Differentially Private Hierarchical Clustering with Provable Approximation Guarantees

Jacob Imola [1] **Alessandro Epasto** [2] **Mohammad Mahdian** [2] **Vincent Cohen-Addad** [2] **Vahab Mirrokni** [2]

Abstract

Hierarchical Clustering is a popular unsupervised machine learning method with decades of history and numerous applications. We initiate the study of *differentially private* approximation algorithms for hierarchical clustering under the rigorous framework introduced by Dasgupta (2016). We show strong lower bounds for the problem: that any ϵ-DP algorithm must exhibit $O(|V|^2/\epsilon)$-additive error for an input dataset V. Then, we exhibit a polynomial-time approximation algorithm with $O(|V|^{2.5}/\epsilon)$-additive error, and an exponential-time algorithm that meets the lower bound. To overcome the lower bound, we focus on the stochastic block model, a popular model of graphs, and, with a separation assumption on the blocks, propose a private $1 + o(1)$ approximation algorithm which also recovers the bottom-level blocks exactly. Finally, we perform an empirical study of our algorithms and validate their performance.

1. Introduction

Hierarchical Clustering is a staple of unsupervised machine learning with more than 60 years of history (Ward Jr, 1963). Contrary to *flat* clustering methods (such as k-means, Jain (2010)), which provide a single partitioning of the data, *hierarchical* clustering algorithms produce a recursive refining of the partitions into increasingly fine-grained clusters. The clustering process can be described by a tree (or dendrogram), and the objective of the tree is to cluster the most similar items in the lowest possible clusters, while separating dissimilar items as high as possible.

The versatility of such methods is apparent from the widespread use of hierarchical clustering in disparate areas of science, such as social networks analysis (Leskovec et al., 2014; Mann et al., 2008), bioinformatics (Diez et al., 2015), phylogenetics (Sneath & Sokal, 1962; Jardine & Sibson, 1968), gene expression analysis (Eisen et al., 1998), text classification (Steinbach et al., 2000) and finance (Tumminello et al., 2010). Popular hierarchical clustering methods (such as linkage (Jain, 2010)) are commonly available in standard scientific computing packages (Virtanen et al., 2020) as well as large-scale production systems (Bateni et al., 2017; Dhulipala et al., 2022).

Despite the fact that many of these applications involve private and sensitive user data, all research on hierarchical clustering (with few exceptions (Kolluri et al., 2021; Xiao et al., 2014) discussed later) has ignored the problem of defining *privacy-preserving* algorithms. In particular, to the best of our knowledge, no work has provided *differentially-private (DP)* (Dwork et al., 2014a) algorithms for hierarchical clustering with provable approximation guarantees.

In this work, we seek to address this limitation by advancing the study of differentially-private approximation algorithms for hierarchical clustering under the rigorous optimization framework introduced by Dasgupta (2016). This celebrated framework introduces an objective function for hierarchical clustering (see Section 3 for a formal definition) formalizing the goal of clustering similar items lower in the tree.

Our algorithms are edge-level *Differentially Private (DP)* on an input similarity graph, which is relevant when edges of the input graph represents sensitive user information. Designing an edge-level DP algorithm requires proving that the algorithm is insensitive to changes to a single edge of the similarity graph. As we shall see, this is especially challenging for hierarchical clustering. In fact, commonly-used hierarchical clustering algorithms (such as linkage-based ones (Jain, 2010)) are *deterministically* sensitive to a single edge, thus leaking directly the input edges. Moreover, as we show, strong inapproximability bounds exist for Dasgupta's objective under differential privacy, highlighting the technical difficulty of the problem.

Main contributions First, we show in Section 4 that no edge-level ϵ-DP algorithm (even with exponential time)

[1] Department of Computer Science and Engineering, UCSD, La Jolla, USA. Work partially done while an intern at Google. [2] Google, New York City, USA. Correspondence to: Jacob Imola <jimola@eng.ucsd.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

exists for Dasgupta's objective with less than $O(|V|^2/\epsilon)$ additive error. This prevents defining private algorithms with meaningful approximation guarantees for *arbitrary* sparse graphs.

Second, on the positive side, we provide the first polynomial time, edge-level approximation algorithm for Dasguta's objective with $O(|V|^{2.5}/\epsilon)$ additive error and multiplicative error matching that of the best non-private algorithm (Agarwal et al., 2022). This algorithm is based on recent advances in private cut sparsifiers (Eliáš et al., 2020). Moreover, we show an (exponential time) algorithm with $O(|V|^2 \log n/\epsilon)$ additive error, almost matching the lower bound.

Third, given the strong lower bounds, in Section 6 we focus on a popular model of graphs with a planted hierarchical clustering based on the *Stochastic Block Model (SBM)* (Cohen-Addad et al., 2017). For such graphs, we present a private $1 + o(1)$ approximation algorithm recovering almost exactly the hierarchy on the blocks. Our algorithm uses, as a black-box, any reconstruction algorithm for the stochastic block model.

Fourth, we introduce a practical and efficient DP SBM community reconstruction algorithm (Section 6). This algorithm is based on perturbation theory of graph spectra combined with dimensionality reduction to avoid adding high noise in the Gaussian mechanism. Combined with our clustering algorithm, this results in the first private approximation algorithm for hierarchical clustering in the hierarchical SBM.

Finally, we show in Section 7 that this algorithm can be efficiently implemented and works well in practice.

2. Related Work

Our work spans the areas of differential privacy, hierarchical clustering and community detection in stochastic block model. For a complete discussion, see Appendix A.

Graph algorithms under DP Differential privacy (Dwork et al., 2006) has recently the gold standard of privacy. We refer to Dwork et al. (2014a) for a survey. Relevant to this work is the area of differential privacy in graphs. Definitions based on edge-level (Epasto et al., 2022; Eliáš et al., 2020) and node-level (Kasiviswanathan et al., 2013) privacy have been proposed. The most related work is that on graph cut approximation (Eliáš et al., 2020; Arora & Upadhyay, 2019), as well as that of private correlation clustering (Bun et al., 2021; Cohen-Addad et al., 2022c).

Hierarchical Clustering Until recently, most work on hierarchical clustering were heuristic in nature, with the most well-known being the linkage-based ones (Jain, 2010; Bateni et al., 2017). Dasgupta (2016) introduced a combinatorial objective for hierarchical clustering which we study in this paper. Since this work, many authors have designed algorithms for variants of the problem with no privacy (Cohen-Addad et al., 2017; 2019; Charikar & Chatziafratis, 2017; Moseley & Wang, 2017; Agarwal et al., 2022; Chatziafratis et al., 2020).

Limited work has been devoted to DP hierarchical clustering algorithms. One paper (Xiao et al., 2014) initiates private clustering via MCMC methods, which are not guaranteed to be polynomial time. Follow-up work (Kolluri et al., 2021) shows that sampling from the Boltzmann distribution (essentially the exponential mechanism (McSherry & Talwar, 2007) in DP) produces an approximation to the maximization version of Dasgupta's function, which is a different problem formulation. Again, this algorithm is not provably polynomial time.

Private flat clustering Contrary to hierarchical clustering, the area of private *flat* clustering on metric spaces has received large attention. Most work in this area has focused on improving the privacy-approximation trade-off (Ghazi et al., 2020; Balcan et al., 2017) and on efficiency (Hegde et al., 2021; Cohen-Addad et al., 2022b;a).

Stochastic block models The Stochastic Block Model (SBM) is a classic model for random graphs with planted partitions which has received a significant attention in the literature (Guédon & Vershynin, 2016; Montanari & Sen, 2016; Moitra et al., 2016; Fei & Chen, 2020; Ding et al., 2022; Liu & Moitra, 2022). For our work, we focus on a variant which has nested ground-truth communities arranged in hierarchical fashion. This model has received attention for hierarchical clustering (Cohen-Addad et al., 2017).

The study of private algorithms for SBMs is instead very recent. One of the only results known for private (non-hierarchical) SBMs is the work of Seif et al. (2022) which provides quasi-polynomial time community detection algorithms for some regimes of the model. Finally, concurrently to our work, the manuscript of Chen et al. (2023) provides strong approximation guarantees using semi-definite programming for recovering SBM communities. Community detection is a distinct problem from hierarchical clustering, and this work is independent of ours.

The connection between existing work in the SBM and ours is that, in Section 6, we design a hierarchical clustering algorithm (Algorithm 1) which uses community detection as a black-box. Moreover, we show a novel algorithm for hierarchical SBM community detection (Algorithm 2), independent of Chen et al. (2023), which is of practical interest because it uses SVDs, instead of semidefinite programming, and thus does not have a large polynomial run-time.

3. Preliminaries

Our results involve the key concepts of hierarchical clustering and differential privacy. We define these two concepts in the next sections.

3.1. Hierarchical Clustering

Hierarchical clustering seeks to produce a tree clustering a set V of n items by their similarity. It takes as input an undirected graph $G = (V, E, w)$, where $E \subseteq V \times V$ is the set of edges and $w : V \times V \to \mathbb{R}^+$ is a weight function indicating similarity; i.e. a higher $w(u, v)$ indicates u, v are more similar. We extend the weight function w and say that $w(u, v) = 0$ if $w(u, v) \notin E$.

A hierarchical clustering (HC) of G is a tree T whose leaves are V. The tree can be viewed as a sequence of merges of subtrees of T, with the final merge being the root node. A good hierarchical clustering merges more similar items closer to the bottom of the tree. The cost function $\omega_G(T)$ of Dasgupta (Dasgupta, 2016), captures this intuition. We have

$$\omega_G(T) = \sum_{(u,v)\in V^2} w(u, v)|\text{leaves}(T[u \wedge v])|, \tag{1}$$

where $T[u \wedge v]$ indicates the smallest subtree containing u, v in T and $|\text{leaves}(T[u \wedge v])|$ indicates the number of leaves in this subtree. This cost function charges a tree T for each edge based on the similarity $w(u, v)$ and how many leaves are in the subtree in which it is merged.

Additional Notation We let $\omega^*_G = \min_T \omega_G(T)$ denote the best possible cost attained by any tree T. We write $w(A, B) = \sum_{a\in A,b\in B} w(a, b)$ and we say $w(G) = w(G, G)$. Let $\mathcal{A}(G)$ be a hierarchical clustering algorithm. We say $\mathcal{A}$ is an (a_n, b_n)-approximation if

$$\mathbb{E}[\omega_G(\mathcal{A}(G))] \leq a_n\omega^*_G + b_n, \tag{2}$$

where the expectation is over the random coins of $\mathcal{A}$. If an algorithm is a $(a_n, 0)$-approximation algorithm, we often refer to it as simply an a_n-approximation.

3.2. Differential Privacy

For hierarchical clustering we use the notion of graph privacy known as edge differential privacy. Intuitively, our private algorithm behaves similarly whether or not the adjacency matrix of G is altered in L_1 distance by up to 1. Specifically, we say $G = (V, E, w)$ and $G' = (V, E', w')$ are *adjacent graphs* if $\sum_{u,v\in V} |w(u, v) - w'(u, v)| \leq 1$, meaning that the adjacency matrices have L_1 distance at most one [1]. This notion has been used before by Eliáš et al. (2020); Blocki et al. (2012) and it has many real-world applications, such as when the graph is a social network and the edges between users encode relationships between them (Epasto et al., 2022). The definition of edge-DP is as follows:

Definition 1. *An algorithm $\mathcal{A} : \mathcal{G} \to \mathcal{Y}$ satisfies (ϵ, δ)-edge DP if, for any $G = (V, E, w), G' = (V, E', w')$ that are adjacent, and any set of trees $\mathcal{T}$,*

$$\Pr[\mathcal{A}(G') \in \mathcal{T}] \leq e^{\epsilon} \Pr[\mathcal{A}(G) \in \mathcal{T}] + \delta.$$

Edge DP states that given any output $\mathcal{T}$ of $\mathcal{A}$, it is provably hard to tell whether an adjacent G or G' was used. For 0/1 weighted graphs, Definition 1 is equivalent to standard edge DP for unweighted graphs (c.f. Definition 2.2.1 in (Pinot, 2018)).

4. Lower Bounds

We show that for the both objective functions considered, there are unavoidable lower bounds on the objective function for any differentially private algorithm. Our theorem applies a packing-style argument (Hardt & Talwar, 2010), in which we construct a large family $\mathcal{F}$ of graphs such that no tree can cluster more than one graph in $\mathcal{F}$ well. However, a DP algorithm $\mathcal{A}$ is forced to place mass on all trees. This limits its utility as significant mass must be placed on trees which do not cluster the input graphs well. Formally, we prove the following theorem:

Theorem 1. *For any $\epsilon \leq \frac{1}{20}$ and n sufficiently large, let $\mathcal{A}(G)$ be a hierarchical clustering algorithm which satisfies ϵ-edge differential privacy. Then, there is a weighted graph G with $\omega^*_G \leq O(\frac{n}{\epsilon})$ such that*

$$\mathbb{E}[\omega_G(\mathcal{A}(G))] \geq \Omega(\tfrac{n^2}{\epsilon}).$$

We prove this theorem in Section 4.1; we discuss the implications of the theorem here. Since there exists a graph such that $\omega^*_G \leq O(\frac{n}{\epsilon})$, yet $\omega_G(\mathcal{A}(G)) \geq \Omega(\frac{n^2}{\epsilon})$, this means that no differentially private algorithm $\mathcal{A}$ can be a $(O(n^\alpha), O(\frac{n^{2\alpha}}{\epsilon}))$ approximation to hierarchical clustering for any $\alpha < 1$. It is possible for $\mathcal{A}$ to be a $(1, O(\frac{n^2}{\epsilon}))$-approximation— in this case, for graphs with W total weight, it easy to see that $\omega^*_G \leq O(nW)$ and can be as small as $O(W)$. Thus, it is necessary for W to be much bigger than $\frac{n}{\epsilon}$, meaning that G cannot be too sparse.

