DirGSN: A framework for increasing accuracy in heterophilic graphs

Table 1: Notation.

σ	An element-wise non-linearity.
t	Iteration, or layer t.
$d^{(t)}$	The dimension of a vector at iteration t.
d	The dimension of a vector, and an abbreviation for $d^{(0)}$.
$\mathbf{1}^d \in \mathbb{R}^d$	A d-dimensional vector of all 1's.
$\mathbb{I}^d \subseteq \mathbb{R}^d$	The set of d-dimensional one-hot vectors.
$\mathbb{B}$	Boolean domain $\{0, 1\}$.
$\mathbf{b}^{(t)} \in \mathbb{R}^d$	A bias vector.
$\mathbf{x}_u \in \mathbb{R}^d$	The feature of a node $u \in V$.
$\mathbf{h}_u^{(t)} \in \mathbb{R}^{d^{(t)}}$	The representation of a node $u \in V$ at layer t.
$\mathbf{z}_u = \mathbf{h}_u^{(T)} \in \mathbb{R}^{d^{(T)}}$	The final representation of a node $u \in V$ after T layers/iterations.
$\mathbf{W}_x^{(t)} \in \mathbb{R}^{d^{(t+1)} \times d^{(t)}}$	Learnable parameter matrix at layer t.
MLP	A multilayer perceptron with ReLU as nonlinearity.

In this project we attempt to study the expressive power of an extension of the Graph Substructure Network (GSN) [1] by adapting it to the Dir-GNN [2], a framework for deep learning on directed graphs which can extend any MPNN and provide potential for better accuracy in heterophilic settings while having negligible impact on homophilic settings. While the GSN model provides the ability to capture internal structures of the graph, we hypothesize that it still suffers from the MPNN limitations of propagating messages along the edges of the graph, thus making the model have a hard time capturing the information in heterophilic datasets. To this end, we propose the extension model Dir-GSN, which we hypothesize to be strictly more powerful than both GSN and Dir-GNN and to provide better performance in tasks where graphs are heterophilic and nodes have structural roles. Such tasks are found in datasets related to bioinformatics, where atoms of different classes tend to have bonds between them. Since the GSN has been designed for graph classification tasks and Dir-GNN for node classification tasks, we take care to adapt the Dir-GNN framework in the context of graph classification.

We will be trying to empirically validate our hypotheses by evaluating our model in comparison with GSN on heterophilic datasets where nodes have structural meaning, namely *MUTAG*, *PROTEINS* and *NCI1* from TUDataset.

We will also measure the *effective homophily gain* of these datasets. Considering the directionality of a graph substantially increases its effective homophily in heterophilic settings, a metric is used to measure the potential homophily gained in a graph by performing several iterations of a GNN, which corresponds to message propagation over higher-order hops. We call this metric *effective homophily* of a graph and the effective homophily gain measures the difference between the effective homophily of a graph in undirected state and the graph in its directed version. As such, the higher this gain, the more likely this dataset is to benefit from considering its directionality, since higher homophily in graphs empirically tends to lead to better accuracy. Thus, we will be calculating the average effective homophily for undirected graphs h_u^{eff} and the average effective homophily for directed graphs h_d^{eff} and see how much homophily the graphs in a dataset gain by changing them from being undirected to being directed. In terms of computations, we define the *weighted node homophily* to be:

$$h(S) = \frac{1}{||V||} \sum_{i \in V} \frac{\sum_{j \in V} s_{ij} I[x_i = x_j]}{\sum_{j \in V} s_{ij}}$$

Here x_i are feature vectors $I[x_i = x_j]$ is an indicator function equal to 1 if $x_i = x_j$ or 0 otherwise. Since we will be working with datasets where feature vectors are one-hot vectors, this metric makes sense as the feature vectors act as labels.

As we increase the size of these higher-order hops, the hops tend to be more homophilic since the graph starts receiving messages from more nodes, thus increasing the probability of receiving from similar nodes. The effective homophily for a graph can thus be defined as the maximum weighted node homophily observable at any hop of the graph:

$$h^{eff} = \max_{k \geq 1} \max_{C \in B^k} h(C)$$

Here B^k denotes the set of all k-hop matrices for a graph. Because computing diffusion matrices is expensive, we only consider the case where $k = 2$ and $k = 1$. In particular, h_d^{eff} and h_u^{eff} are instances of h^{eff} where graphs are directed, respectively undirected.

The model is designed to incorporate the properties from both GSN and Dir-GNN. The t-th layer in the GSN model is given by the following equations:

$$\mathbf{h}_v^{(t)} = UP^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{m}_v^{(t-1)})$$
$$\mathbf{m}_v^{(t)} = M^{(t)}(\{\{(\mathbf{h}_v^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{x}_v^{(V)}, \mathbf{x}_u^{(V)}) : (u, v) \in E\}\})$$

Here x_u^V is the vertex structural feature vector, where each entry is the number of times that vertex acts as an orbit in a subgraph isomoprhic to one of the graphs from a fixed list of substructres. The $UP^{(t)}$ is the update function and $M^{(t)}$ is the aggregation function (an arbitrary function on multisets).

