Hierarchical Clustering Beyond the Worst-Case

Vincent Cohen-Addad
University of Copenhagen
vcohenad@gmail.com

Varun Kanade
University of Oxford
Alan Turing Institute
varunk@cs.ox.ac.uk

Frederik Mallmann-Trenn
MIT
frederik.mallmann@googlemail.com

Abstract

Hiererachical clustering, that is computing a recursive partitioning of a dataset to obtain clusters at increasingly finer granularity is a fundamental problem in data analysis. Although hierarchical clustering has mostly been studied through procedures such as linkage algorithms, or top-down heuristics, rather than as optimization problems, recently Dasgupta [1] proposed an objective function for hierarchical clustering and initiated a line of work developing algorithms that explicitly optimize an objective (see also [2, 3, 4]). In this paper, we consider a fairly general random graph model for hierarchical clustering, called the hierarchical stochastic block model (HSBM), and show that in certain regimes the SVD approach of McSherry [5] combined with specific linkage methods results in a clustering that give an $O(1)$ approximation to Dasgupta's cost function. We also show that an approach based on SDP relaxations for balanced cuts based on the work of Makarychev *et al.* [6], combined with the recursive sparsest cut algorithm of Dasgupta, yields an $O(1)$ approximation in slightly larger regimes and also in the semi-random setting, where an adversary may remove edges from the random graph generated according to an HSBM. Finally, we report empirical evaluation on synthetic and real-world data showing that our proposed SVD-based method does indeed achieve a better cost than other widely-used heurstics and also results in a better classification accuracy when the underlying problem was that of multi-class classification.

1 Introduction

Computing a recursive partitioning of a dataset to obtain a finer and finer classification of the data is a classic problem in data analysis. Such a partitioning is often refered to as a *hierarchical clustering* and represented as a rooted tree whose leaves correspond to data elements and where each internal node induces a cluster of the leaves of its subtree. There exists a large literature on the design and analysis of algorithms for hierarchical clustering (see *e.g.,* [7]). Two main approaches have proven to be successful in practice so far: on the one hand *divisive* heuristics compute the hierarchical clustering tree in a top-down fashion by recursively partitioning the data (see *e.g.,* [8]). On the other hand, *agglomerative* heuristics produce a tree by first defining a cluster for each data elements and successively merging clusters according to a carefully defined function (see *e.g.,* [9]). These heuristics are widely used in practice and are now part of the data scientists' toolkit—standard machine learning libraries contain implementations of both types of heuristics.

Agglomerative heuristics have several appealing features: they are easy to implement, easy to tune,

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

and their running time is $\widetilde{O}(n^2)$[1] on a dataset of size n. Standard divisive heuristics based on graph partitioning or clustering methods (like for example the bisection k-means or the recursive sparsest-cut approaches) often involve solving or approximating NP-hard problems.[2] Therefore, it is natural to ask how good the solution output by an agglomerative method is compared to the solution output by a top-down method.

From a qualitative perspective, this question has been addressed in a large body of work (see *e.g.,* [10]). However, from a quantitative perspective little is known. As Dasgupta observes in his recent work [1], both agglomerative and divisive heuristics are defined procedurally rather than in term of an objective function to optimize, a reason why a quantitative comparision of the different heuristics is rather difficult. Dasgupta introduced an objective function to model the problem of finding a hierarchical clustering of a similarity graph—such an objective can be used to explicitly design optimization algorithms that minimize this cost function as well as serve as a quantitative measure of the quality of the output.

Given a similarity graph *i.e.,* a graph where vertices represent data elements and edge weights similarities between data elements, Dasgupta's objective function associates a cost to any hierarchical clustering tree of the graph. He showed that his objective function exhibits several desirable properties: For example, if the graph is disconnected *i.e.,* data elements in different connected components are very dissimilar, a tree minimizing this objective function will first split the graph according to the connected components.

This axiomatic approach to defining a "meaningful" objective function for hierarchical clustering has been further explored in recent work by Cohen-Addad *et al.* [4]. Roughly speaking, they characterize a family of cost functions, which includes Dasgupta's cost function, that when the input graph has a "natural" ground-truth hierarchical clustering tree (in other words a natural classification of the data), this tree has optimal cost (and any tree that is not a "natural" hierarchical clustering tree of the graph has higher cost). Therefore, the results by Dasgupta and Cohen-Addad *et al.* indicate that Dasgupta's cost function provides a sound framework for a rigorous quantitative analysis of agglomerative and divisive heuristics.

A suitable objective function to measure the quality of a clustering also allows one to explicitly design algorithms that minimize the cost. Dasgupta showed that the recursive sparsest-cut heuristic is an $O(\log^{3/2} n)$-approximation algorithm for his objective function. His analysis has been improved by Charikar and Chatziafratis [2] and Cohen-Addad et al. [4] to $O(\sqrt{\log n})$. Unfortunately, Charikar and Chatziafratis [2] and Roy and Pokutta [3] showed that, for general inputs, the problem cannot be approximated within any constant factor under the Small-Set Expansion hypothesis. Thus, as suggested by Charikar and Chatziafratis [2], a natural way to obtain a more fine-grained analysis of the classic agglomerative and divisive heuristics is to study *beyond-worst* case scenarios.

Random (and semi-random) Graph Model for Hierarchical Clustering. A natural way to analyse a problem beyond the worst-case is to consider a suitable random input model, which is the focus of this paper. More precisely, we introduce a random graph model and a semi-random graph model which are based on the notion of "hierarchical stochastic block model" (HSBM) introduced by Cohen-Addad *et al.*, which is a natural extensions of the stochastic block model introduced. Our random graph model relies on the notion of *ultrametric*, a metric in which the triangle inequality is strengthened by requiring $d(x,y) \leqslant \max(d(x,z),d(y,z))$. This is a key concept as ultrametrics exactly capture the notion of data having a "natural" hierarchical structure (cf. [10]). The random graphs are generated from data that comes from an ultrametric, but the randomness hides the natural hierarchical structure. Two natural questions are: Given a random graph generated in such a fashion, when is it possible to identify the underlying ultrametric and is the optimization of Dasgupta's cost function easier for graphs generated according to such a model. The former question was partially addressed by Cohen-Addad *et al.* and our focus is primarily on developing algorithms that achieve an $O(1)$ approximation to the expected Dasgupta cost, not on recovering the underlying ultrametric.

More formally, assume that the data elements lie in an unknown ultrametric space (A,dist) and so exhibit a natural hierarchical clustering defined by this ultrametric. The input is a random graph generated as follows: an edge is added between nodes $u,v \in A$ with probability $p = f(\text{dist}(u,v))$, where f is an (unknown) non-increasing function with range $(0,1)$.Thus, vertices that are very close in the ultrametric (and so very similar) have a higher probability to have an edge between them than vertices

[1] The $\widetilde{O}$ notation hides polylogarithmic factors.

[2] In some cases, it may be possible to have a very fast algorithms based on heuristics to compute partitions, however, we are unaware of any such methods that would have provable guarantees for the kinds of graphs that appear in hierarchical clustering.

that are further apart. Given such a random graph, the goal is to obtain a hierarchical clustering tree that has a good cost for the objective function. The *actual* ground-truth tree is optimal in expectation and we focus on designing algorithms that with high probability output a tree whose cost is within a constant factor of the expected cost of the ground-truth tree. Although, we do not study it in this work, the question of exact recovery is also an interesting one and the work of Cohen-Addad *et al.* [4] addresses this partially in certain regimes. We also consider the semi-random case, where an adversary may remove edges from the random graph generated as above, but not add any edges. Such a model has been considered by Makarychev *et al.* [6] in the context of planted partition problems. The goal is still to obtain a constant factor approximation to the expected cost of the ground-truth tree.

Algorithmic Results. Even in the case of random graphs, the linkage algorithms may perform quite poorly, mainly because ties may be broken unfavourably at the very bottom, when the clusters are singleton nodes; these choices cannot be easily compensated later on in the algorithm. We thus consider the LINKAGE++ algorithm which first uses a *seeding step* using a standard SVD approach to build clusters of a significant size, which is an extension of the algorithm introduced in [4]. Then, we show that using these clusters as starting point, the classic single-linkage approach achieves a $(1+\varepsilon)$-approximation for the problem (cf. Theorem 2.3).

We also consider the semi-random model and show that by recursively computing an $O(1)$-approximation to the problem of computing a (roughly) balanced min-cut produces an $O(1)$-approximation to the hierarchical clustering problem. To do so we harness an algorithm introduced by Makarychev et al. [6] for the Small-Set Expansion problem in a semi-random version of the stochastic blockmodel (cf. Theorem 2.6).

Experimental Results. We evaluate the performance of LINKAGE++ on real-world data (Scikit-learn) as well as on synthetic hierarchical data. The measure of interest is the Dasgupta cost function and for completeness we also consider the classification error (see *e.g.,* [3]). Our experiments show that 1) LINKAGE++ performs well on all accounts and 2) that a clustering with a low Dasgupta cost appears to be correlated with a good classification. On synthetic data LINKAGE++ seems to be clearly superior.

Related Work. Our work follows the line of research initiated by Dasgupta [1] and further studied by [3, 2, 4]. Dasgupta [1] introduced the cost function studied in this paper and showed that the recursive sparsest-cut approach yields an $O(\log^{3/2} n)$. His analysis was recently improved to $O(\sqrt{logn})$ by [2, 4]. Roy and Pokutta [3] and Charikar also considered LP and SDP formulations with spreading constraints to obtain approximation algorithms with approximation factor $O(\log n)$ and $O(\sqrt{\log n})$ respectively. Both these works also showed the infeasibility of constant factor approximations under the small-set expansion hypothesis. Cohen-Addad *et al.* [4] took an axiomatic approach to identify suitable cost functions for data generated from ultrametrics, which results in a natural ground-truth clustering. They also looked at a slightly less general hierarchical stochastic blockmodel (HSBM), where each bottom-level cluster must have a linear size and with stronger conditions on allowable probabilities. Their algorithm also has a "seeding phase" followed by an agglomerative approach. We go beyond their bounds by focusing on approximation algorithms (we obtain a $(1+\varepsilon)$-approximation) whereas they aim at recovering the underlying ultrametric. As the experiments show, this trade-off seem not to impact the classification error compared to classic other approaches.