4.1. Proof of Theorem 1

To construct our lower bound, we consider the family of graphs $\mathcal{P}(n, 5)$ consisting of $\frac{n}{5}$ cycles of size 5. We observe the following facts:

- Each $G \in \mathcal{P}(n, 5)$ has n edges. Thus, any $G_1, G_2 \in \mathcal{P}(n, 5)$ differ in at most $2n$ edges.

[1] the constant one may be changed to any constant to match the application, and our results carry over easily.

- For any $G \in \mathcal{P}(n, 5)$, any binary tree which splits the graph into its cycles before splitting any edges in the cycles incurs a cost of at most $\frac{n}{5} W_5$, where $W_5 = \omega^*_{C_5} \leq 18$.

It will be convenient to use the following definition:

Definition 2. *For a graph G, a* balanced cut *is partition (A, B) of V such that $\frac{n}{3} \leq |A|, |B| \leq \frac{2n}{3}$.*

Any hierarchical clustering T can be mapped to a balanced cut on G in the following way:

Definition 3. *For a binary tree T whose leaves are V, let the sequence $N_0, N_1, \ldots, N_r$ denote a recursive sequence of internal nodes such that N_0 is the root node, and N_i is child of N_{i-1} with more leaves in its subtree. Finally, N_r is the first node in the sequence with fewer than $\frac{2n}{3}$ leaves in its subtree. Then, the balanced cut (A, B) induced by T is the partition $(leaves(N_r), V \setminus leaves(N_r))$.*

It is easy to see that (A, B) in the above definition is indeed a balanced cut of G, and for any edge (u, v) crossing (A, B), we have $|\text{leaves}(T[u \wedge v])| \geq \frac{2n}{3}$.

Our class $\mathcal{C}$ of graphs is a subset of $\mathcal{P}(n, 5)$ for which no tree clusters more than one element of $\mathcal{C}$ well. We characterize a condition for which a tree T definitely does not cluster $G \in \mathcal{P}(n, 5)$ well:

Definition 4. *For a binary tree T, let (A, B) be its balanced cut. We say (A, B)* misses *a cycle $C \subseteq G$ if at least one vertex of C lies in A and at least one vertex lies in B.*

Now, we show that if T misses many cycles in its balanced cut, it must incur high cost.

Lemma 1. *For a graph $G \in \mathcal{P}(n, 5)$, let T be a HC with balanced cut (A, B), and suppose that B misses at least $\alpha \frac{n}{5}$ of the cycles in G, for $0 < \alpha \leq 1$. Then,*

$$\omega_G(T) \geq \frac{4\alpha}{15} n^2.$$

Proof: From the given information, we have that $w(A, B) \geq 2\alpha \frac{n}{5}$, as a missed cycle implies at least two edges are cut. Thus,

$$\begin{aligned}\omega_G(T) &\geq \sum_{u \in A, v \in B} w(u, v) |\text{leaves}(T[u \wedge v])| \\ &\geq \tfrac{2n}{3} w(A, B) \geq \tfrac{4\alpha}{15} n^2. \quad \square\end{aligned}$$

We generate graphs from $\mathcal{P}(n, 5)$ at random, showing that the probability that there exists a balanced cut (A, B) which misses few cycles in both G_1, G_2 is exponentially small. This will allow us to generate a large family of graphs such that no balanced cut misses few cycles in more than one graph. This results in the following lemma—in the following, let $\mathcal{B}(G, r) = \{T \in \mathcal{T}_n : \omega_G(T) < r\}$.

Lemma 2. *For n sufficiently large, there exists a family $\mathcal{F} \subseteq \mathcal{P}(n, 5)$ of size $2^{0.2n}$ such that $\mathcal{B}(G, r) \cap \mathcal{B}(G', r) = \emptyset$ for any $G, G' \in \mathcal{F}$ with $r = \frac{n^2}{400}$.*

The proof of this lemma appears in Appendix B. Thus, no tree can cluster more than one of our random graphs well, and we can apply the packing argument to obtain Theorem 1. We prove it as follows.

Proof of Theorem 1: Let $\mathcal{F}$ be the set of graphs guaranteed by Lemma 2. We have $|\mathcal{F}| = 2^{0.2n}$. Let $\mathcal{F}_W$ contain the same graphs of $\mathcal{F}$, but with each edge weighted by a positive integer W satisfying $0.02 \leq \epsilon W < 0.07$. Each $G, G' \in \mathcal{F}$ differs by up to $2n$ edges, and applying group privacy W times, we have that an algorithm A which satisfies ϵ-DP satisfies $2nW\epsilon$-DP on the graphs in $\mathcal{F}_W$.

Now, suppose A satisfies $\mathbb{E}[\text{cost}_G(A(G))] < \frac{W}{800} n^2$ for any $G \in \mathcal{F}_W$. This implies $\Pr[\text{cost}_G(A(G)) \in \mathcal{B}(G, \frac{W}{400} n^2)] \geq \frac{1}{2}$ for all $G \in \mathcal{F}_W$. However, we know these balls are disjoint because of the disjointness property on $\mathcal{F}$. Furthermore, we have that $\Pr[A(G) \in \mathcal{B}(G', \frac{W}{400} n^2)] \geq e^{-2nW\epsilon} \frac{1}{2} > 2^{-0.2n}$ for all $G' \in \mathcal{F}_W$.

$$\begin{aligned}1 &\geq \sum_{G' \in \mathcal{F}_W} \Pr[A(G) \in \mathcal{B}(G', \tfrac{W}{400} n^2)] \\ &> 2^{0.2n} 2^{-0.2n} = 1.\end{aligned}$$

This is a contradiction, and thus the algorithm A must have error higher than $\frac{W}{800} n^2 \geq \Omega(\frac{n^2}{\epsilon})$ on some graph. □

5. Algorithms for Private Hierarchical Clustering

In this section, we design private algorithms for hierarchical clustering which work on any input graph. In Section 5.1, we propose a polynomial time $(\alpha, O(\frac{n^{2.5}}{\epsilon}))$ approximation algorithm, where α is the best approximation ratio of a black-box, *non-private* hierarchical clustering algorithm. Then, in Section 5.2, we show that the exponential mechanism is a $(1, O(\frac{n^2 \log n}{\epsilon}))$-approximation algorithm, implying our lower bound is tight. The proofs of the results in this section appear in Appendix C.2

5.1. Polynomial-Time Algorithm

Our algorithm makes use of a recent algorithm which releases a sanitized, synthetic graph G' that approximates the cuts in the private graph G (Eliáš et al., 2020; Arora & Upadhyay, 2019). Via post-processing, it is then possible to run a non-private, black-box clustering algorithm. We are able to relate the cost in G' to that of G by reducing the cost $\omega_G(T)$ to a sum of cuts. We start by defining the notion of G' approximating the cuts in G.

Definition 5. *For a given graph $G = (V, E, w)$, we say $G' = (V, E', w')$ is an (α_n, β_n)-approximation to cut queries in G if for all $S \subseteq V$, we have*

$$\begin{aligned}&(1-\alpha_n)w(S,\overline{S}) - \beta_n \min\{|S|, n-|S|\}\\ &\le w'(S,\overline{S}) \le (1+\alpha_n)w(S,\overline{S}) + \beta_n \min\{|S|, n-|S|\}.\end{aligned}$$

As we alluded, earlier work shows that it is possible to release an $(\tilde{O}(\frac{1}{\epsilon\sqrt{n}}), \tilde{O}(\frac{\sqrt{n}}{\epsilon}))$-approximation to cut queries while satisfying differential privacy. Using this result, we are able to run any blackbox hierarchical clustering algorithm, and by post-processing, the final clustering T' will still satisfy privacy. Even though T' is computed only viewing G', we are able to relate $\omega_G(T')$ to ω^*_G using the fact that G' approximates the cuts in G, and a decomposition of $\omega_{G'}(T')$ into a sum of cuts. This idea recently appeared in Agarwal et al. (2022), and is a critical component of our theorem. In the end, we obtain the following:

Theorem 2. *Given an $(a_n, 0)$-approximation to the cost objective of hierarchical clustering, there exists an (ϵ, δ)-DP algorithm which, with probability at least 0.8, is a $((1 + o(1))a_n, O(n^{2.5}\frac{\log^2 n \log^2 \frac{1}{\delta}}{\epsilon}))$-approximation algorithm to the cost objective.*

Plugging in a state-of-the-art, $\sqrt{\log n}$ hierarchical clustering algorithm of Charikar & Chatziafratis (2017), we obtain a $((1+o(1))\sqrt{\log n}, \tilde{O}(\frac{n^{2.5}}{\epsilon}))$-approximation. In a graph with total edge weight W, we have $W \le \omega_G(T) \le nW$, and thus an approximation is possible if $W > \frac{n^{1.5}}{\epsilon}$. This means the graph can have an average degree of $\frac{\sqrt{n}}{\epsilon}$.

5.2. Exponential Mechanism

We consider an algorithm based on the well-known exponential mechanism (McSherry & Talwar, 2007). This algorithm takes exponential time, but achieves greater performance that is nearly tight with our lower bound (showing that the lower bound can't be improved significantly from an information-theoretic point of view).

The exponential mechanism $M : \mathcal{X} \to \mathcal{Y}$ releases an element from $\mathcal{Y}$ with probability proportional to

$$\Pr[M(X) = Y] \propto e^{\epsilon u_X(Y)/(2S)},$$

where $u_X(Y)$ is a utility function, and $S = \max_{X,X',Y} |u_X(Y) - u_{X'}(Y)|$ is the sensitivity of the utility function in X. This ubiquitous mechanism satisfies $(\epsilon, 0)$-DP.

In our setting, we use the utility function $u_G(T) = -\omega_G(T)$. The sensitivity is bounded in the following fact.

Fact 1. *For two adjacent input graphs $G = (V, E, w)$ and $G' = (V, E, w')$, we have for all trees T that $|\omega_G(T) - \omega_{G'}(T)| \le n$.*

Proof: We can write the difference as as

$$\begin{aligned}&|\omega_G(T) - \omega_{G'}(T)|\\ &= \left|\textstyle\sum_{u,v\in V^2}(w(u,v) - w'(u,v))|\texttt{leaves}(T[u\wedge v])|\right|\\ &\le \textstyle\sum_{u,v\in V^2}|w(u,v) - w'(u,v))| \cdot |\texttt{leaves}(T[u\wedge v])|\\ &\le n\textstyle\sum_{u,v\in V^2}|w(u,v) - w'(u,v)| \le n. \quad \square\end{aligned}$$

Having controlled the sensitivity, we can apply utility results for the exponential mechanism.

Lemma 3. *There exists an $(\epsilon, 0)$-DP, $(1, O(\frac{n^2 \log n}{\epsilon}))$-approximation algorithm for hierarchical clustering.*

Thus, the exponential mechanism improves on the cost, and shows that private hierarchical clustering can be done on graphs with average degree $O(\frac{n}{\epsilon})$.

6. Private Hierarchical Clustering in the Stochastic Block Model

In this section, we propose a hierarchical clustering algorithm designed for input graph generated from the hierarchical stochastic block model (HSBM), a graph model with planted communities arranged in a hierarchical structure. We define this model in Section 6.1. Next, in Section 6.2, we outline DPHCBlocks, a lightweight private hierarchical clustering algorithm in the HSBM, which uses community detection as a black box. This approach enables any DP community detection algorithm to be used as a sub-routine. Finally, in Section 6.3, we propose a practical, private community detection algorithm which is the first to work in the general HSBM. Combining the results in Sections 6.2 and 6.3, we obtain a private, $1 + o(1)$-approximation algorithm to the Dasgupta cost function.

6.1. Hierarchical Stochastic Block Model of Graphs

In this section, we consider unweighted graphs (V, E) where each edge has weight 1. Observe that differential privacy (Definition 1) corresponds to adding or removing an edge from G. In the HSBM (Cohen-Addad et al., 2017), there is a partition of V into blocks (communities) $B_1, B_2, \dots, B_k$ of V with the properties that two items in the same block have the same set of edge probabilities, and that items in different blocks are less likely to be connected with these probabilities following a hierarchical structure.

The probabilities of the edges in B are specified by a tree P with leaves $B = B_1, \dots, B_k$, internal nodes N, and a function $f : N \cup B \to [0, 1]$. To capture the decreasing probability of edges, f must satisfy $f(n_1) < f(n_2)$ whenever n_1 is an ancestor of n_2 in P. Formally, we have (Cohen-Addad et al., 2017):

Definition 6. *Let $B = B_1, \dots, B_k$; P be a tree with leaves in B and internal nodes N; and $f : N \cup B \to [0, 1]$ be a* *function satisfying that $f(n_1) < f(n_2)$ whenever n_1 is an ancestor of n_2 in P. We refer to the triplet (B, P, f) as a ground-truth tree. Then,* $\mathrm{HSBM}(B, P, f)$ *is a distribution over graphs G whose edges are drawn independently, such that for $u, v \in P$, we have*

$$\Pr[(u, v) \in G] = f(LCA_P(B_u, B_v)),$$

where LCA_P denotes the least common ancestor of the blocks B_u, B_v containing u, v in P.

Due to the randomness of the graph G, it would be unreasonable to expect to be able to recover the exact (B, P, f) from G. Our algorithms will recover an approximate ground-truth tree, according to the following definition:

Definition 7. *(From Cohen-Addad et al. (2017)): Let (B, P, f) be a ground-truth tree, and let (B, T, f') be another ground-truth tree with the same set of blocks. We say (B, T, f') is a γ approximate ground-truth tree if for all $u, v \in B$, $\gamma^{-1} f(LCA_P(u, v)) \leq f'(LCA_{P'}(u, v)) \leq \gamma f(LCA_P(u, v))$.*

For $\gamma \approx 1$, an approximate ground-truth tree means that $\mathrm{HSBM}(B, P, f)$ and $\mathrm{HSBM}(B, P', f')$ are essentially the same distribution.

