of them is an (η, V_i^*)-biased set and corresponds to a different underlying cluster. Here, we choose $\eta = \frac{1}{4} + \frac{p}{2}$. By Hoeffding's inequality, with probability at least $1 - 1/\text{poly}(n)$, the corresponding subset B_i of $\hat{T}_i$ is a (η', V_i^*)-biased set, where $\eta' = \frac{p+1}{3} \in (p, \eta)$. Lemma 9 then implies that we can use B_i to recover V_i^*. This shows that every element in the output is an underlying cluster. On the other hand, if there is an underlying cluster of size $\Omega\left(\frac{k^4 \log n}{(1-2p)^6}\right)$ that has not been recovered, then at the end of the algorithm we have $|V| = \Omega\left(\frac{k^4 \log n}{(1-2p)^6}\right)$. However, after sampling T from V at most $O(\log k)$ times, T contains a subcluster of size s_t and the output will be updated. This gives the correctness of Algorithm 4. From the above argument, we can see that, throughout Algorithm 4, we have $t = O(k)$, and thus $|T| = O(ks_k)$. This also implies that we sample T at most $O(k \log k)$ times. So the number of queries we spend on constructing sample sets is $O\left(k \log k |T|^2\right)$. From the correctness of the algorithm, we invoke Algorithm 1 a total of $O(nk)$ times to recover the underlying clusters. The total number of queries we perform to recover the underlying clusters is $O\left(\frac{nk \log n}{(1-2p)^2}\right)$. Therefore, the query complexity of the algorithm can be bounded by $O\left(\frac{nk \log n}{(1-2p)^2}\right) + O\left(k \log k |T|^2\right) = O\left(\frac{nk \log n}{(1-2p)^2} + \frac{k^9 \log k \log^2 n}{(1-2p)^{12}}\right)$.

5. An efficient parameter-free algorithm under the fully-random model

In this section, we explain how Algorithm 4 can be used to design a parameter-free algorithm for the fully-random model. Since we have an efficient algorithm for the semi-random model, in principle, we can design an algorithm by guessing the parameter $(1-2p)$ from 0 to 1 and applying Algorithm 4 for each guess, until we are very close to the true parameter. However, in this way we will need to pay an extra $\log(\frac{1}{1-2p})$ factor for the query complexity and it will require us to test when to stop the guess. To overcome these problems, we sample a constant number of points $A \subseteq V$, and query $A \times V$ before doing clustering. We will see that, by counting the disagreements of point pairs in A, we can estimate a good upper bound $\bar{p}$ for the true error parameter p. Then, we can run CLUSTERING$(V, \bar{p})$ to recover large underlying clusters efficiently. We present the following algorithm whose performance is stated in Theorem 6 presented in the introduction and we leave the proof of Theorem 6 to Appendix E.1.

Algorithm 6 FCLUSTERING(V) (Parameter-free algorithm under fully-random model)

Randomly select $A \subseteq V$ such that $|A| = 9$.
for any pair of vertices $u, v \in A$ **do**
  Set count_{uv} to be the number of vertices $w \in V$ such that $\mathcal{O}(u, w) \neq \mathcal{O}(v, w)$.
Let $\bar{p} := \frac{1}{2} - \frac{1}{4}\sqrt{1 - \frac{2M}{n}}$ for $M := \min\{\text{count}_{uv} \mid u, v \in A, u \neq v\}$.
return CLUSTERING$(V, \bar{p})$

Acknowledgments

A. Del Pia is partially funded by ONR grant N00014-19-1-2322. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

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

Nir Ailon and Zohar Karnin. A note on: No need to choose: How to get both a ptas and sublinear query complexity. *arXiv preprint arXiv:1204.6588*, 2012.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):1–27, 2008.

Nir Ailon, Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Approximate clustering with same-cluster queries. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on Optimization*, 5(1):13–51, 1995.

Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures & Algorithms*, 13(3-4):457–466, 1998.

Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David. Clustering with same-cluster queries. *Advances in neural information processing systems*, 29, 2016.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1): 89–113, 2004.

Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of computational biology*, 6(3-4):281–297, 1999.

Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1):35–52, 1998.

Marco Bressan, Nicolò Cesa-Bianchi, Silvio Lattanzi, and Andrea Paudice. Exact recovery of mangled clusters with same-cluster queries. *Advances in Neural Information Processing Systems*, 33:9324–9334, 2020.

Michael J Brzozowski, Tad Hogg, and Gabor Szabo. Friends and foes: ideological social networking. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 817–820, 2008.

Moira Burke and Robert Kraut. Mopping up: modeling wikipedia promotion decisions. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 27–36, 2008.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005.

Sitan Chen, Frederic Koehler, Ankur Moitra, and Morris Yau. Classification under misspecification: Halfspaces, generalized linear models, and evolvability. *Advances in Neural Information Processing Systems*, 33:8391–8403, 2020.

Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *The Journal of Machine Learning Research*, 15(1):2213–2238, 2014.

Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd international conference on World Wide Web*, pages 285–294, 2013.

Erik D Demaine and Nicole Immorlica. Correlation clustering with partial information. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pages 1–13. Springer, 2003.

Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent PAC learning of halfspaces with Massart noise. *Advances in Neural Information Processing Systems*, 32, 2019.

Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

Buddhima Gamlath, Sangxia Huang, and Ola Svensson. Semi-supervised algorithms for approximately optimal and accurate clustering. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1167–1176, 2006.

Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F Naughton, Narasimhan Rampalli, Jude Shavlik, and Xiaojin Zhu. Corleone: Hands-off crowdsourcing for entity matching. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 601–612, 2014.

Kasper Green Larsen, Michael Mitzenmacher, and Charalampos Tsourakakis. Clustering with a faulty oracle. In *Proceedings of The Web Conference 2020*, pages 2831–2834, 2020.

Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

Wasim Huleihel, Arya Mazumdar, Muriel Médard, and Soumyabrata Pal. Same-cluster querying for overlapping clusters. *Advances in Neural Information Processing Systems*, 32, 2019.

Thorsten Joachims and John Hopcroft. Error bounds for correlation clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 385–392, 2005.

David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. Neural Information Processing Systems, 2011.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650, 2010.

Yi Li, Yan Song, and Qin Zhang. Learning to cluster via same-cluster queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 978–987, 2021.

Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Correlation clustering with noisy partial information. In *Conference on Learning Theory*, pages 1321–1342. PMLR, 2015.

Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006.

Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 712–728. SIAM, 2010.

Arya Mazumdar and Barna Saha. Clustering with noisy queries. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5790–5801, 2017a.

Arya Mazumdar and Barna Saha. A theoretical analysis of first heuristics of crowdsourced entity resolution. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017b.

Pan Peng and Jiapeng Zhang. Towards a query-optimal and time-efficient algorithm for clustering with a faulty oracle. In *Proceedings of Thirty Fourth Conference on Learning Theory*, pages 3662–3680, 2021.

Barna Saha and Sanjay Subramanian. Correlation clustering with same-cluster queries bounded by optimal cost. In *27th Annual European Symposium on Algorithms (ESA 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

Ron Shamir and Dekal Tsur. Improved algorithms for the random cluster graph model. *Random Structures & Algorithms*, 31(4):418–449, 2007.

Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *SODA*, volume 4, pages 526–527. Citeseer, 2004.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Norases Vesdapunt, Kedar Bellare, and Nilesh Dalvi. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment*, 7(12):1071–1082, 2014.

Van Vu. A simple svd algorithm for finding hidden partitions. *Combinatorics, Probability and Computing*, 27(1):124–140, 2018.

Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11), 2012.

Appendix A. Proof of Lemma 9

For $u \in B, v \in V$ let x_{uv} be the random variable such that

$$x_{uv} = \begin{cases} 1 \text{ if } (u,v) \text{ is not corrupted} \\ 0 \text{ otherwise.} \end{cases}$$

We first assume $v \in V_i^*$. It is sufficient to show $\sum_{u \in B \cap V_i^*} x_{uv} > |B|/2$ with high probability, because for every realization of $\mathcal{O}$, we have

$$\sum_{u \in B} \mathcal{O}(u,v) = \sum_{u \in B \cap V_i^*} \mathcal{O}(u,v) + \sum_{u \in B \setminus V_i^*} \mathcal{O}(u,v) \geq \sum_{u \in B \cap V_i^*} x_{uv}.$$

In expectation, we have

$$\mathbf{E} \sum_{u \in B \cap V_i^*} x_{uv} = (1-p)|B \cap V_i^*| \geq (1-p)\left(\frac{1}{2} + \eta\right)|B| = \left(\frac{1}{2} - \frac{1}{2}p + \eta(1-p)\right)|B| > \frac{1}{2}|B|, \tag{1}$$

where the first inequality holds because B is an (η, V_i^*)-biased set. The second inequality holds by the following calculation.

$$\begin{aligned}\left(\frac{1}{2} - \frac{1}{2}p + \eta(1-p)\right)|B| - \frac{1}{2}|B| = \left(\eta - \eta p - \frac{1}{2}p\right)|B| &= \frac{1}{2}(p+\eta)\left(\frac{1}{2} - p\right)|B| + \left(\frac{3}{4} - \frac{1}{2}p\right)(\eta - p)|B| \\ &\geq \frac{\eta}{2}\left(\frac{1}{2} - p\right)|B| > 0,\end{aligned} \tag{2}$$

where the first inequality holds because $0 < p < \eta \leq \frac{1}{2}$.

Since for every $u \in B \cap V_i^*$, (u,v) is corrupted independently, by Hoeffding's inequality and (1), we have

$$\begin{aligned}\Pr\left(\sum_{u \in B \cap V_i^*} x_{uv} < \frac{1}{2}|B|\right) &\leq \exp\left(-2\frac{\left(\mathbf{E}\sum_{u \in B \cap V_i^*} x_{uv} - \frac{1}{2}|B|\right)^2}{|B \cap V_i^*|}\right) \\ &\leq \exp\left(-\frac{1}{8}\eta^2(1-2p)^2|B|\right) \leq \frac{1}{n^{10}}.\end{aligned}$$

Here, the second inequality holds by (2) and the last inequality follows by $|B| > \frac{80 \log n}{\eta^2(1-2p)^2}$. So every single point $v \in V_i^*$ has probability at most $1/n^{10}$ to be misclassified by DEGREETEST(v, B).

