Studying the Effect of GNN Spatial Convolutions On The Embedding Space's Geometry (Supplementary Material)

Claire Donnat[1] **So Won Jeong**[1]

[1]Department of Statistics, The University of Chicago, Chicago, Illinois, USA

A COMPARISON OF GNN OPERATORS

Method	Operator Type	Convolution Family	Operator
GCNConv [Kipf and Welling, 2016]	Spatial	$\mathcal{F}_{\alpha=0.5,\beta=1}$	$x_i' = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}XW$ $x_i' = \sum_{j\in\mathcal{N}(i)} \frac{e_{ij}}{\sqrt{\hat{d}_i \hat{d}_j}} x_j$
ChebConv [Defferrard et al., 2016]	Spectral	NA	$X = XW_1 + \hat{L}XW_2 + (2\hat{L}^2X - X)W_3$ with $\hat{L} = \frac{2}{\lambda_{\max}}L - I$
SAGEConv [Hamilton et al., 2017]	Spatial	$\mathcal{M}_{\alpha=0,\beta=1}$	$x = W_1 x_i + W_2 \bar{X}_{j\in\mathcal{N}(i)}$ with $\bar{X}_{j\in\mathcal{N}(i)} = \frac{\sum_{j\in\mathcal{N}(i)} x_j}{d_i}$
GraphConv[Morris et al., 2019]	Spatial	$\mathcal{F}_{\alpha=0,\beta=0}$	$x_i' = W_1 x_i + W_2 \sum_{j\in\mathcal{N}(i)} e_{ij} x_j$
GatedGraphConv [Li et al., 2015]	Spatial	Variant of $\mathcal{F}_{\alpha=0,\beta=0}$	$h_i^{(0)} = x_i \| 0$ $m_i^{(l+1)} = \sum_{j\in\mathcal{N}(i)} e_{j,i} W h_j^{(l)}$ $h_i^{(l+1)} = GRU(m_i^{(l+1)}, h_i^{(l)})$
ResGatedGraphConv [Bresson and Laurent, 2017]	Spatial	Variant of $\mathcal{F}_{\alpha=0,\beta=0}$	$x_i' = W_1 x_i + \sum_{j\in\mathcal{N}(i)} \eta_{ij} \circ W_2 x_j$ with $\eta_{ij} = \sigma(W_3 x_i + W_4 x_j)$
GAT [Veličković et al., 2017] GATv2Conv [Brody et al., 2021]	Spatial	Variant of $\mathcal{M}_{\alpha=0,\beta=1}$	$x_i' = \alpha_{ii} W x_i + \sum_{j\in\mathcal{N}(i)} \alpha_{ij} W x_j$ with $\alpha_{ij} = \frac{\exp\{\text{LeakyReLU}(a^T[\Theta x_i \| \Theta x_j]\}}{\sum_{k\in\mathcal{N}(i)\cup\{i\}} \exp\{\text{LeakyReLU}(a^T[\Theta x_i \| \Theta x_j]\}}$
AGNN [Thekumparampil et al., 2018]	Spatial	$\mathcal{M}_{\alpha=0,\beta=1}$	$X' = PX$ $P_{ij} = \frac{\exp\{\beta \dot{\cos}(x_i, x_j)\}}{\sum_{k\in\mathcal{N}(i)\cup\{i\}} \exp\{\beta \dot{\cos}(x_i, x_k)\}}$
Transformer Conv [Shi et al., 2020]	Spatial	Variant of $\mathcal{M}_{\alpha=0,\beta=0}$	$x_i' = W_1 x_i + \sum_{j\in\mathcal{N}(i)} \alpha_{ij} W_2 x_j$ $\alpha_{ij} = \text{Softmax} \frac{(W_3 x_i)^T (W_4 x_j)}{\sqrt{d}}$
TAGConv [Du et al., 2017]	Spectral	NA	$X' = \sum_{k=0}^{K} (D^{-1/2} A D^{-1/2})^k X W_k$
GINConv [Xu et al., 2018]	Spatial	$\mathcal{F}_{\alpha=0,\beta=1+\epsilon}$	$X' = h_\theta\Big((\mathbf{A} + (1+\epsilon)\mathbf{I})X\Big)$
GINEConv [Hu et al., 2019]	Spatial	Variant of $\mathcal{F}_{\alpha=0,\beta=0}$	$x_i' = h_\theta\Big(((1+\epsilon)x_i + \sum_{j\in\mathcal{N}(i)} \text{ReLU}(x_j + e_{ij})\Big))$
ARMAConv [Bianchi et al., 2021]	Spectral	NA	$X' = \frac{1}{K}\sum_{k=1}^{K} X_k^{(T)}$
SGCConv [Wu et al., 2019]	Spatial	$\mathcal{F}_{\alpha=0.5,\beta=1}$	$X' = (\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2})^K XW$

