Attraction/Repulsion Dynamics versus Homophily in Graph Neural Networks

Nikolas Kirschstein, University of Oxford

Abstract

Most established Graph Neural Network architectures rely on the input graphs to be homophilic. On heterophilic graphs, they often perform worse than structure-agnostic models. The recently proposed Gradient Flow Framework (GraFF) generalizes many architectures as discretizations of gradient flow differential equations. Governed by the spectrum of one of its channel-mixing matrices, GraFF allows for both attractive and repulsive interactions between node representations in the latent space of a GNN. In this work, we examine the natural relationship between this attraction/repulsion behavior and the homophily ratio of the input graph. Our findings suggest that this relationship is much weaker than expected.

1. Introduction

Due to their inductive relational bias, Graph Neural Networks (GNNs) are successful in a variety of node classification tasks (Zitnik et al., 2018; Ying et al., 2018; Dou et al., 2020). However, the underlying graph in these applications is usually homophilic, i.e., adjacent nodes are very likely to share a class label. Dealing with the heterophilic case where predominantly nodes of unequal classes connect remains challenging. Often in these settings, a simple multilayer perceptron that ignores the graph topology altogether outperforms GNNs.

According to some researchers, the crux of the matter consists in GNNs implicitly assuming homophily as they typically consider a notion of neighborhood that includes only the node itself and its direct neighbors. Consequently, there is a line of research suggesting extensions, modifications, and tweaks to GNN architectures: Zhu et al. (2020) propose three design techniques (ego- and neighbor-embedding separation, higher-order neighborhoods, combination of intermediate representations) to explicitly address heterophily, leading to the H_2GCN architecture. Zhu et al. (2021) propose to incorporate a class compatibility matrix that explicitly describes the pair-wise connectivity between classes, leading to the CPNN framework. Finally, Chien et al. (2021) propose to adaptively learn Generalized PageRank weights, leading to the GPR-GNN architecture.

Contrary to the specialized architectures targeting heterophily with ad-hoc techniques, Di Giovanni et al. (2022) devise a more principled approach called Gradient Flow Framework (GraFF). It is a physics-inspired generalization of vanilla Graph Convolutional Networks (GCNs). Starting from the interpretation of residual neural networks as discrete dynamical systems, they derive several existing architectures as special cases of the discretization of the gradient flow of a specific energy functional. They find that the framework manages to adapt to both homophilic and heterophilic settings via the spectrum of one of its channel-mixing matrices. Its positive and negative eigenvalues correspond to attractive and repulsive forces, respectively, between adjacent node representations in latent space. Hence, GraFF models are able to align representations in certain subspaces while separating them in other subspaces, supposedly enabling them to deal with any degree of homophily.

In this work, we investigate how the spectrum of the relevant channel-mixing matrix and, thus, the attraction/repulsion dynamics behave under varying homophily of the input graph. We hypothesize that for strongly homophilic graphs, the spectrum will be skewed toward positive and for heterophilic graphs toward negative eigenvalues. To this end, we run empirical experiments on both synthetic and real-world datasets with varying homophily (Abu-El-Haija et al., 2019; Platonov et al., 2023). Section 2 provides the required mathematical background, Section 3 establishes the connection between attraction/repulsion and homophily, and Section 4 discusses our empirical results.

2. Theoretical Background

We consider semi-supervised node classification on a connected, unweighted input graph $G = (V, E)$ with $n = |V|$ nodes, $m = |E|$ edges, feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ consisting of rows $\boldsymbol{x}_i^\top \in \mathbb{R}^d$, class label vector $\boldsymbol{y} \in \{1, \dots, c\}^n$ and symmetrically normalized adjacency matrix $\hat{\mathbf{A}} \in \mathbb{R}^{n \times n}$. We denote by $d_{i,\text{in}}$ and $d_{i,\text{out}}$ the in-degree and out-degree of node $i \in V$, respectively, and define the balanced degree $d_i \coloneqq \frac{1}{2}(d_{i,\text{in}} + d_{i,\text{out}})$ which due to the connectedness assumption is strictly positive. For an arbitrary matrix $\mathbf{M}$, we define its symmetrization as $\mathbf{M}^{\text{sym}} \coloneqq \frac{1}{2}(\mathbf{M} + \mathbf{M}^\top)$. If $\mathbf{M}$ is positive semi-definite, the weighted inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbf{M}} \coloneqq \boldsymbol{x}^\top \mathbf{M} \boldsymbol{y}$ and its induced weighted norm

$\|\boldsymbol{x}\|_{\mathbf{M}} \coloneqq \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathbf{A}}}$ are well-defined.

