A Code Repository and Licensing

The code written for this research work is available at `https://github.com/Stefa1994/GeDi-HNN` and freely distributed under the Apache 2.0 license.[6]

The `Texas`, `Wisconsin`, `Cornell`, `WikiCS`, and `Telegram` datasets were obtained from the PyTorch Geometric Signed Directed (He et al., 2022b) library (distributed under the MIT license). The `Cora`, `Citeseer`, and `PubMed` datasets are available at `https://linqs.org/datasets/`. The `email-Eu` and `email-Enron` datasets are available at `https://www.cs.cornell.edu/~arb/data/`.

The code for the baselines used in the experimental analysis is available at `https://github.com/Graph-COM/ED-HNN` and `https://github.com/yxzwang/PhenomNN` under the MIT license.[7]

B Properties of Our Proposed Laplacian $\vec{L}_N$

This section contains the proofs of the theorems, corollaries, propositions, and lemma reported in the main paper.

Theorem 1. *If $\mathcal{H}$ is an undirected hypergraph, $\vec{L}_N = \Delta$ and $\vec{Q}_N = Q_N$.*

Proof. Since $\mathcal{H} = (V, E)$ is an undirected hypergraph, $\vec{B}$ is binary and only takes values 0 and 1 (rather than being ternary and taking values $0, 1, -\mathrm{i}$, which is the case in general). In particular, for each edge $e \in E$ we have $\vec{B}_{ue} = 1$ if either $u \in H(e)$ or $u \in T(e)$ and $\vec{B}_{ue} = 0$ otherwise. Consequently, the directed incidence matrix $\vec{B}$ is identical to the non-directed incidence matrix B, i.e., $\vec{B} = B$. Thus, by construction, $\vec{L}_N = \Delta$ and $\vec{Q}_N = Q_N$. □

Corollary 1. *If $\mathcal{H}$ is an undirected 2-uniform hypergraph, $\vec{L}_N = \frac{1}{2} L_N$ and $\vec{Q}_N = \frac{1}{2} Q_N$.*

Proof. Since $\mathcal{H}$ is an undirected 2-uniform hypergraph, it follows that:

$$\begin{cases} \vec{B} W \vec{B}^* & = D_v + A \\ D_e^{-1} & = \frac{1}{2} I \end{cases}$$

Based on this, we can rewrite $\vec{Q}_N$ as follows:

$$\begin{aligned}
\vec{Q}_N &= D_v^{-\frac{1}{2}} \vec{B} W D_e^{-1} \vec{B}^* D_v^{-\frac{1}{2}} \\
&= D_v^{-\frac{1}{2}} \vec{B} \left(\frac{1}{2} W \right) \vec{B}^* D_v^{-\frac{1}{2}} \\
&= \frac{1}{2} \left(D_v^{-\frac{1}{2}} \left(D_v + A \right) D_v^{-\frac{1}{2}} \right) \\
&= \frac{1}{2} \left(I + D_v^{-\frac{1}{2}} A D_v^{-\frac{1}{2}} \right) \\
&= \frac{1}{2} \left(I + A_N \right) \\
&= \frac{1}{2} Q_N.
\end{aligned}$$

This proves the second part of the result. Since $\vec{Q}_N = \frac{1}{2} Q_N$ and, due to equation 5, $\frac{1}{2} L_N = I - \frac{1}{2} Q_N$, it follows that $\frac{1}{2} L_N = I - \vec{Q}_N = \vec{L}_N$. □

Theorem 2. *If $\mathcal{H}$ is a directed 2-uniform hypergraph with no antiparallel edges, we have $\vec{L}_N = \frac{1}{2} L_N^{\sigma}$ with $A_s = A + A^\top$.*

[6] `https://www.apache.org/licenses/LICENSE-2.0`
[7] `https://choosealicense.com/licenses/mit/`