6.2. Producing a DP Hierarchical Clustering Given Communities

Given the blocks (communities) of an HSBM, we now propose DPHCBlocks, a lightweight, private algorithm for returning a $1 + o(1)$-approximation to the Dasgupta cost. Our algorithm uses some ideas from the non-private algorithm proposed in Cohen-Addad et al. (2017; 2019).

DPHCBlocks takes in G generated from $\mathrm{HSBM}(B, P, f)$, as well as the blocks B. To produce an approximate ground-truth tree, it considers similarities $sim(B_i, B_j) = \frac{w_G(B_i, B_j)}{|B_i||B_j|}$ for every pair of blocks. It then performs a process similar to single linkage: until all blocks are merged, it greedily merges the groups with the highest similarity, and considers the similarity between this new group and any other groups to be the maximum similarity of any pair of blocks between the groups. Privacy comes from addition of Laplace noise in the similarity calculation, which is the only place in which the private graph G is used. DPHCBlocks appears as Algorithm 1.

DPHCBlocks accesses the graph via the initial similarities $sim(B_i, B_j)$. By observing the sensitivity $\max_{B_i, B_j} |w_{G'}(B_i, B_j) - w_G(B_i, B_j)|$ is at most 1, we are able to prove its privacy. We also use the fact that adding an edge can only affect $sim(B_i, B_j)$ for just one choice of B_i, B_j.

Theorem 3. *DPHCBlocks satisfies ϵ-edge DP in the parameter G.*

Proof. Observe the algorithm can be viewed as a post-processing of the set $\mathcal{B} = \{sim(B_i, B_j) + \mathcal{L}_{ij} : i, j \in k\}$ where $\mathcal{L}_{ij} \sim Lap(\frac{1}{\epsilon})$ i.i.d. Suppose an edge is added between B_i, B_j. Then, $sim(B_i, B_j) + \mathcal{L}_{ij}$ is protected by ϵ-edge DP by the Laplace mechanism, observing the sensitivity of $w_G(B_i, B_j)$ is 1. The other quantities in $\mathcal{B}$ follow the same distribution, so $\mathcal{B}$ itself satisfies ϵ-edge DP. □

We stress that, crucially, Algorithm 1 and all our algorithms are DP for any input graph G, even if the graphs do not come from the HSBM model. We will use the input distribution assumptions only in the utility proofs.

We are also able to show a utility guarantee that DPHCBlocks is a $(1 + o(1), 0)$-approximation to the cost objective. In order to prove this, we need to assume that the blocks in the HSBM are sufficiently large (at least $n^{2/3}$) and that the edge probabilities are at least $\frac{\log n}{\sqrt{n}}$. These assumptions are necessary to ensure concentration of the graph cuts between blocks, so that an accurate approximate tree may be formed. Also, it requires that $\epsilon \geq \frac{1}{\sqrt{n}}$—this is an extremely light assumption, and it still permits us to use a small, constant value of ϵ to guarantee strong privacy. Formally,

Theorem 4. *For $\epsilon \geq \frac{1}{\sqrt{n}}$ and a graph G drawn from* $\mathrm{HSBM}(B, P, f)$ *such that $|B_i| \geq n^{2/3}$ and $f \geq \frac{\log n}{\sqrt{n}}$, with probability $1 - \frac{2}{n}$, the tree T outputted by DPHCBlocks satisfies $\omega_G(T) \leq (1 + o(1))\omega_G(T')$.*

In fact, we show a stronger result that the tuple (B, T, f') returned by DPHCBlocks is a $1 + o(1)$-approximate ground-truth tree for $\mathrm{HSBM}(B, P, f)$. By a result from Cohen-Addad et al. (2019), this implies it achieves the approximation guarantee. We defer the proof to Appendix D.1.

6.3. DP Community Detection in the HSBM

We now develop a DP method of identifying the blocks B of graph drawn from the HSBM. Combined with our clustering algorithm DPHCBlocks, this forms an end-to-end algorithm for hierarchical clustering in the HSBM in which the communities are not known.

In order to describe our algorithm, DPCommunity, we introduce some notation. For a model $\mathrm{HSBM}(B, P, f)$, we associate an $n \times n$ expectation matrix A given by the probabilities that edge (i, j) appears in G. We then let $\hat{A}$ be a randomized rounding of A to $\{0, 1\}$ which is simply the adjacency matrix of G. DPCommunity recovers communities when they are separated in the sense defined by

$$\Delta = \min_{u \in B_i, v \in B_j : i \neq j} \|A_u - A_v\|_2,$$

where A_u is the uth column of A. Next, we let $\sigma_1(A), \ldots, \sigma_n(A)$ denote the singular values of A in or-

Algorithm 1 DPHCBlocks, a hierarchical clustering algorithm in the HSBM given the blocks.

```
Input: $G = (V, E)$ drawn from the HSBM; blocks $B_1, \dots B_k$ partitioning $V$, privacy parameter $\epsilon$
Output: Tree $T$.
for $i = 1$ to $k$ do
    $T_i$ is a random HC with leaves $B_i$
end for
$sim(B_i, B_j) \leftarrow \frac{w_G(B_i, B_j) + \mathcal{L}_{ij}}{|B_i||B_j|}$, where $\mathcal{L}_{ij} \sim Lap(\frac{1}{\epsilon})$.
$\mathcal{C} = \{B_1, \dots, B_k\}$
$T = forest(T_1, \dots, T_k)$
while $|\mathcal{C}| \geq 1$ do
    $A_1, A_2 = \arg\max_{A_1, A_2 \in \mathcal{C}} sim(A_1, A_2)$
    Merge $A_1, A_2$ in $T$; $C = A_1 \cup A_2$
    $f'(C) = sim(A_1, A_2)$
    $\mathcal{C} = (\mathcal{C} \setminus \{A_1, A_2\}) \cup \{C\}$
    for $S \in \mathcal{C} \setminus \{C\}$: do
        $sim(S, C) \leftarrow \max_{B_i \in S, B_j \in C} sim(B_i, B_j)$
    end for
end while
Return: $(B, T, f')$.
```

der of decreasing magnitude. Finally, we let $\Pi_A^{(k)}$ denote the projection onto the top k left singular values of A—formally, if U_k consists of the top k singular values of A, then $\Pi_A^{(k)} = U_k U_k^T$.

DPCommunity is given the adjacency matrix $\hat{A}$ of a graph drawn from HSBM(B, P, f), as well as k, the number of blocks. In practice, k may be treated as a hyperparameter to be optimized. DPCommunity uses the spectral method (McSherry, 2001; Vu, 2014) to cluster the columns of $\hat{A}$. These results show that the columns in $F = \Pi_{\hat{A}}^{(k)}(\hat{A})$ forms a clustering of the points into their original blocks. To make this private, we use stability results of the SVD to compute (an upper bound of) the sensitivity Γ of F, and add noise N via the Gaussian mechanism. Since N, F are both $n \times n$ matrices, the l_2 error introduced by N grows with $\sqrt{n}$, which is large. Our final observation is that, since the distances in F are all that matter, we may project F to $\log(n)$-dimensional space using Johnson-Lindenstrauss (Johnson, 1984), and then add Gaussian noise whose error grows with $\sqrt{\log n}$. DPCommunity is shown in Algorithm 2.

There are two important remarks about DPCommunity. First, to ensure an accurate, private upper bound on Γ, we need the mild assumption that the spectral gap $\sigma_k(\hat{A}) - \sigma_{k+1}(\hat{A})$ is not too small, and if it is, the algorithm returns $\perp$. For most choices of parameters in the SBM, the spectral gap is always much larger than needed—the check is only to ensure privacy even for input graphs not from the SBM. Second, due to ease of theoretical analysis, $\hat{A}$ is split into two parts, and one part is projected onto the top k singular vectors of the other. This removes probabilistic dependence between variables, but the high level ideas are the same.

Algorithm 2 DPCommunity, a community recovery Algorithm

```
Input: $\hat{A}$, adjacency matrix generated from HSBM$(B, P, f)$, privacy parameter $\epsilon$.
Output: $f_z$, an estimate of blocks on a set $Z_2 \subseteq V$.
Compute a random partition $Y \sqcup Z_1 \sqcup Z_2$ of $V$ such that $|Y| = \frac{n}{2}, |Z_1| = |Z_2| = \frac{n}{4}$.
$\tilde{A}_1 \leftarrow \tilde{A}_{YZ_1}$ (submatrix of $\hat{A}$ with rows $Y$, cols. $Z_1$).
$\tilde{A}_2 \leftarrow \tilde{A}_{YZ_2}$
$\tilde{d}_k \leftarrow \sigma_k(\hat{A}_1) - \sigma_{k+1}(\hat{A}_1) - \frac{8}{\epsilon} \ln \frac{4}{\delta} + Lap(\frac{8}{\epsilon})$
$\tilde{\sigma}_1 \leftarrow \sigma_1(\hat{A}_2) + \frac{4}{\epsilon} \ln \frac{4}{\delta} + Lap(\frac{4}{\epsilon})$
if $\hat{d}_k \leq 10(\frac{8}{\epsilon} \ln \frac{4}{\delta})$ then
    return $\perp$
end if
$\tilde{\Gamma} \leftarrow \frac{\tilde{\sigma}_1}{\hat{d}_k}, m \leftarrow 64 \ln \frac{2n}{\delta}$.
$F \leftarrow P\Pi_{\hat{A}_1}^{(k)}(\hat{A}_2)$, where $P \sim \mathcal{N}(0, \frac{1}{\sqrt{m}})^{m \times n/2}$.
$\tilde{F} \leftarrow F + N$, where $N \sim \frac{3k\tilde{\Gamma}}{\epsilon}\sqrt{2 \ln \frac{5}{\delta}}\mathcal{N}(0, 1)^{m \times n/4}$.
return $\hat{F}$
```

We now analyze privacy and utility. Full proofs of the results in this section appear in Appendix 6. Our privacy analysis involves analyzing the release of the singular values $\sigma_1, \sigma_k, \sigma_{k+1}$, and $\tilde{F}$. The bulk of this analysis comes from analyzing the sensitivity of $\tilde{F}$, which uses the accuracy of the Johnson-Lindenstrauss transform and spectral perturbation bounds.

Theorem 5. *(Privacy): For $\epsilon < 1$, Algorithm 2 satisfies (ϵ, δ)-DP with respect to a change of one edge in $\hat{A}$.*

To prove the utility of DPCommunity, we prove that recovery is possible provided that Δ is larger than some threshold depending on ϵ, the singular values of A, the minimum edge probability, and the minimum block size, along with other mild assumptions on k and the block sizes. These assumptions are necessary, as there will be too little data for concentration otherwise. Formally,

Theorem 6. *(Utility): Let $\hat{A}$ be drawn from HSBM(B, P, f), $\tau = \max f(x)$, and $s = \min_{i=1}^{k} |B_i|$. There is a universal constant C such that if $\tau \geq C\frac{\log n}{n}$, $s \geq C\sqrt{n \log n}$, $k < n^{1/4}$, $\delta < \frac{1}{n}$, $\sigma_k(A) \geq C \max\{\sqrt{n\tau}, \frac{1}{\epsilon} \ln \frac{4}{\delta}\}$, and*

$$\Delta > C \max\left\{\frac{k(\ln \frac{1}{\delta})^{3/2}}{\epsilon} \frac{\sigma_1(A)}{\sigma_k(A)}, \sqrt{\frac{n\tau}{s}} + \sqrt{k\tau \log n} + \frac{\sqrt{nk\tau}}{\sigma_k}\right\},$$

then with probability at least $1 - 3n^{-1}$, DPCommunity returns a set of points $\tilde{F} = \{f_i : i \in Z_2\}$ such that

$$\|f_i - f_j\|_2 \leq \tfrac{2\Delta}{5} \quad \textit{if } \exists u.\ i, j \in B_u$$
$$\|f_i - f_j\|_2 \geq \tfrac{4\Delta}{5} \quad \textit{otherwise}.$$

Thus, if the assumptions are met, then $\tilde{F}$ consists of k well-separated clusters which indicate the communities of each point in the sampled set $Z_2 \subset V$. These communities can be found using a simple routine such as k-centers. In order to cluster all of V, we can simply divide the privacy budget into $\log n$ parts, run DPCommunity $\log n$ times, and merge the clusters.

To illustrate our theorem in a simple example, consider the HSBM with k equal-sized blocks, and let $f_P(n) = p$ when n is a parent of a leaf in P, and $f_P(n) = q$ otherwise, with $p \geq q$. This corresponds to probability p of an edge within a block and probability q of an edge between any two blocks. In this case, we obtain the following.

Corollary 1. *In the above HSBM, DPCommunity recovers the exact communities when $\delta \leq \frac{1}{n}$, $k < n^{1/4}$, and $\sqrt{p} - \sqrt{q} \geq \Omega(\frac{k \ln \frac{1}{\delta}}{\sqrt{\epsilon} n^{1/4}})$.*

Compared to previous work in the SBM with privacy, our algorithm requires a larger assumption on $\sqrt{p} - \sqrt{q}$ (Seif et al. (2022); Chen et al. (2023) require $\sqrt{p} - \sqrt{q} \geq \sqrt{\frac{k}{\epsilon n}}$). However, previous work either uses semi-definite programming or does not run in polynomial time, whereas DPCommunity is a practical use of the significantly more efficient Singular Vector Decomposition. Furthermore, our algorithm works in the fully-general HSBM, whereas previous work has no analogue of Theorem 6.

