— For **Reviewer #1**. Thank you for the comments! We address your specific concerns in detail below. —

Response to Q1: We want to highlight that our main contribution is the novel flexible generative framework which takes advantage of both network models and GNNs. Our technical contributions lie in the derivation of the variational method when bridging the network models with GNNs. We intend to use simple existing building blocks to avoid unnecessary confounding factors for proof-of-concept of the general framework, as mentioned in lines 142 to 144.

Response to Q2: First, the model we used, $P(G, Y, X) = P(G|X, Y)P(Y|X)P(X)$ is as much a "fully generative model" as $P(G|X, Y)P(X|Y)P(Y)$. Note that specifying $P(X)$ only results in an extra additive term to the loss at line 177 (together as $L(\theta, \phi) - log(P(X))$), which is irrelevant to the model we will use for prediction. Therefore, removing this term will lead to no loss of generality. We agree $P(G|X, Y)P(X|Y)P(Y)$ could be another design choice for the generative model. We did try both and found the one in our paper works better at this task.

Response to Q3: For LSM models, the ELBO is fully differentiable w.r.t. all model parameters, therefore the parameters can be learned through SGD. For the simple instantiation of SBM used in this paper, p_0 and p_1 are the only learnable parameters of $P(e_{ij}|y_i, y_j)$. In principle, they can also be learned through SGD, but given just two parameters, we found grid-search with limited combinations (line 220) works better in practice. In future work, we will attempt to further improve the estimation of the parameters in the SBM, taking advantage of the vast literature in this domain. As proof-of-concept, the simple learning procedure is sufficient to obtain desirable improvements on the benchmarks.

Response to Q4: We acknowledge that the GNN-based approximate posterior models do not accommodate Y. We made this choice mainly as a trade-off for computational efficiency. Even with this approximation, the GNNs can leverage the generative model and improve the inference performance. Interestingly, we recently became aware of a concurrent study (Graph Markov Neural Networks, ICML 2019) after our submission which used the same approximation (see their Eq.3). We plan to design more inclusive model structures that can efficiently handle the labels in future work.

Response to Q5: We did further analysis to investigate the reasons. First, note GAT is no better than GCN on Pubmed. Second, the number of classes is smaller on Pubmed (3) than on Cora (7) and Citeseer (6). The average number of 2-hop neighbors is much larger on Pubmed (57.1) than on Cora (35.0) and Citeseer (13.5). When the number of classes is small and the graph is relatively dense, GCN is already quite capable of propagating feature information from neighbors, which makes it difficult to further improve. When there are more classes or the graph is relatively sparse (e.g., Cora, Citeseer, and the missing-edge setting of Pubmed), the advantage of our proposed method is more evident.

— For **Reviewer #2**. Thank you for the encouraging comments! —

— For **Reviewer #3**. Thank you for the comments! We address your specific concerns in detail below. —

Response to Q1: We agree that Bayesian GCN (B-GCN) should have been another proper baseline. However, the code of B-GCN is not released and we were not able to reproduce the results in [22] despite our due diligence. We instead test LSM_GCN closely following the experimental setup of [22] and compare with the results reported in [22]. Specifically, we use fixed hyper-parameters (HPs) for LSM_GCN, where those of the GCN part are the same as B-GCN. We also fix the LSM part with hidden size 16 and $\eta = 1$. To assure fair comparison, we use the official (fixed) split and report averaged results of 50 runs. As an evidence of fair comparison, below we report the GCN performances from [22] and from our implementation, which are very similar. The results show that LSM_GCN outperforms both B-GCN and GCN in most cases. Besides the empirical comparisons, we also want to highlight that when it comes to more complex graphs, the proposed framework (modeling the relationship between the graph, node labels, and features) can better utilize the information of the data than B-GCN (modeling the graph alone).

Methods	Cora 20	Cora 10	Citeseer 20	Citeseer 10	Pubmed 20	Pubmed 10
GCN ([22]) / GCN (ours)	81.6 | 81.5	74.9 | 74.8	70.8 | 71.4	65.8 | 66.8	78.9 | 79.0	72.8 | 71.7
B-GCN([22]) / LSM_GCN	81.2 | 82.4	76.6 | 78.6	72.2 | 73.8	70.8 | 68.7	76.6 | 78.0	72.3 | 70.6

Response to Q2: First, we clarify that the test data (Planetoid split) in each dataset has only 1,000 nodes, which account for about 1/2 of the edges in Cora and Citeseer, and 1/9 in Pubmed. Second, as we mentioned in 4.2.1, this setting is realistic and actually has important practical value: these types of predictions are usually applied to new users in social media networks who are likely to have few or no connections. There are a large number of low degree nodes in real world networks given the power-law degree distribution. Finally, this setting signifies another advantage of the proposed method over B-GCN: B-GCN cannot transfer knowledge to isolated nodes as it models the graph alone.

Response to Q3: Combining the experiments in the response to Q1 and the experiments in our paper, we have evaluated our model and baselines in two ways: with and without tuning the HPs using the validation set. In both ways, our method outperformed the baselines in most cases. We also want to clarify that both the GCN and the GAT papers did use the validation set (see their experiment section for details), and B-GCN borrowed the HPs from the GCN paper.