Table 1. Comparison of some of the different convolution operators. We report here some of the most famous existing convolutions — but we deliberately omitted those applicable to edges, dynamic graphs, heterogeneous graphs, hypergraphs and other extensions. We report the type of convolution family (as defined in Section 2) corresponding to each of the proposed convolution. The term " variant" denotes some adaptation of the base family (for instance, learning the appropriate edge weights as part of the training procedure, or treating the source node differently than the sum of the neighbors).

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

B PROOFS OF SECTION 3

Proof[Lemma 3.1] As per section 3, we analyse the embedding that is fed into the last linear layer, denoted as:

$$H^{(K)} = S\sigma(H^{(K-1)}) = \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}}{(d_u + \beta)^{\alpha}(d_v + \beta)^{\alpha}} Z_{v\cdot}$$

where $\mathcal{N}(u)$ denotes the neighbourhood of node u, A_{uv} is the (potentially weighted) adjacency matrix, with diagonal equal to β, and $Z_{v\cdot} = \sigma(H_{v\cdot}^{(K-1)})$.

Writing $\Delta_v = d_v - d_u$, note that $H^{(K)}$ can be rewritten as:

$$H^{(K)} = \frac{1}{(d_u + \beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}}{(1 + \frac{\Delta_v}{d_u + \beta})^{\alpha}} Z_{v\cdot} \tag{1}$$

Note that $\frac{\Delta_v}{d_u+\beta} \geq -1$ as long as $d_v - d_u \geq -d_u - \beta \implies d_v \geq -\beta$, which holds necessarily, since $d_v \geq 1$. Since the function $x \to (x+1)^{-\alpha}$ is infinitely differentiable for $x \in (-1, \infty)$, using the Maclaurin expansion of $(x+1)^{-\alpha}$ around 0, we know that there exists $\xi \in [\min(0, x), \max(0, x)]$ such that:

$$\frac{1}{(x+1)^{\alpha}} = 1 - \alpha x + \frac{\alpha(\alpha+1)}{2} \frac{x^2}{(\xi+1)^{\alpha+2}} \tag{2}$$

It is easy to check that if $\frac{\Delta_v}{d_u+\beta} \geq 0$, then $\frac{1}{(1+\xi)^{\alpha+2}} \leq 1$. Conversely, if $d_v \leq d_u$, then $\frac{1}{(\xi+1)^{\alpha+2}} \leq \frac{1}{(1+\frac{\Delta_v}{d_u+\beta})^{\alpha+2}} \leq \frac{1}{(1+\frac{1-d_u}{d_u+\beta})^{\alpha+2}} \leq \frac{1}{(\frac{\beta+1}{d_u+\beta})^{2+\alpha}} \leq (d_{\max} + \beta)^{2+\alpha} = M$.

Equation 1 thus becomes:

$$\begin{aligned}
||H^{(K)}||_2 = &||SZ||_2 \leq \frac{1}{(d_u + \beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}}{(1 + \frac{\Delta_v}{d_u+\beta})^{\alpha}} ||Z_v||_2 \\
&\leq \frac{||Z||_{2,\infty}}{(d_u + \beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} A_{uv}(1 - \alpha \frac{\Delta_v}{d_u + \beta} + \frac{\alpha(\alpha+1)M}{2} \frac{\Delta_v^2}{(d_u+\beta)^2}) \\
&= ||Z||_{2,\infty} \Big((d_u + \beta)^{1-2\alpha} - \alpha \frac{\bar{\Delta}_v}{(d_u+\beta)^{2\alpha}} + \frac{\alpha(\alpha+1)M}{2} \frac{\overline{\Delta^2}_v}{(d_u+\beta)^{1+2\alpha}} \Big)
\end{aligned}$$