There is also a vast literature on graph partitionning problems in random and semi-random models. Most of this work (see *e.g.,* [5, 11]) focuses on recovering a hidden subgraph *e.g.,* a clique, whereas we address the problem of obtaining good approximation guarantees w.r.t. an objective function. At a high-level our approach is related to the work of Makarychev et al. [6, 12] in the semi-random model for graph partitioning objectives like balanced cut, multicut, etc.

The reader may refer to [13, 14] for the definitions and the classic properties on agglomerative and divisive heuristics. Agglomerative and divisive heuristics have been widely studied from either a qualitative perspective or for classic "flat" clustering objective like the classic k-median and k-means, see *e.g.,* [15, 16, 17, 18, 19]. For further background on hierarchical clustering and its application in machine learning and data science, the reader may refer to *e.g.,* [20, 21, 22, 23].

Preliminaries In this paper, we work with undirected weighted graph $G=(V,E,w)$, where V is a set of vertices, E a set of edges, and $w: E \to \mathbb{R}_+$. In the random and semi-random model, we work with unweighted graphs. We slightly abuse notation and extend the function w to subsets of V. Namely, for any $A, B \subseteq V$, let $w(A,B) = \sum_{a\in A, b\in B} w(a,b)$. We use weights to model similarity, namely $w(u,v) > w(u,w)$ means that data element u is more similar to v than to w. When G is clear from

the context, we let $|V| = n$ and $|E| = m$. For any subset S of vertices of a graph G, let $G[S]$ be the subgraph induced by the nodes of S.

In the following, let $G = (V,E,w)$ be a weighted graph on n vertices. A *cluster tree* or *hierarchical clustering* T for G is a rooted binary tree with exactly $|V|$ leaves, each of which is labeled by a distinct vertex $v \in V$. We denote $\text{LCA}_T(u,v)$ the lowest common ancestor of vertices u,v in T. Given a tree T and a node N of T, we say that the subtree of N in T is the the connected subgraph containing all the leaves of T that are descendant of N and denote this set of leaves by $V(N)$. A metric space (X,d) is an ultrametric if for every $x,y,z \in X$, $d(x,y) \leqslant \max\{d(x,z),d(y,z)\}$.

We borrow the notion of a (similarity) *graph generated from an ultrametric* and *generating tree* introduced by [4]. A weighted graph $G = (V,E,w)$ is a generated from an ultrametric, if there exists an ultrametric (X,d), such that $V \subseteq X$, and for every $x,y \in V, x \neq y$, $e = \{x,y\}$ exists, and $w(e) = f(d(x,y))$, where $f : \mathbb{R}_+ \to \mathbb{R}_+$ is a non-increasing function.

Definition 1.1 (Generating Tree)**.** *Let $G = (V,E,w)$ be a graph generated by a minimal ultrametric (V,d). Let T be a rooted binary tree with $|V|$ leaves; let $\mathcal{N}$ denote the internal nodes and L the set of leaves of T and let $\sigma : L \to V$ denote a bijection between the leaves of T and nodes of V. We say that T is a generating tree for G, if there exists a weight function $W : \mathcal{N} \to \mathbb{R}_+$, such that for $N_1,N_2 \in \mathcal{N}$, if N_1 appears on the path from N_2 to the root, $W(N_1) \leqslant W(N_2)$. Moreover for every $x,y \in V$, $w(\{x,y\}) = W(LCA_T(\sigma^{-1}(x),\sigma^{-1}(y)))$.*

As noted in [4], the above notion bear similarities to what is referred to as a *dendrogram* in the machine learning literature (see *e.g.,* [10]).

Objective Function. We consider the objective function introduced by Dasgupta [1]. Let $G = (V,E,w)$ be a weighted graph and $T = (\mathcal{N},\mathcal{E})$ be any rooted binary tree with leaves set V. The cost induced by a node N of T is $\text{cost}_T(N) = |V(N)| \cdot w(V(C_1),V(C_2))$ where C_1,C_2 are the children of N in T. The cost of T is $\text{cost}_T = \sum_{N \in \mathcal{N}} \text{cost}_T(N)$. As pointed out by Dasgupta [1], this can be rephrased as $\text{cost}_T = \sum_{(u,v) \in E} w(u,v) \cdot |V(\text{LCA}_T(u,v))|$.

2 A General Hierarchical Stochastic Block Model

We introduce a generalization of the HSBM studied by [4] and [24]. Cohen-Addad et al. [4] introduce an algorithm to recover a "ground-truth" hierarchical clustering in the HSBM setting. The regime in which their algorithm works is the following: (1) there is a set of hidden clusters that have linear size and (2) the ratio between the minimum edge probability and the maximum edge probability is $O(1)$. We aim at obtaining an algorithm that "works" in a more general setting. We reach this goal by proposing on $(1+\varepsilon)$-approximation algorithms. Our algorithm very similar to the widely-used linkage approach and remains easy to implement and parallelize. Thus, the main message of our work is that, on "structured inputs" the agglomerative heuristics perform well, hence making a step toward explaining their success in practice.

The graphs generated from our model possess an underlying, hidden (because of noise) "ground-truth hierarchical clustering tree" (see Definition 2.1). This aims at modeling real-world classification problem for which we believe there is a natural hierarchical clustering but perturbed because of missing information or measurement erros. For example, in the tree of life, there is a natural hierarchical clustering hidden that we would like to reconstruct. Unfortunately because of extinct species, we don't have a perfect input and must account for noise. We formalize this intuition using the notion of generating tree (Def 1.1) which, as hinted at by the definition, can be associated to an ultrametric (and so a "natural" hierarchical clustering). The "ground-truth tree" is the tree obtained from a generating tree on k leaves to which we will refer as "bottom"-level clusters containing $n_1,n_2,...,n_k$ nodes (following the terminology in [4]). Each edge of a generated graph has a fixed probability of being present, which only depends on the underlying ground-truth tree. This probability is a function of the clusters in which their endpoints lie and the underlying graph on k vertices for which the generating tree is generating (as in Def 1.1).

Definition 2.1 (Hierarchical Stochastic Block Model – Generalization of [4])**.** *Let n be a positive integer. A hierarchical stochastic block model with k bottom-level clusters is defined as follows:*

1) Let $\widetilde{G}_k = (\widetilde{V}_k,\widetilde{E}_k,w)$ be a graph generated from an ultrametric, where $|\widetilde{V}_k| = k$ for each $e \in \widetilde{E}_k$, $w(e) \in (0,1)$. let $\widetilde{T}_k$ be a tree on k leaves, let $\widetilde{\mathcal{N}}$ denote the internal nodes of $\widetilde{T}$ and $\widetilde{L}$ denote the leaves; let $\widetilde{\sigma} : \widetilde{L} \to [k]$ be a bijection. Let $\widetilde{T}$ be generating for $\widetilde{G}_k$ with weight function $\widetilde{W} : \widetilde{\mathcal{N}} \to [0,1)$.

2) For each $i \in [k]$, let $p_i \in (0,1]$ be such that $p_i > \widetilde{W}(N)$, if N denotes the parent of $\widetilde{\sigma}^{-1}(i)$ in $\widetilde{T}$.

3) For each $i \in [k]$, there is a positive integer n_i such that $\sum_{i=1}^{k} n_i = n$.
Then a random graph $G = (V,E)$ on n nodes is defined as follows. Each vertex $i \in [n]$ is assigned a label $\psi(i) \in [k]$, so that exactly n_j nodes are assigned the label j for $j \in [k]$. An edge (i,j) is added to the graph with probability $p_{\psi(i)}$ if $\psi(i) = \psi(j)$ and with probability $\widetilde{W}(N)$ if $\psi(i) \neq \psi(j)$ and N is the least common ancestor of $\widetilde{\sigma}^{-1}(i)$ and $\widetilde{\sigma}^{-1}(j)$ in $\widetilde{T}$. The graph $G = (V,E)$ is returned without any labels.

We use, for a generating tree $\widetilde{T}$, the notation $p_{\min}$ to denote $\widetilde{W}(N_0)$, where N_0 is the root node of $\widetilde{T}$. Let $n_{\min}$ be the size of the smallest cluster (of the k clusters) As in [4], we will use the notion of *expected graph*. The *expected graph* as the is the weighted complete graph $\bar{G}$ in which an edge (i,j) has weight $p_{i,j}$, where $p_{i,j}$ is the probability with which it appears in the random graph G. We refer to any tree that is generating for the expected graph $\bar{G}$ as a *ground-truth tree* for G. In order to avoid ambiguity, we denote by $\text{cost}_T(G)$ and $\text{cost}_T(\bar{G})$ the costs of the cluster tree T for the unweighted (random) graph G and weighted graph $\bar{G}$ respectively. Observe that due to linearity of expectation for any tree T and any admissible cost function, $\text{cost}_T(\bar{G}) = \mathbb{E}[\text{cost}_T(G)]$, where the expectation is with respect to the random choices of edges in G. We investigate the cost of a ground-truth tree in Proposition C.2 and the following theorem.