Proof. Since $\mathcal{H}$ is a directed 2-uniform hypergraph without antiparallel edges, it follows that:

$$\begin{cases} \vec{B}W\vec{B}^* & = \bar{D}_s + H^\sigma \\ D_e^{-1} & = \frac{1}{2}I. \end{cases}$$

Since $\mathcal{H}$ has no digons, the assumption $A_s = A + A^\top$ implies $\bar{D}_s = D_v$. Thus, we can rewrite $\vec{L}_N$ as follows:

$$\begin{aligned} \vec{L}_N &= I - D_v^{-\frac{1}{2}}\vec{B}WD_e^{-1}\vec{B}^*D_v^{-\frac{1}{2}} \\ &= I - D_v^{-\frac{1}{2}}\vec{B}\left(\frac{1}{2}W\right)\vec{B}^*D_v^{-\frac{1}{2}} \\ &= I - \frac{1}{2}\left(D_v^{-\frac{1}{2}}\left(\bar{D}_s + H^\sigma\right)D_v^{-\frac{1}{2}}\right) \\ &= I - \frac{1}{2}\left(I + D_v^{-\frac{1}{2}}H^\sigma D_v^{-\frac{1}{2}}\right) \\ &= \frac{1}{2}L_N^\sigma. \end{aligned}$$

□

Corollary 2. *If $\mathcal{H}$ is a directed 2-uniform unweighted hypergraph with no antiparallel edges, we have $\vec{L}_N = \frac{1}{2}L_N^{(q)}$ with $q = \frac{1}{4}$ and $A_s = A + A^\top$.*

Proof. Since $\mathcal{H}$ is a directed 2-uniform unweighted hypergraph, $A \in \{0,1\}^{n\times n}$. Thus, as shown by Fiorini et al. (2023), with $q = \frac{1}{4}$ we have $L^\sigma = L^{(q)}$. Since Theorem 2 states that $\vec{L}_N = \frac{1}{2}L_N^\sigma$, it follows that

$$\vec{L}_N = \frac{1}{2}L_N^\sigma = \frac{1}{2}L_N^{(\frac{1}{4})}.$$

□

Theorem 3. *$\vec{L}_N$ and $\vec{Q}_N$ are diagonalizable with real eigenvalues.*

Proof. This follows from the fact that the two matrices are, by construction, Hermitian. □

Theorem 4. *$\vec{Q}_N$ is positive semidefinite.*

Proof.

$$\begin{aligned} x^*\vec{Q}_N x :=& x^*\left(D_v^{-\frac{1}{2}}\vec{B}WD_e^{-1}\vec{B}^*D_v^{-\frac{1}{2}}\right)x \\ &\left(x^*D_v^{-\frac{1}{2}}\vec{B}W^{\frac{1}{2}}D_e^{-\frac{1}{2}}\right)\left(D_e^{-\frac{1}{2}}W^{\frac{1}{2}}\vec{B}^*D_v^{-\frac{1}{2}}x\right) \\ &\left(D_e^{-\frac{1}{2}}W^{\frac{1}{2}}\vec{B}^*D_v^{-\frac{1}{2}}x\right)^*\left(D_e^{-\frac{1}{2}}W^{\frac{1}{2}}\vec{B}^*D_v^{-\frac{1}{2}}x\right) \\ &||\left(D_e^{-\frac{1}{2}}W^{\frac{1}{2}}\vec{B}^*D_v^{-\frac{1}{2}}x\right)^*||_2^2 \geq 0. \end{aligned}$$