where $\bar{\Delta}_u$ (respectively $\bar{\Delta^2}_u$) are the weighted averages of the degree differences: $\bar{\Delta}_u = \frac{\sum_{v \in \mathcal{N}(u) \cup \{u\}} A_{uv} \Delta_v}{d_u + \beta}$ (respectively, squared degree differences: $\bar{\Delta^2}_u = \frac{\sum_{v \in \mathcal{N}(u) \cup \{u\}} A_{uv} \Delta_v^2}{d_u + \beta}$). In the previous equation, we have also introduced the notation $||Z||_{2,\infty} = \max_v ||Z_v||_2$.

□

C PROOFS OF SECTION 4

C.1 SECTION 4: PROOF OF THE OBSERVATIONS

We begin by revisiting in greater details the observations made in section 4. To see how the two families of spatial operators differ in the importance they attribute to topology and node feature information, consider a simple two-layer GCN such as suggested by Kipf et al Kipf and Welling [2016]. In this setting, node embeddings can be written as: $H = S\sigma(SXW + b)$, so that the output of the network is $Y = S\sigma(SXW + b)W^{(2)} + b^{(2)} = HW^{(2)} + b^{(2)}$. We also choose the non-linearity σ to be the ReLU function. In this case, for the directions in which the term is positive, the embedding H (ie, the transformed features that are being fed into the last linear layer) can be re-written as:

$$H_{u\cdot} = \sum_{k=1}^{d} \sum_{\substack{v \in \tilde{\mathcal{N}}(u) \\ (SXW+b)_{vk} \geq 0}} \big(S_{uv}(SXW)_{vk} W_{k\cdot}^{(2)} + S_{uv} b_{vk} W_{k\cdot}^{(2)} \big) \tag{3}$$

The embedding is thus the sum of two components: a function of a (subset of) neighbouring feature vector and a term that has the potential to encode local topology. To see why this is the case, consider a scenario where nodes in $\mathcal{N}(u)$ are all such that $(SXW+b)_{vk} \geq 0$ for all k or $(SXW+b)_{vk} < 0$ for all k. Denote $\tilde{A}(u) = \{v \in \tilde{N}(u) : \quad (SXW+b)_{vk} \geq 0 \quad \text{for all } k\}$. In this case, Equation 3 becomes: $H_{u\cdot} = \sum_{v\in\tilde{A}(u)} \left(S_{uv}(SX\tilde{W})_{v\cdot} + S_{uv}\tilde{b}_{v\cdot}\right)$, with $\tilde{b} = bW^{(2)}$ and $\tilde{W} = WW^{(2)}$. Therefore, for symmetric convolutions, the term $(\sum_{v\in\tilde{A}(u)} S_{uv})b$ encodes information about the neighborhood (it is proportional to the number of terms in the sum $|\tilde{A}(u)|$.) Conversely, for row-symmetric convolutions, this term is identically equal to b, resulting in an embedding that is less sensitive to topology.

C.2 PROOF OF LEMMA 4.1: SYMMETRIC CONVOLUTIONS

In this subsection, we prove the results stated in lemma 4.1 for symmetric convolutions. We remind the reader of the setting of lemma 4.1: we consider two structurally equivalent neighbourhoods (meaning that there exists a mapping ϕ that transforms each node in the neighborhood of v into its corresponding one in the neighborhood of u — see Figure 1), but the feature vectors are different. Mathematically, we model this situation as:

$$\forall j \in N(v), \quad X_j = X_{u\phi(j)} + \epsilon$$

where ϵ is a vector with independent centered Gaussian entries with parameter σ. The purpose of this subsection is to analyze the effect of the convolution on the relative distance between embeddings.