Next, we assume that $v \in V_j^*$ for some $j \neq i$. It is sufficient to show with high probability $\sum_{u \in B \setminus V_j^*} (1 - x_{uv}) \leq \eta |B|$, because

$$\sum_{u \in B} \mathcal{O}(u,v) = \sum_{u \in B \cap V_j^*} \mathcal{O}(u,v) + \sum_{u \in B \setminus V_j^*} \mathcal{O}(u,v) \leq |B \cap V_j^*| + \sum_{u \in B \setminus V_j^*} (1 - x_{uv})$$
$$\leq \left(\frac{1}{2} - \eta\right) |B| + \sum_{u \in B \setminus V_j^*} (1 - x_{uv}) .$$

In expectation, we have

$$\mathbf{E} \sum_{u \in B \setminus V_j^*} (1 - x_{uv}) = p|B \setminus V_j^*| < \eta |B|. \tag{3}$$

Since for every $u \in B \cap V_j^*$, (u, v) is corrupted independently, by Hoeffding's inequality and (3), we have

$$\Pr\left(\sum_{u \in B \setminus V_j^*} (1 - x_{uv}) > \eta |B| \right) \leq \exp\left(-2 \frac{\left(\eta |B| - p|B \setminus V_j^*|\right)^2}{|B \setminus V_j^*|} \right)$$
$$\leq \exp\left(-2 (\eta - p)^2 |B| \right) \leq \frac{1}{n^{10}}.$$

Thus, for every $v \in V$, with probability at most $1/n^{10}$, v will be misclassified by DEGREETEST(v, B). By union bound, we know DEGREETEST(v, B) correctly classifies every $v \in V$ with probability at least $1 - 1/n^9$. ■

Appendix B. Technical discussion of efficient algorithm

In this section, we give a discussion of previous techniques for designing efficient algorithms under the noise model. We will take a disagreement counting method as an example and show how it can be applied to design algorithms under the semi-random model and where its limitation is. We will consider a slightly weaker model here.

Definition 13 *(Non-adaptive semi-random model of clustering with the faulty oracle) Under definition 1, the oracle $\mathcal{O}$ is defined in the following way. There is a set of unknown parameter $\{p_{uv} \geq 0 \mid u, v \in V\}$ and a known error parameter $p \in (0, 1/2)$ such that, for every point pair u, v, with probability $1 - p_{uv}$, $\mathcal{O}(u, v)$ outputs the true label of (u, v) and with probability p_{uv}, $\mathcal{O}(u, v)$ outputs the wrong label of (u, v), where $0 \leq p_{uv} \leq p$.*

B.1. A disagreement counting method to obtain (η, V_i^*)-biased sets

From Lemma 9, we know if we get an (η, V_i^*)-biased set of size $O(\log n)$, we can use it to recover V_i^*, by making $O(n \log n)$ queries. Thus, the key technique for designing an efficient algorithm is to obtain such (η, V_i^*)-biased sets by making a small number of queries. To address this problem,

we start with a simple disagreement counting method, which has been heavily used under the fully-random model Bansal et al. (2004); Mazumdar and Saha (2017a); Green Larsen et al. (2020). We consider the following simple procedure. Lemma 14 gives the theoretical guarantee for this simple procedure.

Algorithm 7 DISAGREEMENTTEST(u, v, T) (Check if u, v are in the same cluster via a set $T \subseteq V$)

if $\text{count}_v = |\{w \in T \mid \mathcal{O}(u, w) \neq \mathcal{O}(v, w)\}| > \frac{|T|}{2}$ **then return** "No" **else return** "Yes"

Lemma 14 *Under the non-adaptive semi-random model, suppose $k = 2$. If $|T| \geq \frac{100 \log n}{(1-2p)^4}$, then for every point pair (u, v), with probability at least $1 - 1/poly(n)$, the following event happens.*

- *If $u, v \in V_i^*$ for some $i \in [2]$,* DISAGREEMENTTEST(u, v, T) *returns "Yes".*
- *If $u \in V_i^*, v \in V_j^*$ for $i \neq j$,* DISAGREEMENTTEST(u, v, T) *returns "No".*

Proof We first assume $u, v \in V_i^*$ for some $i \in [2]$. We have for every point pair (u, v) and for every $w \in V$,

$$\Pr\left(\mathcal{O}(u, w) \neq \mathcal{O}(v, w)\right) = p_{uw}(1 - p_{vw}) + p_{vw}(1 - p_{uw}) \leq 2p(1 - p) = \frac{1}{2} - \frac{1}{2}(1 - 2p)^2,$$

since $\mathcal{O}(u, w) \neq \mathcal{O}(v, w)$ happens if and only if $\mathcal{O}$ gives a wrong answer to exactly one of (u, w) and (v, w). So when u, v in the same cluster V_i^*, in expectation, we have

$$\mathbf{E}\text{count}_v \leq \frac{|T|}{2} - \frac{1}{2}(1 - 2p)^2|T|.$$

By Hoeffding's inequality, we have

$$\Pr\left(\text{count}_v > \frac{|T|}{2}\right) \leq \exp\left(-\frac{(1 - 2p)^4 |T|}{2}\right) \leq \frac{1}{n^{50}}.$$

Next, we assume that u, v belong to different clusters. For every such point pair (u, v) and for every $w \in V$, we have

$$\Pr\left(\mathcal{O}(u, w) \neq \mathcal{O}(v, w)\right) = p_{uw}p_{vw} + (1 - p_{uw})(1 - p_{vw}) \geq 1 - 2p(1 - p) = \frac{1}{2} + \frac{1}{2}(1 - 2p)^2.$$

By Hoeffding's inequality, we have

$$\Pr\left(\text{count}_v \leq \frac{|T|}{2}\right) \leq \exp\left(-\frac{(1 - 2p)^4 |T|}{2}\right) \leq \frac{1}{n^{50}}.$$

■

Based on Lemma 14, we get the following simple algorithm for the special case where $k = 2$ under the non-adaptive semi-random model.

Algorithm 8 BICLUSTERING(V, p) (Exactly recover 2 underlying clusters)

Randomly select a point $u \in V$
Let $T_1 = \{u\}, T_2 = \emptyset$
for $i \in [\frac{640 \log n}{(1-2p)^2}]$ **do**
  Randomly select a subset $T \subseteq V$ of size $\frac{100 \log n}{(1-2p)^4}$
  Select $v \in V \setminus (T_1 \cup T_2)$
  if DISAGREEMENTTEST(u, v, T)="Yes" **then**
    $T_1 \leftarrow T_1 \cup \{v\}$
  else
    $T_2 \leftarrow T_2 \cup \{v\}$
Let $B = \text{argmax}\{|T_1|, |T_2|\}$
Let $\tilde{V}_1 = \{v \in V \mid \text{DEGREETEST}(v, T_1) = \text{“Yes”}\}$ $\tilde{V}_2 = \{v \in V \mid \text{DEGREETEST}(v, T_1) = \text{“No”}\}$
return $\tilde{V}_1, \tilde{V}_2$

Theorem 15 *There is an algorithm* BICLUSTERING(V, p)*, such that under the non-adaptive semi-random model, suppose* $k = 2$*, with probability at least* $1 - 1/poly(n)$*,* BICLUSTERING(V, p) *exactly recovers* V^**. Furthermore, the query complexity of* BICLUSTERING(V, p) *is* $O\left(\frac{n \log n}{(1-2p)^2} + \frac{\log^2 n}{(1-2p)^6}\right)$ *and the running time of* BICLUSTERING(V, p) *is* $O\left(\frac{n \log n}{(1-2p)^2} + \frac{\log^2 n}{(1-2p)^6}\right)$.

B.2. Proof of Theorem 15

According to Lemma 14 and union bound, we know that with probability at least $1 - 1/\text{poly}(n)$, every point $v \in T_1$ belongs to the same underlying cluster as u and every point $v \in T_2$ belongs to the different underlying cluster from u. Thus, we know that $B \subseteq V_i^*$ for some $i \in [2]$. Furthermore, since $|B| \geq \frac{320 \log n}{(1-2p)^2}$, according to Lemma 9, with $\eta = 1/2$, we know that with probability at least $1 - 1/\text{poly}(n)$, all points in $\tilde{V}_i$, for $i \in [2]$, come from the same underlying cluster. Combine the above argument together, with probability at least $1 - 1/\text{poly}(n)$, BICLUSTERING(V, p) exactly recover V^*.

It is simple to check the running time and the query complexity of BICLUSTERING(V, p) are the same. Since in BICLUSTERING(V, p), we invoke DISAGREEMENTTEST(u, v, T) at most $O\left(\frac{640 \log n}{(1-2p)^2}\right)$ times, and each time DISAGREEMENTTEST(u, v, T) queries $O\left(\frac{\log n}{(1-2p)^4}\right)$ times. The total number of queries is $O\left(\frac{\log^2 n}{(1-2p)^6}\right)$ in this stage. Since $|B| \leq \frac{640 \log n}{(1-2p)^2}$ and we invoke DEGREETEST n times, in this stage the total number of queries is $O\left(\frac{n \log n}{(1-2p)^2}\right)$. Thus, the query complexity of BICLUSTERING(V, p) is $O\left(\frac{n \log n}{(1-2p)^2} + \frac{\log^2 n}{(1-2p)^6}\right)$. ■

We can see that our algorithm achieves the same query complexity as the algorithm in Green Larsen et al. (2020) does, but our algorithm can succeed in a stronger model. It seems that using similar ideas we can design an efficient algorithm for more general settings. Unfortunately, as it turns out, this is a wrong approach. Although the naive disagreement counting method works very well under the fully-random model, under the semi-random model it only works on some very restrictive cases.

