Firstly we would like to thank all the reviewers for their very insightful comments and suggestions. We hope these points appropriately address the reviewers' concerns, and they will be incorporated in more detail in the paper.

Baseline with mean, max and min aggregators. As Reviewer 2 suggested, we have added the PNA without *std* and scalers to the results below to better highlight the improvement brought by those components which are, to the best of our knowledge, entirely novel in the graph machine learning literature. As expected, its performance lies between the model with also the *std* (PNA no scaler) and the one with just *max* aggregator (MPNN max).

Most important aggregator. Answering to Reviewer 4, experiments showed that the choice of aggregator is very much task-dependent, e.g., for the graph theory artificial dataset (Figure 3) we found the *mean* was the best performing aggregator, whereas in computer vision tasks (Figure 5) is the *max* aggregator. The result achieved in tasks where we found out one aggregator was significantly more important than the others may suggest the PNA is able to focus on such aggregator.

Structure of the GNNs. We understand Reviewers 2 and 3's concern with the non-standard architecture using GRU, S2S, and repeated convolutions. We will clarify in the paper that (1) this is only used in the synthetic benchmarks, while in the real-world benchmarks, we kept the same architecture from Dwivedi *et al.*, (2) this architecture was chosen to provide a fairer comparison between the models as later explained. However, for completeness, we reckoned it important to run the models on a standard GNN architecture and report the results below. The GRU helps to avoid over-smoothing, and the models that do not have a skip connection across the aggregation (GAT, GIN and GCN) are those benefiting the most from it; therefore, to still provide a fair comparison in the results below, we added skip connections from every convolutional layer to the readout, in all the models. The S2S (as opposed to a mean readout used in the results below) most helps architectures without scalers as it can provide an alternative counting mechanism. Finally, the repeated convolutions are a parameter-saving prior which works well in these tasks but does not change the rank between the various models. We will clarify better the thought process behind choosing the architecture and add these results in the appendix to address these types of concerns in our final version.

Single task results. As Reviewer 1 correctly suggested, the multi-task approach offers a regularization opportunity that some models capture more than others. In particular, we found that models without scalers (or *sum* aggregator) are those benefiting the most from the approach; we hypothesise that the reason for this lies in some supervision that specific tasks give to recognise the size of a model neighbourhood. Moreover, more complex models are more prone to overfit when training on a single task. Due to space limitations, we only report the average performance, the detailed per-task performance and analysis will be added to the appendix of the paper.

Framework	PNA	PNA no scalers	mean, max & min	MPNN sum	MPNN max	GAT	GIN	GCN
multi-task	-3.130	-2.770	-2.570	-2.530	-2.500	-2.260	-1.990	-2.040
multi-task standard	-2.970	-2.550	-2.430	-2.780	-2.410	-2.000	-2.030	-2.140
single task	-2.860	-2.070	-1.850	-2.680	-2.100	-2.460	-1.960	-2.130

Graph type results. Following the suggestion by Reviewer 1, we have tested the models' performance across the various types of graphs in the synthetic benchmark. The results show that the PNA improves across all types; however, it performs the worst on the graphs with higher diameter (especially graphs close to lines), suggesting that the number of layers is not enough to reach the complete graph. Therefore, the main limitation to the PNA performance seems to be the message passing framework; this could motivate future research to try to improve the framework itself.

Model	Erdos-Rényi	Barabási-Albert	Grid	Caveman	Tree	Ladder	Line	Star	Caterpillar	Lobster
PNA	-3.377	-3.495	-2.770	-3.000	-3.097	-3.131	-2.371	-3.252	-2.879	-2.790
MPNN-sum	-2.085	-2.347	-1.955	-1.872	-2.237	-2.024	-1.991	-2.790	-2.219	-2.190
MPNN-max	-2.807	-2.943	-2.383	-2.523	-2.484	-2.721	-1.980	-3.066	-2.379	-2.339
GAT	-2.361	-2.578	-2.111	-2.027	-2.161	-2.250	-1.892	-2.678	-2.134	-2.114
GIN	-1.840	-2.084	-1.769	-1.679	-1.912	-1.842	-1.672	-1.927	-1.913	-1.877
GCN	-1.930	-2.187	-1.740	-1.536	-2.039	-1.841	-1.691	-2.088	-1.997	-1.974

As Reviewer 1 highlighted, indeed, the multi-task benchmark is undoubtedly not the main contribution of our paper; we will try to clarify that is not the case, but instead to motivate our need for the creation of this flexible benchmarking tool.

Standard datasets. We had initially omitted considering Cora and other datasets given their known oversaturation and the lack of potential for comparing existing GNN approaches. As recommended by Reviewer 2, we have conducted preliminary tests on them and found that the results are consistent with the existing state-of-the-art ($> 85\%$ on Cora). We will incorporate this discussion within our paper.

Finally, we want to thank Reviewer 4 for bringing to our attention the interesting work by Lee *et al.* on mixed pooling in computer vision. Although in a different field, the motivations behind it are similar to those that led us to this work; therefore, we will add a discussion of the connection between the two fields in our introduction.