We thank all the reviewers for acknowledging the novelty and concreteness of the path-integral idea presented in this paper. We find it very encouraging that the reviewers found the presentation clearly connects our graph learning framework with its deep roots in physics, and that the experimental results are convincing. We indeed believe in the conceptual and practical values of the PAN framework. We hope the new data set can also contribute to the community. We are very grateful for the reviewers' thoughtful and constructive comments. By incorporating many of these suggestions, we believe the quality of the manuscript is further improved. We discuss the major comments of the reviews, as follows.

PANConv for node classification (Reviewers #1, #2, #4) We appreciate Reviewers #1 and #4's interests in the performance of PANConv on node classification tasks, which we also agree are the best places to test the effectiveness of the convolution unit. In fact, as Reviewer #2 pointed out (also mentioned in related works), the advantage of the PAN convolution idea in node classification tasks have been reported in a short ICML workshop paper, see arXiv:1904.10996. The authors showed that path-integral based convolution could achieve excellent/top performance on benchmark Citation Networks datasets as compared to many existing convolution methods such as GCN. Moreover, this convolution method converges faster, which is consistent with what we found here in graph classification tasks. The current work distinguishes itself from the workshop paper by focusing on graph-level tasks and providing a pooling method. Additionally, as Reviewer #2 kindly pointed out, the discussion here is much more in-depth and more detailed. A new data set from statistical physics is also presented.

Ablation study (Reviewers #1, #3) We thank Reviewer #1 for this great suggestion. As we mentioned above, the effectiveness of PANConv is already observed in node classification tasks. We have run extra experiments and here we show some preliminary results to this end. In Table 1 (shown below), we fix the convolution unit (PANConv)/pooling unit (PANPool) and replace the other unit with TopKPool/GCNConv, respectively. We find with PANConv or PANPool, the GNNs achieve superior performance, which provides further evidence on the effectiveness of both units. We hope these results also help to answer Reviewer #3's question on the usefulness of PANPool alone.

Table 1: Ablation test for PANConv (L=4) and PANPool on graph classification benchmarks

Method	PROTEINS	PROTEINSF	NCI1	AIDS	MUTAGEN
PANConv+TopKPool	61.61	59.82	55.47	93.50	57.70
GCNConv+**PANPool**	66.96	66.07	66.67	88.50	66.90
PANConv+PANPool	69.64	70.24	74.30	99.50	68.82

More complex energy/weight functions (Reviewers #3, #4) This is a fascinating question which we would like to test in future. Considering the scope of the paper, for simplicity, here we only assign a scalar to the weight of paths with a certain length. This practice should be understood as the easiest implementation of the general PAN framework. We do not see fundamental difficulties in extending the weights to incorporate information of edges, such as replacing them with a neural network that takes the nodes on a path as input.

Some clarifications on the new dataset, codes, etc (Reviewer #3, #4) For Reviewer #3, in the generation process of point patterns, the number of nodes for each class is drawn from the same distribution. Figure 3 simply shows some samples with similar numbers of nodes (736, 703, 784, respectively). We will add more details in the supplementary material (SM) as well as in the paper if space permits. As mentioned in Section 5.2, when we increase the value of ϕ closer to 0.5, it is harder to distinguish the RSA pattern from HD. It thus adds to the difficulty of the classification. We may also add more complex point processes in the future (e.g., random organization), but their interpretations are less straightforward. In the main text, we have used "Hybrid PANPool" in all experiments. So "PANPool" is used as a general term but also refers to Hybrid PANPool in implementations. In the SM, we also studied many variations of PANPool. The cutoff L for each layer is chosen to be the same. In principle, there is no constraint on it, but this practice significantly simplifies the training.

For Reviewer #4, we provided the PyTorch codes for demonstration in the SM. More details are contained in README.md, which shows how to run the program. We have prepared a Github page and will release it immediately after the review process. We also updated the validation information in the main text.

Ongoing/Future work: Information Bottleneck (Reviewers #1, #4) We are very thankful to Reviewer #1 for directing us to the interesting reference on information bottleneck, which we now cite. This is a common challenge for all GNNs, which echoes Reviewer #4's comments on the limitation of expressiveness of all message-passing GNNs. Interestingly, Reviewer #1 pointed out that PAN may actually alleviate this problem compared to other methods. Guided by this insight, we are now carrying out experiments for different depths. We will include those results if time permits. We also plan to study the comparison between a "wide" PAN and a deep GCN in the future.