2.1. Measuring Homophily

While homophily originally is a concept from chemistry, we are concerned with its analogon in network analysis. There, it describes the degree to which adjacent vertices in a graph are similar to each other (McPherson et al., 2001). In the case of a classification task, the most common quantification is the *homophily ratio* (Zhu et al., 2020; Ma et al., 2022), the fraction of edges that connect nodes of the same class:

$$\text{homophily}(G, \boldsymbol{y}) = \frac{|\{(i,j) \in E \mid y_u = y_v\}|}{|E|} \tag{1}$$

This definition provides a scalar characterization of homophily but to do so necessarily averages effects across classes. If there are more than two classes, some pairs might exhibit strong homophily while others exhibit strong heterophily. For these cases, more fine-grained definitions exist that capture pair-wise class interactions, like the class-compatibility matrix (Zhu et al., 2020). Favoring a rough characterization that lives on a linear scale, we stick with Equation (1) in this study.

2.2. GNNs as Dynamical Systems

A residual GNN[1] can be viewed as simulating a discretized ordinary differential equation (ODE) (Chen et al., 2018). Let $\text{GNN}_\theta : \mathbb{R}^d \to \mathbb{R}^d$ be a GNN layer with parameters θ. Now consider a first-order ODE of the form

$$\dot{\mathbf{X}}(t) = \text{GNN}_\theta(\mathbf{X}(t)) \tag{2}$$

evolving the initial state $\mathbf{X}(0) = \mathbf{X}$ for $t \geq 0$. To discretize this, we choose a *step size* $\tau > 0$ and construct the Taylor expansion around $\mathbf{X}(t)$:

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \tau\, \text{GNN}_\theta(\mathbf{X}(t)) + \mathcal{O}(\tau^2).$$

By dropping all higher-order terms, we obtain the forward Euler method, which is the most fundamental numerical algorithm for approximately solving ODEs like Equation (2). It starts at $\mathbf{X}^{(0)} \coloneqq \mathbf{X}$ and iteratively updates

$$\mathbf{X}^{(k+1)} \coloneqq \mathbf{X}^{(k)} + \tau\, \text{GNN}_\theta(\mathbf{X}^{(k)}), \tag{3}$$

for $k = 0, \ldots, N-1$. This is reminiscent of an N-layer residual neural network with the parameters θ *shared* across layers. If the step size τ is sufficiently small, we have $\mathbf{X}^{(k)} \approx \mathbf{X}(k\tau)$. The ODE dynamics can be recovered for $\tau \to 0$ as the continuous-time limit of the GNN.

2.3. Restricting to Gradient Flows

The ODE (2) is called a *gradient flow* if there exists an energy functional $\mathcal{E}_\theta : \mathbb{R}^n \to \mathbb{R}$ such that the right-hand side is its negative gradient:

$$\text{GNN}_\theta(\mathbf{X}(t)) = -\nabla \mathcal{E}_\theta(\mathbf{X}(t)). \tag{4}$$

By construction, the energy $\mathcal{E}_\theta$ decreases monotonically along a solution since

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \mathcal{E}_\theta(\mathbf{X}(t)) &= \langle -\nabla \mathcal{E}_\theta(\mathbf{X}(t)), \dot{\mathbf{X}}(t) \rangle \\ &= -\|\nabla \mathcal{E}_\theta(\mathbf{X}(t))\|^2 \\ &\leq 0. \end{aligned} \tag{5}$$

and the corresponding forward Euler step simulation algorithm obtained by plugging (4) into (3) is gradient descent:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - \tau \nabla \mathcal{E}_\theta(\mathbf{X}^{(k)}) \tag{6}$$

Due to the energy minimization perspective, gradient flow ODEs are ubiquitous in physics. As they provide a high amount of interpretability, Di Giovanni et al. exclusively consider gradient flow discretizations. The only remaining design choice then is the definition of $\mathcal{E}_\theta$.