□

Theorem 5. *Let $x = a + ib \in \mathbb{C}^n$, with $a, b \in \mathbb{R}^n$. The 2-Dirichlet energy function $||x||^2_{\vec{L}_N} = x^*\vec{L}_N x$ of x induced by $\vec{L}_N$ is the following quadratic form:*

$$\begin{aligned} \frac{1}{2}\sum_{e\in E}\frac{w(e)}{\delta(e)}\sum_{u,v\in E}\Bigg(&\left(\left(\frac{a_u}{\sqrt{d_u}} - \frac{a_v}{\sqrt{d_v}}\right)^2 + \left(\frac{b_u}{\sqrt{d_u}} - \frac{b_v}{\sqrt{d_v}}\right)^2\right)\mathbf{1}_{u,v\in H(e)\vee u,v\in T(e)} \\ &+\left(\left(\frac{a_u}{\sqrt{d_u}} + \frac{b_v}{\sqrt{d_v}}\right)^2 + \left(\frac{a_v}{\sqrt{d_v}} - \frac{b_u}{\sqrt{d_u}}\right)^2\right)\mathbf{1}_{u\in H(e),v\in T(e)} \\ &+\left(\left(\frac{a_u}{\sqrt{d_u}} - \frac{b_v}{\sqrt{d_v}}\right)^2 + \left(\frac{a_v}{\sqrt{d_v}} + \frac{b_u}{\sqrt{d_u}}\right)^2\right)\mathbf{1}_{v\in H(e),u\in T(e)}\Bigg), \end{aligned} \tag{12}$$

where **1** *is the indicator function.*

Proof.

$$
\begin{aligned}
x^* \vec{L}_N x &= \sum_{u \in V} x_u^* x_u - \sum_{u,v \in V} \sum_{e \in E} \frac{w(e)}{\delta(e)} \frac{\bar{B}(u,e)\bar{B}(v,e)^*}{\sqrt{d(u)}\sqrt{d(v)}} x_u x_v^* \\
&= \sum_{u \in V} x_u^* x_u - \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{\delta(e)} \frac{\bar{B}(u,e)\bar{B}(v,e)^*}{\sqrt{d(u)}\sqrt{d(v)}} x_u x_v^* \\
&= \sum_{u \in V} x_u^* x_u - \sum_{e \in E} \frac{w(e)}{\delta(e)} \sum_{u,v \in V : u \leq v} \left(\bar{B}(u,e)\bar{B}(v,e)^* \frac{x_u x_v^*}{\sqrt{d(u)}\sqrt{d(v)}} + \bar{B}(v,e)\bar{B}(u,e)^* \frac{x_v x_u^*}{\sqrt{d(v)}\sqrt{d(u)}} \right) \\
&= \sum_{e \in E} \frac{w(e)}{\delta(e)} \sum_{u,v \in E : u \leq v} \left(\frac{x_u^* x_u}{d(u)} + \frac{x_v^* x_v}{d(v)} \right) \\
&\quad - \sum_{e \in E} \frac{w(e)}{\delta(e)} \sum_{u,v \in V : u \leq v} \left(\bar{B}(u,e)\bar{B}(v,e)^* \frac{x_u x_v^*}{\sqrt{d(u)}\sqrt{d(v)}} + \bar{B}(v,e)\bar{B}(u,e)^* \frac{x_v x_u^*}{\sqrt{d(v)}\sqrt{d(u)}} \right) \\
&= \sum_{e \in E} \frac{w(e)}{\delta(e)} \sum_{u,v \in V : u \leq v} \left(\frac{x_u^* x_u}{d(u)} + \frac{x_v^* x_v}{d(v)} - \bar{B}(u,e)\bar{B}(v,e)^* \frac{x_u x_v^*}{\sqrt{d(u)}\sqrt{d(v)}} - \bar{B}(v,e)\bar{B}(u,e)^* \frac{x_v x_u^*}{\sqrt{d(v)}\sqrt{d(u)}} \right).
\end{aligned}
$$

Let us analyze the three possible cases for the summand.

Case 1.a: $u \in H(e) \wedge v \in H(e) \Leftrightarrow \bar{B}(u,e) = 1, \bar{B}(v,e) = 1$. We have $\bar{B}(u,e)\bar{B}(v,e)^* = \bar{B}(v,e)\bar{B}(u,e)^* = 1$.

