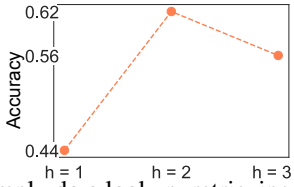
We thank the reviewers for their time and valuable feedback. Overall, reviewers found our method G-META *"novel"*, *"scalability is particularly appealing"*, *"theoretical analysis is great, substantial, valid, correct"*, experiments are *"comprehensive"*, and the paper *"well-written"*. Below, we clarify important points raised by reviewers: **(1) relation to existing work**, **(2) local subgraphs in G-META vs. computation graphs in GNNs**, **(3) size of local subgraphs**, **(4) ablation**, and **(5) others.** We believe these clarifications, together with our new analyses, resolve all key issues raised.

(1) Relation to existing work. **R1** questioned the difference between Meta-GNN and G-META, saying it *"remains moderate."* We respectfully disagree. The key difference are local subgraphs, and this is crucial because local subgraphs enable i) accurate and fast adaptation to new tasks, ii) learning in few-shot settings, and iii) theoretical justification. In contrast to our G-META, Meta-GNN and, similarly, Meta-Graph, use *entire graphs* for meta-learning. Because of that, they are unable to propagate label information across large graphs when only a few node labels are given. Theoretically, Meta-GNN/Meta-Graph works only on 1 out of 3 graph meta-learning problems (see Appendix B) and does not have any theoretical motivation whereas G-META works on all 3 problems and is theoretically justified. Empirically, G-META outperforms Meta-GNN, for example, by 65% on the ogbn-arxiv dataset (Meta-GNN cannot even be used on other datasets and meta-learning regimes). As suggested by reviewers, we will carefully discuss this in the final version.

(2) Local subgraphs vs. computation graphs. **R3** raised a critical concern that *"the idea of using local subgraphs to compute node representations is not novel."* This points to a **critical misunderstanding—computation graphs in GNNs are used to generate node embeddings vs. local subgraphs in our G-META are used to transfer knowledge for graph meta-learning.** We are not claiming novelty in *"the idea of using local subgraphs to compute node representations."* Instead, as we write in the paper, we are claiming novelty in the idea of using local subgraphs for graph meta-learning, as recognized by **R1** and **R2**. This innovation has important implications, which we show theoretically (i.e., proofs, solving classes of graph meta-learning problems not solved before) and empirically (i.e., considerable boost in accuracy over 9 baselines and 7 datasets). For example, a baseline method Meta-GNN, which uses a standard GNN on an entire graph together with MAML, performs 42.5% worse than G-META-MAML, a simplified variant of our G-META. Note that the only difference between Meta-GNN and this simplified G-META's variant is that it uses the entire graph vs. local subgraphs. We will clearly mention this contribution, which we agree is crucial for G-META.

(3) Size of local subgraphs and computational complexity. **(3.1)** **R2**, **R3**, and **R4** nicely point that G-META's performance can vary with local subgraph size h. To address this, we evaluate G-META for $h = 1, 2, 3$ (Figure for Fold-PPI). We find that $h = 2$ gives the best performance, which nicely corroborates **R2**'s hypothesis on the inverted U-shape relationship. We will include this analysis in our final version. **(3.2)** **R4** also raises an important concern on computational complexity of subgraph extraction. We would like to clarify that we don't need to compute *"shortest paths ahead of the training"*. Instead, we simply do a lookup, retrieving neighbors of neighbors. Empirically, we find the subgraph construction takes 14.7% of training time, and this can be further reduced by implementing techniques for improving training efficiency, e.g., GraphSAINT [Zeng et al., ICLR 2020]. Regarding **R4**'s comment of *"evaluating each node label individually"*, we note that each mini-batch consists only of a few labels (e.g., 9 in 3-way 3-shot learning) and as such is a cheap operation.



(4) Further ablations. **All reviewers** raise an important point on ablation, which we agree is crucial especially for G-META. While we already included ablation in the form of baselines, we will make it explicit in the next version following **R3** and **R4**. We thank **R3** and **R4** for nicely pointing out that baselines ProtoNet and MAML can be seen as G-META's ablations. In response to **R2**, new results in Table show that gradient-based meta-learning aids G-META more than metric-based meta-learning but both are indispensable for G-META's performance.

Method	ogbn-arxiv	Tissue-PPI	Fold-PPI
G-META	**0.451**	**0.768**	**0.561**
- MAML	0.372	0.546	0.382
- Prototype	0.389	0.745	0.482

(5) Baselines and further clarifications. **(5.1)** **R1** raised an important point about performance under varying size of the training set, i.e., K in the K-shot problem. We conduct experiments and observe, as expected, a linear trend between K and performance (e.g., on arxiv-ogbn, accuracy goes from 0.373, 0.442 to 0.484 for $K = 1, 3$, and 10, respectively). We will include the full study in the final version. **(5.2)** **R1** correctly pointed out G-META *"covers node classification task and link prediction task where the target is discrete class label."*. In contrast, existing Meta-GNN and Meta-Graph only work for one of these two tasks—G-META is the first to work on both tasks. Also, to avoid confusion, we will update reference for graph-level molecular prediction [Hu et al., ICLR 2019] to edge-level interaction prediction [Zitnik et al., Bioinf. 2018]. **(5.3)** We thank **R4** for rightly pointing out that local subgraphs can alleviate over-smoothing because, in each iteration, different subgraphs are fed into GNN, which promotes inductive generalization. **(5.4)** **R4** raised a question about hyper-parameters. We use random search on validation set to select hyper-parameters and find that model performance is stable for a broad range of values. We will include the recommended set of constant parameters in the next version. **(5.5)** **R4** raised an important point on experimental setup. We follow standard episode training and semi-supervised setting in which most nodes are not labeled, i.e., few-shot learning. In K-shot N-way setup, for "Multiple Graphs and Shared Labels" problem, each task samples K nodes for each label N_i in the same label set of size N from one graph, and different tasks are associated with different graphs. For "Multiple Graph and Disjoint Labels" problem, each task defines an N-size label set, and samples K nodes for each label N_i.