We thank all reviewers for their thorough reading and valuable comments! Please find below our responses.

Response to Review #2:

On the downstream tasks: It seems like there is a misunderstanding on the downstream tasks we considered in this paper. We do not dedicate our approach to the link prediction task even though our approach aims to jointly learn graph structures and node embeddings (that are optimized towards the downstream node/graph classification tasks). In some scenarios where the initial graph structure (could be noisy or incomplete) is provided, our goal is to learn "optimized" graph structures and node embeddings for the downstream task. In addition, in many other scenarios, no ground-truth graph structure is available and thus neither linkage information at hand. Therefore, in our experiments, we choose node-level and graph-level prediction tasks such as node classification, graph classification and graph regression.

On the significance of experimental results: our baselines include both state-of-the-art GNN models (which can only be used when graph structure is available) and graph learning models. In scenarios where the graph structure is available, compared to the state-of-the-art GNNs, our proposed models achieve either significantly better (e.g., 2.9% absolute performance gain on Pubmed data) or competitive results, even though the underlying GNN component of our models is just a vanilla GCN. In scenarios where the graph structure is not available (thus regular GNNs are not directly applicable), compared to graph learning baselines, our proposed models consistently achieve much better results on almost all datasets. Compared to our main graph learning baseline LDS, our models not only achieve significantly better performance, but also are more scalable (i.e., our IDGL-ANCH model). More importantly, our IDGL models are able to handle both transductive and inductive problems (whereas LDS can only handle transductive problems).

On the effectiveness of iterative learning: as shown in Table 2, the performance gain of utilizing our iterative learning strategy is actually significant and consistent by comparing IDGL vs. IDGL w/o IL, and IDGL-ANCH vs. IDGL-ANCH w/o IL on six datasets. In some datasets, the performance gap is even huge (e.g., more than 3% absolute performance gain on Citeseer data).

New results of other kNN-GNN variants on Wine, Cancer, Digits: kNN-GAT: 95.8 (3.1) | 88.6 (2.7) | 89.8 (0.6), kNN-GraphSAGE: 96.5 (1.1) | 92.8 (1.0) | 88.4 (1.8). For your reference, below are the reported results of kNN-GCN: 95.9 (0.9) | 94.7 (1.2) | 89.5 (1.3) and IDGL: **97.8 (0.6) | 95.1 (1.0)| 93.1 (0.5)**. It is easy to see that our IDGL consistently outperforms the state-of-the-art in kNN graph.

On the training time: as shown in Table 4, the training time of our proposed IDGL model is comparable to our main graph learning baseline LDS. In addition, the training time of our IDGL-ANCH variant is much lower than LDS, and is even comparable to the GAT model which is already fairly fast.

On data splits: we did mention the data split statistics in Line 242-244. As for the seven datasets in Table 1, we followed the experimental setup of previous works. As for the two datasets in Table 3, due to the space limit, we put the data split statistics in Appendix D.1. We will move the necessary data statistics to the main text in the revision.

On reproducibility: we described our experimental settings in Line 237-244 as well as our model settings in Appendix D.2. In addition, we provided the source code and instructions as supplementary files for better reproducibility.

Response to Review #3:

On the number of anchor nodes: Thanks for the great suggestions. we reported the sensitivity of the number of anchor nodes in Table 6 in the Appendix. We will add more discussions regarding that. We will report the effect of the number of anchor nodes on the convergence in the revision.

On adversarial GNNs: we will add more discussions on the connections and difference between our work and adversarial defenses in GNNs in the revision.

Response to Review #4:

On combining A(1) and A(t): Thanks for the great suggestions. As mentioned in Line 147-150, in order to combine the advantages of both A(1) (computed from the raw node features) and A(t) (computed from the updated node embeddings that are optimized toward the downstream task), we compute a weighted sum of them, as shown in Eq. (3). We will add more discussions in the revision.

On GNN components: in principle, our proposed framework is agnostic to any GNN that takes as input node feature matrix and adjacency matrix to compute node embeddings. We choose a vanilla GCN because it is simple and widely used. It will be interesting to apply our framework to other GNN variants. We will discuss this point in the revision.

On explanations of hyperparameters: we will add more explanations on important hyperparameters.

Final remark: we hope our replies resolve all reviewers' concerns and are helpful in making the final recommendation.