Algorithm 3 DPClusterHSBM a hierarchical clustering algorithm in the HSBM given the blocks.

```
Input: $\hat{A}$, adjacency matrix generated from HSBM$(B, P, f)$, number of blocks $k$, privacy parameter $\epsilon$.
Output: An hierarchical clustering $T$ of $\hat{A}$.
for $i \in \{1, \ldots, \log n\}$ do
    $\hat{F} \leftarrow$ DPCommunity$(\hat{A}, \frac{\epsilon}{2 \log n})$
    $B_1^i, \ldots, B_k^i \leftarrow$ k-centers$(\hat{F}, k)$
end for
$B_1, \ldots, B_k \leftarrow$ Union-Find$(B_1^1, \ldots, B_k^1, \ldots, B_k^{\log n})$
$T \leftarrow$ DPHCBlocks$(\hat{A}, \{B_1, \ldots, B_k\}, \frac{\epsilon}{2})$
return $T$
```

Combining Theorems 4 and 6, we are able to obtain DPClusterHSBM, an end-to-end hierarchical clustering algorithm in the HSBM (Algorithm 3). This algorithm runs DPCommunity $\log n$ times, using k-centers each run to find the well-separated communities in the subset $Z_2 \subseteq V$ returned by DPCommunity. Running $\log n$ times ensures that with high probability, each point in V will participate in at least one Z_2; these clusters may then be merged using a union-find data structure.

Corollary 2. *Let $\hat{A}$ be drawn from HSBM(B, P, f), and let $\tau = \max f(x)$ and $s = \min_{i=1}^{k} |B_i|$. Then, if $\epsilon > \frac{1}{\sqrt{n}}$, $\delta < \frac{1}{n}$, $s \geq n^{3/4}$, $f \geq \frac{\log n}{\sqrt{n}}$, and the parameters s, τ, A, Δ satisfy the conditions of Theorem 6, then DPClusterHSBM satisfies (ϵ, δ)-edge DP and is a $1 + o(1)$ approximation to the Dasgupta cost.*

Corollary 2 gives a $1 + o(1)$ multiplicative approximation the the Dasgupta cost for the given parameter regimes of the HSBM. This is a nearly-optimal cost that avoids the additive error of the algorithms in Section 5.

7. Experiments

The purpose of this section is evaluate Algorithm 1 designed for the HSBM model. First, we outline our methods and then we discuss our results.

Experimental Setup We tested our clustering algorithms on a real-world graph and generated synthetic graphs from the HSBM model. We compared the performance of DPClusterHSBM to several baseline algorithms. We ran algorithms at $\epsilon \in \{0.5, 1.0, 2.0\}$, as well as with no privacy.

To enable the replication of our work, we make the code available **open-source** [2].

Datasets Our real-world graph was generated from the MNIST digits dataset (LeCun, 1998) (with 1797 digits) by, for each digit, adding an undirected edge corresponding to one of its 120 nearest neighbors in pixel space. We generated graphs from HSBM(B, P, f) with $n = 2048$ nodes, $k = \{4, 8\}$ blocks, with block sizes chosen proportional to $\{1, \gamma, \ldots, \gamma^{k-1}\}$, where $\gamma^{k-1} = 3$. This has the effect of creating differently-sized blocks. We selected P to be a balanced tree over the blocks, and f that increases uniformly in the interval $[0.1, 0.9]$ as the tree is descended.

Algorithms We ran DPClusterHSBM and several baseline algorithms. In the implementation of DPClusterHSBM, we used a modified version of DPCommunity for practical considerations. This does not affect the privacy guarantees but it simplifies the algorithm. In particular, we privately release $\tilde{A}_1$ using the Laplace mechanism, and compute $\Pi_{\tilde{A}_1}(\hat{A}_2)$ without projection. We are then able to add Gaussian noise tailored to the sensitivity of $\Pi_{\tilde{A}_1}$, rather than to Γ which proved to be a rough upper bound in practice.

For our baselines, we considered a naive private approach in which we release A using the Laplace mechanism and truncate these values to be non-negative to form a sanitized, weighted graph. Then, we ran single, complete, and average linkage, and recorded the best of these methods. We refer collectively to these baselines as Linkage. Second, we formed a tree by recursively partitioning the graph into

[2] https://bitbucket.org/jjimola/dphc/src/master/

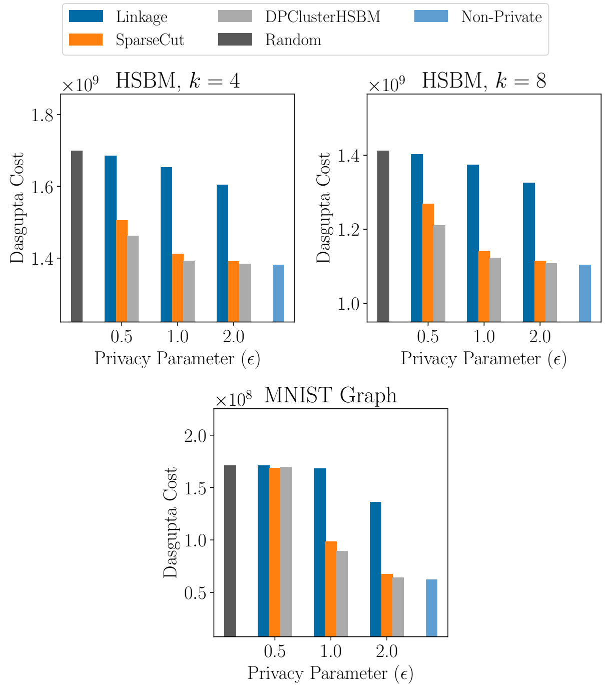

Figure 1. Cost for HSBM graphs with 2048 nodes and k clusters and MNIST graph with 1797 nodes.

its (approximately) sparsest cut. As shown in Charikar & Chatziafratis (2017), this is an $O(\sqrt{\log n}, 0)$-approximation in the *sanitized* graph. We refer to this baseline as SparseCut.

Metrics For each graph and clustering algorithm, and the value of ϵ, we computed $\omega_G(T)$, averaged over 5 runs.

7.1. Results

Our results appear in Figure 1. In addition to the cost for each algorithm, we included the cost of a random tree. The data had low variance: for each of the 5 runs used to compute each bar, the values were within 0.5% of each other.

For all trials, the cost of Linkage was much higher than the other two algorithms; even with $\epsilon = 2$, Linkage did not offer improvement of more than 10% reduction in cost over the random tree. Thus, the rest of our discussion focuses on DPClusterHSBM and SparseCut.

For the synthetic graphs, the cost of DPClusterHSBM is lower than SparseCut, particularly when $\epsilon = 0.5$. In this case, when $k = 4$ (resp. 8), DPClusterHSBM offered a 14.4% (resp. 14.2%) reduction in cost over the random tree, whereas SparseCut offered an 11.5% (resp. 10.3%) reduction. Thus, DPClusterHSBM offers up to 38% more reduction in cost than SparseCut, over the cost of a random tree. Even when $\epsilon = 0.5$, the cost of DPClusterHSBM is just 5.8% (resp. 9.6%) higher than the cost of the best tree with no privacy.

For $\epsilon = 1, 2$ on HSBM graphs, the costs of SparseCut and DPClusterHSBM fall to within 1% of each other, though DPClusterHSBM consistently outperforms the former for all values of ϵ. Moreover, notice that for $\epsilon = 2$, the costs of both algorithms are within 1% of the non-private tree, indicating that for higher ϵ the cost of privacy becomes negligible.

For the graph generated from MNIST, all algorithms perform as poorly as a random tree for $\epsilon = 0.5$. This indicates that the noise introduced by the high privacy constraint destroys the clusters, which are less-well structured than those of the HSBM graphs. At $\epsilon = 1$, the error of SparseCut is 10% higher than DPClusterHSBM. For $\epsilon = 2$, the cost of SparseCut is 5% higher than that of DPClusterHSBM, and DPClusterHSBM attains error within 3% of the best tree with no privacy. This is consistent with our previous observation that DPClusterHSBM offers improvement over the baselines, particularly when ϵ is not too high.

8. Conclusion

We have considered hierarchical clustering under differential privacy in Dasgupta's cost framework. While strong lower bounds exist for the problem, we have proposed algorithms with nearly matching approximation guarantees. Furthermore, we showed the lower bounds can be overcome in the HSBM, and nearly optimal trees can be found in this setting using efficient methods. For future work, one could consider private hierarchical clustering in a less structured model than the HSBM in hopes of overcoming the lower bound here as well.

References

Agarwal, A., Khanna, S., Li, H., and Patil, P. Sublinear algorithms for hierarchical clustering. *arXiv preprint arXiv:2206.07633*, 2022.

Arora, R. and Upadhyay, J. On differentially private graph sparsification and applications. *Advances in neural information processing systems*, 32, 2019.

Balcan, M.-F., Dick, T., Liang, Y., Mou, W., and Zhang, H. Differentially private clustering in high-dimensional euclidean spaces. In *International Conference on Machine Learning*, pp. 322–331. PMLR, 2017.

Bateni, M., Behnezhad, S., Derakhshan, M., Hajiaghayi, M., Kiveris, R., Lattanzi, S., and Mirrokni, V. Affinity clustering: Hierarchical clustering at scale. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in*

Neural Information Processing Systems 30, pp. 6864–6874. Curran Associates, Inc., 2017.

Bhatia, R. *Matrix Analysis*, volume 169. Springer Verlag, 1997.

Blocki, J., Blum, A., Datta, A., and Sheffet, O. The johnson-lindenstrauss transform itself preserves differential privacy. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 410–419. IEEE, 2012.

Bun, M., Elias, M., and Kulkarni, J. Differentially private correlation clustering. In *International Conference on Machine Learning*, pp. 1136–1146. PMLR, 2021.

Charikar, M. and Chatziafratis, V. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 841–854. SIAM, 2017.

Chatziafratis, V., Yaroslavtsev, G., Lee, E., Makarychev, K., Ahmadian, S., Epasto, A., and Mahdian, M. Bisect and conquer: Hierarchical clustering via max-uncut bisection. In *International Conference on Artificial Intelligence and Statistics*, pp. 3121–3132. PMLR, 2020.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Chen, H., Cohen-Addad, V., d'Orsi, T., Epasto, A., Imola, J., Steurer, D., and Tiegel, S. Private estimation algorithms for stochastic block models and mixture models. *arXiv preprint arXiv:2301.04822*, 2023.

Cohen-Addad, V., Kanade, V., and Mallmann-Trenn, F. Hierarchical clustering beyond the worst-case. *Advances in Neural Information Processing Systems*, 30, 2017.

Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., and Mathieu, C. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):1–42, 2019.

Cohen-Addad, V., Epasto, A., Lattanzi, S., Mirrokni, V., Munoz, A., Saulpic, D., Schwiegelshohn, C., and Vassilvitskii, S. Scalable differentially private clustering via hierarchically separated trees. *arXiv preprint arXiv:2206.08646*, 2022a.

Cohen-Addad, V., Epasto, A., Mirrokni, V., Narayanan, S., and Zhong, P. Near-optimal private and scalable k-clustering. In *Advances in Neural Information Processing Systems*, 2022b.

Cohen-Addad, V., Fan, C., Lattanzi, S., Mitrović, S., Norouzi-Fard, A., Parotsidis, N., and Tarnawski, J. Near-optimal correlation clustering with privacy. *arXiv preprint arXiv:2203.01440*, 2022c.

Dasgupta, S. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 118–127, 2016.

Dhulipala, L., Eisenstat, D., Łacki, J., Mirronki, V., and Shi, J. Hierarchical agglomerative graph clustering in poly-logarithmic depth. In *Neurips 2022*, 2022.

Diez, I., Bonifazi, P., Escudero, I., Mateos, B., Muñoz, M. A., Stramaglia, S., and Cortes, J. M. A novel brain partition highlights the modular skeleton shared by structure and function. *Scientific reports*, 5:10532, 2015.

Ding, J., d'Orsi, T., Nasser, R., and Steurer, D. Robust recovery for stochastic block models. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 387–394. IEEE, 2022.

Dwork, C. Differential privacy and the us census. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI symposium on principles of database systems*, pp. 1–1, 2019.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014a.

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 11–20, 2014b.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

Eliáš, M., Kapralov, M., Kulkarni, J., and Lee, Y. T. Differentially private release of synthetic graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 560–578. SIAM, 2020.

Epasto, A., Mirrokni, V., Perozzi, B., Tsitsulin, A., and Zhong, P. Differentially private graph learning via sensitivity-bounded personalized pagerank. In *Neurips*, 2022.

Erlingsson, Ú., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

Fei, Y. and Chen, Y. Achieving the Bayes error rate in synchronization and block models by SDP, robustly. *IEEE Trans. Inform. Theory*, 66(6):3929–3953, 2020. ISSN 0018-9448. doi: 10.1109/TIT.2020.2966438. URL https://doi.org/10.1109/TIT.2020.2966438.

Ghazi, B., Kumar, R., and Manurangsi, P. Differentially private clustering: Tight approximation ratios. *Advances in Neural Information Processing Systems*, 33:4040–4054, 2020.

Guédon, O. and Vershynin, R. Community detection in sparse networks via Grothendieck's inequality. *Probab. Theory Related Fields*, 165(3-4):1025–1049, 2016. ISSN 0178-8051. doi: 10.1007/s00440-015-0659-z. URL http://dx.doi.org/10.1007/s00440-015-0659-z.

Hardt, M. and Talwar, K. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 705–714, 2010.

Hegde, A., Möllering, H., Schneider, T., and Yalame, H. Sok: Efficient privacy-preserving clustering. *Proceedings on Privacy Enhancing Technologies*, 2021(4):225–248, 2021.

Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. doi: 10.1016/j.patrec.2009.09.011. URL https://doi.org/10.1016/j.patrec.2009.09.011.

Jardine, N. and Sibson, R. A model for taxonomy. *Mathematical Biosciences*, 2(3-4):465–482, 1968.

Johnson, W. B. Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.*, 26:189–206, 1984.