Algorithm LINKAGE++, a $(1+\varepsilon)$-Approximation Algorithm in the HSBM. We consider a simple algorithm, called LINKAGE++, which works in two phases (see Alg. 1 for more details in Section C):

1) Apply an SVD to the input data and apply single-linkage using the Euclidean distance to build big enough clusters.

2) Consider these bottom clusters in the original input and apply single-linkage using the edge weights of the input graph to finish building the hierarchical clustering tree.

We use a result of [5] who considers the planted partition model. His approach however does not allow to recover directly a hierarchical structure when the input has it.

Theorem 2.2 ([5], Observation 11 and a simplification of Theorem 12)**.** *Let δ be the confidence parameter. Assume that for all u,v belonging to different clusters with adjacency vectors $\mathbf{u},\mathbf{v}$ (i.e., u_i is 1 if the edge (u,i) exists in G and 0 otherwise) satisfy*

$$\|\mathbb{E}[\mathbf{u}] - \mathbb{E}[\mathbf{v}]\|_2^2 \geqslant c \cdot k \cdot \left(\sigma^2 n / n_{\min} + \log(n/\delta)\right) \tag{1}$$

for a large enough constant c, where $\mathbb{E}[\mathbf{u}]$ is the entry-wise expectation and $\sigma^2 = \omega(\log^6 n/n)$ is an upper bound on the variance. Then, the algorithm of [5, Thm. 12] with parameters G,k,δ projects the columns of the adjacency matrix of G to points $\{\zeta(1),\ldots,\zeta(|V|)\}$ in a k-dimensional subspace of $\mathbb{R}^{|V|}$ such that the following holds w.p. at least $1-\delta$ over the random graph G and with probability $1/k$ over the random bits of the algorithm. There exists $\eta > 0$ such that for any u in the ith cluster and v in the jth cluster: 1) if $i = j$ then $\|\zeta(u) - \zeta(v)\|_2^2 \leqslant \eta$ and 2) if $i \neq j$ then $\|\zeta(u) - \zeta(v)\|_2^2 > 2\eta$.

In the remainder we assume $\delta = 1/|V|^2$. We are ready to state our main theorem.

Theorem 2.3. *Let n be a positive integer and $\varepsilon > 0$ a constant. Assume that the separation of bottom clusters given by* (1) *holds, $p_{\min} = \omega(\sqrt{\log n/n})$, and $n_{\min} \geqslant \sqrt{n} \cdot \log^{1/4} n$. Let k be a fixed constant and G be a graph generated from an* HSBM *(as per Defn. 2.1) where the underlying graph $\widetilde{G}_k$ has k nodes with satisfying the above constraints.*

With high probability, Algorithm 1 with parameter k on graph G outputs a tree T' that satisfies $\text{cost}_{T'} \leqslant (1+\varepsilon)OPT$.

We note that k might not be known in advance. However, different values of k can be tested and an $O(1)$-estimate on k is enough for the proofs to hold. Thus, it is possible to run Algorithm 1 $O(\log n)$ times with different "guesses" for k and take the best of these runs.

Let $G = (V,E)$ be the input graph generated according to an HSBM. Let T be the tree output by Algorithm 1. We divide the proof into two main lemmas that correspond to the outcome of the two phases mentioned above.

The algorithm of [5, Thm. 12] might fail for two reasons: The first reason is that the random choices by the algorithm result in an incorrect clustering. This happens w.p. at most $1-1/k$ and we can simply repeat the algorithm sufficiently many times to be sure that at least once we get the desired result, *i.e.,* the projections satisfy the conclusion of Thm. 2.2. Lemmas 2.4, 2.5 show that in this case, Steps 6 to 11 of LINKAGE++ produce a tree that has cost close to optimal. Ultimately, the algorithm simply

outputs a tree that has the least cost among all the ones produced (and one of them is guaranteed to have cost $(1+\varepsilon)$OPT) with high probability.

The second reason why the McSherry's algorithm may fail is that the generated random graph G might "deviate" too much from its expectation. This is controlled by the parameter δ (which we set to $1/|V|^2$). Deviations from expected behaviour will cause our algorithm to fail as well. We bound this failure probability in terms of two events. The first bad event is that McSherry's algorithm fails for either of the aforementioned reasons. We denote the complement of this event $\mathcal{E}_1$. The second bad event it that the number of edges between the vertices of two nodes of the ground-truth tree deviates from it's expectation. Namely, that given two nodes N_1,N_2 of T^*, we expect the cut to be $E_{(N_1,N_2)} = |V(N_1)|\cdot|V(N_1)|\cdot W(\text{LCA}_{T^*}(N_1,N_2))$. Thus, we define $\mathcal{E}_2$ to be the event that $|w(V(N_1),V(N_2)) - E_{(N_1,N_2)}| < \varepsilon^2 E_{(N_1,N_2)}$ for all cuts of the k bottom leaves. Note that the number of cuts is bounded by 2^k and we will show that, due to size of $n_{\min}$ and $p_{\min}$ this even holds w.h.p.. The assumptions on the ground-truth tree will ensure that the latter holds w.h.p. allowing us to argue that both events hold w.p. at least $\Omega(1/k)$ Thus, from now on we assume that both "good" events $\mathcal{E}_1$ and $\mathcal{E}_2$ occur. We bound the probability of event $\mathcal{E}_1$ in Lemma C.1. We now prove a structural properties of the tree output by the algorithm, we introduce the following definition. We say that a tree $T=(\mathcal{N},\mathcal{E})$ is a γ-*approximate ground-truth tree* for G and T^* if there exists a weight function $W':\mathcal{N}\mapsto\mathbb{R}_+$ such that for any two vertices a,b, we have that

1. $\gamma^{-1}W'(\text{LCA}_T(a,b)) \leqslant W(\text{LCA}_{T^*}(a,b)) \leqslant \gamma W'(\text{LCA}_T(a,b))$ and
2. for any node N of T and any node N' descendant of N in T, $W(N)\leqslant W(N')$.

Lemma 2.4. *Assume that the separation of bottom clusters given by* (1) *holds,* $p_{\min}=\omega(\sqrt{\log n/n})$, *and* $n_{\min}\geqslant\sqrt{n}\cdot\log^{1/4}n$. *Let G be generated according to an* HSBM *and let T^* be a ground-truth tree for G. Assume that events $\mathcal{E}_1$ and $\mathcal{E}_2$ occur, and that furthermore, the clusters obtained after Step 4 correspond to the assignment ψ, i.e., there exists a permutation $\pi:[k]\to[k]$ such that for each $v\in C_i$, $\psi(v)=\pi(i)$. Then, the output by the algorithm is a $(1+\varepsilon)$-approximate ground-truth tree.*

The following lemma allows us to bound the cost of an approximate ground-truth tree.

Lemma 2.5. *Let G be a graph generated according to an* HSBM *and let T^* be a ground-truth tree for G. Let $\bar{G}$ be the expected graph associated to T^* and G. Let T be a γ-approximate ground-truth tree. Then, $cost_T\leqslant\gamma^2 OPT$.*

This allows us to bound the outcome of the second phase using the following lemma and prove the main theorem of this section.

Proof of Theorem 2.3. Conditioning on $\mathcal{E}_1$ and $\mathcal{E}_2$ which occur w.h.p. and combining Lemmas C.1, 2.5, and 2.4 together with Theorem C.3 yields the result. As argued before, $\mathcal{E}_1$ holds at least w.p. $1/k$ and it is possible to boost part of this probability by running Algorithm 1 multiple times. Running it $\Omega(k\log n)$ times and taking the tree with the smallest cost yields the result. Moreover, $\mathcal{E}_2$ also holds w.h.p.: Note that due to our assumption the expected number of edges on each of these cuts is $n_{\min}^2 p_{\min} = \omega(\sqrt{n}\log n)$ and hence, Chernoff bounds, give us a probability of at least $1-2^{-k\log n}$. Taking union bound over all 2^k cuts yields the result, where we used that $k\leqslant\sqrt{n}$ due to the bound on $n_{\min}$. □

2.1 Algorithm for Semi-Random Model using SDP Relaxations

We show that in random and semi-random graph models generated according to an HSBM and SDP-based algorithm can be used to guarantee an $O(1)$-approximation with high-probability in a regime beyond that proved in Theorem 2.3. The proof of the following result follows using the technique of Makarychev *et al.* [6] to obtain $O(1)$-approximations to problems such as sparsest cut and small-set expansion (SSE) in random and semi-random settings combined with a result in Cohen-Addad *et al.*. [4] that shows that approximations to (roughly) balanced min-cut problems can be used to obtain an equivalent approximation ratio for the problem of finding a minimum cost hierarchical cluster tree.

To generate a random graph, $G=(V,E)$, we use an HSBM (Defn. 2.1), however, we allow allow $k=|\widetilde{V}_k|$ to be as large as n, *i.e.,* bottom-level clusters may be individual nodes, and allow the weights $w(e)$ to depend on n. The semi-random model simply considers a random graph generated as above and an adversary is allowed to remove edges from G, but not add any. Note that in either case the comparison is to the cost of the generating tree on the graph $\widetilde{G}$ (cf. Defn. 2.1).