Case 1.b: $u \in T(e) \wedge v \in T(e) \Leftrightarrow \bar{B}(u,e) = -\mathrm{i}, \bar{B}(v,e) = -\mathrm{i}$. We have $\bar{B}(u,e)\bar{B}(v,e)^* = \bar{B}(v,e)\bar{B}(u,e)^* = (-\mathrm{i})(-\mathrm{i})^* = (-\mathrm{i})(\mathrm{i}) = 1$.

In both cases, we have:

$$
\frac{x_u^* x_u}{d(u)} + \frac{x_v^* x_v}{d(v)} - \frac{x_u x_v^*}{\sqrt{d(u)}\sqrt{d(v)}} - \frac{x_v x_u^*}{\sqrt{d(v)}\sqrt{d(u)}} = \left(\frac{x_u}{\sqrt{d(u)}} - \frac{x_v}{\sqrt{d(v)}} \right)^* \left(\frac{x_u}{\sqrt{d(u)}} - \frac{x_v}{\sqrt{d(v)}} \right).
$$

Letting $x_u = a_u + \mathrm{i}b_u$ and $x_v = a_v + \mathrm{i}b_v$, we have:

$$
\left(\frac{a_u}{\sqrt{d_u}} - \frac{a_v}{\sqrt{d_v}} \right)^2 + \left(\frac{b_u}{\sqrt{d_u}} - \frac{b_v}{\sqrt{d_v}} \right)^2.
$$

Case 2.a: $u \in H(e) \wedge v \in T(e) \Leftrightarrow \bar{B}(u,e) = 1, \bar{B}(v,e) = -\mathrm{i}$. We have $\bar{B}(u,e)\bar{B}(v,e)^* = (1)(-\mathrm{i})^* = \mathrm{i}$ and $\bar{B}(v,e)\bar{B}(u,e)^* = (-\mathrm{i})(1)^* = -\mathrm{i}$.

Thus:

$$
\frac{x_u^* x_u}{d(u)} + \frac{x_v^* x_v}{d(v)} - \mathrm{i} \frac{x_u x_v^*}{\sqrt{d(u)}\sqrt{d(v)}} + \mathrm{i} \frac{x_v x_u^*}{\sqrt{d(v)}\sqrt{d(u)}}
$$

Let $x_u = a_u + \mathrm{i}b_u$ and $x_v = a_v + \mathrm{i}b_v$, then we have:

$$
\left(\frac{a_u}{\sqrt{d_u}} + \frac{b_v}{\sqrt{d_v}} \right)^2 + \left(\frac{a_v}{\sqrt{d_v}} - \frac{b_u}{\sqrt{d_u}} \right)^2.
$$

Case 2.b: $u \in T(e) \wedge v \in H(e) \Leftrightarrow \bar{B}(u,e) = -\mathrm{i}, \bar{B}(v,e) = 1$. We have $\bar{B}(u,e)\bar{B}(v,e)^* = (-\mathrm{i})(1)^* = -\mathrm{i}$ and $\bar{B}(v,e)\bar{B}(u,e)^* = (1)(-\mathrm{i})^* = \mathrm{i}$. We have:

$$
\frac{x_u^* x_u}{d(u)} + \frac{x_v^* x_v}{d(v)} + \mathrm{i} \frac{x_u x_v^*}{\sqrt{d(u)}\sqrt{d(v)}} - \mathrm{i} \frac{x_v x_u^*}{\sqrt{d(v)}\sqrt{d(u)}}
$$

Let $x_u = a_u + \mathrm{i}b_u$ and $x_v = a_v + \mathrm{i}b_v$, then we have:

$$\left(\frac{a_u}{\sqrt{d_u}} - \frac{b_v}{\sqrt{d_v}}\right)^2 + \left(\frac{a_v}{\sqrt{d_v}} + \frac{b_u}{\sqrt{d_u}}\right)^2.$$