B.3. Disagreement counting method fails under general semi-random model

In fact, we can construct examples to show the disagreement counting method can easily fail, even if we have a small number of clusters and a constant level of noise rate. In the first example, we will show in Lemma 14, the assumption of the non-adaptive model is necessary. If we run Algorithm 7 under the adaptive semi-random model, we will fail even if we have only two clusters.

Example 1 *Under the adaptive semi-random model, suppose $k = 2$ and $p \geq 1 - \frac{\sqrt{2}}{2}$. There is an instance such that for every subset $T \subseteq V$ and for every point pair u, v, with probability $1/2$, the following event happens*

- *If $u, v \in V_i^*$ for some $i \in [2]$,* DISAGREEMENTTEST(u, v, T) *returns "No".*
- *If $u \in V_i^*, v \in V_j^*$ for $i \neq j$,* DISAGREEMENTTEST(u, v, T) *returns "Yes".*

Proof It is sufficient to show that for every $q \in [(1-p)^2, 1-p+p^2]$ and $r \in [p(1-p), p(2-p)]$, there is an adversary such that for every u, v in the same underlying cluster, $\Pr(\mathcal{O}(u, w) = \mathcal{O}(v, w)) = q, \forall w \in V$ and for every u, v in different underlying clusters, $\Pr(\mathcal{O}(u, w) = \mathcal{O}(v, w)) = r, \forall w \in V$. This is because when $p \geq 1 - \frac{\sqrt{2}}{2}$, we have

$$p(1-p) \leq (1-p)^2 \leq \frac{1}{2} \leq p(2-p) \leq 1 - p + p^2.$$

By setting $q = r = 1/2$, we know for every tuple u, v, w, $\Pr(\mathcal{O}(u, w) = \mathcal{O}(v, w)) = 1/2$, which directly implies that

$$\begin{aligned}\Pr\left(\text{count}_v > \frac{|T|}{2}\right) &= \Pr(\text{DISAGREEMENTTEST}(u, v, T) = \text{"No"}) \\ &= \Pr(\text{DISAGREEMENTTEST}(u, v, T) = \text{"Yes"}) = \Pr\left(\text{count}_v \leq \frac{|T|}{2}\right) = \frac{1}{2},\end{aligned}$$

for every choice of u, v, T. This implies the adversary can make it impossible to distinguish if two point u, v in the same underlying cluster or not by counting the disagreement. Now we show this.

We consider an adversary that works in the following way. For every point pair u, v in a same underlying cluster and any point $w \in V$, let (u, w) be the first point pair we query. The adversary always chooses to output a wrong label of (u, w), when (u, w) is corrupted. If $\mathcal{O}(u, w)$ outputs the correct label, the adversary chooses to output a correct label of (v, w) with probability $p_{cc} \in [0, 1]$, when (v, w) is corrupted. If $\mathcal{O}(u, w)$ outputs the wrong label, the adversary chooses to output a correct label of (v, w) with probability $p_{nc} \in [0, 1]$, when (v, w) is corrupted. Thus,

$$\Pr(\mathcal{O}(u, w) = \mathcal{O}(v, w)) = (1-p)(1-p+pp_{cc}) + p^2(1-p_{nc}) \in [(1-p)^2, 1-p+p^2],$$

where both the upper bound and the lower bound are achievable. By continuity, for every $q \in [(1-p)^2, 1-p+p^2]$, we can select proper parameters p_{cc}, p_{nc} to exactly match the probability. Similarly, for every (u, v) in different clusters, let (u, w) be the first point pair we query. The adversary always chooses to output a wrong label of (u, w), when (u, w) is corrupted. If $\mathcal{O}(u, w)$ outputs the correct label, the adversary chooses to output a correct label of (v, w) with probability

$p'_{cc} \in [0,1]$, when (v,w) is corrupted. If $\mathcal{O}(u,w)$ outputs the wrong label, the adversary chooses to output a correct label of (v,w) with probability $p'_{nc} \in [0,1]$, when (v,w) is corrupted. Thus,

$$\Pr\left(\mathcal{O}(u,w) = \mathcal{O}(v,w)\right) = (1-p)p(1-p'_{cc}) + p(1-p+pp'_{nc}) \in [p(1-p), p(2-p)],$$

where both the upper bound and the lower bound are achievable. By continuity, for every $r \in [p(1-p), p(2-p)]$, we can select proper parameter p'_{cc}, p'_{nc} to exactly match the probability. ■

We have seen Algorithm 7 can easily fail under the adaptive semi-random model, because of the power of the adversary. However, even if we work on the non-adaptive semi-random model, it is still hard to apply the disagreement counting technique to cases where we have more clusters.

Example 2 *Under the non-adaptive semi-random model, there is an instance with $k = 3$, $|V_i^*| = n/3$, for $i \in [3]$ and $p = 1/3$ such that for a subset T uniformly selected from V and for every u, v that are in different underlying clusters, with probability $1/2$,* DISAGREEMENTTEST(u, v, T) *returns "Yes".*

Proof We design the oracle in the following way. $\mathcal{O}(u,v) = 0$ with probability p_{in} for every u, v in the same underlying cluster. $\mathcal{O}(u,v) = 1$ with probability p_{out} for every u, v in different underlying clusters. Let (u,v) be a pair of points in different underlying clusters. Denote by V_u^* the underlying cluster that u belongs to and denote by V_v^* the underlying cluster that v belongs to. Let $w \in V_u^* \cup V_v^*$, Then

$$\Pr\left(\mathcal{O}(u,w) = \mathcal{O}(v,w)\right) = (1-p_{in})\, p_{out} + (1-p_{out})\, p_{in}.$$

Let $w \in V \setminus (V_u^* \cup V_v^*)$, then

$$\Pr\left(\mathcal{O}(u,w) = \mathcal{O}(v,w)\right) = p_{out}^2 + (1-p_{out})^2.$$

We pick $p_{in} = \frac{1}{2} - \frac{1}{4\sqrt{2}}$ and $p_{out} = \frac{1}{2} - \frac{\sqrt{2}}{4}$. It can be checked that $p_{out} < p_{in} < 1/3$. If we uniform pick some $w \in V$, we have

$$\Pr\left(\mathcal{O}(u,w) = \mathcal{O}(v,w)\right) = \frac{2}{3}\left((1-p_{in})\, p_{out} + (1-p_{out})\, p_{in}\right) + \frac{1}{3}\left(p_{out}^2 + (1-p_{out})^2\right) = \frac{1}{2}.$$

This implies for every subset T uniformly selected from V and every u, v in different underlying clusters,

$$\Pr\left(\text{DISAGREEMENTTEST}(u,v,T) = \text{"Yes"}\right) = \Pr\left(\text{count}_v \le \frac{|T|}{2}\right) = \frac{1}{2}.$$

■

Appendix C. Missing proofs in Section 3

C.1. Proof of Theorem 4

We first show the correctness of Algorithm 2. Let $\textsc{Estimation}(V,p) = \{\tilde{V}_1,\dots,\tilde{V}_\ell\}$. We start by showing every $\tilde{V}_i$ is an underlying cluster. Every $\tilde{V}_i$ is constructed by adding points from V to some $T' \in \textsc{FindBigClusters}(T)$, for some T through the algorithm. According to Theorem 10, $T' = T \cap V_i^*$ for some $i \in [k]$ and $|T'| \geq \frac{320\log n}{(1-2p)^2}$. Thus, according to Lemma 9, a point v is added to T' if and only if $v \in V_i^*$, which implies $\tilde{V}_i \subseteq V_i^*$. On the other hand, every $v \in V_i^*$ must be added to $\tilde{V}_i$. This is because at the time the first $T' = T \cap V_i^*$ is found by $\textsc{FindBigClusters}(T)$, v is in either T or V. If $v \in T$, then $v \in T'$, otherwise, v will be added to T' according to Lemma 9. So every element in the output of Algorithm 2 is an underlying cluster. Next, if there is an underlying cluster V_i^* of size at least $\frac{321\log n}{(1-2p)^2}$ that is not recovered, then $V_i^* \subseteq T$ at the end of the algorithm. However, in this case, according to Theorem 10, with probability at least $1 - 1/\text{poly}(n)$, $\textsc{FindBigClusters}(T) \neq \emptyset$ and the output will be updated. Thus, Algorithm 2 recovers all underlying clusters of size at least $\Omega\left(\frac{\log n}{(1-2p)^2}\right)$ with probability at least $1 - 1/\text{poly}(n)$.

Finally, we show the query complexity of Algorithm 2. Given a point v, before we put v into T, we check if we can assign v to some cluster in C and suppose $|C| = k'$. To check if v can be assigned to a cluster in C, we need to query at most $O\left(\frac{k'\log n}{(1-2p)^2}\right)$ times. If v can be assigned to a cluster in C, we do not need to query v with other points anymore. If v cannot be assigned to a cluster in C, we add v to T and query $O(T)$ times. We notice that $|T| = O\left(\frac{(k-k')\log n}{(1-2p)^2}\right)$, because otherwise T will contain a subcluster of size $\Omega\left(\frac{\log n}{(1-2p)^2}\right)$ and T will be updated. This implies to assign a given point v, we need to query

$$O\left(\frac{k'\log n}{(1-2p)^2}\right) + O\left(\frac{(k-k')\log n}{(1-2p)^2}\right) = O\left(\frac{k\log n}{(1-2p)^2}\right)$$

times. Thus, the query complexity of Algorithm 2 is $O\left(\frac{nk\log n}{(1-2p)^2}\right)$. ■

C.2. Proof of Theorem 10

In this part, we give the proof of Theorem 10. We first state two technical lemmas and use these two lemmas to prove Theorem 10. Then we give a full proof for the two technical lemmas.

The key part for the proof of Theorem 10 is to show the following two technical lemmas. Intuitively, we want to show if $T \subseteq V$ contains a large subcluster, then with high probability, this subcluster has no negative cut, while on the other hand, with high probability, any large subset of T that is not a subcluster must have a negative cut.