Lemma 4.1 is re-written here, to make this appendix self-contained:

Lemma 4.1 *For symmetric convolutions, with probability at least $1-\delta$, with M as in 3.1, we have:*

$$||H_u - H_{u'}||^2 \leq \mu + 2\sqrt{2}\sigma||W||_{2,\infty}(d_u+\beta)^{1-2\alpha} \times \sqrt{1 + 2\alpha|\overline{\Delta}_u| + \frac{\alpha(2\alpha+1)M}{d_u}}\log(1/\delta)$$

where $\mu = \sigma^2||W||^2\left((d_u+\beta)^{2-4\alpha} + 2\alpha|\overline{\Delta}_u| + \alpha(2\alpha+1)M\frac{\overline{\Delta^2}_u}{d_u}\right)$. Conversely, for row-symmetric embeddings:

$$||H_u - H_{u'}||^2 \leq \mu + 2\sqrt{2}\sigma||W||_2\sqrt{\sum_{v\in\tilde{\mathcal{N}}(u)}\frac{1}{(d_v+\beta)^{2\alpha}}}\log(1/\delta),$$
$$\mu = \frac{\sigma^2||W||^2}{\sum_{v\in\tilde{N}(u)}(d_v+\beta)^{-2\alpha}}\frac{1}{1+\beta}$$

As previously stated, the purpose of this subsection is to analyze the effect of the convolution on the relative distance between embeddings. Consequently, we consider a simplified one-layer setting, with no non-linearities. We argue that this is indeed sufficient to characterize the effect of the convolution on the organization of the data, and we expect results for deeper networks to follow by induction, and to hold by 1-Lipschitzness of the ReLU activation for ReLU non-linear GNNs.

Proof of Lemma 4.1(symmetric convolutions). Therefore, in the simplified setting, the distance between the outputs of a GCN layer for nodes u and v can be written as:

$$\begin{aligned} H_u^{(k)} - H_{u'}^{(k)} &= (SX_u - SX_{u'})W \\ &= \sum_{v\in\mathcal{N}(u)\cup\{u\}} \frac{A_{uv}}{(d_u+\beta)^\alpha(d_v+\beta)^\alpha}(X_v - X_{\phi(v)})W \\ &= \sum_{v\in\mathcal{N}(u)\cup\{u\}} \frac{A_{uv}}{(d_u+\beta)^\alpha(d_v+\beta)^\alpha}\epsilon_v W \end{aligned}$$

(4)

Since $\epsilon_v \sim \mathcal{N}(0, \sigma^2)$, each entry of the vector $H_u^{(k)} - H_{u'}^{(k)}$ is Gaussian:

$$\begin{aligned} H_{uj}^{(k)} - H_{u'j}^{(k)} &= \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}}{(d_u+\beta)^\alpha (d_v+\beta)^\alpha} \sum_{k=1}^{d} \epsilon_{vk} W_{kj} \\ &\sim \mathcal{N}(0, \frac{\sigma^2 ||W_{\cdot j}||^2}{(d_u+\beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}^2}{(d_v+\beta)^{2\alpha}}) \end{aligned}$$

The mean of $||H_u^{(k)} - H_{u'}^{(k)}||^2$ is simply given by:

$$\begin{aligned} \mu &= \mathbb{E}[||H_u^{(k)} - H_{u'}^{(k)}||^2] \\ &= \frac{\sigma^2 ||W||^2}{(d_u+\beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}^2}{(d_v+\beta)^{2\alpha}} \\ &\leq \frac{\sigma^2 ||W||^2}{(d_u+\beta)^{4\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} A_{uv}^2 (1 - 2\alpha \frac{\Delta_v}{d_u+\beta} + \alpha(2\alpha+1) M \frac{\Delta_v^2}{(d_u+\beta)^2}) \end{aligned}$$

for some constant M, using the same Taylor expansion reasoning as for Lemma 3.1. Therefore, denoting as $\tilde{d}_u = \sum_{v \in \mathcal{N}(u)} A_{uv}^2$ and $\bar{\tilde{\Delta}} = \frac{\sum_{v \sim \mathcal{N}(u)} A_{uv}^2 \Delta_v}{\tilde{d}_u}$, $\overline{\tilde{\Delta}^2} = \frac{\sum_{v \sim \mathcal{N}(u)} A_{uv}^2 \Delta_v^2}{\tilde{d}_u}$, we have:

$$\begin{aligned} \mu &\leq \frac{\sigma^2 ||W||^2}{(d_u+\beta)^{4\alpha}} \Big((\tilde{d}_u + \beta^2) - 2\alpha \overline{\Delta}_u \frac{\tilde{d}_u}{d_u+\beta} + \alpha(2\alpha+1) M \overline{\tilde{\Delta}^2}_u \frac{\tilde{d}_u}{(d_u+\beta)^2} \Big) \\ &\overset{(i)}{\leq} \frac{\sigma^2 ||W||^2}{(d_u+\beta)^{4\alpha}} \Big((d_u+\beta)^2 + 2\alpha |\bar{\tilde{\Delta}}_u| \frac{d_u}{d_u+\beta} + \alpha(2\alpha+1) M \frac{\overline{\tilde{\Delta}^2}_u}{d_u+\beta} \Big) \\ &= \sigma^2 ||W||^2 \Big((d_u+\beta)^{2-4\alpha} + 2\alpha |\bar{\tilde{\Delta}}_u| (d_u+\beta)^{-4\alpha} + \alpha(2\alpha+1) M \overline{\tilde{\Delta}^2}_u (d_u+\beta)^{-1-4\alpha} \Big) \end{aligned}$$

where line (i) follows from the fact that, assuming the edge weights are less than 1, $A_{uv}^2 \leq A_{uv}$, implying that $\tilde{d}_u \leq d_u$.

Let us now turn to the analysis of the concentration of this norm. By Gaussianity of each of its coordinate, the squared norm $||H_u^{(k)} - H_{u'}^{(k)}||^2 = \sum_{j=1}^{p} \big(\sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}}{(d_u+\beta)^\alpha (d_v+\beta)^\alpha} \epsilon_{v \cdot} W_{\cdot j} \big)^2$ is sub-exponential.

To see this, note that since each of the p coordinate of the vector $H_u^{(k)} - H_{u'}^{(k)}$ is Gaussian with variance $\tilde{\sigma}_j^2 = \frac{\sigma^2 ||W_{\cdot j}||^2}{(d_u+\beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}^2}{(d_v+\beta)^{2\alpha}})$, its square is sub-Exponential with parameter $(2\tilde{\sigma}_j^2, 4\tilde{\sigma}_j^2)$ (Wainwright [2019]), so the squared norm (ie the sum of the squared entries) is sub-Exponential with parameter:

$$(2\sum_{j=1}^{p} \tilde{\sigma}_j^2, 4 \max_j \tilde{\sigma}_j^2) = \Big(2 \frac{\sigma^2 ||W||^2}{(d_u+\beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}^2}{(d_v+\beta)^{2\alpha}}, 4 \frac{\sigma^2 ||W||_{2,\infty}^2}{(d_u+\beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}^2}{(d_v+\beta)^{2\alpha}} \Big).$$

By property of the sub-exponential tail, we know that:

$$\mathbb{P}[||H_u^{(k)} - H_{u'}^{(k)}||^2 - \mu \geq t] \leq \min(e^{-t^2/(4\sum_{j=1}^{p} \tilde{\sigma}_j^2)}, e^{-t/(2\sqrt{2}\sqrt{\sum_{j=1}^{p} \tilde{\sigma}_j^2})}) \tag{5}$$

Therefore, with probability at least $1-\delta$, for any $\delta \in (0,1)$, we must have:

$$\begin{aligned} &||H_u^{(k)} - H_{u'}^{(k)}||^2 \\ &\leq \mu + 2\sqrt{2} \sqrt{\frac{\sigma^2 ||W||_{2,\infty}^2}{(d_u+\beta)^{2\alpha}} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{A_{uv}^2}{(d_v+\beta)^{2\alpha}}} \log(1/\delta) \\ &\leq \mu + 2\sqrt{2}\sigma ||W||_{2,\infty} (d_u+\beta)^{1-2\alpha} \\ &\quad \times \sqrt{1 + 2\alpha |\overline{\Delta}_u| + \alpha(2\alpha+1) M \frac{\overline{\Delta^2}_u}{d_u+\beta}} \log(1/\delta) \end{aligned}$$

The concentration is thus a function of the node degree: the leading term is in $(d_u + \beta)^{1-2\alpha}$, and we observe again the existence of a critical threshold at $\alpha = 0.5$. □

C.3 PROOF OF LEMMA 4.1: THE CASE OF ROW-NORMALIZED CONVOLUTIONS

We now turn to the proof of Lemma 4.1 for row-normalized convolutions.