Kasiviswanathan, S. P., Nissim, K., Raskhodnikova, S., and Smith, A. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*, pp. 457–476. Springer, 2013.

Kolluri, A., Baluta, T., and Saxena, P. Private hierarchical clustering in federated networks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2342–2360, 2021.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Leskovec, J., Rajaraman, A., and Ullman, J. D. *Mining of massive datasets*. Cambridge university press, 2014.

Liu, A. and Moitra, A. Minimax rates for robust community detection. *CoRR*, abs/2207.11903, 2022. doi: 10.48550/arXiv.2207.11903. URL https://doi.org/10.48550/arXiv.2207.11903.

Machanavajjhala, A., He, X., and Hay, M. Differential privacy in the wild: A tutorial on current practices & open challenges. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1727–1730, 2017.

Mann, C. F., Matula, D. W., and Olinick, E. V. The use of sparsest cuts to reveal the hierarchical community structure of social networks. *Social Networks*, 30(3):223–234, 2008.

McSherry, F. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 529–537. IEEE, 2001.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, 2007.

Moitra, A., Perry, W., and Wein, A. S. How robust are reconstruction thresholds for community detection? In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 828–841, 2016.

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

Moseley, B. and Wang, J. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. *Advances in neural information processing systems*, 30, 2017.

Murtagh, F. and Contreras, P. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

Pinot, R. Minimum spanning tree release under differential privacy constraints. *arXiv preprint arXiv:1801.06423*, 2018.

Roy Chowdhury, A., Wang, C., He, X., Machanavajjhala, A., and Jha, S. Crypte: Crypto-assisted differential privacy on untrusted servers. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 603–619, 2020.

Seif, M., Nguyen, D., Vullikanti, A., and Tandon, R. Differentially private community detection for stochastic block models. *arXiv preprint arXiv:2202.00636*, 2022.

Sneath, P. H. and Sokal, R. R. Numerical taxonomy. *Nature*, 193(4818):855–860, 1962.

Steinbach, M., Karypis, G., Kumar, V., et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pp. 525–526. Boston, 2000.

Tumminello, M., Lillo, F., and Mantegna, R. N. Correlation, hierarchies, and networks in financial markets. *Journal of Economic Behavior & Organization*, 75(1):40–58, 2010.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Vu, V. A simple svd algorithm for finding hidden partitions. *arXiv preprint arXiv:1404.3918*, 2014.

Vu, V. H. Spectral norm of random matrices. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pp. 423–430, 2005.

Ward Jr, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

Xiao, Q., Chen, R., and Tan, K.-L. Differentially private network data release via structural inference. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 911–920, 2014.

A. Related Work

Differential Privacy Differential privacy (Dwork et al., 2006) has recently become the gold standard of privacy used by institutions such as the US census (Dwork, 2019) and large tech companies (Erlingsson et al., 2014). In a nutshell, DP algorithms provide plausible deniability for the input data of any user. There is a vast literature on DP algorithms for a disparate range of problems and many different models for differential privacy (Dwork et al., 2006; McSherry & Talwar, 2007; Chaudhuri et al., 2011; Roy Chowdhury et al., 2020; Machanavajjhala et al., 2017; Dwork, 2019) (we refer to Dwork et al. (2014a) for a survey).

Among this rapidly growing literature, our work builds on multiple work on differentially privacy, namely DP PCA algorithms (Dwork et al., 2014b), DP Johnson Lindenstrauss projections (Blocki et al., 2012), DP cut sparsification in graphs (Eliáš et al., 2020) as well as DP stochastic block model reconstruction (reviewed later).

Private graph algorithms Especially relevant to this work is the area of differential privacy in graphs. DP has been declined in graph problems both as the edge-level (Epasto et al., 2022; Eliáš et al., 2020) and node-level model (Kasiviswanathan et al., 2013). The most related work in this area is that on graph cut approximation (Eliáš et al., 2020; Arora & Upadhyay, 2019), as well as that of graph clustering with DP in correlation clustering model (Bun et al., 2021; Cohen-Addad et al., 2022c).

Hierarchical Clustering As we discussed in the introduction, hierarchical clustering has been studied for decades in multiple fields. For this reason, a significant number of algorithms for hierarchical clustering have been introduced (Murtagh & Contreras, 2012). Up until recently (Dasgupta, 2016), most work on hierarchical clustering has been heuristic in nature, defining algorithms based on procedures without specific theoretical guarantees in terms of approximation. Most well-known among such algorithms are the linkage-based ones (Jain, 2010; Bateni et al., 2017). Dasgupta (2016) introduced for the first time a combinatorial approximation objective for hierarchical clustering which is the one studied in this paper. Since this work, many authors have designed algorithms for variants of the problem (Cohen-Addad et al., 2017; 2019; Charikar & Chatziafratis, 2017; Moseley & Wang, 2017; Agarwal et al., 2022; Chatziafratis et al., 2020) exploring maximization/minimization versions of the problem on dissimilarity/similarity graphs.

Limited work has been devoted to DP hierarchical clustering algorithms. One paper (Xiao et al., 2014) initiates private clustering via MCMC methods, which are not guaranteed to be polynomial time. Follow-up work (Kolluri et al., 2021) shows that sampling from the Boltzmann distribution (essentially the exponential mechanism (McSherry & Talwar, 2007) in DP) produces an approximation to the maximization version of Dasgupta's function, which is a different problem formulation. Again, this algorithm is not provably polynomial time.

Private flat clustering Contrary to hierarchical clustering, the area of private *flat* clustering on metric spaces has received large attention. Most work in this area has focus on improving the privacy-approximation trade-off (Ghazi et al., 2020; Balcan et al., 2017) and on efficiency (Hegde et al., 2021; Cohen-Addad et al., 2022b;a).

Stochastic block models The Stochastic Block Model (SBM) is a classic model for random graphs with planted partitions which has received significant attention in the literature. Most work in this area has focus on providing exact or approximate recovery of communities for increasingly more difficult regimes of the model (Guédon & Vershynin, 2016; Montanari & Sen, 2016; Moitra et al., 2016; Fei & Chen, 2020; Ding et al., 2022; Liu & Moitra, 2022). Specifically for our work, we focus on a variant of the model which has nested ground-truth communities arranged in a hierarchical fashion. This model has received attention for hierarchical clustering (Cohen-Addad et al., 2017).

The study of private algorithms for SBMs is instead very recent and no work has addressed private recovery for hierarchical SBMS. One of the only results known for private (non-hierarchical) SBMs is the work of Seif et al. (2022) which provides a quasi-polynomial time algorithm for some regimes of the model. This paper require either non-poly time or $\epsilon \in \Omega(\log(|V|))$. Finally, very recently and currently to our work, the manuscript of Chen et al. (2023) has been published. This work provides strong approximation guarantees using semi-definite programming for recovering SBM communities. None of these papers can be used directly to approximate hierarchical clustering on HSBMs. For this reason in Section 6 we design a hierarchical clustering algorithm (Algorithm 1) which uses as subroutine a DP SBM community detection algorithm. Moreover, we show a novel algorithm for SBMs (Algorithm 2) (independent to that of Chen et al. (2023)) which is of practical interest as it does not require procedure with large polynomial dependency on the size of the input, such solving a complex semi-definite program.

B. Omitted proofs from Section 4

B.1. Proof of Lemma 2

We start with the following lemma:

Lemma 4. *Let G_1, G_2 be two graphs drawn uniformly at random from $\mathcal{P}(n, 5)$. Let $\alpha = \frac{1}{100}$. The probability that there exists a balanced cut (A, B) which misses at most $\frac{\alpha}{5}n$ of the cycles for both G_1, G_2 is at most $2^{-0.4n}$.*

Proof. Let (A, B) be any balanced cut with $|A| = \beta n$, for $\frac{1}{3} \leq \beta \leq \frac{2}{3}$. Let $\mathcal{E}_1(A, B)$ be the event that (A, B) misses at most $\frac{\alpha}{X}n$ cycles in G_1, and define $\mathcal{E}_2(A, B)$ similarly for G_2. We observe the desired probability can be upper bounded by

$$\sum_{(A,B) \text{ a balanced cut}} \Pr[\mathcal{E}_1(A, B)] \Pr[\mathcal{E}_2(A, B)]. \tag{3}$$

In the above sum, the balanced cuts (A, B) are fixed, and the graphs G_1, G_2 are generated independently. We consider an equivalent random process, where $G_1 \in \mathcal{P}(n, 5)$ is fixed, and then (A, B) is generated by picking a uniformly random string $S \in \{0, 1\}^n$ with βn 1s. There are $\binom{n}{\beta n}$ possible strings. We will now upper bound the number of strings for which $\mathcal{E}_1(A, B)$ holds. When $\mathcal{E}_1(A, B)$ holds, we can choose c cycles which are monochromatic 1s, where c is a non-negative integer such that $5c < n$, plus $\frac{\alpha n}{5}$ cycles which are not necessarily monochromatic. Within these $\frac{\alpha n}{5}$ cycles, there are αn vertices from which we can choose $d \leq \alpha n$ remaining 1s. The total number of 1s is $5c + d$, and thus $5c + d = \beta n$. Thus, the total number of admissible strings is at most

$$\sum_{5c+d=\beta n, d \leq \alpha n} \binom{n/5}{c} \binom{n/5}{\alpha n/5} \binom{\alpha n}{d}.$$

We make the simple observation that $\binom{\alpha n}{d} \leq 2^{\alpha n}$. Furthermore, we observe that there are $\frac{\alpha n}{5}$ admissible choices of c, d. In the following, we use the fact that $2^{H_2(\beta)n - \ln n} \leq \binom{n}{\beta n} \leq 2^{H_2(\beta)n}$, where $H_2(p)$ is the binary entropy function. We upper bound the number of admissible strings with

$$\begin{aligned} \frac{\alpha n}{5} \max_{(\beta-\alpha)n \leq 5c \leq \beta n} \binom{n/5}{c} \binom{n/5}{\alpha n/5} 2^{\alpha n} &\leq \frac{\alpha n}{5} \max_{(\beta-\alpha)n \leq 5c \leq \beta n} 2^{H_2(5c/n)n/5} 2^{H_2(\alpha)n/5} 2^{\alpha n} \\ &\leq n 2^{H_2(\beta)n/5} 2^{H_2(\alpha)n/5} 2^{\alpha n}. \end{aligned}$$

Dividing this number by $\binom{n}{\beta n}$, the total possible number of strings, we obtain

$$\begin{aligned} \Pr[\mathcal{E}_1(A, B)] &\leq \frac{n 2^{(H_2(\beta)+H_2(\alpha))n/5 + \alpha n}}{2^{H_2(\beta)n - \ln n}} \\ &\leq 2^{\left(\frac{H_2(\beta)+H_2(\alpha)}{5} + \alpha - H_2(\beta)\right)n + \ln n} \\ &\leq 2^{-0.7n}, \end{aligned}$$

where the last line follows from the fact that $\frac{1}{3} \leq \beta \leq \frac{2}{3}$ and that $\alpha = \frac{1}{100}$ so that $H_2(\alpha) \leq 0.081$. By a similar argument, we have $\Pr[A_2(B)] \leq 2^{-0.7n}$.

Thus, (3) can be upper bounded by

$$2^n \Pr[\mathcal{E}_1(A, B)] \Pr[\mathcal{E}_2(A, B)] \leq 2^n 2^{-2 \times 0.7n} \leq 2^{-0.4n}.$$

□

Having shown the result for two random graphs, we apply the union bound to show that for exponentially many random graphs, it is unlikely that any tree can cluster more than one graph in the family well. We now prove Lemma 2.

Proof. Let $\mathcal{F}$ consist of $2^{0.2n}$ graphs generated uniformly at random $\mathcal{P}(n,5)$. For each pair of graphs G_1, G_2, we have by Lemma 4 every balanced cut will miss at least $\frac{\alpha}{5}n$ cycles in either G_1 or G_2 with probability $1-2^{-0.4n}$. By the union bound applied $\frac{1}{2}2^{0.4n}$ times for each pair of graphs, we have with probability $\frac{1}{2}$ that every balanced cut will miss at least $\frac{\alpha}{5}n$ cycles in all but at most one graph in $\mathcal{F}$.

Every tree can be mapped to a balanced cut, so by Lemma 1, any tree will cost at least $\frac{4\alpha}{15}n^2 \geq \frac{n^2}{400}$ on all but at most one member of $\mathcal{F}$. This allows us to conclude that the sets $\mathcal{B}(G,r)$ are disjoint for all $G \in \mathcal{F}$. □

C. Omitted proofs from Section 5

C.1. Proof of Theorem 7

First, we state a theorem about private graph sparsification.

Theorem 7. *There is a polynomial-time, (ϵ,δ)-edge differentially private algorithm which, on input graph $G=(V,E,w)$, outputs a graph G' which with probability 0.9 is a $(z, O(nz))$-approximation to cut queries in G, where $z = O(\frac{\log^2\frac{1}{\delta}}{\epsilon}\frac{\log n}{\sqrt{n}})$.*

Proof. We apply an edge sparsification algorithm of Arora & Upadhyay (2019), which given a graph with Laplacian L, outputs a graph with Laplacian L' with $O(\frac{n}{\gamma^2})$ edges such that

$$(1-\gamma)((1-z)L + zL_n) \preceq L' \preceq (1+\gamma)((1-z)L + zL_n),$$

where L_n is the Laplacian of an unweighted K_n. The value of the cut $w(S,\overline{S})$ is given by by $\mathbf{1}_S^T L \mathbf{1}_S$; therefore, we have

$$(1-\gamma)((1-z)w(S,\bar(S)) - z|S|(n-|S|)) \leq w'(S,\overline{S}) \leq (1+\gamma)((1-z)w(S,\overline{S}) + z|S|(n-|S|))$$

Using the fact that $|S|(n-|S|) \leq n\min\{|S|, n-|S|\}$ and letting $\gamma \to 0$, we estabish that G' is a (z, nz) approximation to cut queries in G. □

Next, we reduce the cost to a sum of cuts. This idea appeared in Agarwal et al. (2022).