Lemma 16 *Under the semi-random model, let $S \subseteq V_i^*$ for some $i \in [k]$ such that $|S| \geq \frac{320\log n}{(1-2p)^2}$, then with probability at least $1 - 1/\text{poly}(n)$, $val_S = \min_{A\subseteq S}\sum_{u\in A}\sum_{v\in S\setminus A} w_{uv} > 0$.*

Lemma 17 *Let $T \subseteq V$ such that $|T| \geq \frac{320\log n}{(1-2p)^2}$. For $i \in [k]$, let $T_i := T \cap V_i^*$. Denote by $t := \max_{i\in[k]}|T_i|$ and $s := \frac{320\log n}{(1-2p)^2}$. Under the semi-random model, with probability at least $1 - 1/\text{poly}(n)$, any subset $S \subseteq T$ such that $|S| \geq \max\{t,s\}$ and S is not a subcluster satisfies $val_S = \min_{A\subseteq S}\sum_{u\in A}\sum_{v\in S\setminus A} w_{uv} \leq 0$.*

Now we prove Theorem 10 using Lemma 16 and Lemma 17.

Proof of Theorem 10 For $i \in [k]$, let $T_i := T \cap V_i^*$. Denote by $t := \max_{i\in[k]} |T_i|$ and $s := \frac{320 \log n}{(1-2p)^2}$. If $t < s$, according to Lemma 17, with probability at least $1 - 1/\text{poly}(n)$, any subset S of T, such that $|S| \geq s$ must have $val_S \leq 0$. In this case, no element will be added to the output. If $t \geq s$, let T' be the largest subcluster contained in T. According to Lemma 16, $val_{T'} > 0$. Furthermore, according to Lemma 17, any subset of T that has a larger size than T' will contain a negative cut. Thus, the largest subcluster T' is the largest subset of T that contains no negative cut and will be added to the output. Since there are $k \leq n$ underlying clusters, by union bound, we know with probability at least $1 - 1/\text{poly}(n)$, the output of Algorithm 3 is $\{T \cap V_i^* \mid |T \cap V_i^*| \geq \frac{320 \log n}{(1-2p)^2}\}$. ■

C.2.1. PROOF OF LEMMA 16

Let $(A, S \setminus A)$ be a partition of S such that $|A| \leq |S|/2$. For $u \in A, v \in S \setminus A$, let x_{uv} be the random variable such that

$$x_{uv} = \begin{cases} 1 \text{ if } (u,v) \text{ is not corrupted} \\ -1 \text{ otherwise.} \end{cases}$$

Clearly, for every realization of the query result,

$$\sum_{u\in A} \sum_{v\in S\setminus A} w_{uv} \geq \sum_{u\in A} \sum_{v\in S\setminus A} x_{uv}.$$

Since

$$\mathbf{E} \sum_{u\in A} \sum_{v\in S\setminus A} x_{uv} = (1-2p)\,|A||S \setminus A|,$$

by Hoeffding's inequality, we have

$$\Pr\left(\sum_{u\in A} \sum_{v\in S\setminus A} w_{uv} \leq 0\right) \leq \Pr\left(\sum_{u\in A} \sum_{v\in S\setminus A} x_{uv} \leq 0\right) \leq \exp\left(-\frac{(1-2p)^2\,(|A||S \setminus A|)}{2}\right).$$

Suppose $|A| = t$, where $t \in [|S|/2]$. By union bound, we have

$$\begin{aligned}
\Pr\left(\exists A \subseteq S, |A| = t, \sum_{u\in A} \sum_{v\in S\setminus A} w_{uv} \leq 0\right) &\leq \binom{|S|}{t} \exp\left(-\frac{(1-2p)^2\,(|A||S \setminus A|)}{2}\right) \\
&\leq \exp\left(t \log |S| - \frac{(1-2p)^2\, t(|S| - t)}{2}\right) \\
&\leq \exp\left(-t\left(\frac{(1-2p)^2|S|}{4} - \log |S|\right)\right) \\
&\leq \exp\left(-24 \log n\right) = \frac{1}{n^{24}},
\end{aligned}$$

where the third inequality follows by $t \leq |S|/2$ and the last inequality holds because $\frac{320 \log n}{(1-2p)^2} \leq |S| \leq n$ and $t \geq 1$. Using union bound again, we can get

$$\Pr\left(val_S \leq 0\right) \leq \sum_{t=1}^{|S|/2} \Pr\left(\exists A \subseteq S, |A| = t, \sum_{u \in A} \sum_{v \in S \setminus A} w_{uv} \leq 0\right) \leq \frac{1}{\text{poly}(n)}.$$

So with probability at least $1 - 1/\text{poly}(n)$, $val_S > 0$. ■

C.2.2. PROOF OF LEMMA 17

Let $S \subseteq T$. Denote by $m = |S| \geq \max\{t, s\}$. Denote by S^* the largest subcluster contained in S. Let $(A, S \setminus A)$ be a partition of S such that $|A| \leq |S|/2$. For $u \in A, v \in S \setminus A$, let x_{uv} be the random variable such that

$$x_{uv} = \begin{cases} 1 \text{ if } (u, v) \text{ is not corrupted} \\ -1 \text{ otherwise.} \end{cases}$$

We first show with probability at least $1 - 1/\text{poly}(n)$, any subset S that satisfies the statement of Lemma 17 and $|S^*| \leq m/4$, must have $val_S \leq 0$. We first fix such a set S. We show with high probability we can construct a subset $A \subseteq S$ such that $\sum_{u \in A} \sum_{v \in S \setminus A} w_{uv} \leq 0$. Without loss of generality, we assume that $0 \leq |S \cap V_1^*| \leq \cdots \leq |S \cap V_k^*| \leq |S|/4$.

We define $A := \cup_{j=1}^{i^*} \left(S \cap V_j^*\right)$, where i^* is the largest index such that $|\cup_{j=1}^{i^*} \left(S \cap V_j^*\right)| \leq |S|/2$. We can see $|A| > |S|/4$, otherwise, we can put $S \cap V_{i^*+1}^*$ into A and keep $|A| \leq |S|/2$. This implies $|A||S \setminus A| \geq |S|^2/8$. In expectation, we have

$$\mathbf{E} \sum_{u \in A} \sum_{v \in S \setminus A} -x_{uv} = -\left(1 - 2p\right)|A||S \setminus A|.$$

By Hoeffding's inequality, we have

$$\begin{aligned} \Pr\left(val_S \geq 0\right) &\leq \Pr\left(\sum_{u \in A} \sum_{v \in S \setminus A} w_{uv} \geq 0\right) \leq \Pr\left(\sum_{u \in A} \sum_{v \in S \setminus A} -x_{uv} \geq 0\right) \\ &\leq \exp\left(-\frac{(1-2p)^2 |A||S \setminus A|}{2}\right) \leq \exp\left(-\frac{(1-2p)^2 m^2}{16}\right). \end{aligned}$$

By union bound, we have

$$\begin{aligned} \Pr\left(\exists S, |S| = m, |S^*| \leq m/4, val_S > 0\right) &\leq \tbinom{n}{m} \exp\left(-\frac{(1-2p)^2 m^2}{16}\right) \\ &\leq \exp\left(m \log n - \frac{(1-2p)^2 m^2}{16}\right) \\ &= \exp\left(-m\left(\frac{(1-2p)^2 m}{16} - \log n\right)\right) \\ &\leq \exp(-4m \log n) \leq \frac{1}{n^{80}}, \end{aligned}$$

where in the third inequality, we use the fact $\frac{(1-2p)^2 m}{16} \geq \frac{(1-2p)^2 s}{16} \geq 5 \log n$.

Again, using union bound over m, we get

$$\Pr\left(\exists S, |S| \geq \max\{t, s\}, |S^*| \leq |S|/4, val_S > 0\right) \leq n \Pr\left(\exists S, |S| = m, |S^*| \leq m/4, val_S > 0\right) \leq \frac{1}{\text{poly}(n)}.$$

Thus, with probability at least $1 - 1/\text{poly}(n)$, any subset S that satisfies the statement of Lemma 17 and $|S^*| \leq m/4$ must have $val_S \leq 0$.

Next, we show with probability at least $1 - 1/\text{poly}(n)$, any subset S that satisfies the statement of Lemma 17 and $m/4 \leq |S^*| \leq m - 1$ must have $val_S \leq 0$. To simplify the notation, we denote by $r := |S \setminus S^*|$. We fix such a set S. We show with high probability $\sum_{u \in S^*} \sum_{v \in S \setminus S^*} w_{uv} \leq 0$. By Hoeffding's inequality, we have

$$\begin{aligned}
\Pr\left(val_S \geq 0\right) &\leq \Pr\left(\sum_{u \in S^*} \sum_{v \in S \setminus S^*} w_{uv} \geq 0\right) \leq \Pr\left(\sum_{u \in S^*} \sum_{v \in S \setminus S^*} -x_{uv} \geq 0\right) \\
&\leq \exp\left(-\frac{(1-2p)^2 |S^*||S \setminus S^*|}{2}\right) = \exp\left(-\frac{(1-2p)^2 (m-r) r}{16}\right).
\end{aligned}$$

Then we upper bound the number of such S. We first choose an index $i \in [k]$ such that $S \cap V_i^* = S^*$. The number of choices of the index is at most k. Then we choose $m - r$ points from $T \cap V_i^*$. The number of choices of the points is at most $\binom{t}{m-r}$. Finally, we choose r points from the rest points in T. The number of such choices is at most $\binom{|T|}{r}$. So the number of such S is at most

$$k\tbinom{t}{m-r}\tbinom{|T|}{r} \leq k\tbinom{m}{m-r}\tbinom{n}{r} = k\tbinom{m}{r}\tbinom{n}{r} \leq k \exp\left(r(\log m + \log n)\right) \leq n \exp(2r \log n),$$

where the first inequality follows by $t \leq m$ and $|T| \leq n$ and the last inequality follows by $k \leq n$ and $m \leq n$. By union bound, we have

$$\begin{aligned}
\Pr\left(\exists S, |S| = m, |S^*| = m - r, val_S > 0\right) &\leq n \exp\left(2r \log n - \frac{(1-2p)^2 (m-r) r}{16}\right) \\
&= n \exp\left(-2r\left(\frac{(1-2p)^2 (m-r)}{32} - \log n\right)\right) \\
&\leq n \exp\left(-2r\left(\frac{3(1-2p)^2 m}{128} - \log n\right)\right) \\
&\leq n \exp\left(-10 r \log n\right) \leq \frac{1}{n^9}.
\end{aligned}$$