Proof of Lemma 4.1(row-normalized convolutions). In the case of row-normalized convolutions, we have instead:

$$(SX_u - SX_{u'})W = \sum_{v \in \mathcal{N}(u) \cup \{u\}} s_{uv} \epsilon_v W \tag{6}$$

where, as highlighted in section 3, s_{uv} is proportional to $\frac{1}{(d_v+\beta)^\alpha}$, but does not depend on d_u. In this case, following a similar reasoning to the previous subsection:

$$\begin{aligned}
\mu &= \mathbb{E}[||H_u^{(k)} - H_{u'}^{(k)}||^2] \\
&= \frac{\sigma^2 ||W||^2}{Z^2} \sum_{v \in \mathcal{N}(u) \cup \{u\}} s_{uv}^2 \quad \text{with } Z = \sum_{v \in \mathcal{N}(u) \cup \{u\}} s_{uv} \\
&\leq \frac{\sigma^2 ||W||^2}{Z^2} \max_{v \in \mathcal{N}(u) \cup \{u\}} \{s_{uv}\} \quad \text{by Holder's inequality} \\
&\leq \frac{\sigma^2 ||W||^2}{Z^2} \beta \quad \text{assuming } \beta \geq 1 \implies \max_{v \in \mathcal{N}(u) \cup \{u\}} \{s_{uv}\} \leq \beta
\end{aligned}$$

□

C.4 TOY EXAMPLE 2

Conversely, u and u' have radically different neighborhoods from a topological perspective, but have similar features:

$$\forall j \in \mathcal{N}(u) \cup \mathcal{N}(u'), \quad X_j = \bar{X}_u$$

In the symmetric case:

$$
\begin{aligned}
&||(SX)_{u\cdot} - (SX)_{u'\cdot}||^2 \\
&= || \sum_{v \in \tilde{\mathcal{N}}(u)} \frac{A_{uv}}{(d_u+\beta)^\alpha (d_v+\beta)^\alpha} \bar{X} \\
&\qquad - \sum_{v' \in \tilde{\mathcal{N}}(u')} \frac{A_{u'v'}}{(d_{v'}+\beta)^\alpha (d_{u'}+\beta)^\alpha} \bar{X} ||^2 \\
&= \Big(\sum_{v \in \tilde{\mathcal{N}}(u)} \frac{A_{uv}}{(d_u+\beta)^\alpha (d_v+\beta)^\alpha} \\
&\qquad - \sum_{v' \in \tilde{\mathcal{N}}(u')} \frac{A_{u'v'}}{(d_{v'}+\beta)^\alpha (d_{u'}+\beta)^\alpha} \Big)^2 ||\bar{X}||^2 \\
&= \Big(\sum_{v \in \tilde{\mathcal{N}}(u)} \frac{A_{uv}}{(d_u+\beta)^{2\alpha} (1+\frac{\Delta_v}{d_u+\beta})^\alpha} \\
&\qquad - \sum_{v' \in \tilde{\mathcal{N}}(u')} \frac{A_{u'v'}}{(d_{u'}+\beta)^{2\alpha} (1+\frac{\Delta_v}{d_{u'}+\beta})^\alpha} \Big)^2 ||\bar{X}||^2 \\
&= \Big((d_u+\beta)^{1-2\alpha} - \alpha \overline{\Delta}_u \\
&\quad + \frac{\alpha(\alpha+1)}{2(d_u+\beta)^{2+2\alpha}} \sum_{v \in \tilde{\mathcal{N}}(u)} \frac{A_{uv}(d_v-d_u)^2}{((1-t_v)+t_v \frac{\Delta_v}{d_u+\beta}+\beta)^{2+2\alpha}} \\
&\quad - (d_{u'}+\beta)^{1-2\alpha} + \alpha \overline{\Delta}_{u'} \\
&\quad - \frac{\alpha(\alpha+1)}{2(d_{u'}+\beta)^{2+2\alpha}} \sum_{v' \in \tilde{\mathcal{N}}(u')} \frac{A_{u'v'}(d'_v-d'_u)^2}{((1-t_{v'})+t_{v'} \frac{\Delta_{v'}}{d_{u'}+\beta}+\beta)^{2+2\alpha}} \Big)^2 ||\bar{X}||^2
\end{aligned}
$$

where $t_v, t_{v'} \in [0,1]$. In this case, note that:

- When $\alpha = 0$, this difference writes as: $||(SX)_{u\cdot} - (SX)_{u'\cdot}||^2 = d_u - d_{u'}$, and is thus extremely sensitive to the degree of the nodes,
- When $\alpha = 1$, the difference can be written as:

$$
\begin{aligned}
&||(SX)_{u\cdot} - (SX)_{u'\cdot}||^2 \\
&= \Big(\frac{1}{d_u+\beta} - \overline{\Delta}_u \\
&\quad + \frac{1}{(d_u+\beta)^4} \sum_{v \in \tilde{\mathcal{N}}(u)} \frac{A_{uv}(d_v-d_u)^2}{((1-t_v)+t_v \frac{\Delta_v}{d_u+\beta}+\beta)^4} \\
&\quad - \frac{1}{d_{u'}+\beta} + \overline{\Delta}_{u'} \\
&\quad - \frac{1}{(d_{u'}+\beta)^4} \sum_{v' \in \tilde{\mathcal{N}}(u')} \frac{A_{u'v'}(d'_v-d'_u)^2}{((1-t_{v'})+t_{v'} \frac{\Delta_{v'}}{d_{u'}+\beta}+\beta)^4} \Big)^2 ||\bar{X}||^2
\end{aligned}
$$

  In this case, the leading terms are functions of the inverse of the node degrees $\frac{1}{d_u+\beta} - \frac{1}{d_{u'}+\beta}$ and the difference in local homogeneity of topology $\overline{\Delta}_u - \overline{\Delta}_{u'}$. Consequently, the distance is still sensitive to topological properties of the neighborhood.
- When $\alpha = 0.5$: in this case, the distance writes as:

$$
\begin{aligned}
||(SX)_{u\cdot} - (SX)_{u'\cdot}||^2 = \Big(&\frac{1}{2}\overline{\Delta}_{u'} - \frac{1}{2}\overline{\Delta}_u + \frac{3}{8(d_u+\beta)^3} \sum_{v \in \tilde{\mathcal{N}}(u)} \frac{A_{uv}(d_v-d_u)^2}{((1-t_v)+t_v \frac{\Delta_v}{d_u+\beta}+\beta)^3} \\
&- \frac{3}{8(d_{u'}+\beta)^3} \sum_{v' \in \tilde{\mathcal{N}}(u')} \frac{A_{u'v'}(d_{v'}-d_{u'})^2}{((1-t_{v'})+t_{v'} \frac{\Delta_{v'}}{d_{u'}+\beta}+\beta)^3} \Big)^2 ||\bar{X}||^2
\end{aligned}
$$