2.4. The GraFF Energy

Consider the node features as particles in $\mathbb{R}^d$ with energy $\mathcal{E}_\theta$. Generalizing the well-known Dirichlet energy stemming from the heat diffusion equation, Di Giovanni et al. propose the following generic quadratic energy parameterized by $\theta = \{\mathbf{W}, \boldsymbol{\Omega}\} \subseteq \mathbb{R}^{d \times d}$:

$$\mathcal{E}_\theta(\mathbf{X}) \coloneqq \underbrace{\tfrac{1}{2} \sum_{i \in V} \|\boldsymbol{x}_i\|^2_{\boldsymbol{\Omega}}}_{\text{external energy}} - \underbrace{\tfrac{1}{2} \sum_{(i,j) \in E} \hat{A}_{ij} \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle_{\mathbf{W}}}_{\text{internal energy}}. \tag{7}$$

The first term is independent of $\hat{\mathbf{A}}$ and describes an external energy component, while the second term accounts for pair-wise interactions. By differentiation, we can derive the corresponding gradient flow equation which is

$$\dot{\mathbf{X}}(t) = -\nabla \mathcal{E}_\theta(\mathbf{X}(t)) = \hat{\mathbf{A}} \mathbf{X}(t) \mathbf{W}^{\text{sym}} - \mathbf{X}(t) \boldsymbol{\Omega}^{\text{sym}}. \tag{8}$$

Note that the parameter matrices $\mathbf{W}, \boldsymbol{\Omega}$ appear in their symmetrized form. This can be seen by vectorizing the equation and using that the Hessian of $\mathcal{E}_\theta$ must be symmetric (Di Giovanni et al., Appendix B). Without loss of generality, we thus drop the superscript and, from now on, assume $\mathbf{W}, \boldsymbol{\Omega}$ to be symmetric in the first place. The resulting GraFF-NN definition by plugging (8) into (3) reads:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \tau(\hat{\mathbf{A}} \mathbf{X}^{(k)} \mathbf{W} - \mathbf{X}^{(k)} \boldsymbol{\Omega}). \tag{9}$$

We call $\mathbf{W}$ the *internal* and $\boldsymbol{\Omega}$ the *external* channel-mixing matrix, respectively. By setting $\tau = 1$ and $\boldsymbol{\Omega} = \mathbf{0}$, we obtain a vanilla GCN with residual connection, shared symmetric weights, and without non-linear activation. However,

[1] In fact, any residual neural network

Di Giovanni et al. prove that $\mathcal{E}_\theta$ also decreases along solutions of an "upgraded" version of Equation (8) that includes a non-linearity. Other inductive biases, i.e., restrictions to the channel-mixing matrices, yield further established architectures as special cases.

3. Attraction and Homophily

By virtue of Equation (9), we have a GNN at hand that simulates the gradient flow of the energy functional $\mathcal{E}_\theta$ defined in Equation (7). We can use this connection to gain more insight into the dynamics of such a GNN. To this end, we require a notion of derivatives on graphs: the *graph gradient* is the edge signal $\nabla\mathbf{X} \in \mathbb{R}^{m\times d}$ with rows given by

$$\nabla \boldsymbol{x}_{(i,j)} \coloneqq \tfrac{1}{\sqrt{d_j}}\boldsymbol{x}_j - \tfrac{1}{\sqrt{d_i}}\boldsymbol{x}_i$$

which compares the normalized features of the endpoints of each edge. To make a connection with Equation (7), consider the summation

$$\begin{aligned}\tfrac{1}{2}\sum_{(i,j)\in E}\|\nabla \boldsymbol{x}_{(i,j)}\|^2_{\mathbf{W}} &= \tfrac{1}{2}\sum_{(i,j)\in E}\|\tfrac{1}{\sqrt{d_j}}\boldsymbol{x}_j - \tfrac{1}{\sqrt{d_i}}\boldsymbol{x}_i\|^2_{\mathbf{W}}\\ &= \tfrac{1}{2}\sum_{(i,j)\in E}\Big(\tfrac{1}{d_i}\|\boldsymbol{x}_i\|^2_{\mathbf{W}} + \tfrac{1}{d_j}\|\boldsymbol{x}_j\|^2_{\mathbf{W}} - \tfrac{2}{\sqrt{d_i d_j}}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle_{\mathbf{W}}\Big)\\ &= \sum_{i\in V}\|\boldsymbol{x}_i\|^2_{\mathbf{W}} - \sum_{(i,j)\in E}\hat{A}_{ij}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle_{\mathbf{W}} \qquad (10)\end{aligned}$$

