# Response to the Reviewers

We thank the reviewers for their critical assessment and insightful comments of our work. We have made extensive modifications to our manuscript. In the following we address their concerns point by point.

# Reviewer 2

Reviewer Point P2.1 — In the paper, authors accomplished a unique study and analysis on GNN models training complexity. The articles first review and development history of GNNs and creatively model all architectures as input layers, intermediate layers of graph neurons and prediction layers. And they quantitatively summarize the time and space complexity of 4 representative GNNs, including graph convolution, gated recurrent graph net, graph attention net and GraphSage. Most importantly, the article first break down complexity into operator level and offered analysis of good granularity, giving reader more guidance in future study. At last, the solid experiments included the study of effects of hyper-parameters and a comparison of two major sampling techniques: neighbor sampling and cluster sampling.

Reply: Thank you for your positive comments on our manuscript. We have carefully revised the manuscript according to your kindly comments and suggestions in the following points.

Reviewer Point P2.2 — In general, the paper was well written and organized with good structure and clear narratives. Just some minor language errors like line Page 8, Line 208, ”In active graph neurons” =*>*”Inactive graph neurons”.

Reply: We feel sorry for our carelessness. We have proofread our revised manuscript to eliminate such language errors.

Reviewer Point P2.3 — I was impressed by the way that authors categorize layers and operators in GNNs, very clear and instructive.

It is also pretty neat to divide layer time complexity into two buckets: vertex calculation and edge calculation. The data model pretty well summarizes mainstream GNN layer architectures. And this analysis is very insightful for layer profiling.

And the experimental evaluation were done over 6 large graph-structured datasets.

Reply: Thank you for the positive comment.

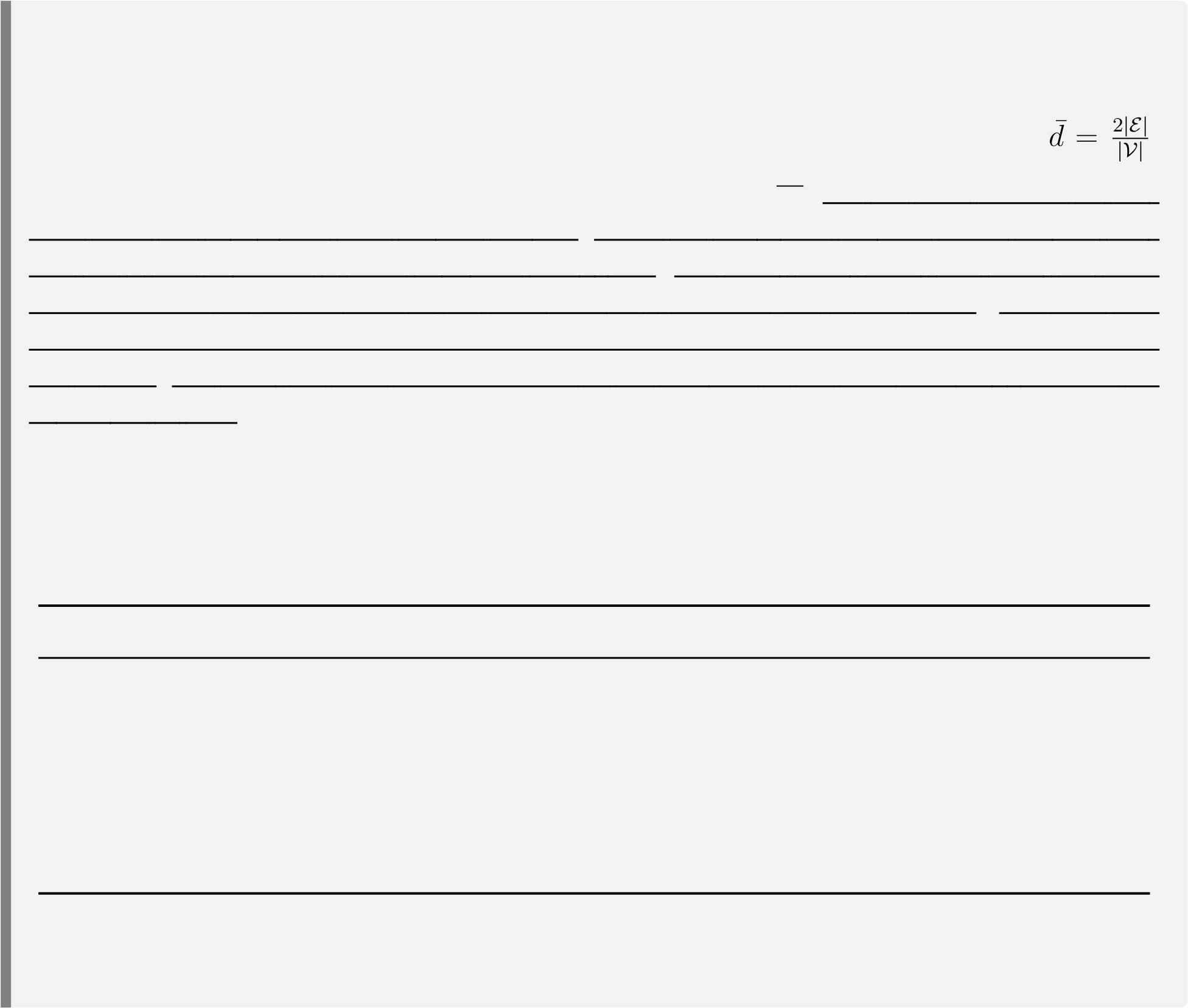
Reviewer Point P2.4 — While, one major drawback is that I did not clearly see the analysis complexity v.s. accuracy. For example, in Figure 19 and 20, I did not see network accuracy from those 4 GNNs. There is always tradeoff between model complexity and model performance, and in some scenarios where high complexity is allowed, a sophisticated model of more powerful representation capability is still needed.