Theorem 2.6. *Let $\widetilde{G}=(\widetilde{V}_n,\widetilde{E}_n,w)$ be a graph generated from an ultrametric where with $|\widetilde{V}_n|=n$ and $w:\widetilde{E}_n\to(0,1)$ satisfying $p_{\min}:=\min_{(u,v)\in\widetilde{E}_n}w(e)=\Omega(\log n/n^{2/3})$. Let $\widetilde{T}$ be a generating tree for $\widetilde{G}$. Suppose $G=(V,E)$ is a random graph with $V=\widetilde{V}_n$ generated as follows: an edge $e=(u,v)$ is added to E with probability $w(e)$. Then, there exists a randomized polynomial time algorithm that with probability $1-o(1)$ outputs a tree T such that, $cost_T=O(OPT(G))$, where $OPT(G)$ denotes the value of the optimal tree for G. Furthermore, the above holds even in the semi-random case, i.e., when an adversary is allowed to remove any subset of the edges from E (though the adversary cannot add any edges).*

3 Empirical Evaluation

In this section, we evaluate the effectiveness of LINKAGE++ on real-world and synthetic datasets. We compare our results to the classic agglomerative heuristics for hierarchical clustering both in terms of the cost function and the classification error. Our goal is answering the question: *How good is* LINKAGE++ *compared to the classic agglomerative approaches on real-world and synthetic data that exhibit a ground-truth clustering?*

Datasets. The datasets we use are part of the standard Scikit-learn library [25] (and most of them are available at the UCI machine learning repository [26]). Most of these datasets exhibit a "flat" clustering structure, with the exception of the `newsgroup` datasets which is truly hierarchical. The goal of the algorithm is to perform a clustering of the data by finding the underlying classes. The datasets are: `iris`, `digits`, `newsgroup`[3], `diabetes`, `cancer`, `boston`. For a given dataset, we define similarity between data elements using the *cosine similarity*, this is a standard approach for defining similarity between data elements (see, *e.g.,* [3]) This induces a weighted similarity graph that is given as input to LINKAGE++.

Synthethic Data. We generate random graphs of sizes $n\in\{256,512,1024\}$ according to the model described in Section 2.1. More precisely, we define a binary tree on $\ell\in\{4,8\}$ bottom clusters/leaves. Each leaf represents a "class". We create n/ℓ vertices for each class. The probability of having an edge between two vertices of class a and b is given by the probability induced by lowest common ancestor between the leaves corresponding to a and b respectively. We first define $p_{\min}=2\log n\cdot\ell/n$. The probability induced by the vertices of the binary tree are the following: the probability at the root is $p=p_{\min}+(1-p_{\min})/\log(\ell)$, and the probability induced by a node at distance d from the root is $(d+1)p$. In particular, the probability induced by the leaves is $p_{\min}+\log(\ell)(1-p_{\min})/\log(\ell)=1$. We also investigate a less structured setting using a ground truth tree on three nodes.

Method. We run LINKAGE++ with 9 different breakpoints at which we switch between phase 1 and phase 2 (which corresponds to "guesses" of k). We output the clustering with the smallest cost. To evaluate our algorithm, we compare its performances to classic agglomerative heuristics (for the similarity setting): single linkage, complete linkage, (see also [13, 14] for a complete description) and to the approach of performing only phase 1 of LINKAGE++ until only one cluster remains; we will denote the approach as PCA+. Additionally, we compare ourselves to applying only phase 2 of LINKAGE++, we call this approach density-based linkage. We observe that the running times of the algorithms are of order $\widetilde{O}(n^2)$ stemming already from the agglomerative parts.[4] This is close to the $\widetilde{O}(n^2))$ running time achieved by the classic agglomerative heuristics.

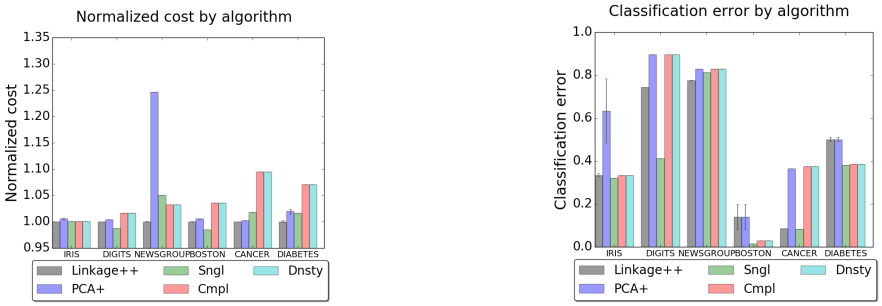
We compare the results by using both the cost of the output tree w.r.t. the hierarchical clustering cost function and the *classification error*. The classification error is a classic tool to compare different (usually flat) clusterings (see, *e.g.,* [3]). For a k-clustering $C:V\mapsto\{1,...,k\}$, the classification error w.r.t. a ground-truth flat clustering $C^*:V\mapsto\{1,...,k\}$ is defined as $\min_{\sigma\in S_k}\left(\sum_{x\in V}\mathbf{1}_{C(x)\neq\sigma(C^*(x))}\right)/|V|$, where S_k is the set of all permutations σ over k elements.

We note that the cost function is more relevant for the `newsgroup` dataset since it exhibits a truly hierarchical structure and so the cost function is presumably capturing the quality of the classification at different levels. On the other hand, the classification error is more relevant for the others data sets as they are intrinsically flat. All experiments are repeated at least 10 times and standard deviation is shown.

Results. The results are summarized in Figure 1, 2, and 3 (App. A). Almost in all experiments LINKAGE++ performs extremlely well w.r.t. the cost and classification error. Moreover, we observe

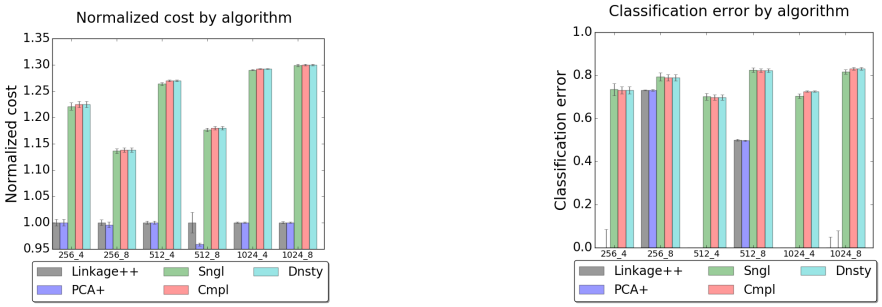
[3]Due to the enormous size of the dataset, we consider a subset consisting of 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'rec.sport.baseball', 'rec.sport.hockey'

[4]Top k singular vectors of an $n\times n$ matrix can be approximately computed in time $\widetilde{O}(kn^2)$.



(a) (b)

Figure 1: A comparison of the algorithms on real-world data. (a) The figure shows the cost cost$(\cdot)$ of the algorithm normalized by the the cost of LINKAGE++. (b) The figure shows the percentage of misclassified nodes. By looking more closely at the output of the algorithm, one can see that a large fraction of the misclassifications happen in subgroups of the same group.



(a) (b)

Figure 2: A comparison of the algorithms on synthetic data for highly structured ground-truth for different n,k. PCA+ performs well on these inputs and we conjecture that this due to the highly structured nature of the ground-truth. (a) The cost of LINKAGE++ and PCA+ are well-below the costs' of the standard linkage algorithms. (b) We see a threshold phenomena for $k=8$ from $n=512$ to $n=1024$. Here the classification error drops from 0.5 to 0, which is explained by concentration of the eigenvalues allowing the PCA to separated the bottom clusters correctly.

that a low cost function correlates with a good classification error. For synthetic data, in both LINKAGE++ and PCA+, we observe in Figure 2b that classification error drops drastically from $k=4$ to $k=8$, from 0.5 to 0 as the size is number of nodes is increased from $n=512$ to $n=1024$. We observe this threshold phenomena for all fixed k we considered. We can observe that the normalized cost in Figure 2a for the other linkage algorithms increases in the aforementioned setting.

Moreover, the only dataset where LINKAGE++ and PCA+ differ significantly is the hierarchical dataset `newsgroup`. Here the cost function of PCA+ is much higher. While the classification error of all algorithm is large, it turns out by inspecting the final clustering of LINKAGE++ and PCA+ that the categories which were being misclassified are mostly sub categories of the same category. On the dataset of Figure 3 (App. A) only LINKAGE++ performs well.

Conclusion. Overall both algorithms LINKAGE++ and Single-linkage perform considerably better when it comes to real-world data and LINKAGE++ and PCA+ dominate on our synthetic datasets. However, in general there is no reason to believe that PCA+ would perform well in clustering truly hierarchical data: there are regimes of the HSBM for which applying only phase 1 of the algorithm might lead to a high missclassification error and high cost and for which we can prove that LINKAGE++ is an $(1+\varepsilon)$-approximation.

This is exemplified in Figure 3 (App. A). Moreover, our experiments suggest that one should use in addition to LINKAGE++ other linkage algorithm and pick the algorithm with the lowest cost function, which appears to correlate with the classification error. Nevertheless, a high classification error of hierarchical data is not a bad sign per se: A misclassification of subcategories of the same categories (as we observe in our experiments) is arguably tolerable, but ignored by the classification error. On the other hand, the cost function captures such errors nicely by its inherently hierarchical nature and we thus strongly advocate it.

Acknowledgement The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 748094.

References

[1] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pages 118–127, New York, NY, USA, 2016. ACM.

[2] Moses Charikar and Vaggos Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 841–854, 2017.

[3] Aurko Roy and Sebastian Pokutta. Hierarchical clustering via spreading metrics. In *Advances In Neural Information Processing Systems*, pages 2316–2324, 2016.

[4] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. *CoRR*, abs/1704.02147, 2017.

[5] Frank McSherry. Spectral partitioning of random graphs. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 529–537, 2001.

[6] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 367–384. ACM, 2012.

[7] Chandan K Reddy and Bhanukiran Vinzamuri. A survey of partitional and hierarchical clustering algorithms. *Data Clustering: Algorithms and Applications*, 87, 2013.

[8] Alain Guénoche, Pierre Hansen, and Brigitte Jaumard. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *Journal of classification*, 8(1):5–30, 1991.

[9] Fionn Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359, 1983.

[10] Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11:1425–1470, 2010.

