We thank the reviewers for the time they spent evaluating our manuscript and for their valuable comments. All the answers given in this document will be incorporated in the manuscript and/or in its appendix. In the following we took the liberty to group and reword some of the reviewers comment (in blue italic) to save space.

General answer on the usefulness of gradient descent, its theoretical guarantees, and its scalability.

The proposed gradient-based approach has two important advantages: (i) it is extremely flexible, allowing us to easily incorporate new cost terms; (ii) it provides a generic "ultrametric layer" that can be used to build end-to-end learning pipelines, for example with methods for graph embedding in hyperbolic spaces (Nickel and Kiela, NeurIPS 2017).

We agree that having theoretical guarantees would be a big plus. However, we know, in particular with recent advances on deep neural networks, that obtaining any insight toward theoretical guarantees with gradient descent algorithms applied to nonconvex problems is extremely challenging, and we hope to be able to get theoretical results in future works. In this regard, apart from what mentioned in the manuscript, an interesting line of research arises from recent works on the implicit bias of gradient descent by Soudry *et al.* (ICLR 2018) and Ji *et al.* (COLT 2019).

As for scalability, the bottleneck of our method is the single-linkage algorithm. One way to address this issue (up to a certain limit) would be to use recent progresses on parallel single-linkage algorithms (out-of-core algorithms can process billions of vertices). Nonetheless, true scalability will require to adapt mini-batch gradient descent to ultrametric fitting, which is a topic that we are currently working on. Similarly to Monath *et al.* (NeurIPS 2017), our idea consists of only updating sub-trees at each iteration. However, this leads to a biased approximation of the gradient, that we plan to correct by maintaining an exponentially-weighted average of the Jacobian matrix, like in Pfau *et al.* (ICLR 2019). Given the significant body of additional material, we feel that this topic is best left to a future publication.

R1: *Experiments on larger datasets for comparison of hierarchical clustering quality.* As scalable solutions for dealing with massive datasets require nontrivial extensions, we believe that they deserve a separate publication.

R2: *Line 8,56,70,93: I would suggest a more cautious usage of the word "equivalent".* Your are right that the transformed problem is not strictly equivalent to the original one, we will thus remove the world "equivalent". Moreover, a proof of equivalence between ultrametric and hierarchical clustering can be found in [29, Th. 9].

R2: *A differentiation between the ultrametric d and the ultrametric u would make their different usages clearer.* We will improve the beginning of Section 2 and its illustration in Figure 1 in order to better explain the difference and the importance of modeling the ultrametric d_u by its compact "restriction" u defined only on the edges of the graph.

R2: *Line 186 ff.: A usage of the subdominant ultrametric for the cluster-size regularization, would make the algorithms part more consistent with the following considerations in this paper,* ***and*** *One more sentence to the relationship between (13) and (17) would make the transition with the use of the subdominant ultrametric compared to before clearer.* We will emphasize how Eq (14) can be used to rewrite cluster size regularization and Dasgupta's cost to suit the proposed optimization framework. In particular, we will give the full derivation of Eq (13) to Eq (17) in the appendix.

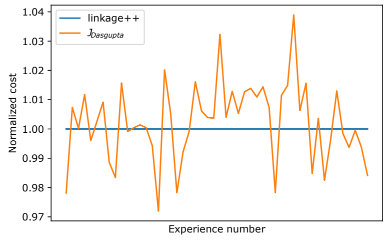
R2: *How is the weighting λ of the regularization term chosen? Were also other values used?* We tried several values on an exponential scale and we observed that the constant value reported in the article was sufficient for all the experiments.

R2: *Line 149: $J_{Dasgupta}$ calculated with the Heaviside function H or the sigmoid function?* ***and*** *Equation (17): what is ℓ?* A continuous approximation of the Heaviside function is mandatory for Dasgupta's cost, as the cluster sizes is the only element of the cost that depends of the ultrametric u. With the Heaviside function, the derivative of Dasgupta's cost will be null almost everywhere. The undefined symbol ℓ in (17) was indeed meant to refer to the sigmoid function.

R2: *Notations and clarifications. 1) Line 128: labels C_v, why C is used? C already used for cycles in G. 2) Algorithm 2 Title: defined in (14) instead of (5)?, 3) Line 144: $x \in n$. x is a node and n also? 4) Line 149: Was the Dasgupta cost function for the visualization in Figure 2 also combined with a regularization as in the experimental part?* 1) We will replace C_v with $\mathcal{L}_v$ (label of v) to avoid any confusion. 2) Yes, the algorithm computes the min-max operator defined in (5) with the formulation given in (14). 3) You are right, nodes are subsets of V, so $x \in V$ is an element of a node n, and the singleton $\{x\}$ is a leaf node. 4) No, Dasgupta's cost was used without regularization in all the toy examples.

R3: *I think that it would be good to more clearly state the process (and its complexity) of going from the ultrametric fit to data to a dendrogram.* ***and*** *Clarifications of how a hierarchical clustering is extracted from a fit ultrametric.* The dendrogram associated to a given ultrametric is a byproduct of Algorithm 2 (tree computed on line 2). The final dendrogram is thus obtained for free at the end of Algorithm 1, when computing the min-max operator on the estimated edge weights. Thresholding the dendogram node altitudes yields a flat clustering. In practice, one can efficiently find the threshold that would produce a given number of clusters by browsing the dendrogram from the root to the leaves.

R3: *Experiments that compare the optimization of hierarchical clustering methods wrt Dasgupta's cost.* The figure beside compares Dasgupta's cost obtained with Linkage++ [11] and the proposed relaxed cost function $J_{Dasgupta}$ on 50 random graphs with cosine similarities (blobs-like structure with 1024 points organized in 16 clusters). These preliminary results shows that the proposed relaxation is likely close to the exact Dasgupta's cost. Note that this is consistent with Fig 1(a) in [11]. We plan to further explore this question with real data in the final version.


linkage++
$J_{Dasgupta}$
Normalized cost
Experience number