Reply: Thanks very much for your valuable suggestions. The model complexity directly affects both the accuracy and the training time. In order to analyze the relationship between model complexity and accuracy, we have conducted two kinds of extra experiments in the revised manuscript: (1) how the hyper-parameters of the GNNs (like the dimension of hidden vectors and the number of heads) affect the accuracy of GNNs (in Section 4.1); (2) how the batch size in the sampling methods affects the accuracy of GNNs (in Section 4.4).

We have added more dataset description related to accuracy evaluation in Section 3.1 “Experimental Setting”. Newly added sentences are marked with underlines in the following quotation.

## Section 3.1 “Experimental Setting”

*Datasets*. We used six real-world graph datasets as listed in Table 1 (in the reply) that were popular in the GNN evaluation [28,31,34,36]. For directed graphs, PyG converts them into undirected ones during data loading. Thus, the average degree of a directed graph .

For an undirected graph, the average degree is defined as *d*¯= |V||E|. For the cam dataset, its vertices were not associated with feature vectors. Thus, we generated random dense feature vectors for it and excluded it from accuracy evaluation. For amp, amc, cph, and cam, we used 70%/15%/15% of the vertices as the training/evaluation/test set, respectively. For pub, we used 500/1000 vertices as the evaluation/test set and the remained as the training set according to [28]. For fli, we used 50%/25%/25% of the samples as the training/evaluation/test set according to [31]. We also used random graphs generated by the R-MAT graph generator [35] in some experiments, to explore the effects of graph topological characteristics (like average degrees) on performance bottlenecks. Input feature vectors of random graphs were random dense vectors with the dimension of 32. Vertices of random graphs were classified into 10 classes randomly.

Dataset |V| |E| *d*¯ *dim*(*v*) #Class Directed

pubmed (pub) [28] 19,717 44,324 4.5 500 3 Yes amazon-photo (amp) [34] 7,650 119,081 31.1 745 8 Yes amazon-computers (amc) [34] 13,752 245,861 35.8 767 10 Yes coauthor-physics (cph) [34] 34,493 247,962 14.4 8415 5 Yes flickr (fli) [31] 89,250 899,756 10.1 500 7 No com-amazon (cam) [36] 334,863 925,872 2.8 32 10 No

Table 1: Dataset overview. *d*¯represents the average vertex degree. *dim*(*v*) is the dimension of the input feature vector.

In this reply, we focus on the experiments related to hyper-parameters in Section 4.1. We introduce the experimental results of the sampling techniques in Section 4.4 in the next reply.

Hyper-parameters determine the model complexity of a GNN. Generally speaking, higher values of hyper-parameters bring higher model complexity and increase representation capability. In the revised manuscript, we add a new subsection Section 4.1.4 “Effects on Accuracy” at the end of Section 4.1 “Effects of Hyper-parameters on Performance” to analyze how hyper-parameters affect the accuracy. We have two main findings. First, the accuracy of GNNs is much more sensitive to the dimension of hidden vectors (for GCN/GGNN/GaAN) and the dimension of each head *dhead* (for GAT) than the other hyper-parameters. Second, the relative accuracy of the four typical GNNs varies greatly with different datasets. We quote from Section 4.1.4 of the revised manuscript to present our results.