Here, the second inequality follows by $1 \leq r \leq m/4$. The third inequality holds because $m \geq s$ and the last inequality holds since $r \geq 1$. Since $1 \leq r \leq m \leq n$, by applying union bound over r and m, we have

$$\Pr\left(\exists S, |S| \geq \max\{t, s\}, |S|/4 < |S^*| < |S|, val_S > 0\right) \leq n^2 \Pr\left(\exists S, |S| = m, |S^*| = m - r, val_S > 0\right) \leq \frac{1}{\text{poly}(n)}.$$

Thus, with probability at least $1 - 1/\text{poly}(n)$, any subset $S \subseteq T$ such that $|S| \geq \max\{t, s\}$ and S is not a subcluster satisfies $val_S = \min_{A \subseteq S} \sum_{u \in A} \sum_{v \in S \setminus A} w_{uv} \leq 0$. ■

C.3. Example where Algorithm 1 in Mazumdar and Saha (2017a) fails

In this part, we present an example where Algorithm 1 in Mazumdar and Saha (2017a) fails. We remark that the main difference of Algorithm 2 in this paper and Algorithm 1 in Mazumdar and Saha (2017a) is that given a sampled set T, we find the largest subset of T that contains no negative cut, while they compute the heaviest subgraph of T.

Example 3 *Under the semi-random model, there is an instance with $k = 2$ and $p \geq \frac{2}{5}$ such that with probability at least $1 - o_n(1)$, Algorithm 1 in Mazumdar and Saha (2017a) fails to recover any cluster.*

Proof Consider $V = V_1^* \cup V_2^*$, where $|V_1^*| = |V_2^*| = \frac{n}{2}$. Let $p = 2/5$ be the error parameter. We run Algorithm 1 in Mazumdar and Saha (2017a) on this example. The first step of the algorithm is to sample a set T of size $s = \frac{16 \log n}{(1-2p)^2}$. By Chernoff bound, with probability at least $1 - o_n(1)$, $|T \cap V_1^*| \geq \frac{3s}{7}$ and $|T \cap V_1^*| \geq \frac{3s}{7}$. After getting such a sampled set T, the adversary works in the following way. For every corrupted point pair (u, v), the adversary outputs the true label if u, v are in the same underlying clusters and otherwise outputs a wrong label.

We will next show with probability at least $1-o_n(1)$, the largest subgraph of T is T. To simplify the notation, let $A = T \cap V_1^*$ and $B = T \cap V_2^*$. We know for every (u, v) such that $u \in A$ and $v \in B$, $w_{uv} = -1$ with probability p. By Hoeffding's inequality and union bound, with probability at least $1 - o_n(1)$, for every $u \in A$, we have $\sum_{v \in B} w_{uv} \geq -\frac{|B|}{3}$ and for every $v \in B$, we have $\sum_{u \in A} w_{uv} \geq -\frac{|A|}{3}$. Now, let S be an arbitrary subset of T and assume of V_i^* is the underlying cluster that the majority of points of S come from. It is not hard to see, adding another point $v \in T \cap V_i^*$ will not decrease the total weight of S. So we can without loss of generality assume $A \subseteq S$ or $B \subseteq S$. We deal with the case when $A \subseteq S$ and the proof is the same when $B \subseteq S$. Suppose $|S \cap B| = x|B|$, where $x \in [0, 1]$. Now we add the rest $(1 - x)|B|$ points from B to S and we get T. Since for every $v \in B$, we have $\sum_{u \in A} w_{uv} \geq -\frac{|A|}{3}$, we know the increment of weight is at least

$$x(1-x)|B|^2 + \frac{(1-x)^2|B|^2}{2} - \frac{(1-x)|A||B|}{3} \geq x(1-x)|B|^2 + \frac{(1-x)^2|B|^2}{2} - \frac{4(1-x)|B|^2}{9} > 0,$$

where the first inequality follows by the fact $\frac{|A|}{|B|} \leq \frac{4}{3}$. This implies, with probability at least $1 - o_n(1)$, T itself is the largest subset of T. Furthermore, since $|T| \geq s$, we will extract T. In this case, we have already fail to recover any cluster. ■

Appendix D. Missing proof and discussion in Section 4

D.1. Proof of Theorem 11

In this part, we discuss Theorem 11 in detail. We will see where Theorem 11 and the results in Mathieu and Schudy (2010) are different and why Theorem 11 is true. To start with, we summarize the rounding step in Algorithm 5 in the following algorithm. Let $T \subseteq V$ be a set of points and F be a binary function over $T \times T$. We say a symmetric matrix $\hat{X}$ is good if

- $0 \leq \hat{X}_{uv} \leq 1$ for every $u, v \in T$,

- The distance between $\hat{X}$ and the feasible region of (SDP(F)) is at most $1/\text{poly}(|T|)$,
- $d(\hat{X}, F) \leq d(X^*, F) + 1/\text{poly}(|T|)$, where X^* is an optimal solution to (SDP(F)).

Algorithm 9 SDPCLUSTER(T, F) (Algorithm 2 in Mathieu and Schudy (2010))

Let $C = \emptyset$
Let $\hat{X}$ be a good solution to (SDP(F)).
while $T \neq \emptyset$ **do** ▷ Use $\hat{X}$ to do rounding
  Randomly select a point T from V
  Let $U = \{v\}$
  for $u \in T \setminus \{v\}$ **do**
    Add u to U with probability $\hat{X}_{uv}$
  $C \leftarrow C \cup \{U\}, T \leftarrow T \setminus \{U\}$
return C.

We remark that the only difference between Algorithm 9 and Algorithm 2 in Mathieu and Schudy (2010) is that we use a good solution to do rounding, while they use an optimal solution to do rounding. Currently, we do not know a polynomial time algorithm that can solve general semi-definite programmings exactly. This is to say we do not know how to obtain an optimal solution to (SDP(F)) in polynomial time. Current theoretical guarantee for solving an SDP Grötschel et al. (1981); Alizadeh (1995) is that for every $\epsilon \in (0, 1)$, we can find a solution ϵ-close to the feasible region with additive error at most ϵ in polynomial time. In our case, by choosing $\epsilon = 1/\text{poly}(n)$, this implies we can obtain a good solution to (SDP(F)) in polynomial time via a naive rounding step to make sure the first condition in the definition of a good solution holds. This tiny change can ensure Algorithm 9 definitely runs in polynomial time. In the following discussion, we will show Theorem 11 is still true even if we do not use an optimal solution to do rounding.

Theorem 18 *(Theorem 5 in Mathieu and Schudy (2010)) For every input (T, F), let $\mathcal{A} =$ SDPCLUSTER(T, F). For every clustering function C' over T, we have*

$$\mathbf{E}d(\mathcal{A}, \hat{X}) \leq 3d(C', \hat{X}),$$

where $\hat{X}$ is the good solution used in SDPCLUSTER(T, F).

The proof of Theorem 18 can be found in Mathieu and Schudy (2010). Readers may notice that the statement of Theorem 18 is slightly different from the original statement in Mathieu and Schudy (2010). In the original statement, Mathieu and Schudy, restricted $\hat{X}$ to be an optimal solution to (SDP(F)), while we relax this restriction to good solutions. We remark that, as Mathieu and Schudy claimed in their proof, as long as $\hat{X}$ is a symmetric matrix in $[0, 1]^{|V|\times|V|}$, Theorem 18 holds. An immediately corollary of Theorem 18 is if we do rounding $\Omega(\log \frac{1}{\delta})$ times and pick $\mathcal{A}^*$ to be the clustering that is closest to $\hat{X}$, then with probability $1 - \delta$, $d(\mathcal{A}^*, \hat{X}) \leq 4d(C', \hat{X})$. Next, we will see why $\mathcal{A}^*$ can achieve an additive error $O\left(\frac{|T|^{3/2}}{1-2p}\right)$ with high probability.

Let $M, N \in \mathbb{R}^{|T|\times|T|}$. We define $M \cdot N := \sum_{u,v} M_{uv}N_{uv}$. For $F \in \{0,1\}^{|T|\times|T|}$, We define $\hat{F} \in \{-1,1\}^{|T|\times|T|}$ as follows:

$$\hat{F}_{uv} = \begin{cases} -1 & \text{if } F_{uv} = 1 \\ 1 & \text{if } F_{uv} = 0. \end{cases}$$

Claim 1 *(Claim 16 in Mathieu and Schudy (2010)) For every $M \in \{0,1\}^{|T|\times|T|}$ and $N \in [0,1]^{|T|\times|T|}$, we have*

$$d(M,N) = \frac{1}{2}\left(\hat{M}\cdot N - \hat{M}\cdot M\right).$$

Under the semi-random model, we define a symmetric random matrix $E \in \{0,1\}^{|T|\times|T|}$ in the following way. For every (u,v) such that u,v are in the same cluster in V^*, $E_{uv} = 0$ if and only if (u,v) is corrupted. For every (u,v) such u,v are in the different clusters in V^*, $E_{uv} = 1$ if and only if (u,v) is corrupted. Intuitively, $E_{uv} = \mathcal{O}(u,v)$ for every u,v, where the adversary in the oracle always gives the wrong answer.

Claim 2 *(Lemma 23 in Mathieu and Schudy (2010)) Under the semi-random model, there is a constant $c > 0$, such that with probability at least $1 - 4\exp(-|T|)$,*

$$|\hat{E}\cdot X - \mathbf{E}\hat{E}\cdot X| \le c|T|^{\frac{3}{2}}$$

for every symmetric matrix X with trace at most $2|T|$ and smallest eigenvalue at least $-1/poly(|T|)$.