[11] Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *J. Comput. Syst. Sci.*, 63(4):639–671, December 2001.

[12] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant factor approximation for balanced cut in the PIE model. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 41–49. ACM, 2014.

[13] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *In KDD Workshop on Text Mining*, 2000.

[14] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[15] C Greg Plaxton. Approximation algorithms for hierarchical location problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 40–49. ACM, 2003.

[16] Sanjoy Dasgupta and Philip M Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.

[17] Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P Williamson. A general approach for incremental approximation and hierarchical clustering. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1147–1156. Society for Industrial and Applied Mathematics, 2006.

[18] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 671–680, New York, NY, USA, 2008. ACM.

[19] Maria-Florina Balcan and Yingyu Liang. Clustering under perturbation resilience. *SIAM J. Comput.*, 45(1):102–155, 2016.

[20] N. Jardine and R. Sibson. *Mathematical Taxonomy*. Wiley series in probability and mathematical statistiscs. John Wiley & Sons, 1972.

[21] Peter HA Sneath and Robert R Sokal. Numerical taxonomy. *Nature*, 193(4818):855–860, 1962.

[22] Joseph Felsenstein and Joseph Felenstein. *Inferring phylogenies*, volume 2. Sinauer Associates Sunderland, 2004.

[23] Rui M Castro, Mark J Coates, and Robert D Nowak. Likelihood based hierarchical clustering. *IEEE Transactions on signal processing*, 52(8):2308–2321, 2004.

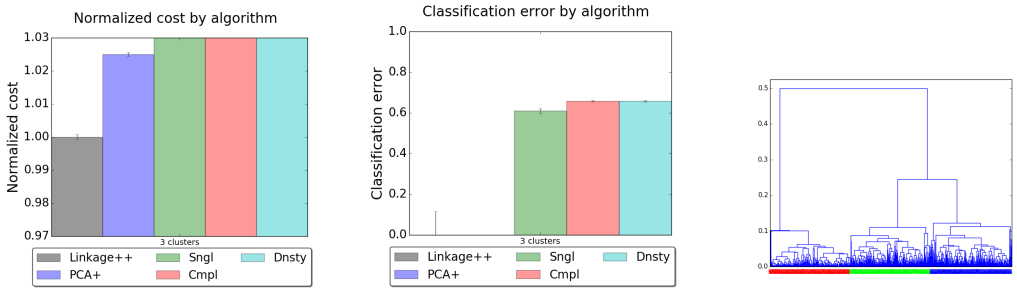
[24] Vince Lyzinski, Minh Tang, Avanti Athreya, Youngser Park, and Carey E Priebe. Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions on Network Science and Engineering*, 4(1):13–26, 2017.

[25] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[26] D. J. Newman A. Asuncion. UCI machine learning repository, 2007.

[27] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

A Additional Experimental Evaluation



(a) (b) (c)

Figure 3: The clustering obtained by PCA+ on a ground truth tree on three nodes induced by the adjacency matrix [[1.,0.49,0.39][0.49,0.49,0.39][0.39,0.39,0.62]] and $n = 999$ nodes split equally. Here only LINKAGE++ and PCA+ classify the bottom clusters of the subtrees correctly. However, the projection to the euclidian space (PCA) does not preserve the underlying ultramtric causing PCA+ to merge incorrectly. (a) LINKAGE++ recovers the ground truth. All other algorithm merge incorrectly. (b) LINKAGE++ and PCA+ classify the bottom clusters correctly causing the classification to be perfect even though PCA+ failed to correctly reconstruct the ground-truth. This suggests that the classification error is less suitable measure for hierarchical data. (c) PCA+ in contrast to LINKAGE++ merges incorrectly two bottom clusters of different branches in the ground-truth tree (green and blue as opposed to green and red).

B Algorithm in Semi-Random Model using SDP Relaxations

This section is dedicated to the proof of the following theorem.

Theorem 2.6. *Let $\widetilde{G} = (\widetilde{V}_n, \widetilde{E}_n, w)$ be a graph generated from an ultrametric where with $|\widetilde{V}_n| = n$ and $w : \widetilde{E}_n \to (0,1)$ satisfying $p_{\min} := \min_{(u,v)\in\widetilde{E}_n} w(e) = \Omega(\log n / n^{2/3})$. Let $\widetilde{T}$ be a generating tree for $\widetilde{G}$. Suppose $G = (V,E)$ is a random graph with $V = \widetilde{V}_n$ generated as follows: an edge $e = (u,v)$ is added to E with probability $w(e)$. Then, there exists a randomized polynomial time algorithm that with probability $1 - o(1)$ outputs a tree T such that,*

$$cost_T = O(OPT(G)), \tag{2}$$

where $OPT(G)$ denotes the value of the optimal tree for G. Furthermore, the above holds even in the semi-random case, i.e., when an adversary is allowed to remove any subset of the edges from E (though the adversary cannot add any edges).

B.1 Background

In this section, we recall the work of Makarychev *et al.* [6]. Essentially all of this section is directly cited from this work and we only provide it in this paper for completeness.

While we don't require to go into details, we define the crude SDP for Small-Set Expansion (SSE) used by Makarychev *et al.* [6] below. $\bar{u}$ denotes some vector representation corresponding to vertex u in the SDP. The reader may refer to SDP solutions φ occurring in the rest of this section to mean feasible solutions to the following SDP. Note that solving the SSE problem gives a (roughly) balanced sparse cut when $\rho = \Theta(1)$.

$$\min \quad \frac{1}{2}\sum_{(u,v)\in E(G)} \|\bar{u}-\bar{v}\|^2$$

subject to

$$\text{for all } u\in V, \quad \sum_v \langle \bar{u},\bar{v}\rangle \leqslant \rho|V| \qquad \text{(Spreading Constraints)}$$

$$\text{for all } u,v,w\in V, \quad \|\bar{u}-\bar{v}\|^2+\|\bar{v}-\bar{w}\|^2 \geqslant \|\bar{u}-\bar{v}\|^2 \qquad (\ell_2^2\text{-triangle inequalities})$$

$$\text{for all } u,v\in V, \quad \langle \bar{u},\bar{v}\rangle \geqslant 0$$

$$\text{for all } u\in V, \quad \|\bar{u}\|^2 = 1$$

Definition B.1 (Heavy Set, $H_{\delta,\varphi}(M)$ [6])**.** *Let V be a set of n vertices and $M\subseteq N$. Consider an SDP solution $\varphi: V\to\mathcal{H}$. We say that a vertex $u\in M$ is δ-heavy in M if the ℓ_2^2-ball of radius δ around $\varphi(u)$ contains at least $\delta^2 n$ vectors from $\varphi(M)$, i.e., $|v\in M \mid \varphi(v)\in \mathrm{Ball}(\varphi(u),\delta)| \geqslant \delta^2 n$. We denote the set of all vertices that are δ-heavy in M by $H_{\delta,\varphi}(M)$.*

Definition B.2 (Geometric Expansion [6])**.** *A graph $G=(V,E)$ satisfies the geometric expansion property with cut value X at scale δ, if for every SDP solution $\varphi: V\to\mathcal{H}$ satisfying $H_{\delta,\varphi}(V)=\varnothing$ (recall that $H_{\delta,\varphi}(V)$ is the set of δ-heavy vertices in V):*

$$\left|\{(u,v)\in E \mid \|\varphi(u)-\varphi(v)\|^2 \leqslant \delta/2\}\right| \leqslant 2\delta^2 X.$$

A graph $G=(V,E)$ satisfies the geometric expansion property with cut value X up to scale 2^{-T} for $T\in\mathbb{N}$ if it satisfies the geometric expansion property for every $\delta\in\{2^{-t} \mid 1\leqslant t\leqslant T\}$.

Theorem B.3 (Theorem 3.4 from [6])**.** *Let $G=(V,E)$ be a graph that satisfies the geometric expansion property with cut value X at scale up to $c\sqrt{\log|V|}$. Then, there exists a randomized polynomial time that with high probability outputs a partition L,R of V such that $|\mathrm{cut}(L,R)|=O(X)$ and $|L|,|R|\geqslant |V|/3$.*

B.2 Geometric Expansion of HSBM

Let $\widetilde{G}_n = (\widetilde{V}_n, \widetilde{E}_n, w)$ be a graph generated according to an ultrametric, where for each $e\in\widetilde{E}_n$, $w(e)\in(0,1)$. In this case, we allow $w(e)$ to be depend on n—in particular it is possible that $w(e)\to 0$ as $n\to\infty$. Let $G=(V,E)$ be an unweighted random graph with $|V|=|\widetilde{V}_n|=n$ generated from $\widetilde{G}$ as follows. An edge (u,v) is added to G with probability $w((u,v))$, for the corresponding vertices $u,v\in\widetilde{V}_n$. Note that this is a special case of the hierarchical stochastic blockmodels (Defn. 2.1), where the number of leaves in the generating graph is the same as that in the random graph generated, *i.e.,* in principle there may be n bottom-level clusters.

For the rest of this discussion we assume that $V=\widetilde{V}_n$ as a natural bijection exists between the two vertex sets. Let $\widetilde{T}$ be a generating tree for $\widetilde{G}$. Let $U\subseteq V$ and let $\widetilde{T}|_U$ be the restriction of $\widetilde{T}$ to leaves in U (removing unnecessary leaves and reducing internal nodes as necessary). Let $N(U)$ be the root of $\widetilde{T}|_U$. Consider the following procedure the nodes appearing as leaves in the left and right subtrees of the root of $\widetilde{T}|_U$. Suppose we follow the convention that the left subtree is never any smaller than the right subtree in $\widetilde{T}|_U$. We say that the canonical node of $\widetilde{T}|_U$ is the first left node N_L encountered in a top-down traversal starting from $N(U)$ such that $2|U|/3\geqslant V(N_L)\geqslant |N|/3$. We define $U_L=V(N_L)$, and $U_R=U\backslash U_L$. We say that (U_L,U_R) is the *canonical cut* of U. It is easy to see that such a cut always exists since the tree is binary and left subtrees are never smaller than right subtrees.