using that

$$\begin{aligned}&\tfrac{1}{2}\sum_{(i,j)\in E}\tfrac{1}{d_i}\|\boldsymbol{x}_i\|^2_{\mathbf{W}} + \tfrac{1}{2}\sum_{(j,i)\in E}\tfrac{1}{d_i}\|\boldsymbol{x}_i\|^2_{\mathbf{W}}\\ &= \tfrac{1}{2}\sum_{i\in V}\tfrac{1}{d_i}\|\boldsymbol{x}_i\|^2_{\mathbf{W}}\underbrace{\sum_{j:(i,j)\in E}1}_{d_{i,\text{out}}} + \tfrac{1}{2}\sum_{i\in V}\tfrac{1}{d_i}\|\boldsymbol{x}_i\|^2_{\mathbf{W}}\underbrace{\sum_{j:(j,i)\in E}1}_{d_{i,\text{in}}}\\ &= \sum_{i\in V}\tfrac{1}{d_i}\tfrac{d_{i,\text{in}}+d_{i,\text{out}}}{2}\|\boldsymbol{x}_i\|^2_{\mathbf{W}} = \sum_{i\in V}\|\boldsymbol{x}_i\|^2_{\mathbf{W}}.\end{aligned}$$

Using the trivial property $\|\cdot\|_{\boldsymbol{\Omega}} = \|\cdot\|_{\boldsymbol{\Omega}-\mathbf{W}} + \|\cdot\|_{\mathbf{W}}$ and the observation (10) yields the following decomposition of the energy functional:

$$\begin{aligned}2\mathcal{E}_\theta(\mathbf{X}) &= \sum_{i\in V}\|\boldsymbol{x}_i\|^2_{\boldsymbol{\Omega}} - \sum_{(i,j)\in E}\hat{A}_{ij}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle_{\mathbf{W}}\\ &= \sum_{i\in V}\|\boldsymbol{x}_i\|^2_{\boldsymbol{\Omega}-\mathbf{W}} + \sum_{i\in V}\|\boldsymbol{x}_i\|^2_{\mathbf{W}}\\ &\quad - \sum_{(i,j)\in E}\hat{A}_{ij}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle_{\mathbf{W}}\\ &= \sum_{i\in V}\|\boldsymbol{x}_i\|^2_{\boldsymbol{\Omega}-\mathbf{W}} + \tfrac{1}{2}\sum_{(i,j)\in E}\|\nabla \boldsymbol{x}_{(i,j)}\|^2_{\mathbf{W}} \qquad (11)\end{aligned}$$

To analyze the second term further, consider the eigenvalue decomposition of the (symmetric!) internal channel mixing matrix $\mathbf{W} = \mathbf{T}\boldsymbol{\Lambda}\mathbf{T}^\top$. Split $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_+ - \boldsymbol{\Lambda}_-$ into the positive part $\boldsymbol{\Lambda}_+ \coloneqq \max\{\mathbf{0}, \boldsymbol{\Lambda}\}$ and the negative part $\boldsymbol{\Lambda}_+ \coloneqq \max\{\mathbf{0}, -\boldsymbol{\Lambda}\}$ of the spectrum to obtain

$$\mathbf{W} = \underbrace{\mathbf{T}\boldsymbol{\Lambda}_+\mathbf{T}^\top}_{=:\mathbf{W}_+} - \underbrace{\mathbf{T}\boldsymbol{\Lambda}_-\mathbf{T}^\top}_{=:\mathbf{W}_-}.$$