Lemma 5. *Suppose G' is an (α_n, β_n)-approximation to cut queries in G for some $\alpha < 1$. Let T' be any tree which satisfies $\omega_{G'}(T') \leq a_n \omega^*_{G'}$. Then,*

$$\omega_G(T') \leq (1+2\alpha_n)a_n\omega^*_G + (4a_n+2)\beta_n n^2.$$

For the revenue objective, let T' be any tree which satisfies $\omega^{\mathsf{MW}}_{G'}(T') \geq a_n\omega^{\mathsf{MW}*}_{G'}$. Then,

$$\omega^{\mathsf{MW}}_G(T') \geq (1-2\alpha_n)a_n\omega^{\mathsf{MW}*}_G - 2(a_n+1)\beta_n n^2 - 2(a_n+1)\alpha_n n^3.$$

A proof of this lemma appears in the next section.

Finally, we are ready to prove the theorem.

Proof. (Of Theorem 7): First, release a private graph G' using Theorem 7, which is a (z, nz)-cut approximation with probability at least 0.9, where $z = O(\frac{\log^2\frac{1}{\delta}}{\epsilon}\frac{\log n}{\sqrt{n}})$. We use the black box hierarchical clustering algorithm, which finds a tree such that $\mathbb{E}[\omega_G(T')] \leq a_n\omega^*_G$. Then, we apply Lemma 5, obtaining

$$\mathbb{E}[\omega_G(T')] \leq (1+2z)a_n\omega^*_G + (4a_n+2)zn^3.$$

For the revenue objective, our black box hierarchical clustering finds a tree T' such that $\mathbb{E}[\omega^{\mathsf{MW}}_G(G')] \geq a_n\omega^{\mathsf{MW}*}_G$. We apply Lemma 5, obtaining

$$\omega^{\mathsf{MW}}_G(T') \geq (1-2z)a_n\omega^{\mathsf{MW}*}_G - 4(a_n+1)zn^3.$$

□

C.2. Proof of Lemma 5

We start with the well-known representation of $\omega_G(T)$ (Dasgupta, 2016):

$$\omega_G(T) = \sum_{S \to (S_1, S_2) \text{ in } T} |S| w(S_1, S_2),$$

where the sum is indexed by internal splits of T, which splits a set S of leaves into two parts S_1, S_2. Using the identity $w(S_1, S_2) = \frac{1}{2} w(S_1, \overline{S_1}) + \frac{1}{2} w(S_2, \overline{S_2}) - \frac{1}{2} w(S, \overline{S})$, we substitute:

$$\omega_G(T) = \frac{1}{2} \sum_{S \to (S_1, S_2) \text{ in } T} |S| w(S_1, \overline{S_1}) + |S| w(S_2, \overline{S_2}) - |S| w(S, \overline{S})$$

In the above sum, if we assign cuts to their respective nodes, then we obtain the following: The root node is assigned $-|S| w(S, \overline{S}) = 0$. Each internal node S_1 which is not a leaf node or the root is assigned $|S| w(S_1, \overline{S_1}) - |S_1| w(S_1, \overline{S_1}) = |S_2| w(S_1, \overline{S_1})$, where $S \to (S_1, S_2)$ is the parent split of S_1. Finally, each leaf node S_1 is assigned $|S| w(S_1, \overline{S_1}) = |S_2| w(S_1, \overline{S_1}) + w(S_1, \overline{S_1})$, using the fact that $|S_1| = 1$. This brings us to the following decomposition (Agarwal et al., 2022):

$$\omega_G(T) = \underbrace{\sum_{S \to (S_1, S_2) \text{ in } T} |S_2| w(S_1, \overline{S_1}) + |S_1| w(S_2, \overline{S_2})}_{\omega_G^1(T)} + \underbrace{\sum_{i=1}^{n} w(v, \overline{v})}_{\omega_G^2}.$$

We refer to the leftmost term of the above as $\omega_G^1(T)$, and the rightmost term as ω_G^2. Observe the second quantity does not depend on T. Now, for any tree T, we have

$$\begin{aligned}
\omega_{G'}^1(T) \le \sum_{S \to (S_1, S_2) \text{ in } T} \Big(& |S_2| ((1 + \alpha_n) w_G(S_1, \overline{S_1}) + \beta_n \min\{|S_1|, n - |S_1|\}) \\
& + |S_1| ((1 + \alpha_n) w_G(S_2, \overline{S_2}) + \beta_n \min\{|S_2|, n - |S_2|\}) \Big) \\
&\le (1 + \alpha_n) \omega_G^1(T) + \beta_n \sum_{S \to (S_1, S_2) \text{ in } T} |S_2| \min\{|S_1|, n - |S_1|\} + |S_1| \min\{|S_2|, n - |S_2|\} \\
&\le (1 + \alpha_n) \omega_G^1(T) + \beta_n \sum_{S \to (S_1, S_2) \text{ in } T} 2 |S_1| |S_2| \\
&\le (1 + \alpha_n) \omega_G^1(T) + \beta_n n^2,
\end{aligned}$$

where the final line comes from an induction argument: if $f(n) \le \max_{1 \le i \le n} f(i) f(n-i) + 2\beta i(n-i)$, then we can show via induction that $f(n) \le \frac{n^2 \beta}{2}$. By a similar process, we can show the following inequalities

$$(1 - \alpha_n) \omega_G^1(T) - \beta_n n^2 \le \omega_{G'}^1(T) \le (1 + \alpha_n) w_G^1(T) + \beta_n n^2 \tag{4}$$

$$(1 - \alpha_n) \omega_G^2 - \beta_n n \le \omega_{G'}^2 \le (1 + \alpha_n) \omega_G^2 + \beta_n n \tag{5}$$

This implies that

$$(1 - \alpha_n) \omega_G(T) - 2\beta_n n^2 \le \omega_{G'}(T) \le (1 + \alpha_n) \omega_G(T) + 2\beta_n n^2.$$

This allows us to derive that

$$\begin{aligned}
\omega_G(T') &\le (1 + \alpha_n) \omega_{G'}(T') + 2\beta_n n^2 \\
&\le (1 + \alpha_n) a_n \omega_{G'}(T^*) + 2\beta_n n^2 \\
&\le (1 + \alpha_n) a_n ((1 + \alpha_n) \omega_G^* + 2\beta_n n^2) + 2\beta_n n^2 \\
&\le (1 + 2\alpha_n) a_n \omega_G^* + (4a_n + 2) \beta_n n^2
\end{aligned}$$

Plugging T^*, the optimal tree for G, into the above, we obtain that $\omega_{G'}^* \le (1 + \alpha_n) \omega_G^* + 2\beta_n n^2$, and therefore,

$$\omega_{G'}(T') \le a_n (1 + \alpha_n) \omega_G^* + 2 a_n \beta_n n^2.$$

We also have that $(1 - \alpha_n) \omega_G(T') - 2\beta_n n^2 \le \omega_{G'}(T')$, and we obtain our result by rearranging.

C.3. Proof of Lemma 3

Using a general lemma about the exponential mechanism (McSherry & Talwar, 2007), we are able to prove a bound on the algorithm error.

Lemma 6. *Let $f(X, Y)$ be a function with sensitivity 1 in X. Suppose we run the exponential mechanism $M : \mathcal{X} \to \mathcal{Y}$ with finite range $\mathcal{Y}$ using utility function $u_X(Y) = f(X, Y)$. Let $OPT(X) = \min_{Y\in\mathcal{Y}} u_X(Y)$. If our privacy budget is ϵ, then for each $X \in \mathcal{X}$, we have*

$$\Pr[u_X(M(X)) \leq OPT(X) + 2\frac{\log(|\mathcal{Y}|)}{\epsilon}] \geq 1 - \frac{1}{|\mathcal{Y}|}.$$

Proof. Let $\mathcal{Z} = \{Y \in \mathcal{Y} : u_X(Y) \leq OPT(X) + 2\frac{\log(|\mathcal{Y}|)}{\epsilon}\}$. We are guaranteed that the optimal element, Z^*, with $u_X(Z^*) = OPT(X)$, is in $\mathcal{Z}$. We want to lower bound the quantity $\Pr[M(X) \in \mathcal{Z}]$. Observe that

$$\begin{aligned}\Pr[M(X) \in \mathcal{Z}] &= \frac{\sum_{Z\in\mathcal{Z}} e^{-\epsilon u_X(Z)/2}}{\sum_{Z\in\mathcal{Z}} e^{-\epsilon u_X(Z)/2} + \sum_{Y\in\mathcal{Y}, Y\notin\mathcal{Z}} e^{-\epsilon u_X(Y)/2}} \\ &\geq \frac{e^{-\epsilon u_X(Z^*)/2}}{e^{-\epsilon u_X(Z^*)/2} + \sum_{Y\in\mathcal{Y}, Y\notin\mathcal{Z}} e^{-\epsilon u_X(Y)/2}} \\ &= \frac{e^{-\epsilon OPT(X)/2}}{e^{-\epsilon OPT(X)/2} + \sum_{Y\in\mathcal{Y}, Y\notin\mathcal{Z}} e^{-\epsilon u_X(Y)/2}}.\end{aligned}$$

The second line holds because the function $g(z) = \frac{z}{z+K}$ for $K > 0$ is decreasing as $z \to 0$. The bottom sum can be upper bounded with $|\mathcal{Y}|e^{-\epsilon(OPT(X)+2\log(|\mathcal{Y}|)/\epsilon)/2} \leq \frac{1}{|\mathcal{Y}|}e^{-\epsilon OPT(X)/2}$. Thus, we are left with

$$\Pr[M(X) \in \mathcal{Z}] \geq \frac{1}{1 + 1/|\mathcal{Y}|} \geq 1 - \frac{1}{|\mathcal{Y}|}.$$

□

For hierarchical clustering, our algorithm is a corollary of the previous result:

Proof. We apply the exponential mechanism with utility function $u_G(T) = -\frac{1}{n}\omega_G(T)$, which has sensitivity 1. The range of the algorithm is the space of trees with n nodes; there are at most n^n trees of this size. By Lemma 6, the utility satisfies $\Pr[\frac{\omega_G^*}{n} \leq \frac{\omega_G(M(G))}{n} + 2\frac{n\log n}{\epsilon}] \geq 1 - o(1)$, and hence the algorithm is a $(1, O(\frac{n^2\log n}{\epsilon}))$-approximation.

For the revenue objective, we apply the exponential mechanism with utility function $u_G(T) = \frac{1}{2n}\omega_G^{\mathsf{MW}}(T)$, which has sensitivity 1. By Lemma 3, the utility satisfies $\Pr[\frac{\omega_G^{\mathsf{MW}}(M(G))}{2n} \leq \frac{\omega_G^{\mathsf{MW}*}}{2n} + 2\frac{n\log n}{\epsilon}] \geq 1 - o(1)$. This establishes $(1, O(\frac{n^2\log n}{\epsilon}))$-approximation. □

D. Omitted proofs from Section 6

D.1. Proof of Theorem 4

In order to prove this theorem, we will show that DPHCBlocks finds a $(1 + o(1))$-approximate ground-truth tree, and then appeal to a result showing the such trees are approximately optimal with high probability (Cohen-Addad et al., 2019):

Lemma 7. *(Lemma 5.10 from Cohen-Addad et al. (2019)) Let G be a graph drawn from* HSBM(B, P, f)*, where $p_{min} = \min_{i\in B\cup N} f(i) \geq \omega(\sqrt{\frac{\log n}{n}})$. Let (B, P', f') be a γ-approximate ground-truth tree. Then, with probability $1 - 2^{-n}$, we have*

$$\text{cost}_G(P') \leq \gamma(1 + o(1))\text{cost}_G^*,$$

We now show that DPHCBlocks outputs an approximate ground-truth tree. We introduce a high-probability event and prove that if it happens, then the output is an approximate ground-truth tree.

Our event $\mathcal{E}$ states that $sim(B_i, B_j)$ as used in DPHCBlocks is a good estimate for $f(LCA_P(B_i, B_j))$. Intuitively, this makes sense, as if one had access to $f(LCA_P(B_i, B_j))$, then it would be easy to construct P (or an equivalent tree) using single linkage. Formally, we let $\mathcal{E}$ denote the event that there exists α such that for all B_i, B_j,

$$\left|sim(B_i, B_j) - f(LCA_P(B_i, B_j))\right| \leq \alpha f(LCA_P(B_i, B_j)). \tag{6}$$

The following lemma shows that $\mathcal{E}$ occurs with high probability.

Lemma 8. *If $|B_i| \geq n^{2/3}$ for all i, j, $\epsilon \geq \frac{1}{n^{1/2}}$, and $f(x) \geq \frac{\log n}{n^{1/2}}$, then the event $\mathcal{E}$ occurs with $\alpha = \frac{8}{n^{1/6}}$ with probability at least $1 - \frac{2}{n}$.*

Proof. The values $w_G(B_i, B_j)$ are distributed according to Binomial(N_{ij}, p_{ij}), where $N_{ij} = |B_i||B_j|$ and $p_{ij} = f(LCA_P(B_i, B_j))$. By Hoeffding's bound, we have that

$$\Pr[|w_G(B_i, B_j) - p_{ij}N_{ij}| \geq 2 \log n \sqrt{N_{ij}}] \leq \frac{1}{n^3}.$$

Furthermore, we have that $\Pr[|\mathcal{L}_{ij}| \geq \frac{6 \log n}{\epsilon}] \leq \frac{1}{n^3}$. Plugging in $sim(B_i, B_j) = \frac{w_G(B_i,B_j)+\mathcal{L}_{ij}}{N_{ij}}$, we obtain

$$\Pr\left[|sim(B_i, B_j) - p_{ij}| \geq \frac{2 \log n}{\sqrt{N_{ij}}} + \frac{6 \log n}{\epsilon N_{ij}}\right] \leq \frac{2}{n^3}.$$

Because $N_{ij} \geq n^{4/3}$ and $\epsilon \geq \frac{1}{n^{1/2}}$, we have $\frac{2 \log n}{\sqrt{N_{ij}}} + \frac{6 \log n}{\epsilon N_{ij}} \leq \frac{8 \log n}{n^{2/3}} \leq \frac{8}{n^{1/6}} p_{ij}$. Thus, we obtain $\Pr[|sim(B_i, B_j) - p_{ij}| \geq \alpha p_{ij}] \leq \frac{2}{n^3}$, with $\alpha = \frac{8}{n^{1/6}}$. Taking a union bound over all $\binom{k}{2} \leq n^2$ choices of i, j, we obtain our result. □

Finally, we show that when $\mathcal{E}$ occurs, then DPHCBlocks finds an approximate ground-truth tree. A similar result was proved in Cohen-Addad et al. (2019), though our lemma statement is sufficiently different that we include a proof here.

