We thank all the reviewers for your thoughtful feedback! We will incorporate our responses into the paper.

–To Reviewer #1 & #3– **Q1** & **Q7:** *To curve the embedding space is required for obtaining good graph embeddings, like kernel tricks in SVM. The proposed regularizations to reduce curvature may adversely affect on the graph embedding.*

Response: We partially agree with your opinion. Curving an embedding space is indeed required, where a useful curvature is of benefit to preserving graph topology. For instance, the green dashed circles in Fig. 1 (from Fig. 1 in paper) indicate useful curvatures that form good distributions, where the connected nodes are close and disconnected nodes are far apart. Hyperbolic space is also an example, whose useful negative curvature is suitable for embedding tree-structured graphs.

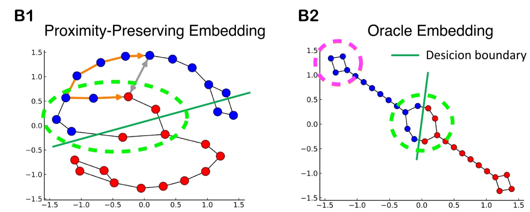

Figure 1: Useful and useless curvatures.

However, useless or harmful curvatures are also inevitable in proximity-preserving embedding because the existing strategies have no restriction on the "non-local" curves of an embedding space, such as the global "swirling" curve in Fig. 1B1. Such harmful curvatures bring undesired distortions because unsupervised graph embedding aims to faithfully preserve graph topology patterns into an embedding space. In other words, any changes/distortions in patterns is not desired, which is fundamentally different from kernel tricks in SVM, which optimize distribution pattern (by maximizing separation) *controlled by* data labels in a supervised manner.

We stress that the proposed method differentiates useful from harmful curvatures because the former will contribute to the reconstruction term in the objective function while the latter will not and thus be eliminated. In Fig. 1B2, to illustrate a desired "oracle", the global swirling curvature is reduced while the green-circle curvature still exists.

Q3: *A more thorough introduction to sectional curvature, especially using figures.*

Response: Thanks for the suggestion. As sectional curvature is defined in the space of more than 3 dimensions, it's hard to visualize in a figure. To provide similar intuition, in a new figure (Fig. 2), we will instead illustrate normal curvature, which can be considered as a degenerate case of sectional curvature in a 3-dimension space. We will add a more thorough introduction of sectional curvature in our revised version.

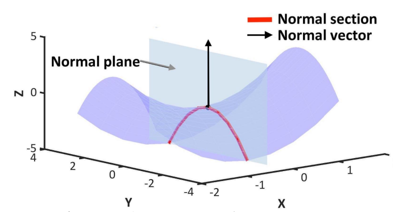

Figure 2: Normal curvature.

Q4: *Connection between the curvature regularization and betweenness centrality.*

Response: Your finding is absolutely right. The proposed regularization constrains more on nodes with high betweenness than others– which we think makes sense to prevent distortion: A node with high betweenness in a graph is corresponding to a "bottleneck" region in an embedding manifold, i.e., a small region where lots of geodesic curves pass through. Taking Fig. 1B2 as an example again, the nodes in the green dashed circle have high betweenness. If we curve or fold the embedding space around these nodes, the Euclidean distances of most node pairs will be changed, which will bring large distortions in terms of Eq. 1 (the divergence between the geodesic and Euclidean metric) in paper. To contrast, if we curve the embedding space around the pink dashed circle (low betweenness), it brings limited distortion. Thus, indeed, high-betweenness nodes should be constrained more.

Q5: *Betweenness and PageRank can be used to inform sampling for the two efficient variants of curvature regularization.*

Response: Thanks for your constructive suggestion. Indeed, our sampled regularization variant Ω_s is equivalent to the betweenness-informed strategy you suggested. It samples shortest paths between nodes, by which nodes with high betweenness naturally have a high probability to be sampled. We believe a PageRank-based method is also meaningful, which we will discuss and experiment in our final version.

Q6: *To see how much worse was the curvature loss in the different setups.*

Response: Great suggestion. We now conducted a comparison experiment to illustrate the curvature loss by using Laplacian Eigenmaps (LE) on the Cora dataset. We ran the comparison 10 times, and Fig. 3 reported the mean and variance. One can see that LE with curvature regularization got lower curvature loss (solid line) and distortion (dashed line) than the original LE after iterations. We will add more extensive/systematic comparisons to our final version.

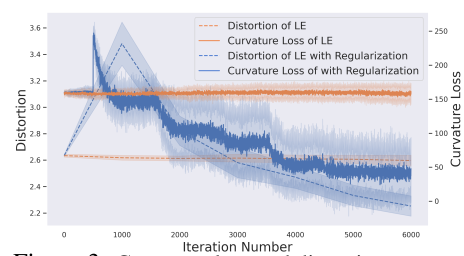

Figure 3: Curvature loss and distortion.

–To Reviewer #2 & #3– **Major concern 1:** *Different statements of Theorem 1.*

Response: Sorry for the mistake. We gave the wrong conditions in the main paper and corrected it in the supplementary material. We will correct the conditions of Theorem 1 in our revised version.

Major concern 2: *Does the condition in Theorem 1 apply to any 2-dimensional subspace of the embedding space? If so, it may introduce a large computational burden during the optimization, which does not mention in the Algorithm 1.*

Response: While the condition in Theorem 1 applies to any 2-dimensional subspace, we stress that it is only for theoretical proof and does *not* introduce computational overhead. The 2-dimensional subspaces are only used theoretically in Theorem 1 to prove the proportional relation. In contrast, Algorithm 1 optimizes directly node embeddings in the overall high-dimensional space, rather than every 2-dimensional subspace.

–To Reviewer #4– Formatting issue: We appreciate your recognition of our work. We will correct our paper to meet the format requirement of NeurIPS.