We remark that the statement of Claim 2 is slightly different from the statement of Lemma 23 in Mathieu and Schudy (2010). In the original statement, Mathieu and Schudy didn't give a concrete bound of the probability of success and X is forced to be positive semi-definite with trace to be $|T|$. We remark that every good solution satisfies the statement of the claim. Here we give a short proof of the claim by slightly modifying the proof of Lemma 23 in Mathieu and Schudy (2010).

Proof of Claim. Write $M = \hat{E} - \mathbf{E}\hat{E}$. Write $X = \sum_{i=1}^{|T|}\lambda_i v_i v_i^T$ by doing spectral decomposition of X. Without loss of generality, we assume that $\lambda_1 \ge \ldots \lambda_r \ge 0 \ge \lambda_{r+1} \ge \lambda_{|T|} \ge -1/\text{poly}(|T|)$. Notice that

$$\begin{aligned}
|M\cdot X| = |\sum_{i=1}^{|T|}\lambda_i v_i^T M v_i| &\le \sum_{i=1}^{r}\lambda_i |v_i^T M v_i| - \sum_{i=r+1}^{|T|}\lambda_i |v_i^T M v_i| \\
&\le \sum_{i=1}^{r}\lambda_i \rho(M) - \sum_{i=r+1}^{|T|}\lambda_i \rho(M) \\
&= \sum_{i=1}^{|T|}\lambda_i \rho(M) - 2\sum_{i=r+1}^{|T|}\lambda_i \rho(M) \\
&\le (2|T| + \frac{1}{\text{poly}(|T|)})\rho(M) \le 3|T|\rho(M).
\end{aligned}$$

It is sufficient to show $\rho(M) = O(\sqrt{T})$ with probability at least $1 - 4\exp(|T|)$. We notice that M is symmetric matrix whose entries on and above the diagonal are independent mean-zero sub-gaussian random variables. By Corollary 4.4.8 in Vershynin (2018), there is a constant c such that $\rho(M) \leq c\sqrt{T}$ with probability at least $1 - 4\exp(|T|)$.

$\diamond$

We know with probability $1 - \delta$, $d(\mathcal{A}^*, \hat{X}) \leq 4d(C', \hat{X})$ for every clustering C' over T. So we have

$$d(\mathcal{A}^*, \bar{T}) \leq d(\mathcal{A}^*, \hat{X}) + d(\hat{X}, \bar{T}) \leq d(\mathcal{A}^*, \hat{X}) + d(T^*, \bar{T}) + \frac{1}{\text{poly}(|T|)} \leq d(T^*, \bar{T}) + 4d(T^*, \hat{X}) + \frac{1}{\text{poly}(|T|)},$$

where the first inequality holds by triangle inequality, the second inequality holds since $\hat{X}$ is a good solution to SDP($\bar{T}$), and T^* is a feasible solution to SDP($\bar{T}$). It remains to upper bound $d(T^*, \hat{X})$. It can be checked easily that

$$\mathbf{E}\hat{E} = (1 - 2p)T^*.$$

By Claim 1, we know

$$\begin{aligned}
d(T^*, \hat{X}) &= \frac{1}{2}\left(\hat{T}^* \cdot \hat{X} - \hat{T}^* \cdot T^*\right) = \frac{1}{2(1-2p)}\left(\mathbf{E}\hat{E} \cdot \hat{X} - \mathbf{E}\hat{E} \cdot T^*\right) \\
&= \frac{1}{2(1-2p)}\left(\mathbf{E}\hat{E} \cdot \hat{X} - \hat{E} \cdot \hat{X} + \hat{E} \cdot \hat{X} - \hat{E} \cdot T^* + \hat{E} \cdot T^* - \mathbf{E}\hat{E} \cdot T^*\right) \\
&\leq \frac{1}{2(1-2p)}\left(\hat{E} \cdot \hat{X} - \hat{E} \cdot T^* + 2c|T|^{\frac{3}{2}}\right) \\
&= \frac{1}{2(1-2p)}\left(\hat{E} \cdot \hat{X} - \hat{E} \cdot E + \hat{E} \cdot E - \hat{E} \cdot T^* + 2c|T|^{\frac{3}{2}}\right) \\
&= \frac{1}{2(1-2p)}\left(2\left(d(\hat{X}, E) - d(T^*, E)\right) + 2c|T|^{\frac{3}{2}}\right) \\
&\leq \frac{1}{2(1-2p)}\left(2\left(d(\hat{X}, \bar{T}) - d(V^*, \bar{T})\right) + 2c|T|^{\frac{3}{2}}\right) \\
&\leq \frac{c'|T|^{\frac{3}{2}}}{(1-2p)}.
\end{aligned}$$

Here, the first inequality follows by Claim 2, the last equality follows by Claim 1 and the last inequality holds because $\hat{X}$ is a good solution. To see why the second last inequality holds, we suppose that the adversary gets the chance to give a wrong label of $e = (u, v)$ but chooses to give the correct label. Then we have

$$|T^*_e - E_e| = |T^*_e - \bar{T}_e| + 1,$$

while

$$|\hat{X}_e - E_e| \leq |\hat{X}_e - \bar{V}_e| + 1,$$

because $\hat{X}_e \in [0, 1]$. So we know for every point pair e,

$$|\hat{X}_e - E_e| - |T^*_e - E_e| \leq |\hat{X}_e - \bar{T}_e| - |T^*_e - \bar{T}_e|.$$

By sum all these inequalities over point pair e, we get the second last inequality. So far, we have shown with probability at least $1-\delta-4\exp(-|T|)$, $d(\mathcal{A}^*,\bar{T}) \leq d(V^*,\bar{T}) + O\left(\frac{|T|^{3/2}}{1-2p}\right)$. In particular, since we do not need to solve (SDP(F)) exactly, $\tilde{T} = \text{APPROXCORRELATIONCLUSTER}(T)$ can be obtained in polynomial time.

D.2. Missing technical theorem

In this section, we prove the following technical theorem, which will be used to prove Theorem 12.

Theorem 19 *Let V be a set of points such that $|V| = ts$, where $t, s > 0$. Let $c > 0, \epsilon \geq 0$ be two numbers such that $c(1-2p)^2 s^2/2 > \epsilon$. Under the semi-random model, with probability at least $1 - \exp\left(ts\log ts - c(1-2p)^3 s^2/8\right)$, for every clustering function V' over V such that $d(V',V^*) \geq c(1-2p)s^2$, we have*

$$d(\bar{V},V') > d(\bar{V},V^*) + \epsilon,$$

where $\bar{V}$ is the binary function over V corresponding to the results that we query every point pair of V and V^* is the underlying clustering function of V.

We first introduce the following notations to simplify the proof. Let $V', \tilde{V}$ be two clustering functions over V. We let $D_{V'\tilde{V}} := \{(u,v) \mid V'(u,v) \neq \tilde{V}(u,v)\}$ be set of point pairs that are labeled differently by V' and $\tilde{V}$. In particular, for every clustering function V', we define $D^{\text{n}}_{V'V^*} = \{(u,v) \in D_{V'V^*} \mid V^*(u,v) \neq \bar{V}(u,v)\}$ and $D^{\text{c}}_{V'V^*} = \{(u,v) \in D_{V'V^*} \mid V^*(u,v) = \bar{V}(u,v)\}$.

To prove Theorem 19, we first prove the following lemma.

Lemma 20 *Let $V = [n]$ be a set of points. Let V^* be the underlying clustering function of V. Let $\bar{V}$ be the binary function corresponding to the results that we query all point pairs of V. Let V' be a clustering function over V. Then*

$$d(\bar{V},V^*) - d(\bar{V},V') = |D^n_{V'V^*}| - |D^c_{V'V^*}|$$

Proof Since $V^*(u,v) \neq V'(u,v)$ if and only if $(u,v) \in D_{V'V^*}$, we know

$$d(\bar{V},V^*) - d(\bar{V},V') = \sum_{(u,v)\in D_{V'V^*}} \left(|\bar{V}(u,v) - V^*(u,v)| - |\bar{V}(u,v) - V'(u,v)|\right).$$

For every $(u,v) \in D^{\text{n}}_{V'V^*}$, we have $|\bar{V}(u,v) - V^*(u,v)| = 1$ and $|\bar{V}(u,v) - V'(u,v)| = 0$. On the other hand, for every $(u,v) \in D^{\text{c}}_{V'V^*}$, we have $|\bar{V}(u,v) - V^*(u,v)| = 0$ and $|\bar{V}(u,v) - V'(u,v)| = 1$. Thus, we have

$$d(\bar{V},V^*) - d(\bar{V},V') = |D^{\text{n}}_{V'V^*}| - |D^{\text{c}}_{V'V^*}|.$$

■

Now we use Lemma 20 to prove Theorem 19.

Proof of Theorem 19 We first fix a clustering function V' over V such that $d(V', V^*) \geq c\,(1-2p)\,s^2$. We first show that with high probability, $d(\bar{V}, V') > d(\bar{V}, V^*) + \epsilon$. For every point pair (u, v), we define random variable

$$x_{uv} = \begin{cases} 1 \text{ if } (u,v) \text{ is not corrupted} \\ -1 \text{ otherwise.} \end{cases}$$

We observe that for every realization of $\bar{V}$, we always have

$$|D^{\mathrm{c}}_{V'V^*}| - |D^{\mathrm{n}}_{V'V^*}| \geq \sum_{e \in D_{V'V^*}} x_e. \tag{4}$$

This is because if an adversary gets a chance to output a wrong label of e, but does not do that, $|D^{\mathrm{n}}_{V'V^*}|$ will increase by 1, while $|D^{\mathrm{c}}_{V'V^*}|$ will decrease by 1.