Lemma 9. *Assume that event $\mathcal{E}$ occurs. Then, the tuple (B, T, f') returned by Algorithm 1 is a $(1 + \alpha)$-approximate ground-truth tree for (B, P, f).*

Proof. We want to show that for all $B_i, B_j \in V$, we have

$$(1 - \alpha) f(LCA_P(B_i, B_j)) \leq f'(LCA_{P'}(B_i, B_j)) \leq (1 + \alpha) f(LCA_P(B_i, B_j)).$$

Let $I = LCA_T(B_i, B_j)$ be the internal node in which B_i, B_j are merged, and let C_i, C_j be the children of I such that $B_i \subseteq C_i$ and $B_j \subseteq C_j$. We have that

$$f'(LCA_{P'}(B_i, B_j)) = sim(C_i, C_j) = \max_{B \in C_i, B' \in C_j} sim(B, B').$$

Thus, it holds that $sim(B_i, B_j) \leq f'(LCA_{P'}(B_i, B_j))$. As event $\mathcal{E}$ holds, we have that $sim(B_i, B_j) \geq (1 - \alpha) f(LCA_P(B_i, B_j))$.

To finish, we show that $sim(C_i, C_j) \leq (1 + \alpha) f(LCA_P(B_i, B_j)$. Let $J = LCA_P(B_i, B_j)$ be the internal node in which B_i, B_j are merged in P, and let D_i, D_j be the children of J such that $B_i \subseteq D_i$ and $B_j \subseteq D_j$. We consider the following two cases.

Case 1: $C_i \subseteq D_i$ and $C_j \subseteq D_j$. Then, we have

$$sim(C_i, C_j) \leq \max_{B \in D_i, B' \in D_j} sim(B, B') \leq (1 + \alpha) \max_{B \in D_i, B' \in D_j} f(LCA_P(B, B')).$$

As D_i, D_j are nodes of the ground-truth tree, it holds that $f(LCA_P(B, B'))$ is the same for any choice of $B \in D_i, B' \in D_j$. In particular, this is true for $f(LCA_P(B_i, B_j))$.

Case 2: There exists B_ℓ such that $B_\ell \subseteq C_i$ and $B_\ell \nsubseteq D_i$ (or the same holds for C_i, D_i replaced by C_j, D_j). WLOG, suppose the former case holds. Then, there exists a child N of I whose children are N_L, N_R, such that $N_L \subseteq D_i$ and $N_R \cap D_i = \emptyset$. It then follows that

$$sim(N_L, N_R) \leq (1+\alpha) \max_{B \in N_L, B' \in N_R} f(LCA_P(B, B')) \leq (1+\alpha) f(LCA_P(B_i, B_j)),$$

where the second inequality holds because f is decreasing as we ascend P. However, we also have that $sim(N_L, N_R) \geq sim(C_i, C_j)$, as sim also obeys this property (if the last inequality did not hold, then N_L, N_R would not have been merged). This finishes the last case. □

The proof follows by applying Lemma 8 and then Lemma 9.

D.2. Proof of Theorem 5

D.2.1. Overview

When running DPCommunity, fix Y, Z_1, Z_2, and let $(\hat{A}_1, \hat{A}_2)$ and $(\hat{A}'_1, \hat{A}'_2)$ be the splits of $\hat{A}$ and an adjacent database $\hat{A}'$. We will view the matrix $F = P(\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2))$ as a vector, and then show that releasing F plus appropriate Gaussian noise satisfies privacy via the Gaussian mechanism. Our proof will bound the L_2 sensitivity of F, given by

$$\Delta_2(F) = \|P(\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2)) - P(\Pi^{(k)}_{\hat{A}'_1}(\hat{A}'_2))\|_F,$$

in terms of the quantity $\Gamma = \frac{\sigma_1(\hat{A}_2)}{\sigma_k(\hat{A}_1) - \sigma_k(\hat{A}_2)}$. Recall that P is a random $m \times \frac{n}{2}$ projection matrix. To control this sensitivity, we will need the fact that P preserves the distances in A via the Johnson-Lindenstrauss projection theorem:

Theorem 8. *(Johnson-Lindenstrauss projection theorem (Johnson, 1984)): Let $0 \leq \alpha < \frac{1}{2}$ and $0 \leq \beta \leq 1$, and $m = 8\frac{\ln \frac{2}{\beta}}{\alpha^2}$. If $x \in \mathbb{R}^n$ is a vector and $P \sim \mathcal{N}(0, \frac{1}{\sqrt{m}})^{m \times n}$ is a random matrix then with probability $1-\beta$, we have*

$$(1-\alpha)\|x\|_2 \leq \|Px\|_2 \leq (1+\alpha)\|x\|_2$$

We use the above theorem to show that the matrix P does not increase the sensitivity $\Delta(F)$ with high probability.

Lemma 10. *Let $0 \leq \delta < 1$ and $m = 64 \ln \frac{2n}{\delta}$. Then, if $P \sim \mathcal{N}(0, \frac{1}{\sqrt{m}})^{m \times n/2}$ the following holds with probability at least $1 - \frac{\delta}{4}$:*

$$\Delta(F) \leq \frac{3}{2} \|\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2) - \Pi^{(k)}_{\hat{A}'_1}(\hat{A}'_2)\|_F.$$

Proof. Let the columns of $\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2)$ be $\{a_1, \ldots, a_{n/4}\}$ and the columns of $\Pi^{(k)}_{\hat{A}'_1}(\hat{A}'_2)$ be $\{a'_1, \ldots, a_{n'/4}\}$. By the union bound, Theorem 8 with $\alpha = \frac{1}{2}$ and $\beta = \frac{\delta}{n}$ applies to all vectors $a_i - a'_i$ with probability at least $1 - \frac{\delta}{4}$. Thus, we have

$$\Delta_2(F)^2 = \sum_{i=1}^{n/4} \|P(a_i) - P(a'_i)\|_2^2 \leq (1+\alpha)^2 \sum_{i=1}^{n/4} \|a_i - a'_i\|_2^2 = (1+\alpha)^2 \|\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2) - \Pi^{(k)}_{\hat{A}'_1}(\hat{A}'_2)\|_F^2.$$

The result follows. □

Finally, we need a bound on the stability of the projection $\Pi^{(k)}_{\hat{A}_1}$ when $\hat{A}_1$ is perturbed. This is the result of the Davis-Kahan Theorem (Bhatia, 1997).

Theorem 9. *Let $\hat{A}_1, \hat{A}'_1$ be matrices where $d_k = \sigma_k(\hat{A}_1) - \sigma_{k+1}(\hat{A}_1) > 0$. Then,*

$$\|\Pi^{(k)}_{\hat{A}_1} - \Pi^{(k)}_{\hat{A}'_1}\|_F \leq \frac{\|\hat{A}_1 - \hat{A}'_1\|_F}{d_k}.$$

Furthermore, the above holds replacing $\|\cdot\|_F$ with $\|\cdot\|_2$.

Having bounded the L_2-sensitivity, we finally use the well-known Gaussian mechanism (Dwork et al., 2014a)

Theorem 10. *If $x \in \mathbb{R}^m$ has L_2 sensitivity at most S, then releasing $x + N$, where $N \sim \frac{S}{\epsilon}\sqrt{2 \ln \frac{1.25}{\delta}}\mathcal{N}(0,1)^m$ satisfies (ϵ, δ)-DP.*

D.2.2. Proof

Let $\hat{A}$ and $\hat{A}'$ be two adjacent inputs, and consider two runs of DPCommunity with fixed Y, Z_1, Z_2, and P; we will show that the outputs satisfy (ϵ, δ)-DP. Let $\hat{A}_1'$ and $\hat{A}_2'$ be the values of $\hat{A}_1$ and $\hat{A}_2$ when $\hat{A}'$ is used instead of $\hat{A}$. DPCommunity can be viewed as a post-processing of the private release of values $d_k = \sigma_k(\hat{A}_1) - \sigma_{k+1}(\hat{A}_1)$, $\sigma_1(\hat{A}_2)$, and F; thus, we will show that releasing each of these values satisfies privacy.

Using Lindskii's inequality (Bhatia, 1997), each rank i singular value of $\hat{A}_1, \hat{A}_2$ can only change by 1 when $\hat{A}$ is changed to $\hat{A}'$. Thus, the sensitivity of d_k is 2, of σ_1 is 1, and thus the release of $\tilde{d}_k = d_k + \frac{8}{\epsilon} \ln \frac{4}{\delta} + Lap(\frac{8}{\epsilon})$ and $\tilde{\sigma}_1 = \sigma_1 + \frac{4}{\epsilon} \ln \frac{4}{\delta} + Lap(\frac{4}{\epsilon})$ both satisfy $(\frac{\epsilon}{4}, 0)$-DP. Thus, we will show that releasing $\tilde{F}$ satisfies $(\frac{\epsilon}{2}, \delta)$-DP, and privacy will follow by composition.

By Lemma 10 with probability at least $1 - \frac{\delta}{4}$, we have

$$\Delta_2(F) \leq \tfrac{3}{2} \|\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2) - \Pi^{(k)}_{\hat{A}_1'}(\hat{A}_2')\|_F$$

We have either $\hat{A}_1 = \hat{A}_1'$ or $\hat{A}_2 = \hat{A}_2'$. We analyze the cases separately.

Case $\hat{A}_1 = \hat{A}_1'$: Then, $\hat{A}_2$ and $\hat{A}_2'$ differ in one bit, so $\hat{A}_2 = \hat{A}_2' + E$, where E is a matrix that is ± 1 in one entry and 0 everywhere else. Then,

$$\tfrac{3}{2} \|\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2) - \Pi^{(k)}_{\hat{A}_1}(\hat{A}_2')\|_F = \tfrac{3}{2} \|\Pi^{(k)}_{\hat{A}_1}(E)\|_F \leq \tfrac{3}{2} \|E\|_F \leq \tfrac{3}{2},$$

where the inequality holds because projecting vectors onto a subspace cannot increase their magnitude.

Case $\hat{A}_2 = \hat{A}_2'$: Then, $\hat{A}_1$ and $\hat{A}_1'$ differ in one bit, so $\|\hat{A}_1 - \hat{A}_1'\|_F \leq 1$. We have

$$\|\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2) - \Pi^{(k)}_{\hat{A}_1'}(\hat{A}_2')\|_F \leq 2k \|(\Pi^{(k)}_{\hat{A}_1} - \Pi^{(k)}_{\hat{A}_1'})(\hat{A}_2)\|_2 \leq 2k \|\Pi^{(k)}_{\hat{A}_1} - \Pi^{(k)}_{\hat{A}_1'}\|_2 \|\hat{A}_2\|_2,$$

where the first inequality holds because each term has rank at most k, so the entire quantity has rank at most $2k$, and the second holds by sub-multiplicativity of $\|\cdot\|_2$. By Theorem 9, we have $\|\Pi^{(k)}_{\hat{A}_1} - \Pi^{(k)}_{\hat{A}_1'}\|_2 \leq \frac{1}{d_k}$. Thus, we have

$$\frac{3}{2} \|\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2) - \Pi^{(k)}_{\hat{A}_1'}(\hat{A}_2')\|_F \leq \frac{3k \|\hat{A}_2\|_2}{d_k} = 3k\Gamma.$$

By concentration of Laplace variables, we have $\tilde{d}_k \leq d_k$ and $\tilde{\sigma}_1 \geq \sigma_1$, so $\Gamma \leq \frac{\tilde{\sigma}_1}{\tilde{d}_k} = \tilde{\Gamma}$ with probability at least $1 - \frac{\delta}{2}$. Thus, the sensitivity $\Delta(F)$ is at most $3k\tilde{\Gamma}$, and $(\frac{\epsilon}{2}, \frac{\delta}{4})$-DP follows via Theorem 10. Factoring in the aformentioned failure probabilities, the entire release of $\tilde{F}$ satisfies $(\frac{\epsilon}{2}, \delta)$-DP.

D.3. Proof of Corollary 6

D.3.1. Overview

Recall that DPCommunity sees a matrix $\hat{A}$ drawn from HSBM(B, P, f), with expectation matrix A. We define $\tau^2 = \max f(x)$, $s = \min_{i=1}^k |B_i|$, and $\Delta = \min_{u \in B_i, v \in B_j, i \neq j} \|A_u - A_v\|_2$. We will show that DPCommunity approximates $\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2)$, which is guaranteed to cluster the original communities via the following result (Vu, 2014). We let the columns of $\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2)$, which is indexed by the set Z_2, be $\{b_i : i \in Z_2\}$.

