We thank all reviewers for their thoughtful comments and suggestions. We address each review separately.

Reviewer #1. Regarding the encoder architecture impact, in Table 1, we vary the classification model making it increasingly more powerful, and demonstrate that our method produces improvements in all cases. For the agreement model encoder, we found that the results are not as sensitive to the encoder choice (e.g., for CIFAR10 switching from an MLP to a CNN did result in significant differences). We will include results for different agreement model architectures.

Reviewer #5. We thank R5 for the relevant references. We were not aware of them—especially the contemporaneous ones from ICLR and ICML 2019. SNTG infers a similarity graph between samples, but it does so in a significantly different way than GAM. Also, in contrast to SNTG, we propose an additional self-training component, and our method is applicable when a graph is provided, whereas SNTG (as published) is not designed to use information from a provided graph. We will include a thorough comparison to SNTG and Fast-SWA in the paper. We will also discuss the following:

- Parameters: Increasing the number of parameters of the baselines to match that of the respective GAMs results in worse performance for the baselines (e.g., in Table 1, $\texttt{MLP}_{256}$ has the same number of parameters as $\texttt{MLP}_{128}$+GAM, as we use an $\texttt{MLP}_{128}$ for the agreement model, but it performs worse than $\texttt{MLP}_{128}$+GAM and $\texttt{MLP}_{128}$). This is expected as GAMs provide a robust form of regularization for training high capacity models that tend to overfit otherwise.
- Convergence: Prior work [e.g., Blum and Mitchell 1998, Balcan et al. 2005] proves that co-training converges if: (i) the majority of the learners perform better than random guessing after the first iteration, and (ii) the mistakes they make are weakly dependent. Our experiments indicate that (i) is true in our case. (ii) is harder to verify due to the coupling between the models. However, our empirical evaluation shows that co-training converges successfully. Note that in Fig. 5, even the worse iterations are well above chance, so it should not diverge under these assumptions.
- Experiments: The missing numbers for $\texttt{GCN}_{1024}$+VAT are 83.4, 68.9, 79.5 on Cora, Citeseer, and Pubmed, while $\texttt{GCN}_{1024}$+VATENT obtains 32.5, 8.5, 18.0, which follow the same trend as our other results. For VATENT, we observed that on the graph datasets the entropy term becomes large and dominates the loss. Decreasing its weight makes the performance to converge to that of VAT. Our implementation works on CIFAR10 and SVHN, thus it seems unlikely to be the reason behind the poor results. Interestingly, [2] reports only GCN+VAT results and not GCN+VATENT.

Regarding comparisons with other methods, we will add the results reported in [2] to Table 1. Their best numbers are lower than our GCN+GAM. [4] tackles the same problem, but their evaluation is on random train/test splits rather than the commonly used Planetoid splits. We observe that the GCN paper reports much better results on random splits than [4], and we have demonstrated that GAM can be applied on top of GCN to improve it further. For completeness, we will report results on random splits and compare with [4]. To compare with SNTG and Fast-SWA, we plan to run the experiments with a 13-layer CNN suggested by R5 for the camera-ready. Note, however, that we do not necessarily see these approaches as competitors to GAM, but rather as additional regularizers that, similar to VAT, can be applied in conjunction with GAM to further improve generalization. To illustrate that GAM works for large networks too, here are results (obtained after the submission deadline) using the WideResnet of Oliver et al. 2018 on CIFAR10-4000: baseline 79.69%, +Π-Model 83.63%, +Mean teacher 84.13%, +VATENT 86.87%, +GAM* 87.42%.

Reviewer #6. R6 suggests a discussion on the challenges in simply replacing classification models in label propagation with deep learning models. We address this through an example from our paper, and then explain how this example is more broadly applicable. Replacing classification models in label propagation with deep learning models is exactly what Neural Graph Machines (NGMs) do (described in Section 2): an NGM is a label propagation model complemented by a deep learning classifier operating on the node features. Setting the regularization coefficients to 0 makes it a pure deep learning model, while increasing their values brings it closer to label propagation. When the graph is noisy, the regularization coefficients need to be small (otherwise the regularization forces connected nodes from different classes to incorrectly have the same label), thereby reducing the effect of the graph on the model. However, with such minimal regularization the model tends to overfit to the few available labeled examples. Our approach combines deep learning with label propagation in a manner that allows us to handle noisy graphs in a robust fashion. Note that other methods besides NGM also suffer from this problem (e.g., GCN, Planetoid)—see *Robustness* section. Our experiments show how GAMs are able to learn in a much more robust manner.

Novelty: The novelty of our algorithm is the interaction between the agreement and classification models, which allows it to benefit from both label propagation and deep learning even when dealing with noisy graphs (where most label propagation algorithms fail), or no graphs at all. It is surprising and interesting that even though the two models learn using the same features and same data, their interplay can produce such large increases in accuracy on a wide variety of base networks (MLP, CNN, Resnet, GCN, and GAT), suggesting they learn complementary information.

Co-Training: We argue that our proposed training algorithm does indeed fit in the co-training framework. While the original paper [Blum and Mitchell, 1998] proposed co-training in the setting described by R6, the same authors subseqeuently proposed co-training settings where some classifiers predict label distributions and others predict coupling constraints over these distributions (like in our setting). Perhaps the most notable and influential example of this is the Never-Ending Language Learning (NELL) system [Mitchell 2015, 2018].