In expectation, we have

$$\mathbf{E} \sum_{e \in D_{V'V^*}} x_e = (1-2p)\,|D_{V'V^*}| = (1-2p)\,d(V', V^*) \geq c\,(1-2p)^2\,s^2 > 2\epsilon. \tag{5}$$

Thus, we have

$$\begin{aligned} \Pr\left(|D^{\mathrm{c}}_{V'V^*}| - |D^{\mathrm{n}}_{V'V^*}| \leq \epsilon\right) &\leq \Pr\left(\sum_{e \in D_{V'V^*}} x_e \leq \epsilon\right) \\ &\leq \Pr\left(\sum_{e \in D_{V'V^*}} x_e \leq \frac{\mathbf{E}\sum_{e \in D_{V'V^*}} x_e}{2}\right) \\ &\leq \exp\left(-\frac{(1-2p)^2\,d(V', V^*)^2}{8 d(V', V^*)}\right) \\ &\leq \exp\left(-\frac{c\,(1-2p)^3\,s^2}{8}\right). \end{aligned}$$

Here, the first inequality follows by (4), the second inequality follows by (5), the third inequality follows by the Hoeffding's inequality and in the last inequality, we use the assumption that $d(V', V^*) \geq c\,(1-2p)\,s^2$. By Lemma 20, we know that with probability most $\exp\left(c\,(1-2p)^3\,s^2/8\right)$,

$$d(\bar{V}, V') = d(\bar{V}, V^*) + |D^{\mathrm{c}}_{V'V^*}| - |D^{\mathrm{n}}_{V'V^*}| \leq d(\bar{V}, V^*) + \epsilon.$$

Since the number of clustering function over V is at most $(ts)^{ts}$, we know that

$$\begin{aligned} \Pr\left(\exists V', d(V', V^*) \geq c\,(1-2p)\,s^2, d(\bar{V}, V') \leq d(\bar{V}, V^*) + \epsilon\right) &\leq (ts)^{ts} \exp\left(-\frac{c\,(1-2p)^3\,s^2}{8}\right) \\ &= \exp\left(ts \log ts - \frac{c\,(1-2p)^3\,s^2}{8}\right). \end{aligned}$$

Thus, with probability at least $1-\exp\left(ts\log ts - c\left(1-2p\right)^3 s^2/8\right)$, for every clustering function V' over V such that $d(V', V^*) \geq c\left(1-2p\right)s^2$, we have

$$d(\bar{V}, V') > d(\bar{V}, V^*) + \epsilon.$$

■

D.3. Proof of Theorem 12

The key part of the proof of Theorem 12, is to show the following three claims.

Claim 3 *In Algorithm 5, if there is some $i \in [k]$ such that $|T_i^*| > s_t$, but $h = 0$, then $d(\tilde{T}, T^*) > s_t^2/8$.*

Proof of Claim. Without loss of generality, we assume that $|T_1^*| > s_t$. We denote by $A_i := \tilde{T}_i \cap T_1^*$. Without loss of generality, we can assume $A_i \neq \emptyset$ if and only if $i \in [\ell]$, where ℓ is a positive integer. We say $\tilde{T}$ makes a negative mistake on (u, v) if $T^*(u, v) = 1$ and $\tilde{T}(u, v) = 0$. It is easy to see that the number of negative mistakes made by $\tilde{T}_i$ over $T_1^* \times T_1^*$ is

$$\sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell}|A_i||A_j|.$$

Since $h = 0$, for every $i \in [\ell]$, $|A_i| \leq s_t/2$. To lower bound the number of negative mistakes, we consider the following family of quadratic programming problems, parameterized by ℓ.

$$\begin{aligned} \min \ & \sum_{i=1}^{\ell}\sum_{j=i+1}^{\ell} x_i x_j \\ \text{s.t.} \ & \sum_{i=1}^{\ell} x_i \geq s_t \\ & 1 \leq x_i \leq \frac{s_t}{2} \quad \forall i \in [\ell]. \end{aligned} \tag{QP(ℓ)}$$

Clearly every choice of $\{A_i\}_{i\in[\ell]}$ is corresponding to a feasible solution to QP(ℓ). Thus, we will show that for every $\ell \geq 2$, the optimal value of QP(ℓ) is at least $s_t^2/8$.

We prove this by induction. For the base case, it is easy to check the optimal value of QP(2) is $s_t^2/4$. Now suppose that the optimal value of QP(ℓ) is at least $s_t^2/8$, we show this also correct for $\ell+1$. Let $x = (x_1, \ldots, x_{\ell+1})$ be a feasible solution to QP($\ell+1$). We consider two cases.

In the first case, there exist $i, j \in [\ell+1]$, such that $y = x_i + x_j \leq s_t/2$. Without loss of generality, we can assume that $i = \ell, j = \ell+1$. Then we know that $x' = (x_1, \ldots, x_{\ell-1}, y)$ is a feasible solution to QP(ℓ). It can be checked that the objective value of x is at least that of x' and thus at least $s_t^2/8$.

In the second case, for every $i, j \in [\ell+1]$, $x_i + x_j > s_t/2$. So we know there is some $i \in [\ell+1]$ such that $s_t/4 < x_i \leq s_t/2$. This implies the objective value of x is at least $s_t^2/8$.

Thus, by induction the number of negative mistakes is at least $s_t^2/8$. So we know

$$d(\tilde{T}, T^*) \geq \frac{s_t^2}{8},$$

as long as $h = 0$. ◇

Claim 4 *In Algorithm 5, if there is some $i \in [k]$ such that $|\tilde{T}_i| > s_t/2$ and $\tilde{T}_i$ is an η-bad set, where $\eta = 1/4 + p/2$, then $d(\tilde{T}, T^*) > (1-2p)s_t^2/64$.*

Proof of Claim. We consider separately two cases. In the first case, we assume that for every $j \in [k]$, $|\tilde{T}_i \cap T_j^*| \leq |\tilde{T}_i|/4$. Let $S := \cup_{j=1}^{i^*} \tilde{T}_i \cap T_j^*$, where i^* is the largest index such that $|S| \leq |\tilde{T}_i|/2$. Thus we know $|\tilde{T}_1 \setminus S| \geq |\tilde{T}_i|/2$. By the choice of i^*, we know that $|S| \geq |\tilde{T}_i|/4$. So every point pair (u, v) such that $u \in S$ and $v \in \tilde{T}_i \setminus S$ is labeled 1 by $\tilde{T}$ but labeled 0 by T^*. The total number of such point pairs is at least $|\tilde{T}_i|^2/8 > s_t^2/32$.

In the second case, we assume that there is some $j \in [k]$ such that $|\tilde{T}_i \cap T_j^*| > |\tilde{T}_i|/4$. We know $|\tilde{T}_i \setminus T_j^*| \geq (1-2p)|\tilde{T}_i|/4$, since $\tilde{T}_i$ is an η-bad set. We notice that every point pair (u, v) such that $u \in \tilde{T}_i \cap T_j^*$ and $v \in \tilde{T}_i \setminus T_j^*$ is labeled 1 by $\tilde{T}$ but labeled 0 by T^*. The total number of such point pairs is at least $(1-2p)|\tilde{T}_i^2|/16 > (1-2p)s_t^2/64$. ◇

Claim 5 *In Algorithm 5, if there is some $i, j, \ell \in [k]$, and $i \neq j$ such that $|\tilde{T}_i|, |\tilde{T}_j| > s_t/2$ and $\tilde{T}_i, \tilde{T}_j$ are both (η, V_ℓ^*)-biased sets, then $d(\tilde{T}, T^*) > (1-2p)s_t^2/16$.*

Proof of Claim. We notice that for every point pair (u, v) such that $u \in \tilde{T}_i \cap V_\ell^*$ and $v \in \tilde{T}_j \cap V_\ell^*$, (u, v) is labeled 0 by $\tilde{T}$ but is labeled 1 by T^*. The total number of such point pairs is at least $s_t^2/16$, since $\tilde{T}_i, \tilde{T}_j$ are both (η, V_ℓ^*)-biased sets and $|\tilde{T}_i|, |\tilde{T}_j| > s_t/2$. Thus, we have $d(\tilde{T}, T^*) > (1-2p)s_t^2/16$. ◇

Now we are able to use the above claims to prove Theorem 12.

Proof We first apply Theorem 19 on the sample set T with $s = s_t, \epsilon = c_1 (ts_t)^{\frac{3}{2}} / (1-2p)$ and $c = 1/64$, where c_1 is a constant that satisfies Theorem 11. We first show that the choice of parameter satisfies the statement of Theorem 19. On the one hand, we have

$$\epsilon = \frac{c_1 (ts_t)^{\frac{3}{2}}}{(1-2p)} = \frac{c_1 (c')^{\frac{3}{2}} t^6 \log^{\frac{3}{2}} n}{(1-2p)^{10}}.$$

On the other hand, we have

$$\frac{c(1-2p)^2 s_t^2}{2} = \frac{(1-2p)^2 s_t^2}{128} = \frac{(c')^2 t^6 \log^2 n}{128 (1-2p)^{10}} > \epsilon,$$

because c' is a large enough constant. So with probability at least

$$1 - \exp\left(ts_t \log ts_t - \frac{(1-2p)^3 s_t^2}{1024}\right) \geq 1 - \exp\left(-\left((c')^2 - \frac{c'}{1024}\right) \frac{t^6 \log^{\frac{3}{2}} n}{(1-2p)^9}\right) \geq 1 - \frac{1}{\text{poly}(n)}, \tag{6}$$

any clustering T' such that $d(T', T^*) > (1-2p)s_t^2/64$ will satisfy

$$d(\bar{T}, T') > d(\bar{T}, T^*) + \frac{c_1 (ts_t)^{\frac{3}{2}}}{(1-2p)}.$$

Here in (6), the first inequality follows by $\log ts_t \le \sqrt{ts_t}$ and the second inequality holds because c' is a large enough constant.