Lemma B.4. *For a random graph G generated according the model described above with probability at least $1-o(1)$, for every subset U of size at least $n^{2/3}\sqrt{\log n}$, let U_L,U_R be the canonical cut of U and let $E_{rnd}=\{(u,v) \mid u\in U_L, v\in U_R\}$. Then the subgraph (U,E_{rnd}) is geometrically expanding with cut cost*

$$X = C\cdot\max\{w(L,R), |U|\cdot D\cdot\log^2 D, |U|\cdot D\cdot\log n\} \tag{3}$$

up to scale $1/\sqrt{D}$. Furthermore, the result also applies in the semi-random setting where an adversary may remove any subset of edges from the random graph G.

Proof. The proof is essentially identical to that of Theorem 5.1 in [6]. However, as there are some minor modifications, we are unable to cite their result directly and hence provide the entire proof here for completeness.

Fix some subset U and let U_L,U_R be the canonical cut of U given by the generating tree $\widetilde{T}$ of $\widetilde{G}$. Let $E_{\text{all}} = \{(u,v) \mid u \in U_L, v \in U_R\}$ and let $E_{\text{rnd}} = E_{\text{all}} \cap E$, where E is the set of edges in the realized random graph $G = (V,E)$. As the adversary in the semi-random graph can only remove edges it suffice to show that (U,E_{rnd}) is geometrically expanding with high probability. We fix the parameter $\delta = 2^{-t}$ (where $1 \leqslant t \leqslant T$), and prove that the graph (U,E_{rnd}) is geometrically expanding with cut value X at scale δ. The probability that this fails to happen will be low enough for us to take a simple union bound over all the posible values of δ.

The condition $H_{\delta,\varphi}(U) = \varnothing$ implies that,

$$|\{(u,v) \in U \times U \mid \|\varphi(u) - \varphi(v)\|^2 \delta\}| \leqslant \delta^2 n^2$$

We need to bound the probability of the bad event, the existence of an SDP solution $\varphi : U \to \mathcal{H}$ such that

$$\left|\left\{(u,v) \in U \times U \mid \|\varphi(u) - \varphi(v)\|^2 \leqslant \delta\right\}\right| \leqslant \delta^2 n^2 \tag{4}$$

$$\left|\left\{(u,v) \in E_{\text{rnd}} \mid \|\varphi(u) - \varphi(v)\|^2 \leqslant \frac{\delta}{2}\right\}\right| \geqslant 2\delta^2 X \tag{5}$$

Makarychev *et al.* [6] show that if φ satisfying Eqn. (4) and (5) exists, then provided $|E_{\text{rnd}}| \leqslant 2X$, there exists $\varphi' : U \to N_\delta$ satisfying:

$$\left|\left\{(u,v) \in U \times U \mid \|\varphi'(u) - \varphi'(v)\|^2 \leqslant \frac{3}{4}\delta\right\}\right| \leqslant \frac{5}{4}\delta^2 n^2 \tag{6}$$

$$\left|\left\{(u,v) \in E_{\text{rnd}} \mid \|\varphi'(u) - \varphi'(v)\|^2 \leqslant \frac{3}{4}\delta\right\}\right| \geqslant \frac{3}{4}\delta^2 X \tag{7}$$

where $N_\delta \subset \mathcal{H}$ is a set of size $\exp(O(\log^2 \delta^{-1}))$.

The remainder of the proof is showing that the existence of φ' is a very low-probability event. First, as $|E_{\text{rnd}}| = w(U_L,U_R) \leqslant X$, $\mathbb{P}[|E_{\text{rnd}}| \geqslant 2X] \leqslant e^{-c_0 X}$. Note that if we fix a $\varphi' : U \to N_\delta$, the probability (over the random choice of E_{rnd}) that Eqns. (6) and (7) is at most $e^{-c_1 \delta^2 X}$, by using the Chernoff bound. Finally, we note that there are at most $|N_\delta|^{|U|}$ such φ', thus we can safely take a union bound provided $X/D \geqslant c_3 \cdot |U| \log^2 D$. Finally, there are $n^{|U|}$ subsets of size $|U|$ and again we can safely apply a union bound provided $X/d \geqslant c_4 |U| \log n$. The choice of X ensures that this happens. □

We can now complete the proof of Theorem 2.6.

Proof of Theorem 2.6. We aim at applying Theorem 4.1 of Cohen-Addad *et al.* [4], where they essentially show that if one obtains a ϕ approximation to the $1/3$-balanced min-cut problem (*i.e.,* minimise cut subject to the constraint that both sides have at least $1/3$ of the vertices being cut), then the recursive algorithm gives a $O(\phi)$ approximation for minimizing Dasgupta's cost function.

We observe that $\text{cost}(\widetilde{T};\widetilde{G}) = \Omega(n^3 \cdot p_{\min}) = \Omega(n^{7/3} \log n)$. Thus, we notice that once we obtain sets U of size $n_0 = n^{2/3}\sqrt{\log n}$, since there are at most n/n_0, even if we use an arbitrary tree on any such U, together this can only add $O(\frac{n}{n_0} \cdot n_0^3) = O(n^{7/3} \cdot \log n)$. Thus, we only need to be able to obtain suitable approximations during the recursive procedure as long as $|U| \geqslant n^{2/3} \log n$. This is precisely given by using Lemma B.4. Observe that in Eq. (3), $w(L,R) = \Omega(|U|^2 \cdot p_m in) = \Omega(n^{2/3} \log^2 n)$ and $|U| D \log^2 D = o(|U| D \log n)$ and $D|U| \log n) = O(n^{2/3} \log^2 n)$. Thus, the algorithm of [6] given by Theorem B.3 returns a cut that is a constant factor approximation to the $1/3$-balanced min-cut problem on the induced subgraph of $\widetilde{G}$ on the vertex set U. This observation together with a slight modification of the charging argument in the proof of Theorem 4.1 of Cohen-Addad *et al.* [4] to account for the case where subgraphs have size less than $n^{2/3} \log n$ finishes the proof. □

C LINKAGE++ (Continuation of Section 2)

The full algorithm is the following.

We show the proof of Lemmas 2.4, 2.5 and C.1.

Algorithm 1 LINKAGE++

1: **Input:** Graph $G=(V,E)$ generated from an HSBM.
2: **Parameter:** An integer k.
3: Apply (SVD) projection algorithm of [5, Thm. 12] with parameters G, k, $\delta=|V|^{-2}$, to get $\zeta(1),...,\zeta(|V|)\in\mathbb{R}^{|V|}$ for vertices in V, where $\dim(\mathrm{span}(\zeta(1),...,\zeta(|V|)))=k$.
4: Run the single-linkage algorithm on the points $\{\zeta(1),...,\zeta(|V|)\}$ until there are exactly k clusters. Let $\mathcal{C}=\{C_1^\zeta,...,C_k^\zeta\}$ be the clusters (of points $\zeta(i)$) obtained. Let $C_i\subseteq V$ denote the set of vertices corresponding to the cluster C_i^ζ.
5: Define $\mathrm{dist}:\mathcal{C}\times\mathcal{C}\mapsto\mathbb{R}_+$: $\mathrm{dist}(C_i^\zeta,C_j^\zeta)=w(C_i^\zeta,C_j^\zeta)/(|C_i^\zeta||C_j^\zeta|)$.
6: **while** there are at least two clusters in $\mathcal{C}$ **do**
7: Take the pair of clusters C_i',C_j' of $\mathcal{C}$ that maximizes $\mathrm{dist}(C_i',C_j')$.
8: Define a new cluster $C'=\{C_i'\cup C_j'\}$.
9: Update dist: $\mathrm{dist}(C',C_\ell')=\max(\mathrm{dist}(C_i',C_\ell'),\mathrm{dist}(C_j',C_\ell'))$
10: $\mathcal{C}\leftarrow\mathcal{C}\setminus\{C_i\}\setminus\{C_j\}\cup\{C'\}$
11: **end while**
12: The sequence of merges in the while-loop (Steps 6 to 11) induces a hierarchical clustering tree on $\{C_1,...,C_k\}$, say T_k' with k leaves (represented by $C_1,...,C_k$). Replace each leaf of T_k' by an arbitrary binary tree on $|C_k|$ leaves labelled according to the vertices C_k to obtain T.
13: Repeat the algorithm $k'=2k\log n$ times. Let $T^1,...T^{k'}$ be the corresponding hierarchical clustering trees.
14: **Output:** Tree T^i (out of the k' candidates) that minimises $\Gamma(T_i)$.

Proof of Lemma 2.5. By Theorem 3.4 of Cohen-Addad et al. [4], the cost of T^* is optimal for $\bar{G}$. Furthermore, by Theorem C.3, we have that the cost of T^* on $\bar{G}$ and the cost of T^* are within a factor of $(1+o(1))$. We thus aim at showing that $\mathrm{cost}_T\leqslant\gamma^2\mathrm{cost}_{T^*}$.

We now define a new graph $\bar{G}(T)$ which has the same set of vertices V than $\bar{G}$. For each pair of vertices $u,v\in V$ we create an edge in $\bar{G}(T)$ with weight $w_{\bar{G}(T)}(u,v)=W'(\mathrm{LCA}_T(u,v))$. By definition of W' follows that T is generating for $\bar{G}(T)$, and so applying Theorem 3.4 of Cohen-Addad et al. [4], we obtain that the cost of T for $\bar{G}(T)$, say $\mathrm{cost}_T^{\bar{G}(T)}$, is less than the cost of T^* for $\bar{G}(T)$, say $\mathrm{cost}_{T^*}^{\bar{G}(T)}$.