## Section 4.1.4 “Effects on Accuracy”

The values of hyper-parameters determined the model complexity of a GNN. The relationship between model complexity and accuracy was complex. Generally speaking, higher model complexity brought more powerful representation capability and might bring higher accuracy, but it also increased the risk of overfitting.

To evaluate the effects of hyper-parameters on accuracy, we measured the accuracy of the typical GNNs with varying hyper-parameters. Figure 11 (in the reply) shows the experimental results. For GCN, its accuracy was sensitive to the dimension of hidden vectors . As increased, the accuracy first increased quickly and then stabilized when 8. For GGNN, its accuracy curves showed similar trends as GCN, but GGNN was more sensitive to than GCN. Its accuracy even decreased when . Since

GGNN had high model complexity (with 13 weight matrices/vectors to train), GGNN might occur overfitting in those cases. For GAT, its accuracy was more sensitive to the dimension of each head *dhead* than the number of heads. For GaAN, only showed obvious impacts on accuracy. The experimental results indicated that the accuracy of the GNNs was often low when (for GCN/GGNN/GaAN) or *dhead* (for GAT) was too low. As or *dhead* increased to a certain threshold, the GNNs gained sufficient learning ability to achieve stable accuracy.

*dim*

(

*h*

1

*x*

)

*dim*

(

*h*

1

*x*

)

*dim*

(

*h*

1

*x*

)

≥

*dim*

(

*h*

1

*x*

)

*dim*

(

*h*

1

*x*

)

≥

1024

*dim*

(

*h*

1

*x*

)

*dim*

(

*h*

1

*x*

)