By design, $\mathbf{W}_+$ and $\mathbf{W}_-$ are symmetric positive definite. Hence, their Cholesky decompositions $\mathbf{W}_+ = \boldsymbol{\Theta}_+^\top\boldsymbol{\Theta}_+$ and $\mathbf{W}_- = \boldsymbol{\Theta}_-^\top\boldsymbol{\Theta}_-$ are well-defined. We use them to partition the gradient norm:

$$\begin{aligned}\|\nabla \boldsymbol{x}_{(i,j)}\|^2_{\mathbf{W}} &= \|\nabla \boldsymbol{x}_{(i,j)}\|^2_{\mathbf{W}_+} - \|\nabla \boldsymbol{x}_{(i,j)}\|^2_{\mathbf{W}_-}\\ &= \|\boldsymbol{\Theta}_+\nabla \boldsymbol{x}_{(i,j)}\|^2 - \|\boldsymbol{\Theta}_-\nabla \boldsymbol{x}_{(i,j)}\|^2\end{aligned}$$

Plugging this back into Equation (11) gives us a final, interpretable expression for the energy $\mathcal{E}_\theta$:

$$\begin{aligned}\mathcal{E}_\theta(\mathbf{X}) = \tfrac{1}{2}\sum_{i\in V}\|\boldsymbol{x}_i\|^2_{\boldsymbol{\Omega}-\mathbf{W}}\\ + \tfrac{1}{4}\underbrace{\sum_{(i,j)\in E}\|\boldsymbol{\Theta}_+\nabla \boldsymbol{x}_{(i,j)}\|^2}_{\text{attraction}}\\ - \tfrac{1}{4}\underbrace{\sum_{(i,j)\in E}\|\boldsymbol{\Theta}_-\nabla \boldsymbol{x}_{(i,j)}\|^2}_{\text{repulsion}}\end{aligned} \qquad (12)$$

Recall that by Equation (5), the energy $\mathcal{E}_\theta$ decreases along a solution of Equation (8), i.e., with deeper layers of the GNN specified by Equation (9). Hence, the terms $\|\boldsymbol{\Theta}_+\nabla \boldsymbol{x}_{(i,j)}\|^2$ are minimized over time, which means adjacent node representations become aligned in the non-null singular subspaces of $\boldsymbol{\Theta}_+$. This behavior can be interpreted as *attractive* forces that lead to a smoothing effect. Analogously, in the orthogonal complement, adjacent representations are pushed apart by *repulsive* forces stemming from the maximization leading to a sharpening effect due to the eventual maximization of $\|\boldsymbol{\Theta}_-\nabla \boldsymbol{x}_{(i,j)}\|^2$.

To prevent confusion, we remark that there are two unrelated minimization processes going on. One is the minimization of $\mathcal{E}_\theta$ by the evolution of the gradient flow ODE Equation (8) during inference, which leads to the attraction and repulsion dynamics discussed above. The other is the minimization of some empirical risk function to optimize the parameters θ of the GNN during training, which is how $\boldsymbol{\Theta}_+$ and $\boldsymbol{\Theta}_-$ are obtained in the first place.

Di Giovanni et al. claim that attractive dynamics in the latent space of GNNs benefit classification on homophilic graphs while repulsive behavior is advantageous in the heterophilic case. Intuitively, this seems reasonable: if adjacent features have similar representations, they should be harder to discriminate, which is helpful on homophilic graphs but detrimental on heterophilic graphs. On the other hand, if

features are very discontinuous along edges, then it should be easier for a classifier to tell them apart on heterophilic graphs, but in turn, on homophilic graphs the "neighbors-are-similar-to-me" information is lost.

4. Experiments

In this section, we empirically investigate the connection between attraction/repulsion dynamics in latent space and graph homophily that the theory of the Gradient Flow Framework suggests (cp. Section 3). Recalling that the internal channel-mixing matrix $\mathbf{W}$ encodes attractive and repulsive behavior via its positive and negative eigenspaces, respectively, we pose the following

Hypothesis. *Based on the attraction-homophily connection, we conjecture that the internal channel-mixing matrix* $\mathbf{W}$ *should exhibit more positive eigenvalues on graphs with higher homophily and more negative eigenvalues on graphs with lower homophily. More precisely:*

1. *The number of positive eigenvalues of* $\mathbf{W}$ *should be proportional to the homophily ratio of the input graph:*

$$\operatorname{rank}(\mathbf{\Lambda}_+) \propto \operatorname{homophily}(G, \boldsymbol{y}).$$

2. *The number of negative eigenvalues of* $\mathbf{W}$ *should be proportional to the heterophily ratio of the input graph:*

$$\operatorname{rank}(\mathbf{\Lambda}_-) \propto 1 - \operatorname{homophily}(G, \boldsymbol{y}).$$

4.1. Setup

To test this hypothesis, we conduct experiments on the MixHop synthetic dataset (Abu-El-Haija et al., 2019) and the heterophilic graphs collected by Platonov et al. (2023). Our source code uses the PyTorch Geometric Python library (Fey & Lenssen, 2019) and is publicly available[2].

