We thank the reviewers for their time and valuable feedback for improving the manuscript. All the reviewers agree that our work provides a novel and general technique to improve GNNs in both theory and practice. Reviewers make several valuable suggestion which we are going to address in the final version of the paper. Specifically, **R1** provides a detailed review that we greatly appreciate and raises concerns on our problem definition and the novelty. **R2 and R3** provide actionable insights on enhancing the experiments and generalizing our theory. We particularly thank **R3** for precisely capturing our problem definition and appreciating the theory. Regarding further generalizing our theory, we refer to some initial attempts described in the Sec.D (supplement). Next, we will focus on the rest questions and suggestions.

***Problem definition. R1** makes an insightful observation that by fixing two node sets from two graphs, the graph isomorphism testing becomes a simpler problem. However, the conclusion as it relates to our work is wrong. In contrast, our formalization is in fact *more general and more interesting* as it gives additional flexibility—the target node set could be of arbitrary size, even being the entire node set. By sightly revising the union bound in Lem C.1, Thm 3.3 can be directly generalized to the case with arbitrary size $|S^{(1)}| = |S^{(2)}| = k$ and the success probability becomes $1 - o(k/n)$. When $k = n$, the problem reduces to the standard graph isomorphism problem without any node-set information but we still successfully beat WLGNNs/1-WL test with high probability.

Although our theory is rather general, just to restrict the potentially too wide scope of this work, as stated in Sec. 2, we leave the study of entire graph representations for the future and focus on learning the representations of node sets with small k based on their contextual structures, which itself is a big, if not bigger, use case of GNNs. Such a problem formulation seems to be made by us, but actually not: Many previous works on node classification [3-6], link prediction [8-10] held the same concept but we are the first one to unify them, provide rigorous definition, and more importantly derive relevant theory and better algorithms. Methodologically, many GNN baselines such as GraphSAGE[21], SEAL[9], also need the access to the information of the queried node sets to perform ego-network sampling, but all of them insufficiently heuristically use such information. Therefore, we see it as an important contribution to systematically study the benefit of such information (Thm 3.3) and how to mostly use it (via DE).

***Novelty.** As **R1, R2** and our Sec. 4 pointed out, previous works proposed some similar idea as DE. However, as claimed, we are the first one to *mathemtically characterize* the power of DE, which governs many empirically successful GNN models (Sec. 4). The importance of theory has already been demonstrated when we try to evaluate **R1**'s trick to annotate the target node set S: With a certain choice of f_3 (Eq. (2)), DE actually reduces to this trick as a special case. However, the annotation trick may lose some representation power, as it does not give an injective f_3 (see its requirement in Def. 3.2): For the case when S is the entire node set ($k = n, S = V$), such a trick obviously fails. For the case when S is small ($k < n, S \subset V$), we may prove that GNN requires much larger L (Thm 3.3) to learn the distance information, and even more architectural effort to achieve the same representation power.

***Technical correctness and clarity. R1** had questions on Thms 3.3 and 3.7. ϵ is independent from n and thus we use "constant". Thm 3.7 generalizes short path distances to other DEs via Lem. E.3. We feel sorry to cause this ambiguity.

***Complexity. R1** questioned the complexity of our model for the whole graph representation (Cor 3.4). Again, please note that the whole graph representation is not our primary focus. For rebuttal, first, DEGNN-n (viewing the entire node set V as the target S) follows the same complexity as WLGNN but with more representation power while one needs to pre-compute distance encoding for every two nodes (with complexity $O(|E|d)$ where d is the graph diameter). Second, we agree that DEGNN-1 (viewing each node as a separate S and then aggregating their representations) for the whole graph representation is more complex than WLGNN. However, RingGNN [29] and PPGN [30], two most recently proposed provably more powerful GNNs, are even more complex, as they require 3-mode tensors. DEGNN-1 can use 3-mode tensors to compute all node representations simultaneously and allows utilizing sparse graph structures to propagate features. However, RingGNN (PPGN), if using sparse graph structures will break their theoretical foundation. **R2** questioned the complexity of DE computation. Actually, in practice, we only need to compute short path distances or random walks within constant hops (3-4 hops) whose complexity is constant times $|E|$, which is the same as that of feature propagation in GNN. In our experiments, the computation of DE costs less than 5% of the cpu time.

***Experiments.** We respectfully disagree with **R2**'s doubt on the safety of our evaluation. Regarding the dataset size, we have larger datasets for the link prediction task with 16k+ links. For node classification, we did not intentionally choose small graphs but used the same benchmarks as Struc2vec[5]. Moreover, we demonstrated the stable advantage of our models via providing the 95% significance scores (20-time tests). We further follow **R2**'s suggestion to compare with two more baselines RolX(RX)[3] and GraphWave(GW)[6] over three Airports datasets and obtain their results (Brazil 62.86(RX), 73.07(GW); Europe 54.50(RX), 56.56(GW); USA 52.52(RX), 54.55(GW)) which are far worse than DE-GNN. We further follow **R2 and R3**'s suggestions to evaluate our model over *an even larger graph (7k+ nodes) with richer features*, Actor (recently in GEOM-GCN (Pei et al, ICLR'20)) and get the results:

GCN 29.9±1.0 | SAGE 35.4±0.4† | GIN 27.4±0.7 | Struc2vec 30.04±0.82 | GEOM-GCN 31.82±0.91 | DE-GNN-SP **36.40±0.75*** | DEA-GNN **36.32±0.37***

DE(A)-GNN here propagates DEs to gather node structural representations and concatenates them with node features to predict node labels. GCN/GIN work bad due to the graph's heterophily. The results justify DEs' further significance to larger graphs, especially heterophilic ones. All the results are obtained under the same setting as in our paper.

In the final version, we will clarify our problem definition, algorithmic complexity and merge the above experiments.