A Applying SBM ablation model

The aim of this ablation model is to preserve community structure while eliminating the degree distribution. To create these graphs, we use a stochastic block model which is a generative model that produces random graphs with communities identified by a stochastic block matrix and with a Poisson degree distribution.
Community detection. In order to achieve this, we first assigned a community id to each node using the Louvain method for community detection. With the resulting communities we were able to produce a stochastic matrix which served as an input to the SBM.
Building the stochastic block matrix. We first calculate edge density within each community and then between each pair of different communities. These edge densities serve as edge probabilities between each pair of nodes in the resulting stochastic matrix. With this, we preserve the inter-community relationships and the density within communities.

B Model settings

Common settings. For all the GNN architectures, we use a model of two hidden layer, with the second layer as the output layer. The first hidden layer for each GNN architecture uses the activation that was used in the corresponding paper. The second hidden layer has the size of the number of labels in the respective dataset, and uses softmax as an activation function. For the training process, we use the negative log likelihood as our loss function. We use Adam optimizer for 200 epochs, with early stopping after the validation accuracy has not improved for 10 consecutive epochs[1]. Then we select the state of the model at the epoch where the highest validation accuracy was achieved[2].
Hyperparameter search. We perform the following grid search to find the hyperparameter setting which maximizes the mean validation accuracy over 100 random splits with 20 random model initialization.

- First hidden layer size: $[12, 24, 48, 96]$
- Learning rate: $[0.001, 0.005, 0.01]$
- Dropout probability: $[0.2, 0.4, 0.6, 0.8]$
- Regularization weight: $[0.0001, 0.001, 0.01, 0.1]$
- Attention dropout probability fo GAT: $[0.2, 0.4, 0.6, 0.8]$

[1] The training typically stops before the 50-th epoch.
[2] We implement our models and evaluations using PyTorch Geometric https://pytorch-geometric.readthedocs.io/