Theorem 11. *(Vu (2014)): There exists a universal constant C such that if $\tau^2 \geq C \frac{\log n}{n}$, $s \geq C \log n$, and $k < n^{1/4}$. $\Delta > C(\tau \sqrt{\frac{n}{s}} + \tau \sqrt{k \log n} + \frac{\tau \sqrt{nk}}{\sigma_k(A)})$, with probability at least $1 - n^{-1}$, then the columns $\{b_i : i \in Z_2\}$ in $\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2)$ satisfy:*

$$\|b_i - b_j\|_2 \leq \frac{\Delta}{4} \quad \textit{if } \exists u.\ i \in B_u, j \in B_u \textit{ (i.e. } i, j \textit{ are in the same community)}$$
$$\|b_i - b_j\|_2 \geq \Delta \quad \textit{otherwise.}$$

Thus, the clusters in $\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2)$ cluster the original communities assuming Δ is large enough. We will show that $\tilde{F}$ clusters the original communities assuming some condition on Δ. Since DPCommunity returns $\tilde{F} = P(\Pi^{(k)}_{\hat{A}_1}(\hat{A}_2)) + N$, where N is Gaussian noise, our proof involves showing that the distances in $\tilde{F}$ approximate those in $\Pi^{(k)}_{\hat{A}_1}$ using the Johnson-Lindenstrauss lemma and concentration of the Gaussian noise.

We formally restate Theorem 6:

Theorem 12. *Let $\hat{A}$ be drawn from $\mathrm{HSBM}(B, P, f)$. There is a universal constant $C > 2000$ such that if $\tau^2 \geq C\frac{\log n}{n}$, $s \geq C\sqrt{n \log n}$, $k < n^{1/4}$, $\delta < \frac{1}{n}$, $\sigma_k(A) \geq C \max\{\tau\sqrt{n}, \frac{1}{\epsilon}\ln\frac{4}{\delta}\}$, and*

$$\Delta > C \max\left\{\frac{k(\ln\frac{1}{\delta})^{3/2}}{\epsilon}\frac{\sigma_1(A)}{\sigma_k(A)}, \tau\sqrt{\frac{n}{s}} + \tau\sqrt{k\log n} + \frac{\tau\sqrt{nk}}{\sigma_k}\right\},$$

then with probability at least $1 - 3n^{-1}$, DPCommunity returns a set of points $\tilde{F} = \{f_i : i \in Z_2\}$ such that

$$\begin{aligned} \|f_i - f_j\|_2 &\leq \frac{2\Delta}{5} \quad \text{if } \exists u.\ i, j \in B_u \\ \|f_i - f_j\|_2 &\geq \frac{4\Delta}{5} \quad \text{otherwise}, \end{aligned}$$

and thus the clusters in $\tilde{F}$ indicate the communities.

D.3.2. PROOF OF THEOREM 12

Let the columns of $\tilde{F}$ be $\{f_i : i \in Z_2\}$. We have $f_i = P(b_i) + n_i$, where $n_i \sim \frac{3k\tilde{\Gamma}}{\epsilon}\sqrt{2\ln\frac{5}{\delta}}N(0,1)^m$. By concentration bounds, we have with probability $1 - \frac{1}{n}$ that each n_i satisfies $\|n_i\|_2 \leq \frac{3k\tilde{\Gamma}}{\epsilon}\sqrt{2\ln\frac{5}{\delta}}\sqrt{2m\ln n} \triangleq K$. Next, applying Theorem 8 on the vectors $b_i - b_j$ for $i, j \in Z_2$, we have $0.9\|b_i - b_j\|_2 \leq \|P(b_i) - P(b_j)\|_2 \leq 1.1\|b_i - b_j\|_2$ with probability $1 - \delta > 1 - \frac{1}{n}$. Thus, if $\exists u.\ i \in B_u, j \in B_u$, then

$$\begin{aligned} \|f_i - f_j\|_2 &\leq \|P(b_i) - P(b_j)\|_2 + \|n_i\|_2 + \|n_j\|_2 \\ &\leq 1.1\|b_i - b_j\|_2 + 2K \\ &\leq 0.275\Delta + 2K, \end{aligned}$$

Otherwise, we have

$$\begin{aligned} \|f_i - f_j\|_2 &\geq \|P(b_i) - P(b_j)\|_2 - \|n_i\|_2 - \|n_j\|_2 \\ &\geq 0.9\|b_i - b_j\|_2 - 2K \\ &\geq 0.9\Delta - 2K. \end{aligned}$$

Finally, we show that K can be upper bounded by the singular values of the expectation matrix A. This can be done with the following two lemmas which are proven implicitly in Vu (2014).

Lemma 11. *Let A be an $m \times n$ (with $m \geq n$) matrix of expectations in $[0, 1]$, and let $\hat{A}$ be a randomized rounding of A to $\{0, 1\}$. Then, with probability at least $1 - \frac{1}{n}$, we have for all $1 \leq i \leq m$, $|\sigma_i(A) - \sigma_i(\hat{A})| \leq 4\tau\sqrt{n} + 4\log n$, where τ^2 is the maximum probability in A.*

Proof. Each $\sigma_{i+1}(A)$ is equal to $\max_{\text{rank}(A_i)=i}\|A - A_i\|_2$. Let $A_i^*, \hat{A}_i^*$ be rank i matrices such that $\sigma_{i+1}(A) = \|A - A_i^*\|_2$ and $\sigma_{i+1}(\hat{A}) = \|\hat{A} - \hat{A}_i^*\|_2$. We have that $\sigma_{i+1}(A) \leq \|A - \hat{A}_i^*\|_2 \leq \|\hat{A} - \hat{A}_i^*\|_2 + \|A - \hat{A}\|_2$.

Thus, it remains to bound $\|A - \hat{A}\|_2$. Let the columns in $A - \hat{A}$ be $a_1, \ldots, a_n$. Using Lemma 7 from Vu (2005), we have that with probability at least $1 - \frac{1}{n^3}$, the length of the projection of a_i onto a basis vector e_i is at most $4(\tau + \frac{\log n}{\sqrt{n}})$. Thus, the total length $\|Ae_i\|_2$ is at most $4(\tau + \frac{\log n}{\sqrt{n}}$, and thus $\|A\|_2 \leq \sqrt{n}4(\tau + \frac{\log n}{\sqrt{n}})$ establishing that $\sigma_{i+1}(A) \leq \sigma_{i+1}(\hat{A}) + 4\sqrt{n}\tau + 4\log n$. Likewise, we can show that $\sigma_{i+1}(A) \geq \sigma_{i+1}(\hat{A}) - 4\sqrt{n}\tau - 4\log n$. □

Lemma 12. *Let A be an expectation matrix of* $\mathrm{HSBM}(B, P, f)$ *with k blocks with minimum block size $s \geq 16\sqrt{n \log n}$, and let C be the submatrix of A with rows Y and columns Z, where $|Y| = \frac{n}{2}$ and $|Z| = \frac{n}{4}$ are chosen randomly from $[n]$ such that $Y \cap Z = \emptyset$. Then, with probability at least $1 - \frac{1}{n}$, for all $1 \leq i \leq k$, we have*

$$(\tfrac{1}{8} - \tfrac{\sqrt{n \log n}}{s})\sigma_i(A_1) \leq \sigma_i(A_1) \leq (\tfrac{1}{8} + \tfrac{\sqrt{n \log n}}{s})\sigma_i(A_1)$$

Proof. Observe that the blocks in C are indexed in rows by $B_1 \cap Y, \ldots, B_k \cap Y$ and in columns by $B_1 \cap Z, \ldots, B_k \cap Z$. By Chernoff's bound, with probability at least $1 - \frac{1}{n^2}$, we have for all i that

$$\frac{1}{2} - \frac{\sqrt{n \log n}}{|B_i|} \leq \frac{|B_i \cap Y|}{|B_i|} \leq \frac{1}{2} + \frac{\sqrt{n \log n}}{|B_i|} \qquad \frac{1}{4} - \frac{\sqrt{n \log n}}{|B_i|} \leq \frac{|B_i \cap Z|}{|B_i|} \leq \frac{1}{4} + \frac{\sqrt{n \log n}}{|B_i|}.$$

We have $\sigma_k(A) = \min_{\text{rank}(A_{k-1})=k-1} \|A - A_{k-1}\|_F$ and $\sigma_k(C) = \min_{\text{rank}(C_{k-1})=k-1} \|C - C_{k-1}\|_F$; let A^*_{k-1} and C^*_{k-1} be the maximizers of the previous expressions. Let A' denote the matrix C^*_{k-1} with rows and columns duplicated such that each element $(A')_{ij}$ is equal to $(C^*_{k-1})_{xy}$, where x, y are any two points in the same block as i, j, respectively. Accounting for the duplication factors of each block, we have

$$\left(\frac{1}{2} - \frac{\sqrt{n \log n}}{s}\right)\left(\frac{1}{4} - \frac{\sqrt{n \log n}}{s}\right) \|A - A'\|_F \leq \|C - C^*_{k-1}\|_F,$$

and thus we see that $(\frac{1}{8} - \frac{\sqrt{n \log n}}{s})\sigma_k(A) \leq \sigma_k(A_1)$. By a similar sampling argument, we can show that $(\frac{1}{8} + \frac{\sqrt{n \log n}}{s})\sigma_k(A) \geq \sigma_k(A_1)$. Repeating the argument for $\sqrt{\sigma_i(A)^2 + \cdots \sigma_k(A)^2} = \min_{\text{rank}(A_{i-1})=i-1} \|A - A_{i-1}\|_F$, we obtain the result for all $1 \leq i \leq k$. □

Let A_1, A_2 be the expectation matrices of $\hat{A}_1, \hat{A}_2$ for fixed Y, Z_2. Using Lemmas 11 and 12, we have that $\sigma_1(\hat{A}_2) \leq \sigma_1(A_2) + 4\tau\sqrt{n} + 4 \log n \leq (\frac{1}{8} + \frac{\sqrt{n \log n}}{s})\sigma_1(A) + 4\tau\sqrt{n} + 4 \log n \leq \frac{3}{32}\sigma_1(A) + 4\tau\sqrt{n} + 4 \log n$. Applying these again, we obtain

$$\begin{aligned} d_k(\hat{A}_1) &= \sigma_k(\hat{A}_1) - \sigma_{k+1}(\hat{A}_1) \\ &\geq \sigma_k(A_1) - \sigma_{k+1}(A_1) - 8\tau\sqrt{n} - 8 \log n \\ &\geq (\frac{1}{8} - \frac{\sqrt{n \log n}}{s})(\sigma_k(A) - \sigma_{k+1}(A)) - 8\tau\sqrt{n} - 8 \log n \\ &\geq \frac{1}{16}\sigma_k(A) - 8\tau\sqrt{n} - 8 \log n \end{aligned}$$

Finally, we have $\tilde{\Gamma} = \frac{\tilde{\sigma}_1(\hat{A}_2)}{\tilde{d}_k(\hat{A}_1)}$, which with probability at least δ, will satisfy

$$\tilde{\Gamma} \leq \frac{\sigma_1(\hat{A}_2) + \frac{8}{\epsilon} \ln \frac{4}{\delta}}{d_k(\hat{A}_1) - \frac{16}{\epsilon} \ln \frac{4}{\delta}} \leq \frac{\frac{3}{32}\sigma_1(A) + 4\tau\sqrt{n} + 4 \log n + \frac{8}{\epsilon} \ln \frac{4}{\delta}}{\frac{1}{16}d_k(A) - 8\tau\sqrt{n} - 8 \log n - \frac{16}{\epsilon} \ln \frac{4}{\delta}}.$$

By our assumption that $\sigma_k(A) \geq 1024 \max\{\tau\sqrt{n}, \frac{1}{\epsilon} \ln \frac{4}{\delta}\}$, we obtain that $\hat{\Gamma} \leq 4\frac{\sigma_1(A)}{\sigma_k(A)}$, This implies that

$$K \leq \frac{12k\sqrt{m \ln \frac{5}{\delta} \ln n}}{\epsilon} \frac{\sigma_1(A)}{\sigma_k(A)} = \frac{48k\sqrt{2 \ln \frac{2n}{\delta} \ln \frac{5}{\delta} \ln n}}{\epsilon} \frac{\sigma_1(A)}{\sigma_k(A)} \leq \frac{96k(\ln \frac{5}{\delta})^{3/2}}{\epsilon} \frac{\sigma_1(A)}{\sigma_k(A)},$$

where the last step follows because $\delta < \frac{1}{n}$. From our assumption, we have $2K \leq 0.1\Delta$, and the result follows.

D.4. Proof of Corollary 1

In this special case, we can write $A = P \otimes \mathbf{1}_B$, where P is a $k \times k$ matrix with p on the diagonal and q everywhere else, $\mathbf{1}_s$ is a $s \times s$ matrix consisting of all 1s, and $\otimes$ denotes the Kronecker product. It is easy to see that the eigenvalues of P are $\{p + q(k-1), p - q, \ldots, p - q\}$, and the eigenvalues of $\mathbf{1}_s$ are $\{s, 0, \ldots, 0\}$. The eigenvalues of A are the product of the two sets of eigenvalues of P and $\mathbf{1}_s$. Thus, the top k largest eigenvalues are $s(p+q(k-1))$ and then $k-1$ copies of $s(p-q)$.

Thus, the following properties of A hold: (1) $\sigma_1 = s(p+q(k-1))] \leq sk(p+q)$, (2) $\sigma_k = s(p-q)$, (3) $\tau = \sqrt{p}$, and (4) $\Delta = (p-q)\sqrt{s}$. We are able to apply Theorem 12 when

$$\begin{aligned}(p-q)\sqrt{s} &\geq \frac{s(p+q)}{s(p-q)}\frac{Ck(\log\frac{1}{\delta})^{3/2}}{\epsilon}\\ \frac{(p-q)^2}{p+q} &\geq \frac{C(k\log\frac{1}{\delta})^{3/2}}{\sqrt{n}}.\end{aligned}$$

This establishes the result.