We thank the reviewers for the constructive feedback, which confirms our contributions of laying a solid theoretical foundation for studying the abilitiy for GNNs to count substructures, proving concrete results for MPNNs and IGNs, and proposing the novel LRP model successful in substructure counting and real tasks. We address the concerns below.

1. Motivation and Theory

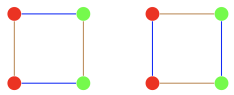
R1 *The advantage of GNNs over traditional subgraph-counting algorithms?* Many challenging prediction tasks on graph data can involve counting an **unknown** set of graph substructures, which calls for a data-driven solution that can not only count substructures but also **discover** which substructures to count. GNNs have such potentials, provided that they have enough expressive power to do so. Moreover, GNNs are known for integrating structural information with node and edge features, which is desirable for counting **attributed** substructures efficiently.

R4 *The theoretical results are not novel given existing results that GNNs are no more expressive than WL.* We respectfully disagree. While the reviewer is correct about prior work on the powers of MPNNs v.s. WL, our results precisely establish substructure-counting (in)abilities of WL which were unknown before our work. These results then carry over to MPNNs by building on prior work. Moreover, we also prove results for IGNs, which are not within the WL hierarchy.

2. The Local Relational Pooling model

R1 and R2 *Time-complexity of LRP.* While enumerating node permutations may sound prohibitive, LRP can be implemented practically, thanks to 1) only enumerating each local egonet; 2) only considering BFS-consistent permutations; 3) tensor cropping; 4) pre-computing node/edge indices for permutations. Its efficiency in practice is shown in Table 1.

R1 *How does GraphSAGE compare with LRP?* Indeed, the expressiveness of GraphSAGE deserves a separate analysis, especially when using multi-hop neighborhood. If we allow GraphSAGE to enumerate all permutations of the neighborhood, as the reviewer suggested, it becomes similar in spirit to LRP, except for aggregating neighborhood information as a sequence rather than a tensor, which results in advantages of LRP in exploiting edge features and high-order structures: 1) While the original GraphSAGE does not consider edge features, even if we can do so via augmenting the node features by applying an invariant function to the features of its immediate edges (e.g. summing or averaging), GraphSAGE cannot distinguish the pairs of graphs shown on the right, while LRP-1-3 can; 2) GraphSAGE cannot distinguish the pair of 12-circular graphs $C_{12}(1,3)$ and $C_{12}(1,5)$ (see [1]), no matter the hop-size being used, because $\forall k$, the k-hop neighborhood of every node in the two graphs has the same size. This means GraphSAGE cannot count squares, while LRP-2-4 can. Further, Table 3 shows the performance on the synthetic tasks of GraphSAGE + LSTM on full 1-hop neighborhood - it can count stars but not triangles, consistent with the limitation of information in 1-hop neighborhood.



R2 *How to decide which subtensor C_k to use for node i?* Let $G^{\text{ego}}_{i,l}$ denote the egonet of depth l centered at node i. For every BFS-consistent permutation, π, of the nodes in $G^{\text{ego}}_{i,l}$, $C_k(\pi \circ \mathbf{B}^{\text{ego}}_{i,l}) \in \mathbb{R}^{k\times k\times d}$ gives the tensor representation of the subgraph induced by the first k nodes under this permutation. On the choice of k - we set k to be an upper bound on the size of the substructures of interest.

R2 *Where does the term $\texttt{MLP}(D_i)/|S_n^{BFS}|$ come from?* If this term were 1 or $1/|S_n^{BFS}|$, we would get sum or mean pooling over all BFS-consistent permutations of the egonet, respectively. However, adding irrelevant edges to node i affects both the total number and fraction of permutations in which a substructure of interest appears, and so neither summing nor averaging over all permutations is fully desirable. Hence, we introduce $\texttt{MLP}(D_i)$ to learn to correct this bias as a function of the degree of node i. Thus, we can learn an invariant function over permutations that extends summation and averaging. From the literature of GNNs, this can be also seen as a generalization of the degree-normalization in GCNs. When $l > 1$, we generalize from $\texttt{MLP}(D_i)$ to an MLP over the list of degrees of all nodes in the permutation.

3. Experiments

R1 *Regarding what "top and median performances" means.* Sorry about not having made it clearer, but here "top" and "median" are with respect to five random seeds in training, and the errors are indeed averaged over all test graphs. The reason we report the top performance is because we are more interested in expressive power than training, and a good top performance suffices to indicates good expressive power.

R4 and R2 *Experiments on more standard datasets; How about LRP with $l > 1$?* Additional results on ZINC and MUTAG are shown below. When $l > 1$, each cropped subtensor contains one node from every depth level $\leq l$. LRP-7-8 almost matches the best performance (by GateGCN-E-PE, which augments node features with top Laplacian eigenvectors) on ZINC benchmarked in [2]. LRP-1-4 also surpasses GIN and 3WLGNN ([3]) on MUTAG.

Table 1: Results for ZINC. †: reported in [2].

Model	Train MAE	Test MAE	Time / Epoch
GraphSAGE†	0.081 ± 0.009	0.398 ± 0.002	16.61s
GIN†	0.319 ± 0.015	0.387 ± 0.015	2.29s
MoNet†	0.093 ± 0.014	0.292 ± 0.006	10.82s
GatedGCN-E-PE†	0.067 ± 0.019	0.214 ± 0.013	10.70s
GatedGCN-E†	0.074 ± 0.016	0.282 ± 0.015	20.50s
3WLGNN†	0.140 ± 0.044	0.256 ± 0.054	334.69s
LRP-7-8	0.028 ± 0.004	0.223 ± 0.008	72s
LRP-5-6	0.020 ± 0.006	0.256 ± 0.033	42s

Table 2: Results for MUTAG. †: reported in [3].

Model	Test Acc (%)
FGSD†	92.12
GIN†	89.4 ± 5.6
3WLGNN†	90.6 ± 9.7
LRP-1-4	91.0 ± 6.4

Table 3: Top & median loss on synthetic tasks.

Model	3-Star	Triangle
GraphSAGE	2.4E-10 / 2.0E-5	**1.3E-1 / 1.5E-1**
LRP-1-4	1.1E-5 / 3.8E-5	**2.8E-5 / 4.8E-5**

[1] Weisstein, "Quartic Graph." [2] Dwivedi et al., "Benchmarking GNNs." [3] Maron et al., "Provably Powerful Graph Networks."