The final equation reported in the statement of the theorem is obtained by combining the four cases we just analyzed. ☐

Corollary 3. *$\vec{L}_N$ is positive semidefinite.*

Proof. Since $\vec{L}_N$ is Hermitian, it can be diagonalized as $U\Lambda U^*$ for some $U \in \mathbb{C}^{n\times n}$ and $\Lambda \in \mathbb{R}^{n\times n}$, where Λ is diagonal and real. We have $x^*\vec{L}_N x = x^*U\Lambda U^*x = y^*\Lambda y$ with $y = U^*x$. Since Λ is diagonal, we have $y^*\Lambda y = \sum_{u\in V} \lambda_u y_u^2$. Thanks to Theorem 5, the quadratic form $x^*\vec{L}_N x$ associated with $\vec{L}_N$ is a sum of squares and, hence, nonnegative. Combined with $x^*\vec{L}_N x = \sum_{u\in V} \lambda_u y_u^2$, we deduce $\lambda_u \geq 0$ for all $u \in V$. ☐

Corollary 4. *$\lambda_{\max}(\vec{L}_N) \leq 1$ and $\lambda_{\max}(\vec{Q}_N) \leq 1$.*

Proof. $\lambda_{\max}(\vec{L}_N) \leq 1$ holds if and only if $\vec{L}_N - I \preceq 0$. Since $\vec{L}_N = I - \vec{Q}_N$ holds by definition, we need to prove $-\vec{Q}_N \preceq 0$, which holds true due to Theorem 4.

Similarly, $\lambda_{\max}(\vec{Q}_N) \leq 1$ holds if and only if $\vec{Q}_N - I \preceq 0$. Since $\vec{Q}_N = I - \vec{L}_N$ holds by definition, we need to prove $-\vec{L}_N \preceq 0$, which holds true due to Corollary 3. ☐

Proposition 1. *The convolution operator obtained from equation 1 by letting $\mathcal{L} = \vec{L}_N$ with parameters θ_0, θ_1 coincides with the one obtained by letting $\mathcal{L} = \vec{Q}_N$ with parameters $\theta'_0 = \theta_0 + \theta_1$, $\theta'_1 = -\theta_1$.*

Proof. Consider the two operators $\theta_0 I + \theta_1 \vec{L}_N$ and $\theta'_0 I + \theta'_1 \vec{Q}_N$. Since $\vec{L}_N = I - \vec{Q}_N$, the first operator reads: $\theta_0 I + \theta_1(I - \vec{Q}_N)$. This is rewritten as $(\theta_0 + \theta_1)I - \theta_1\vec{Q}_N$. By operating the choice $\theta'_0 = \theta_0 + \theta_1$ and $\theta_1 = -\theta'_1$, the second operator is obtained. ☐

C Complexity of GeDi-HNN

The detailed calculations for the (inference) complexity of GeDi-HNN are as follows.

1. The Generalized Directed Laplacian $\vec{L}_N$ is constructed following equation 7 in time $O(n^2m)$, where the factor m is due to the need for computing the product between two rows of $\vec{B}$ to calculate each entry of $\vec{L}_N$. After $\vec{L}_N$ has been computed, the convolution matrix $\hat{Y} \in \mathbb{C}^{n\times n}$ is constructed in time $O(n^2)$. Note that such a construction is carried out entirely in pre-processing and is not required at inference time.