Now recall that the cost of a tree for any given graph G' can be rewritten as follows: $\mathrm{cost}_T^{\bar{G}(T)}=\sum_{u,v}w_{G'}(u,v)\cdot|V(\mathrm{LCA}_T(u,v))|$, where $w_{G'}(u,v)$ is the weight of the edge u,v in G'.

Thus, since by definition of T, we have $\gamma^{-1}W'(\mathrm{LCA}_T(a,b))\leqslant W(\mathrm{LCA}_{T^*}(a,b))\leqslant\gamma W'(\mathrm{LCA}_T(a,b))$. Hence,

$$\begin{aligned}\mathrm{cost}_T^{\bar{G}}&\leqslant\sum_{u,v}w_{\bar{G}}(u,v)\cdot|V(\mathrm{LCA}_T(u,v))|\leqslant\sum_{u,v}\gamma\cdot w_{\bar{G}(T)}(u,v)\cdot|V(\mathrm{LCA}_T(u,v))|\\&=\gamma\cdot\mathrm{cost}_T^{\bar{G}(T)}\leqslant\gamma\cdot\mathrm{cost}_{T^*}^{\bar{G}(T)}.\end{aligned}$$

A similar observation implies that $\mathrm{cost}_{T^*}^{\bar{G}(T)}\leqslant\gamma\cdot\mathrm{cost}_{T^*}^{\bar{G}}$. Combining yields the lemma.

□

Recall that $\psi:V\to[k]$ is the (hidden) labelling assigning each vertex of G to one of the k bottom-level clusters. Let $C_i^*=\{v\in V\mid\psi(v)=i\}$. Recall that $n_i=|V(C_i^*)|$.

Lemma C.1. *Let G be generated by an* HSBM. *Assume that the separation of bottom clusters given by* (1) *holds. Let $C_1^*,...,C_k^*$ be the hidden bottom-level clusters, i.e., $C_i^*=\{v\mid\psi(v)=i\}$. With probability at least $\Omega(1/k)$, the clusters obtained after Step 4 correspond to the assignment ψ, i.e., there exists a permutation $\pi:[k]\to[k]$, such that $C_j=C_{\pi(j)}^*$.*

Proof. The proof relies on Theorem 2.2. Let u,v be two nodes such that $i=\psi(u)\neq\psi(v)=j$. Let $\mathbf{u}$ and $\mathbf{v}$ denote the random variables corresponding to the columns of u and v in the adjacency matrix of G. Let $q=\widetilde{W}(N)$ where N is the $\mathrm{LCA}_{\widetilde{T}_k}(\widetilde{\sigma}^{-1}(i),\widetilde{\sigma}^{-1}(j))$ in $\widetilde{T}_k$, the generating tree for $\widetilde{G}_k$ used in defining the HSBM.

By assumption, if have that (1) holds. for $\delta = \frac{1}{n^2}$. This satisfies the condition of Theorem 2.2. Thus, with probability at least $1/k - \delta = \Omega(1/k)$ the conclusions of Theorem 2.2 hold. In the rest of the proof we assume that the following holds: There exists $\eta > 0$ such that for any pair of nodes u,v we have

1. if $\psi(u) = \psi(v)$ then $\|\zeta(u) - \zeta(v)\|_2^2 \leqslant \eta$;
2. if $\psi(u) \neq \psi(v)$ then $\|\zeta(u) - \zeta(v)\|_2^2 > 2\eta$.

Therefore, any linkage algorithm, *e.g.,* single linkage, performing merges starting from the set $\{\zeta(1),...,\zeta(n)\}$ until there are k clusters will merge clusters at a distance of at most η and hence, the clusters obtained after Step 4 correspond to the assignment ψ. This yields the claim. □

We now condition on Event $\mathcal{E}_1$. Thus, by the above lemma, there exists a 1-to-1 mapping $\pi : \{C_1,...,C_k\} \mapsto \{C_1^*,...,C_k^*\}$ such that $C_i = C^*_{\pi(i)}$. We define $\mathcal{C}^t$ the set of clusters in the variable $\mathcal{C}$ of Alg. 1 at the tth iteration of the while-loop. Finally, observe that since Event $\mathcal{E}_1$ holds, we have that at any iteration t, any cluster $C_i^t \in \mathcal{C}^t$ consists of a union of clusters of $C^* = \{C_1^*,...,C_k^*\}$. Let $L(C_i^t)$ denote the set of clusters of C^* in C_i^t.

Proof of Lemma 2.4. Given a set of hidden bottom clusters S, denote by $\text{LCA}_{T^*}(S)$ the lowest common ancestor of all the clusters in S.

We show by induction on the number of iterations of the while-loop (Step 4) that, with high probability, if $C_i^t, C_j^t \in \mathcal{C}^t$ are merged at iteration t, then for any $C_\ell^* \in L(C_i^t), C_s^* \in L(C_j^t)$,

$$W(\text{LCA}_{T^*}(L(C_i^t) \cup L(C_j^t))) \leqslant W(\text{LCA}_{T^*}(C_\ell^*, C_s^*)) \leqslant (1+\varepsilon) W(\text{LCA}_{T^*}(L(C_i^t) \cup L(C_j^t))),$$

and

$$W(\text{LCA}_{T^*}(L(C_i^t) \cup L(C_j^t))) \leqslant \text{dist}(C_i^t, C_j^t) \leqslant (1+\varepsilon) W(\text{LCA}_{T^*}(L(C_i^t) \cup L(C_j^t))).$$

We note that since the distances "dist" between merged clusters are non-increasing as t is increasing, the above claim implies that T is a $(1+\varepsilon)$-approximate ground-truth tree and so the lemma.

Observe that since Event $\mathcal{E}_2$ occurs, we have that for an $C_\ell^* \in L(C_i^t), C_s^* \in L(C_j^t)$,

$$(1-\varepsilon^2) W(\text{LCA}_{T^*}(C_\ell^*, C_s^*)) \leqslant w(C_\ell^*, C_s^*)/(|C_\ell^*| \cdot |C_s^*|) \leqslant (1+\varepsilon^2) W(\text{LCA}_{T^*}(C_\ell^*, C_s^*))$$

Equipped with this observation, we turn to the proof of the claim. We proceed by induction on the number of merges. At $t = 0$, the claim is true as no merge has been done yet. We now consider the tth merge done by the algorithm. Let R_j be the lowest common ancestor of the nodes in $L(C_j)$ in T^* and let R_i be the lowest common ancestor of the nodes in $L(C_i)$ in T^*. Let R^* be the lowest common ancestor of R_j and R_i.

We differentiate two cases: (1) either $R^* \in \{R_i, R_j\}$ and $R_i \neq R_j$ or (2) either $R^* \notin \{R_i, R_j\}$ or $R_i = R^* = R_j$. First, Assume (2) *i.e.,* R^* is not in $\{R_i, R_j\}$ or that $R_i = R_j = R^*$. Thus by definition of the $\widetilde{T}_k$ we have that for any $C_\ell^* \in L(C_i^t), C_s^* \in L(C_j^t)$, $\text{LCA}_{T^*}(C_\ell^*, C_s^*) = R^*$ and so $W(\text{LCA}_{T^*}(C_\ell^*, C_s^*)) = W(R^*)$. Since Event $\mathcal{E}_2$ holds, the claim is true in that case.

We thus turn to case (1): R^* is either R_i or R_j and $R_i \neq R_j$, and w.l.o.g., we assume that $R^* = R_i \neq R_j$. From this we can provide a lower bound on the edge density between the clusters of C_j^t and C_i^t. Indeed, since Event $\mathcal{E}_2$ occurs, we have that the edge density is at least $(1-\varepsilon^2) W(R_i)$. We now provide an upper bound. Consider $C_k^* \in L(C_i^t)$ and $C_s^* \in L(C_j^t)$ such that $W(\text{LCA}_{T^*}(C_k^*, C_s^*))$ is maximized. By definition of the algorithm we have that

$$(1-\varepsilon^2) W(\text{LCA}_{T^*}(C_k^*, C_s^*)) \leqslant \text{dist}(C_i^t, C_j^t) \leqslant (1+\varepsilon^2) W(\text{LCA}_{T^*}(C_k^*, C_s^*)).$$

Now, by definition of T^* and since $R_i = R^*$, we have that $W(R_i) = W(R^*) \leqslant W(\text{LCA}_{T^*}(C_k^*, C_s^*))$. Furthermore, since clusters with lowest common ancestor R_i were merged before C_j^t and C_i^t and since distances between merged clusters are non-increasing as the number of iterations is increasing, we can conclude:

$$(1-\varepsilon^2) W(\text{LCA}_{T^*}(C_k^*, C_s^*)) \leqslant \text{dist}(C_i^t, C_j^t) \leqslant (1+\varepsilon^2) W(R_i).$$

Assuming ε to be a small enough constant, and recalling that R_i is the lowest common ancestor of all the clusters in $C_i^t \cup C_j^t$, we conclude that for any $C_\ell^* \in C_i^t, C_s^* \in C_j^t$,

$$W(R_i) \leqslant W(\text{LCA}_{T^*}(C_\ell^*, C_s^*)) \leqslant (1+\varepsilon) W(R_i),$$

and

$$W(R_i) \leqslant \text{dist}(C_i^t, C_j^t) \leqslant (1+\varepsilon) W(R_i),$$

and the lemma follows. □

C.1 Objective Functions and Ground-Truth Tree

The proof of these sections are not technically involved and follows the proofs proposed by Cohen-Addad et al. [4] for their less general model. They are included here for completeness. We start with a few observations. Note that $\bar{G}$ itself is generated from an ultrametric and the generating trees for $\bar{G}$ are obtained as follows: Let $\widetilde{T}_k$ be any generating tree for $\widetilde{G}_k$, let $\hat{T}_1, \hat{T}_2, \ldots, \hat{T}_k$ be any binary trees with $n_1, \ldots, n_k$ leaves respectively. Let the weight of every internal node of $\hat{T}_i$ be p_i and replace each leaf l in $\widetilde{T}_k$ by $\hat{T}_{\tilde{\sigma}(l)}$. In particular, this last point allows us to derive Proposition C.2. We will use $\kappa(n)$ to denote the cost of a clique which is the same for all trees for the clique, as argued in [4, Thm. 3.4].

