We sincerely thank the reviewers for their time, feedback, and thoughtful suggestions. The reviewers appreciate the technical novelty in our approach and its theoretical guarantees, and find our work relevant to the NeurIPS community. Reviewer 2 (R2) mainly asked us to add more baselines and experiments, while Reviewer 3 (R3) and Reviewer 4 (R4) asked us to improve our presentation. We respond in more detail below, and took all comments into account in our revised version.

Evaluation (R2, R4) R2 and R4 suggested to include other Hierarchical Clustering (HC) evaluation metrics. We would like to first clarify the claims and evaluation of our work. Our primary contribution is the development of a novel technical approach to optimize over discrete trees, by showing an equivalence between trees and constrained hyperbolic embeddings. Our approach can be used to apply machine learning techniques toward solving any combinatorial search problem involving trees. Thus, our goal is not to show an advantage on different heuristics, but rather to optimize a single well-defined search problem to the best of our abilities. In the context of HC, we focus on Dasgupta's cost (DC), which is a well-studied objective with known guarantees when there is an underlying ground-truth hierarchy [16] (such results have not been established for metrics like dendrogram purity (DP) [24]). We made this clear in our updated draft and included DP scores in the Appendix for completeness, observing that this metric does not correlate well with DC on some datasets.

Approximation Ratio (R3) R3's main concerns are two clarifications about our approximation results. The first asks if the approximation result (Thm 4.1) only holds for the optimal embedding. The answer is no: we prove the stronger result (Lemma C.1) which gives the $(1+\epsilon)$ approximation result when decoding *any* spread embedding, demonstrating the generality of this novel technique. R3's second concern is about the fact that we don't provide an approximation for the continuous optimum. It is known that no constant factor approximation is possible; otherwise, our decoding result would refute known hardness results (under Small-Set Expansion, see [11]). Achieving better (e.g. polylogarithmic) approximations, is an interesting future direction for this work, but currently out of reach, as optimizing non-convex objectives is a recognized difficult problem in the community. We thank R3 for raising these questions and have clarified these in our updated draft.

HC Baselines (R2) We thank R2 for the suggestions to improve our experiments. First, we clarify that the application considered in this work is standard *similarity*-based HC, where the input is only pairwise similarities, rather than features representing the datapoints. This is a well-established setting for analyzing the theoretical guarantees of HC algorithms [16, 33], and also the setting studied by Dasgupta [19]. This setting rules out the Hierarchical K-Means (HKM) (Q2) and decoding (Q3) baselines suggested by R2 as both methods require features as input. In our work, we only compared to methods that have access to the same input information as HYPHC (i.e. similarities), such as Bisecting K-Means, a top-down method which is the direct analog of HKM in a similarity-based context [33]. To address this ambiguity for future readers, we clearly framed the problem setup in our updated draft. To answer R2's question (Q3), we ran the HYPHC decoding without learning embeddings; as expected by R2, this method does significantly worse than HYPHC since input features are not hyperbolic (e.g. 3.411 and 3.288 DC for Zoo and Spambase respectively, versus 2.802 and 3.126 for HYPHC).

End-to-end Task (R2) R2 requested clarification about how the test data was used in the auxiliary end-to-end task. We followed a standard graph-based semi-supervised learning setting, where we have all nodes but only train labels at train time. Note that the purpose of this auxiliary experiment is a simple showcase of the benefits of joint training, which our approach make possible due to its continuous formulation; this can be easily extended to other learning scenarios (e.g. by adjusting embeddings as new examples are provided). We clarified this point in our paper. We also thank R2 for suggesting to measure the performance of a simple classifier that does not perform any clustering step (Q1). Example of such classifiers in a graph context include the Label Propagation (LP) algorithm and we added LP numbers in our updated draft. We find that LP is outperformed by our approach on all datasets (e.g. 76.7 and 46.8 accuracy for LP versus 84.4 and 50.6 for HYPHC on iris and glass respectively), suggesting that clustering learns meaningful partitions of the input similarity graph.

Presentation (R2, R3, R4) We thank the reviewers for their specific comments which helped us re-organize our paper to improve its readability. We included examples and explanations from the Appendix into the main body, such as intuition about new technical concepts (e.g. spread embeddings) (R4), details about the hyperbolic LCA computations (R2) and a more detailed related work (R4). We also clarified the mapping from points to embeddings (R3), which is an embedding lookup (line 154).

Reproducibility (R4) R4 requested more details about the optimization of hyperbolic embeddings. Thanks to the development of Riemannian optimization softwares (e.g. geoopt, 2020), the optimization of hyperbolic embeddings was straightforward, without requiring tricks such as clipping. The hyper-parameters used were learning rate, temperature, batch size, and number of triplets. We added a detailed paragraph describing our implementation and will make it publicly available.