2. Each of the ℓ convolutional layers of GeDi-HNN requires $O(n^2c+nc^2+nc) = O(n^2c+nc^2)$ elementary operations across 3 steps. Let X^{l-1} be the input matrix to layer $l = 1, \ldots, \ell$. The operations that are carried out are the following ones.
   (a) $\vec{L}_N$ is multiplied by the node-feature matrix $X^{l-1} \in \mathbb{C}^{n\times c}$, obtaining $P^{l_1} \in \mathbb{C}^{n\times c}$ in time $O(n^2c)$ (we assume matrix multiplications takes cubic time);
   (b) The matrices $P^{l_0} = IX^{l-1} = X^{l-1}$ and P^{l1} are multiplied by the weight matrices $\Theta_0, \Theta_1 \in \mathbb{R}^{c\times c}$ (respectively), obtaining the intermediate matrices $P^{l_{01}}, P^{l_{11}} \in \mathbb{C}^{n\times c}$ in time $O(nc^2)$.
   (c) The matrices $P^{l_{01}}$ and $P^{l_{11}}$ are additioned in time $O(nc)$ to obtain P^{l_2}.
   (d) The activation function ϕ is applied component-wise to P^{l_2} in time $O(nc)$, resulting in the output matrix $X^l \in \mathbb{C}^{n\times c}$ of the l-th convolutional layer.

3. The unwind operator transforms X^ℓ (the output of the last convolutional layer ℓ) into the matrix $U^0 \in \mathbb{R}^{n\times 2c}$ in linear time $O(nc)$.

4. Call U^{s-1} the input matrix to each linear layer of index $s = 1, \ldots, S$. The application of the s-th linear layer to $U^{s-1} \in \mathbb{C}^{n \times c'}$ requires multiplying U^{s-1} by a weight matrix $M_s \in \mathbb{C}^{c' \times c'}$ (where c' is the number of channels from which and into which the feature vector of each node is projected). This is done in time $O(nc'^2)$.

5. In the last linear layer of index S, the input matrix $U^{S-1} \in \mathbb{R}^{n \times c'}$ is projected into the output matrix $O \in \mathbb{R}^{n \times d}$ in time $O(nc'd)$.

6. The application of the Softmax activation function takes linear time $O(nd)$.

We deduce an overall complexity of $O(\ell(n^2c + nc^2) + nc + (S-1)(nc'^2) + nc'd + nd)$ which, letting $\bar{c} = \max\{c, c', d\}$, coincides with $O(\ell(n^2\bar{c}) + (\ell + S)(n\bar{c}^2))$.

D Further Details on the Datasets

We test GeDi-HNN on ten real-world dataset. `Cora`, `Citeseer`, and `PubMed` (Zhang et al., 2022); `email-Eu`, and `email-Enron` (Benson et al., 2018); `Texas`, `Wisconsin`, and `Cornell` (Pei et al., 2020); `WikiCS` (Mernyei and Cangea, 2020); and `Telegram` (Bovet and Grindrod, 2020).

`Cora`, `Citeseer`, and `PubMed` are citation networks with node labels based on paper topics. In these citation networks, the nodes represent papers, their relationships denote citations of one paper by another, and the node features are the bag-of-words representation of papers.

`Email-Enron` and `email-Eu` are two email datasets—one from communications exchanged between Enron employees (Klimt and Yang, 2004) and the other from a European research institution (Paranjape et al., 2017). The nodes are email addresses and their relationships are of sender-receiver type. Since no node labeling is present in these two datasets, we define the node labels (node classes) using the Spinglass algorithm (Reichardt and Bornholdt, 2006).

`Texas`, `Wisconsin`, and `Cornell` are WebKB data sets extracted from the CMU World Wide Knowledge Base (`Web->KB`) project.[8] WebKB is a webpage data set collected from computer science departments of various universities by Carnegie Mellon University. In these networks, the nodes represent web pages, and the relationship are hyperlinks between them. The node features are the bag-of-words representation of the web pages. The web pages are manually classified into the five categories: student, project, course, staff, and faculty.

`WikiCS` is a directed network whose nodes correspond to Computer Science articles, and the relationships are on hyperlinks. This network has 10 classes representing different branches of the field.

`Telegram` models an influence network built on top of interactions among distinct users who propagate ideologies of a political nature.

The statistic of these ten real-world datasets and of the synthetic datasets we generate are summarized in Tables 3 and 4.