For all experiments, we enclose the GraFF-NN with a two-layer MLP encoder to project the input features into a latent space with fixed dimensionality, and append a two-layer MLP decoder to produce final classification logits. All parameters are initialized using the conventional Glorot initialization scheme (Glorot & Bengio, 2010). We optimize the standard cross-entropy loss with the Adaptive Moments (Adam) SGD variant (Kingma & Ba, 2015). Table 1 details the relevant hyperparameters we use throughout all experiments. As a baseline, we also train a vanilla GCN with the same hyperparameters.

4.2. Synthetic Graphs

The MixHop synthetic dataset contains 10 graphs with homophily ratios varying uniformly between 0 and 0.9. Each graph has 5000 nodes that each belong to one of ten classes. The node features are sampled from a 2D normal distribution per class. Table 2 provides essential dataset analytics.

[2] https://anonymous.4open.science/r/GDL/Attraction_vs_Homophily.ipynb

HYPERPARAMETER	VALUE
NETWORK DEPTH (N)	10
LATENT SPACE DIM (d)	64
STEP SIZE (τ)	1
# EPOCHS	100
LEARNING RATE	0.01

Table 1. Hyperparameter choices for our experiments.

#NODES	#EDGES	#FEATURES	#CLASSES	HOMOPHILY
5000	59,596	2	10	0 TO 0.9

Table 2. Basic facts about the MixHop synthetic dataset.

As a sanity check, Figure 1 compares the accuracies of GraFF-NN and GCN on the validation and test split with increasing homophily ratio. Matching the observations by Di Giovanni et al., GraFF-NN performs better in the heterophilic regime but lags slightly behind GCN's performance in the homophilic regime.

To provide insight into the eigenvalues of $\mathbf{W}$, Table 4 lists their summary statistics and Figure 2 visualizes their distributions as boxplots for different homophily levels. Surprisingly, the distributions are irrefutably similar: centered around a value close to zero, with standard deviations close to one, and a balanced division into positive and negative eigenvalues. We consistently observe equivalent results for various choices of the hyperparameters in Table 1 and regardless of whether or not we apply a non-linearity (ReLU) after each layer.

4.3. Real-world Graphs

To rule out the possibility that the results obtained are specific to the synthetic dataset, we run the same evaluations on a real-world dataset of five heterophilic graphs from Platonov et al.. Their analytics are given in Table 3.

GRAPH	#NODES	#EDGES	#FEATURES	#CLASSES	HOMOPHILY
ROMAN-EMPIRE	22,662	32,927	300	18	0.05
AMAZON-RATINGS	24,492	93,050	300	5	0.38
TOLOKERS	11,758	519,000	10	2	0.59
MINESWEEPER	10,000	39,402	7	2	0.68
QUESTIONS	48,921	153,540	301	2	0.84

Table 3. Basic facts about the real-world datasets.

However, Figure 3, Table 5, and Figure 4 show that we receive analogous results on the real-world graphs. This suggests that our empirical findings generalize and therefore, interestingly, the Hypothesis does *not* hold. Rather,

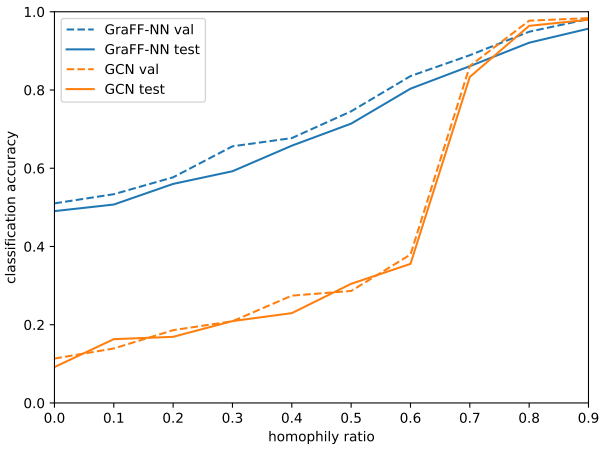

Figure 1. Accuracies of GraFF-NN and GCN on the validation and test split of *synthetic* graphs with varying homophily ratio.