*dim*

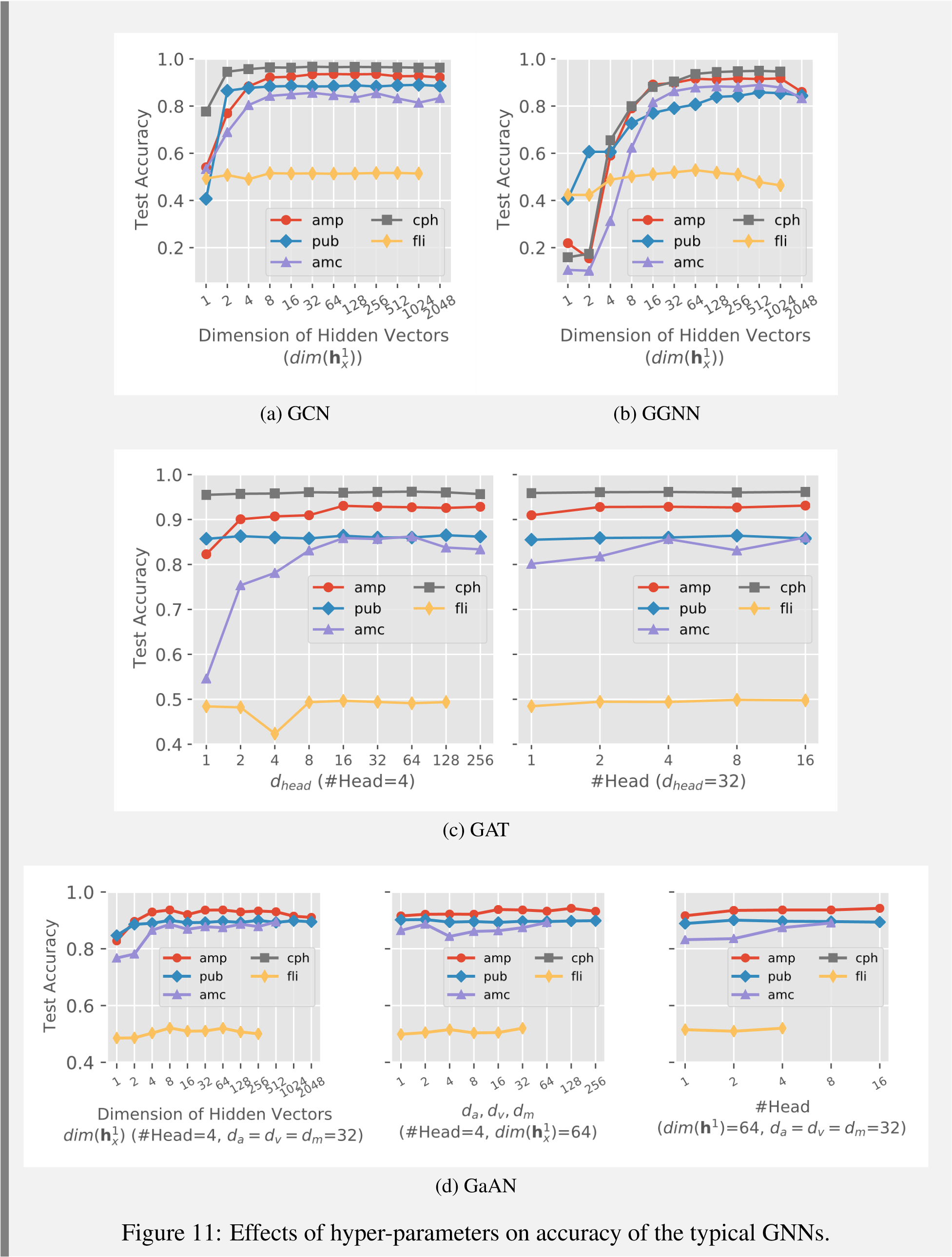
(

*h*

1

*x*

)



Figure

12

in the reply) further compares the best accuracy that each GNN could achieve on

(

different datasets. The best accuracy of the typical GNNs was very close. It was also close to

the accuracy reported in their original references [34, 31]. The results indicated that there was

no clear winner. The relative accuracy between GNNs varied greatly with different datasets.

GaAN achieved the highest accuracy in three out of five datasets. GCN achieved the highest

or second highest accuracy in three out of five datasets, though its model was simplest. Simple

GNN models (such as GCN) could still achieve good accuracy with proper hyper-parameter

settings.

amp

pub

amc

cph

fli

Dataset

0.4

0.5

0.6

0.7

0.8

0.9

1.0

Test Accuracy

0.94

0.89

0.86

0.97

0.52

0.92

0.86

0.89

0.95

0.53

0.93

0.86

0.86

0.96

0.50

0.94

0.90

0.89

Out of Memory

0.52

GCN

GGNN

GAT

GaAN

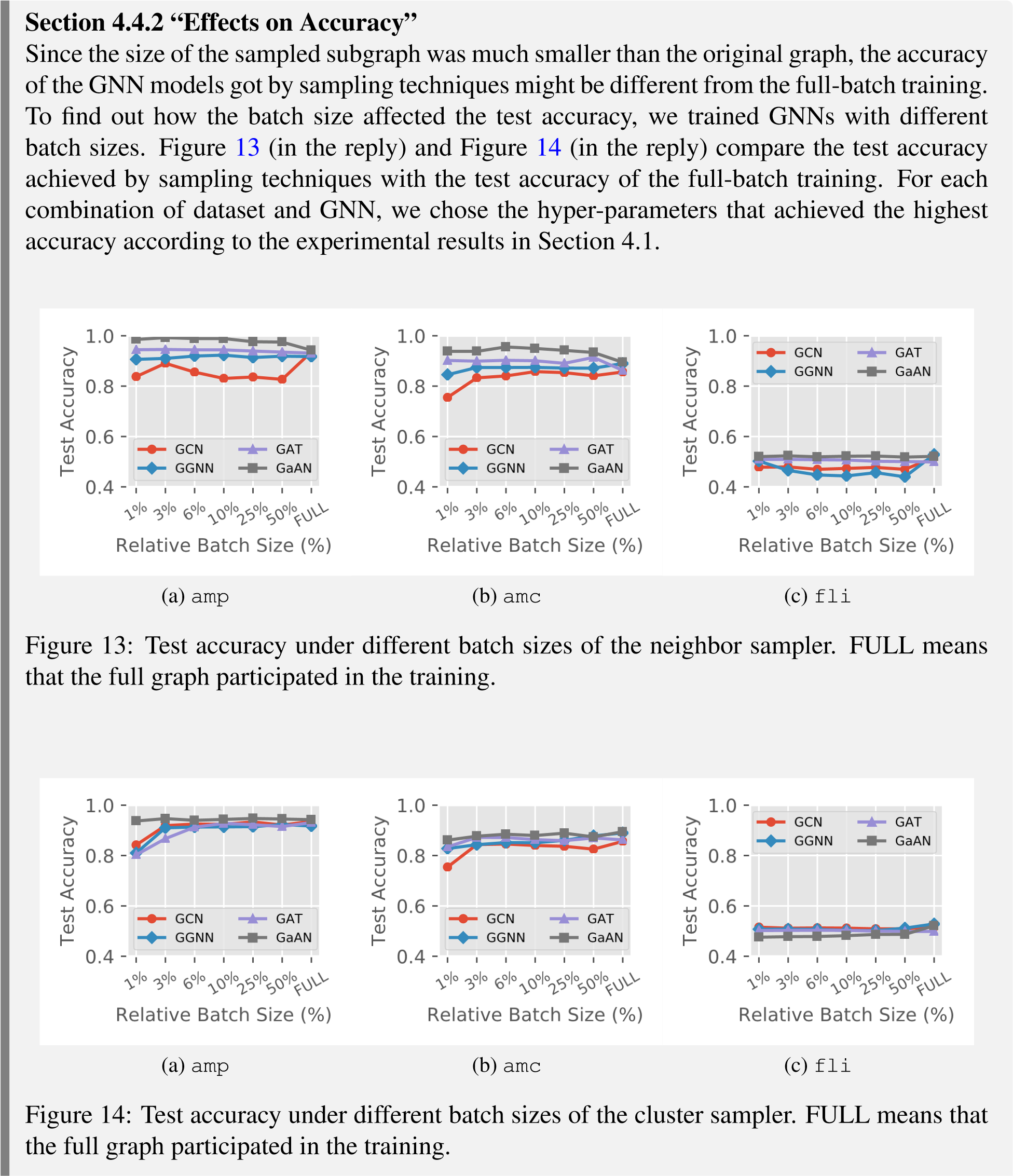
Figure 12: Best accuracy that each GNN achieved on different datasets.