E Experiment Details

Hardware. The experiments were conducted on 2 different machines:

1. An Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz with 380 GB RAM, equipped with an NVIDIA Ampere A100 40GB.

2. A 12th Gen Intel(R) Core(TM) i9-12900KF CPU @ 3.20GHz CPU with 64 GB RAM, equipped with an NVIDIA RTX 4090 GPU.

[8] `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/`

Table 3: Statistics of the real-world datasets

Data set	# node	# hyperedges	# classes	average $\|e\|$
`Cora`	2708	1579	7	3.03
`Citeseer`	3312	1079	6	3.20
`Pubmed`	19717	7963	3	4.35
`email-Eu`	986	873	10	38.01
`email-Enron`	143	128	7	20.03
`Telegram`	245	185	4	48.04
`Texas`	183	40	5	4.45
`Wisconsin`	251	65	5	4.77
`Cornell`	183	41	5	3.88
`WikiCS`	11701	6827	10	42.08

Table 4: Statistics of the synthetic datasets

Data set	# node	# hyperedges	# classes	average $\|e\|$
$I_o = 10$	500	250	5	9.05
$I_o = 30$	500	450	5	10.79
$I_o = 50$	500	650	5	11.63

Model Settings. We trained every learning model considered in this paper for up to 500 epochs. We adopted a learning rate of $5 \cdot 10^{-3}$ and employed the optimization algorithm Adam with weight decays equal to $5 \cdot 10^{-4}$ (in order to avoid overfitting). For all the models that adopt the classification layer, we set it to 2.

We adopted a hyperparameter optimization procedure to identify the best set of parameters for each model. In particular, the hyperparameter values are:

- For AllDeepSets and ED-HNN, the number of basic block is chosen in $\{2, 4, 8\}$, the number of MLPs per block in $\{1, 2\}$, the dimension of the hidden MLP (i.e., the number of filters) in $\{64, 128, 256, 512\}$, and the classifier hidden dimension in $\{64, 128, 256\}$.
- For AllSetTransformer the number of basic block is chosen in $\{2, 4, 8\}$, the number of MLPs per block in $\{1, 2\}$, the dimension of the hidden MLP in $\{64, 128, 256, 512\}$, the classifier hidden dimension in $\{64, 128, 256\}$, and the number of heads in $\{1, 4, 8\}$.
- For UniGCNII, HGNN, HNHN, HCHA/HGNN$^+$, LEGCN, and HCHA with the attention mechanism, the number of basic blocks is chosen in $\{2, 4, 8\}$ and the hidden dimension of the MLP layer in $\{64, 128, 256, 512\}$.
- For HyperGCN, the number of basic blocks is chosen in $\{2, 4, 8\}$.
- For HyperND, the classifier hidden dimension is chosen in $\{64, 128, 256\}$.
- For PhenomNN, the number of basic blocks is chosen in $\{2, 4, 8\}$. We select four different settings:
  1. $\lambda_0 = 0.1$, $\lambda_1 = 0.1$ and prop step= 8,
  2. $\lambda_0 = 0$, $\lambda_1 = 50$ and prop step= 16,
  3. $\lambda_0 = 1$, $\lambda_1 = 1$ and prop step= 16,
  4. $\lambda_0 = 0$, $\lambda_1 = 20$ and prop step= 16.
- For GeDi-HNN and GeDi-HNN `w/o directionality`, the number of convolutional layers is chosen in $\{1, 2, 3\}$, the number of filters in $\{64, 128, 256, 512\}$, and the classifier hidden dimension in $\{64, 128, 256\}$. We tested GeDi-HNN both with the input feature matrix $X \in \mathbb{C}^{n \times c}$ where $\Re(X) = \Im(X) \neq 0$ and with $\Im(X) = 0$.

Node Features. For `Cora`, `Citeseer`, `PubMed`, `Texas`, `Wisconsin`, `Cornell`, `WikiCS`, and `Telegram`, we retain the datasets' original features. For `email-Eu`, `email-Enron`, and the synthetic datasets, the feature vectors are generated using the vertex degree of each node.