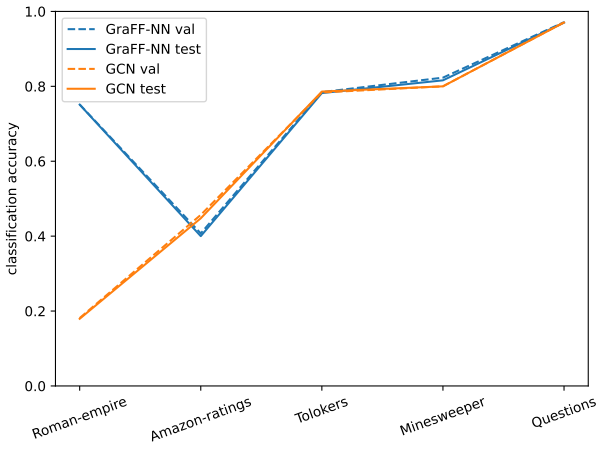

Figure 3. Accuracies of GraFF-NN and GCN on the validation and test split of *real-world* graphs with varying homophily ratio.

HOMOPHILY	MEAN	STD	MIN	MAX	#NEG	#POS
0.0	-0.086	0.969	-2.113	1.713	34	30
0.1	-0.074	1.015	-2.357	1.792	33	31
0.2	-0.096	0.989	-1.958	1.658	34	30
0.3	-0.099	1.004	-1.980	1.786	34	30
0.4	-0.056	0.998	-2.093	1.700	33	31
0.5	-0.076	1.008	-2.228	1.823	34	30
0.6	-0.090	0.998	-1.966	1.785	35	29
0.7	-0.097	0.983	-2.114	1.646	35	29
0.8	-0.076	0.999	-2.197	1.781	34	30
0.9	-0.055	0.995	-1.986	1.812	33	31

Table 4. Eigenvalue statistics of $\mathbf{W}$ when trained on *synthetic* graphs with varying homophily ratio.

GRAPH (HOMOPHILY)	MEAN	STD	MIN	MAX	#NEG	#POS
ROMAN-EMPIRE (0.05)	-0.087	0.974	-2.023	1.740	33	31
AMAZON-RATINGS (0.38)	-0.096	0.967	-1.892	1.745	34	30
TOLOKERS (0.59)	-0.049	0.990	-2.000	1.645	32	32
MINESWEEPER (0.68)	-0.095	0.976	-2.003	1.670	33	31
QUESTIONS (0.84)	-0.030	0.995	-1.908	1.756	32	32

Table 5. Eigenvalue statistics of $\mathbf{W}$ when trained on *real-world* graphs with varying homophily ratio.

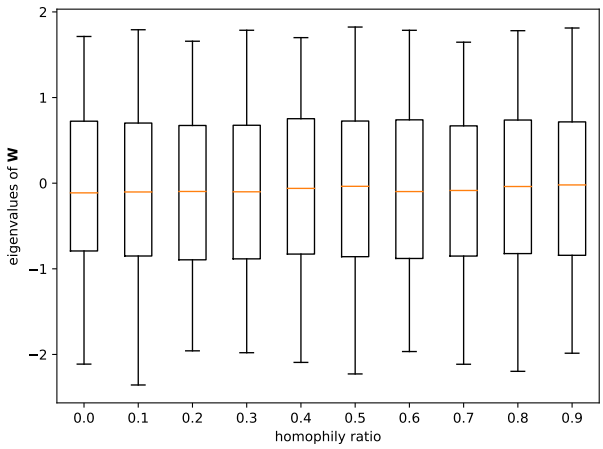

Figure 2. Boxplot of the eigenvalue distributions of $\mathbf{W}$ when trained on *synthetic* graphs with varying homophily ratio.