Proposition C.2 (Slight generalization of [4])**.** *Let G be a graph generated according to an* HSBM *(See Defn. 2.1). Let ψ be the (hidden) function mapping the nodes of G to $[k]$ (the bottom-level clusters). Let T be a ground-truth tree for G Then,*

$$\mathbb{E}[\mathit{cost}(T)] \leqslant \min_{T'} \mathbb{E}\big[\mathit{cost}(T')\big].$$

Moreover, for any tree T', $\mathbb{E}[\mathit{cost}(T)] = \mathbb{E}[\mathit{cost}(T')]$ if and only if T' is a ground-truth tree.

Proof. Recall that we have $n_i > 0$ for all i. Let $\bar{G}$ be a the expected graph, *i.e.,* $\bar{G}$ is complete and an edge (i, j) has weight p_{ij}, the probability that the edge (i, j) is present in the random graph G generated according to the hierarchical model. Thus, by definition of admissibility $\text{cost}(T;\bar{G}) = \min_{T'} \text{cost}(T';\bar{G})$ if an only if T is generating (see [4, Defn. 3.1]). As ground-truth trees for G are precisely the generating trees for $\bar{G}$; the result follows by observing that for any tree T (not necessarily ground-truth) $\mathbb{E}[\text{cost}(T;G)] = \text{cost}(T;\bar{G})$, where the expectation is taken only over the random choice of the edges, by linearity of expectation and the definition of the cost function. □

The following is a generalisation of [4].

Theorem C.3. *Let n be a positive integer and $p_{\min} = \omega(\sqrt{\log n / n})$. Let k be a fixed constant and G be a graph generated from an* HSBM *(as per Defn. 2.1) where the underlying graph $\widetilde{G}_k$ has k nodes and minimum probability is $p_{\min}$. For any binary tree T with n leaves labelled by the vertices of G, the following holds with high probability:*

$$|\mathit{cost}(T) - \mathbb{E}[\mathit{cost}(T)]| \leqslant o(\mathbb{E}[\mathit{cost}(T)]).$$

The expectation is taken only over the random choice of edges. In particular if T^* is a ground-truth tree for G, then, with high probability,

$$\mathit{cost}(T^*) \leqslant (1+o(1)) \min_{T'} \mathit{cost}(T') = (1+o(1))\mathit{OPT}.$$

Proof. Our goal is to show that for any fixed cluster tree T' the cost is sharply concentrated around its expectation with an extremely high probability. We then apply the union bound over all possible cluster trees and obtain that in particular the cost of OPT is sharply concentrated around its expectation. Note that there are at most $2^{c \cdot n \log n}$ possible cluster trees (including labellings of the leaves to vertices of G), where c is a suitably large constant. Thus, it suffices to show that for any cluster tree T' we have

$$\mathbb{P}\big[\big|\text{cost}(T') - \mathbb{E}\big[\text{cost}(T')\big]\big| \geqslant o(\mathbb{E}\big[\text{cost}(T')\big])\big] \leqslant \exp(-c^* n \log n),$$

where $c^* > c$.

Recall that for a given node N of T' with children N_1, N_2, we have $\text{cost}(N) = w(V(N_1), V(N_2)) \cdot (|V(N_1)| + |V(N_2)|)$ and $\text{cost}(T') = \sum_{N \in T'} \text{cost}(N)$. Let $Y_{i,j} = \mathbf{1}_{(i,j) \in E}$ for all $1 \leqslant i, j \leqslant n$ and observe that $\{Y_{i,j} | i < j\}$ are independent and $Y_{i,j} = Y_{j,i}$. Furthermore, let

$Z_{i,j} = (|V(\text{child}_1(N^{i,j}))| + |V(\text{child}_2(N^{i,j}))|) \cdot Y_{i,j}$, where $N^{i,j}$ is the node in T' separating nodes i and j and $\text{child}_1(N^{i,j})$ and $\text{child}_2(N^{i,j})$ are the two children of $N^{i,j}$. We can thus write

$$\text{cost}(T') = \sum_{N \in T'} (|V(\text{child}_1(N))| + |V(\text{child}_2(N))|) \sum_{\substack{i \in V(\text{child}_1(N)) \\ j \in V(\text{child}_2(N))}} Y_{i,j} \tag{8}$$

$$= \sum_{N \in T'} \sum_{\substack{i \in V(\text{child}_1(N)) \\ j \in V(\text{child}_2(N))}} Z_{i,j} \tag{9}$$

$$= \sum_{i<j} Z_{i,j}, \tag{10}$$

where we used that every potential edge $i,j, i \neq j$ appears in exactly one cut and that $Z_{i,j} = Z_{j,i}$. Observe that $\sum_{i<j} Z_{i,j}$ is a sum of independent random variables. Assume that the following claim holds.

Claim C.4 (Slight generalization of [4])**.** *Let $p_{\min}$ be the minimum weight in $\widetilde{T}_k$, the tree generating tree for $\widetilde{G}_k$ (see Defn. 2.1), i.e., $w_{\min} = \min_{N \in \widetilde{T}_k} \widetilde{W}(N)$). We have*

1. $\mathbb{E}[\textit{cost}(T')] \geqslant \kappa(n) \cdot p_{\min}$
2. $\sum_{i<j} (|V(\textit{child}_1(N^{i,j}))| + |V(\textit{child}_2(N^{i,j}))|)^2 \leqslant n \cdot \kappa(n)$

We defer the proof to later and first finish the proof of Theorem C.3. We will make use of the slightly generalized version of Hoeffding bounds (see [27]). For $X_1, X_2, ..., X_m$ independent random variables satisfying $a_i \leqslant X_i \leqslant b_i$ for $i \in [n]$. Let $X = \sum_{i=1}^m X_i$, then for any $t > 0$

$$\mathbb{P}[|X - \mathbb{E}[X]| \geqslant t] \leqslant \exp\left(-\frac{2t^2}{\sum_{i=1}^m (b_i - a_i)^2}\right). \tag{11}$$

By assumption, there exists a function $y_n : \mathbb{N} \to \mathbb{R}_+$ such that $p_{\min} = \omega\left(y_n \cdot \sqrt{\frac{\log n}{n}}\right)$ with $y_n = \omega(1)$.

We apply (11) with $t = \mathbb{E}[\text{cost}(T')] \cdot \frac{y_n \cdot \sqrt{\frac{\log n}{n}}}{p_{\min}} = o(\mathbb{E}[\text{cost}(T')])$ and derive

$$\mathbb{P}\left[\left|\text{cost}(T') - \mathbb{E}\left[\text{cost}(T')\right]\right| \leqslant \mathbb{E}\left[\text{cost}(T')\right] \cdot \frac{y_n \cdot \sqrt{\frac{\log n}{n}}}{p_{\min}}\right] \geqslant$$

$$\geqslant 1 - \exp\left(-\frac{2\left(\mathbb{E}[\text{cost}(T')] \cdot \frac{y_n \cdot \sqrt{\frac{\log n}{n}}}{p_{\min}}\right)^2}{\sum_{i<j} g(|V(N_1^{i,j})|, |V(N_2^{i,j})|)^2}\right)$$

$$\geqslant 1 - \exp\left(-\frac{2 \cdot \kappa(n) \cdot y_n^2 \cdot \log n}{n^2}\right) \geqslant 1 - \exp(-c^* \cdot n \log n),$$

where the last inequality follows by assumption of the lemma and since $y_n = \omega(1)$ and $\kappa(n) = \Theta(n^3)$. □

We now turn to the proof of Claim C.4.

Proof of Claim C.4. Note that for any two vertices i,j of G, the edge (i,j) exists in G with probability at least $p_{\min}$. Thus, we have

$$\mathbb{E}\left[\text{cost}(T')\right] = \sum_{N \in T'} (|V(\text{child}_1(N))| + |V(\text{child}_2(N))|) \sum_{\substack{i \in V(\text{child}_1(N)) \\ j \in V(\text{child}_2(N))}} w(i,j)$$

$$\geqslant p_{\min} \cdot \sum_{N \in T'} (|V(\text{child}_1(N))| + |V(\text{child}_2(N))|) |V(\text{child}_1(N))| \cdot |V(\text{child}_2(N))|$$

$$= p_{\min} \cdot \kappa(n).$$

Furthermore, we have

$$\begin{aligned}
&\sum_{i<j}(|V(\text{child}_1(N^{i,j}))|+|V(\text{child}_2(N^{i,j}))|)^2 = \\
&\quad = \sum_{N\in T'}(|V(\text{child}_1(N))|+|V(\text{child}_2(N))|)^2\cdot|V(\text{child}_1(N))|\cdot|V(\text{child}_2(N)))| \\
&\quad \leqslant n\sum_{N\in T'}(|V(\text{child}_1(N))|+|V(\text{child}_1(N))|)\cdot|V(\text{child}_1(N))|\cdot|V(\text{child}_2(N)))| \\
&\quad = n\cdot\kappa(n).
\end{aligned}$$

□