Reviewer Point P2.5 — Sampling method is definitely going to reduce model complexity, since all models complexity depend on graph node number N, while performance is going to be compromised as well. I would like to see authors resolve the concern of significant accuracy drop after applying aggressive sampling of subgraphs.

Reply:

Thanks for the insightful comment. The sampling methods indeed affect the test accuracy of GNN models [5, 28, 26, 31]. Without sampling, the GNN models are trained with a *batch* gradient descent manner. With sampling, the GNN models are trained with *mini-bach* stochastic gradient descent. The structure of the GNN remains unchanged between batches. Each GNN layer always consists of |V| graph neurons, where |V| is the number of vertices in the graph. But the numbers of *activated* graph neurons are different for different batches. In each batch, only the neurons corresponding to the vertices appearing in the sampled subgraph will be activated. The model parameters are updated according to the gradients calculated on the sampled subgraph in each batch. Thus, the test accuracy of the models trained with sampling techniques may be different from the full-graph training.

To discuss how the sampling techniques affect model accuracy, we have added a new subsection Section 4.2.2 “Effects on Accuracy” in Section 4.4 “Effects of Sampling Techniques” in the revised manuscript. We quote from Section 4.4.2 of the revised manuscript to present our results.



The experimental results confirmed the effectiveness of the sampling methods in terms of accuracy. When the relative batch size was greater than or equal to 3%, the test accuracy of the GNNs trained with sampling was close to the accuracy obtained by full-batch training. In most cases, the accuracy achieved by the sampling techniques was slightly lower than the full-batch training. However, there were some exceptions (like GaAN in Figure 14a and Figure 14b in the reply) that the accuracy achieved by sampling was even higher.

The relationships between batch size and test accuracy were complex. A larger batch size did not always bring higher accuracy. For example, the accuracy of GaAN in Figure 14a (in the reply) and GGNN in Figure 14c (in the reply) decreased as the batch size increased. A smaller batch size sometimes could achieve higher accuracy. For example, GAT achieved a higher accuracy with 1% relative batch size than the full-batch training in Figure 14b (in the reply). Given a sampling method, we found that the optimal batch size highly depended on the dataset and the GNN algorithm. Our observations were similar to [31]. How to automatically select a proper batch size is a topic worth further studying.

Among the two sampling methods, the performance of the cluster sampler was stabler than the neighbor sampler. With the cluster sampler, the test accuracy of different GNNs was very close to the accuracy of full-batch training. With the neighbor sampler, the test accuracy of different GNNs showed more obvious differences.

Reviewer Point P2.6 — Hope authors supplement the effect of sampling and GNNs on accuracy while comparing different complexity of model and sampling methods.

Reply:

We are grateful for your insightful suggestion. As suggested, we additionally evaluated the effects of model complexity on accuracy in Section 4.1 “Effects of Hyper-parameters on Performance” in the revised manuscript. We added experimental results on how the hyper-parameters of GNNs affected the accuracy. Since different GNNs had different model complexity, we further compared the test accuracy of different GNNs on the same dataset. More details are available in the reply to Reviewer Point P2.4.

We also additionally evaluated the effects of sampling techniques on accuracy in Section 4.4 “Effects of Sampling Techniques” in the revised manuscript. We presented the experimental results in a new subsection Section 4.4.2 “Effects on Accuracy”. More details are available in the reply to Reviewer Point P2.5.