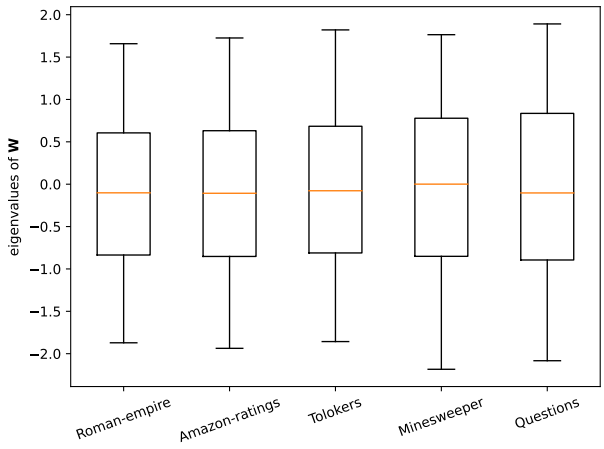

Figure 4. Boxplot of the eigenvalue distributions of $\mathbf{W}$ when trained on *real-world* graphs with varying homophily ratio.

the results show a slight preference for the negative side of the spectrum regardless of the homophily ratio. Future work should investigate whether this bias is statistically significant. Furthermore, the blatant closeness of the eigenvalue distributions to being normalized raises the question if the impact of the initalization of $\mathbf{W}$ might outweigh the attraction/repulsion dynamics.

5. Conclusion

In this work, we investigated the relationship between homophily and attractive/repulsive forces in the latent space of GNNs as described by the Gradient Flow Framework (GraFF). We first revisited the derivation of GraFF from the interpretation of neural networks as discretized ordinary differential equations. Based on the attraction-homophily connection, we then hypothesized that the number of positive (negative) eigenvalues of the internal channel-mixing matrix should be proportional to the homophily (heterophily) ratio. However, our experiments on both synthetic and real-world graphs suggest that this hypothesis is false. If anything, the eigenvalue distribution seems to be stubbornly agnostic of the graph homophily ratio, which is a surprising result.

However, there is a caveat to our study: First, we merely considered the simplest notion of homophily, the edge homophily ratio, and only conducted a small number of experiments with the same model architecture on just two datasets due to severe computational limitations. Future work should explore large-scale cross-validated analyses on a myriad of datasets with various homophily measures and model configurations. In addition, a statistical significance test is needed for the slight overall bias towards negative eigenvalues that the experiments reveal. Finally, the proximity of the eigenvalue distribution to the initialized state calls for future endeavors to compare the effect of different parameter initialization schemes.

References

Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Ver Steeg, G., and Galstyan, A. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pp. 21–29. PMLR, 2019.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural Ordinary Differential Equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*, 2021.

Di Giovanni, F., Rowbottom, J., Chamberlain, B. P., Markovich, T., and Bronstein, M. M. Graph Neural Networks as Gradient Flows: understanding graph convolutions via energy. 2022. Publisher: arXiv Version Number: 3.

Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., and Yu, P. S. Enhancing Graph Neural Network-Based Fraud Detectors against Camouflaged Fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pp. 315–324, New York, NY, USA, 2020. Association for Computing Machinery. event-place: Virtual Event, Ireland.

Fey, M. and Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds*, 2019. event-place: New Orleans, USA.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, May 2010. PMLR.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Ma, Y., Liu, X., Shah, N., and Tang, J. Is Homophily a Necessity for Graph Neural Networks? In *International Conference on Learning Representations*, 2022.

McPherson, M., Smith-Lovin, L., and Cook, J. M. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 2001. _eprint: https://doi.org/10.1146/annurev.soc.27.1.415.

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and Prokhorenkova, L. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*, 2023.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 974–983, New York, NY, USA, 2018. Association for Computing Machinery. event-place: London, United Kingdom.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7793–7804. Curran Associates, Inc., 2020.

Zhu, J., Rossi, R. A., Rao, A., Mai, T., Lipka, N., Ahmed, N. K., and Koutra, D. Graph Neural Networks with Heterophily. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35 (12):11168–11176, May 2021.

Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, June 2018. _eprint: https://academic.oup.com/bioinformatics/article-pdf/34/13/i457/25098429/bty294.pdf.