F From a Directed Hypergraph to the Generalized Directed Laplacian

To illustrate the representation of a directed hypergraph in our Generalized Directed Laplacian, consider a directed hypergraph $\mathcal{H} = (V, E)$ with $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{e_1, e_2\}$. The incidence relationships are defined as follows: $v_1, v_2 \in H(e_1)$, $v_3 \in T(e_1)$, $v_4, v_5 \in H(e_2)$, and $v_1, v_2 \in T(e_2)$. The hyperedges have unit weights (i.e., $W = I$). The hyperedge cardinalities are $\delta_{e_1} = 3$ and $\delta_{e_2} = 4$.

For this hypergraph, we construct our Generalized Directed Laplacian using the following matrices: the incidence matrix $\vec{B}$, its conjugate transpose $\vec{B}^*$, the vertex degree matrix D_v, and the hyperedge degree matrix D_e.

$$\vec{B} = \begin{bmatrix} 1 & -i \\ 1 & -i \\ -i & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \vec{B}^* = \begin{bmatrix} 1 & 1 & i & 0 & 0 \\ i & i & 0 & 1 & 1 \end{bmatrix} \quad D_v = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad D_e = \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}.$$

Based on these matrices, we build $\vec{Q}_N$ as follows:

$$\vec{Q}_N = \begin{bmatrix} 0.29 & 0.29 & \text{i}0.24 & -\text{i}0.18 & -\text{i}0.18 \\ 0.29 & 0.29 & \text{i}0.24 & -\text{i}0.18 & -\text{i}0.18 \\ -\text{i}0.24 & -\text{i}0.24 & 0.33 & 0 & 0 \\ \text{i}0.18 & \text{i}0.18 & 0 & 0.25 & 0.25 \\ \text{i}0.18 & \text{i}0.18 & 0 & 0.25 & 0.25 \end{bmatrix}$$

and then our Generalized Directed Laplacian:

$$\vec{L}_N = \begin{bmatrix} 0.71 & -0.29 & -\text{i}0.24 & \text{i}0.18 & \text{i}0.18 \\ -0.29 & 0.71 & -\text{i}0.24 & \text{i}0.18 & \text{i}0.18 \\ \text{i}0.24 & \text{i}0.24 & 0.66 & 0 & 0 \\ -\text{i}0.18 & -\text{i}0.18 & 0 & 0.75 & -0.25 \\ -\text{i}0.18 & -\text{i}0.18 & 0 & -0.25 & 0.75 \end{bmatrix}$$

By inspecting $\vec{L}_N$, one can observe that it encodes the elements of the hypergraph in the following way:

1. The presence of nodes belonging to the same head or tail set, i.e., $v_1, v_2 \in H(e_1)$, $v_4, v_5 \in H(e_2)$, and $v_1, v_2 \in T(e_2)$, is encoded in the real part. Specifically, $(\vec{L}_N)_{v_1 v_2} = (\vec{L}_N)_{v_2 v_1} = -0.29$ and $(\vec{L}_N)_{v_4 v_5} = (\vec{L}_N)_{v_5 v_4} = -0.25$.

2. The directed hyperedges are encoded via the imaginary part. For example, considering nodes v_1 and v_3, we have $(\vec{L}_N)_{v_1 v_3} = -(\vec{L}_N)_{v_3 v_1} = -\text{i}0.24$.

3. The absence of a relationship between a pair of nodes is encoded by 0. Specifically, $(\vec{L}_N)_{v_3 v_4} = (\vec{L}_N)_{v_4 v_3} = 0$ and $(\vec{L}_N)_{v_3 v_5} = (\vec{L}_N)_{v_5 v_3} = 0$.

4. The "self-loop information" (a measure of how strongly the feature of a node depends on its current value within the convolution operator) is encoded by the diagonal of $\vec{L}_N$.