To adapt it to the Dir-GNN framework, we need to aggregate incoming and outgoing edges separately. Thus, t-th layer in the Dir-GSN model is given by:

$$\mathbf{h}_v^{(t)} = UP^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{m}_{v,\leftarrow}^{(t-1)}, \mathbf{m}_{v,\rightarrow}^{(t-1)})$$
$$\mathbf{m}_{v,\leftarrow}^{(t)} = M_{\leftarrow}^{(t)}(\{\{(\mathbf{h}_v^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{x}_v^{(V)}, \mathbf{x}_u^{(V)}) : (u, v) \in E)\}\})$$
$$\mathbf{m}_{v,\rightarrow}^{(t)} = M_{\rightarrow}^{(t)}(\{\{(\mathbf{h}_v^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{x}_v^{(V)}, \mathbf{x}_u^{(V)}) : (v, u) \in E)\}\})$$

Here, $M_{\leftarrow}^{(t)}$ and $M_{\rightarrow}^{(t)}$ are the two separate in- and out- aggregators.

We have evaluated the GSN model and Dir-GSN model with k-fold cross validation using 10 folds. The setup, as well as the GSN model, are taken from **https://github.com/gbouritsas/GSN/**, where we only changed the message passing functions to consider directionality. We have also provided homophily metric functions in the *./homophily* folder. We are using the hyperparameters proposed in the GSN repository used for training the TUDataset data in their paper. The commonly used hyperparameters in all 3 trainings are: a batch size of 32, training for 128 epochs and 50 iterations each epoch, learning rate of 0.001, a decay rate of 0.5 and decay steps of 50. The number of GNN layers is 4, the number of MLP layers is 2, the size of hidden layers in internal MLPs is 64 and ReLU is used as the activation function. Additionally, all automorphisms and orbits calculated as part of precomputing all subgraph isomorphisms (used for calculating vertices structural feature vectors) are directed. When it comes to *MUTAG*, *PROTEINS* and *NCI1*, the substructures used are cycles of length 12, cliques of size 4 and triangles, respectively.

The GSN model used in training is given by the equations:

$$\mathbf{h}_v^{(t)} = \mathsf{MLP}^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{m}_v^{(t-1)})$$
$$\mathbf{m}_v^{(t)} = \sum_{u \in N(v)} \mathsf{MLP}^{(t)}(\mathbf{h}_v^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{x}_v^{(V)}, \mathbf{x}_u^{(V)})$$

Additionally, the Dir-GSN model used in training is given by:

$$\mathbf{h}_v^{(t)} = \mathsf{MLP}^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{m}_{v,\leftarrow}^{(t-1)}, \mathbf{m}_{v,\rightarrow}^{(t-1)})$$
$$\mathbf{m}_{v,\leftarrow}^{(t)} = \sum_{u : v \in N(u)} \mathsf{MLP}^{(t)}(\mathbf{h}_v^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{x}_v^{(V)}, \mathbf{x}_u^{(V)})$$
$$\mathbf{m}_{v,\rightarrow}^{(t)} = \sum_{u \in N(v)} \mathsf{MLP}^{(t)}(\mathbf{h}_v^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{x}_v^{(V)}, \mathbf{x}_u^{(V)})$$

The results of running the above models over each dataset is given by the following table:

Model	PROTEINS	MUTAG	NCI1
GSN	71.62% +/- 2.32%	90% +/- 7.78%	81.24% +/- 2.31%
Dir-GSN	72.45% +/- 7.23%	91.2% +/- 6.94%	81.09% +/- 1.63%

Here the performance is given by the epoch with the best average accuracy across the 10 folds.

We additionally computed the average effective homophily for each datasets:

Dataset	h_{gain}^{eff}	h_d^{eff}	h_u^{eff}
PROTEINS	2%	0.7271	0.7123
MUTAG	0%	0.8077	0.8077
NCI1	0.005%	0.7746	0.7746

The tests show that Dir-GSN is only slightly better than GSN. Homophily functions show that the average gain in these datasets is small which indicates that these datasets have not much to gain from considering directionality. This probably is what justifies Dir-GSN not having made a huge improvement for these datasets.

To conclude, while Dir-GSN hasn't shown a significant change in accuracy on these datasets, it remains to see whether it can perform better in datasets where the effective homophilic gain is larger. Additionally, we made the assumption that GSN isn't good enough in heterophilic settings, which leads to the question of whether substructures could actually provide homophily. Lastly, there is still need for a theoretical analysis on the expressive power of Dir-GSN in comparison to GSN and Dir-GNN.

References

[1] Giorgos Bouritsas et al. "Improving graph neural network expressivity via subgraph isomorphism counting". In: *arXiv preprint arXiv:2006.09252* (2020).

[2] Emanuele Rossi et al. *Edge Directionality Improves Learning on Heterophilic Graphs.* 2023. arXiv: `2305.10498 [cs.LG]`.