By Claim 3, Claim 4 and Claim 5, we know that if any one of the events in the statement of Theorem 12 does not happen, we will have $d(T', T^*) > (1-2p)s_t^2/64$. However, by Theorem 11, we know that with probability at least $1 - 1/\text{poly}(n)$, we have

$$d(\bar{T}, \tilde{T}) \le d(\bar{T}, T^*) + \frac{c_1 (ts_t)^{\frac{3}{2}}}{(1-2p)}.$$

This implies $d(\tilde{T}, T^*) \le (1-2p)\, s_t^2/64$, with probability at least $1 - 1/\text{poly}(n)$. And thus, the three events must happen together. ■

D.4. Proof of Theorem 5

We first prove the correctness of Algorithm 4. Let $C = \{\tilde{V}_1, \ldots, \tilde{V}_\ell\}$ be the output of Algorithm 4. We first show each element in C is an underlying cluster. We know each $\tilde{V}_i = \{v \in V \mid \text{Test}(v, B_i) = \text{“Yes”}\}$. Also, we know $B_i \subseteq \hat{T}_i$, where $\hat{T}_i \in \text{APPROXCORRELATIONCLUSTER}(T, 1/\text{poly}(n))$ in a certain stage of the algorithm and $|\hat{T}_i| \ge s_t/2$. According to Theorem 5, we know with probability at least $1 - 1/\text{poly}(n)$, $\hat{T}_i$ is an (η, V_i^*)-biased set, where $\eta = \frac{1}{4} + \frac{p}{2}$. By Hoeffding's inequality, by setting $\eta' = \frac{p+1}{3}$, we know

$$\Pr\left(B_i \text{ is not an } (\eta', V_i^*)\text{-biased set}\right) \le \exp\left(-2|B_i|\left(\frac{1-2p}{12}\right)^2\right) \le 1/\text{poly}(n).$$

So with probability at least $1 - 1/\text{poly}(n)$, B_i is an (η', V_i^*)-biased set. According to Lemma 9, with probability at least $1 - 1/\text{poly}(n)$, we have

$$\tilde{V}_i = \{v \in V \mid \text{Test}(v, B_i) = \text{“Yes”}\} = V_i^* \cap V = V_i^*.$$

Here the last equality follows by the fact that $V_i^* \subseteq V$ at the time when B_i is created. This is because no point in V_i^* is put into other underlying clusters before B_i is created. So each element in C is an underlying cluster.

It remains to show every underlying cluster of size $\Omega\left(\frac{k^4 \log n}{(1-2p)^6}\right)$ must be recovered with high probability. Suppose there is some underlying cluster V_i^* such that $|V_i^*| \ge 2ks_{2k} = \Omega\left(\frac{k^4 \log n}{(1-2p)^6}\right)$ not recovered by Algorithm 4. Then at the end of the algorithm, $V_i^* \subseteq V$ and $|V| \ge 2ks_{2k}$. Assume $|C| = h < k$. Then as long as $k - h \le t < 2k$, the sampled set T of size ts_t must contain some underlying cluster of size $\frac{t}{h-t} s_t \ge s_t$. By Theorem 12, with probability at least $1 - 1/\text{poly}(n)$, we will update C again. However, at this time the algorithm has terminated. This gives a contradiction. So every underlying cluster of size at least $O\left(\frac{k^4 \log n}{(1-2p)^6}\right)$ must be recovered by Algorithm 5.

We next prove the sample complexity of Algorithm 4. To show this, we first show that every time we invoke Algorithm 5, we must have parameter $t < 2k$. Suppose $t \geq 2k$, we know that $t/2 \geq k$. Since T is partitioned into at most k underlying clusters, we know that there must be at least one $i \in [k]$ such that $|T_i^*| \geq ts_{t/2}/2k \geq s_{t/2}$. According to Theorem 12, with probability at least $1 - 1/\text{poly}(n)$, $h > 0$. In this case, we will not invoke Algorithm 5 after updating $t/2$ by t. This implies every time we invoke Algorithm 5, we query $O\left(|T|^2\right) = O\left(\frac{k^8 \log^2 n}{(1-2p)^{12}}\right)$ times and in each round we will call Algorithm 5 $O(\log k)$ times. Since there are at most k rounds, the number of queries we spend on Algorithm 5 is $O\left(\frac{k^9 \log k \log^2 n}{(1-2p)^{12}}\right)$.

Next, we see each time we update C, we invoke Algorithm 1 at most n times and each time we query $O\left(\frac{\log n}{(1-2p)^2}\right)$ times. Since we update C at most k times, the number of queries we spend on updating C is $O\left(\frac{nk \log n}{(1-2p)^2}\right)$. So the query complexity of Algorithm 4 is $O\left(\frac{nk \log n}{(1-2p)^2} + \frac{k^9 \log k \log^2 n}{(1-2p)^{12}}\right)$. ■

Appendix E. Missing proof in Section 5

E.1. Proof of Theorem 6

It is sufficient to show with high probability $\bar{p} \geq p$ and $(1 - 2\bar{p}) = O((1 - 2p))$, since in this case $\bar{p}$ is an appropriate upper bound of p and we can use Algorithm 4 to solve the problem. We can assume $(1 - 2p)^4 \geq \frac{10 \log n}{n}$, because if $(1 - 2p)^4 < \frac{10 \log n}{n}$, there is no underlying cluster of size $\Omega\left(\frac{k^4 \log n}{(1-2p)^6}\right)$ and Theorem 6 holds naturally.

We first analyze the set A. For a given point v, denote by V_v^* the underlying cluster that v belongs to. We first show for every sampled set A, either there are two points u, v in the same underlying cluster or there are two points u, v such that $|V_u^*| + |V_v^*| \leq \frac{n}{4}$. For simplicity, we say a such a point pair is good. Let S be the set of points w such that $|V_w^*| \leq \frac{n}{8}$. We know there are at most 8 underlying clusters that have size more than $\frac{n}{8}$. We can without loss of generality assume they are $V_1^*, \ldots, V_i^*$, $i \leq 8$. Since $|A| = 9$, we know there must be two points in S or in the same underlying cluster. In the first case, we have $|V_u^*| + |V_v^*| \leq \frac{n}{4}$, according to the definition of S. So A must contain a good pair.

Next, we show with probability at least $1 - 1/\text{poly}(n)$, $\bar{p} \geq p$. Denote by $\delta = 1 - 2p$ and $\bar{\delta} = 1 - 2\bar{p}$. For every $u, v \in A$ such that u, v in same underlying cluster, we have

$$\mathbf{E}\text{count}_{uv} = 2p(1-p)|V| = 2p(1-p)n = \frac{1-\delta^2}{2}n \leq \left(\frac{1}{2} - \frac{\delta^2}{4}\right)n.$$

On the other hand, for every $u, v \in A$ such that u, v in different underlying clusters, but $|V_u^*| + |V_v^*| \leq \frac{n}{4}$, we have

$$\mathbf{E}\text{count}_{uv} = 2p(1-p)|V| + (1-2p)^2\left(|V_u^*| + |V_v^*|\right) \leq \left(\frac{1}{2} - \frac{\delta^2}{4}\right)n.$$

This implies if point pair (u, v) is good, then

$$\left(\frac{1}{2} - \frac{\delta^2}{2}\right)n \leq \mathbf{E}\text{count}_{uv} \leq \left(\frac{1}{2} - \frac{\delta^2}{4}\right)n.$$

In particular, the lower bound holds for every $u, v \in A$.

Let (u, v) be a point pair in A, by Hoeffding's inequality, we know that

$$\Pr\left(\text{count}_{uv} \le \frac{1-4\delta^2}{2}n\right) \le \exp\left(-2\left(\frac{3\delta^2}{2}n\right)^2\frac{1}{n}\right) = \exp\left(-\frac{9}{2}\delta^4 n\right).$$

By union bound, we know that

$$\Pr\left(M \le \frac{1-4\delta^2}{2}n\right) \le \Pr\left(\exists u \ne v \in A, \text{count}_{uv} \le \frac{1-4\delta^2}{2}n\right) \le n\exp\left(-\frac{9}{2}\delta^4 n\right) \le \frac{1}{n^{44}},$$

where the last inequality follows by $\delta^4 \ge \frac{10\log n}{n}$.

So with probability at least $1 - 1/n^{44}$, we have

$$\bar{p} := \frac{1}{2} - \frac{1}{4}\sqrt{1-\frac{2M}{n}} > \frac{1}{2} - \frac{1}{4}\sqrt{1-\frac{2}{n}\frac{1-4\delta^2}{2}n} = \frac{1}{2}(1-\delta) = p,$$

where the inequality follows by $M > \frac{1-4\delta^2}{2}\sqrt{n}$. Since $\bar{p} > p$, and we know that the fully-random model with parameter p is a special case of the semi-random model with parameter $\bar{p}$, we know that with probability at least $1 - 1/\text{poly}(n)$, Algorithm 4 will recover all clusters of size $\Omega\left(\frac{k^4\log n}{(1-2\bar{p})^6}\right)$ and the query complexity is $O\left(\frac{nk\log n}{(1-2\bar{p})^2} + \frac{k^9\log k\log^2 n}{(1-2\bar{p})^{12}}\right)$.

And it remains to show $\bar{p}$ is not too larger than p, so that we get the correct query complexity. We will show that with probability at least $1 - 1/\text{poly}(n)$, we have $\bar{\delta} > \delta/4$, which implies that $1/(1-2\bar{p}) \le 4/(1-2p)$. Let u, v be a good point pair in A. We have

$$\Pr\left(M \ge \frac{1-\frac{1}{4}\delta^2}{2}n\right) \le \Pr\left(\text{count}_{uv} \ge \frac{1-\frac{1}{4}\delta^2}{2}n\right) \le \exp\left(-2\left(\frac{1}{8}\delta^2 n\right)^2\frac{1}{n}\right) = \exp\left(-\frac{1}{32}\delta^4 n\right).$$

Thus, with high probability we have

$$\bar{\delta} = \frac{1}{2}\sqrt{1-\frac{2M}{n}} > \frac{1}{2}\sqrt{1-\frac{2}{n}\frac{1-\frac{1}{4}\delta^2}{2}n} = \frac{\delta}{4}.$$

■