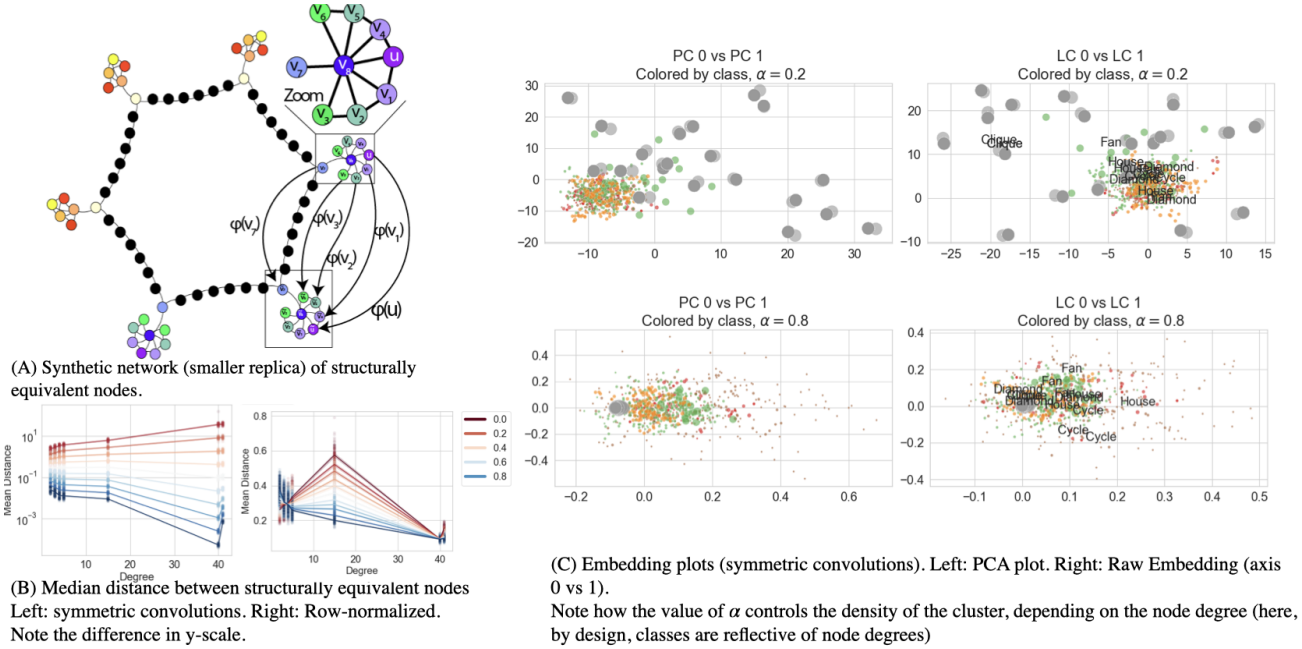
Consequently, this distance is less directly related to the degree of the node, and relies more on the topological traits of the neighborhood.

In the regularised case:

$$||(SXW_1 + b_1)_u - (SXW_1 + b_1)_{u'}||^2 = 0 \tag{7}$$

In this case, the distance is entirely driven by the features.

Results of the experiments



(A) Synthetic network (smaller replica) of structurally equivalent nodes.

(B) Median distance between structurally equivalent nodes
Left: symmetric convolutions. Right: Row-normalized.
Note the difference in y-scale.

(C) Embedding plots (symmetric convolutions). Left: PCA plot. Right: Raw Embedding (axis 0 vs 1).
Note how the value of α controls the density of the cluster, depending on the node degree (here, by design, classes are reflective of node degrees)

Figure 1: Results for our Structural Equivalents experiment

D PROOFS OF SECTION 5

To better formalize our setting, we propose considering a specific family of graphs: the degree-corrected Stochastic Block Model Karrer and Newman [2011] on two classes of equal size n. Let each node have class $Z_i \in \{1, 2\}$, and denote $X_i = \mu^{(Z_i)} + \epsilon_i$ its attributes. According to the DC-SBM model, each edge in the network is sampled according to a Bernouilli distribution: $A_{ij} \sim \text{Bernouilli}(\theta_i \theta_j \omega_{Z_i Z_j})$, where θ_i is a popularity parameter such that, for each group g: $\sum_{i=1}^{n} \theta_i 1_{Z_i=g} = n$, where ω_{ij} is the parameter of the model corresponding to the probability of connection between group i and j. Note that, under this model, the expected number of edges from community (i) to (j) is simply $m_{ij} = n^2 w_{ij}$. Therefore, picking $\forall i, \theta_i = 1$ corresponds to the traditional stochastic block model. We will also assume that $\forall i, \theta_i \in [\frac{1}{\kappa}, \kappa]$ where $\kappa > 1$. In other words, the degree distribution cannot be too skewed.

The degree of each node i can thus be rewritten as:

$$d_i = \sum_{j=1, j\neq i}^{n} A_{ij}$$

$$\mathbb{E}[d_i] = \sum_{j=1,\neq i}^{n} \theta_i \theta_j \omega_{Z_i Z_j} = \theta_i[(n - \theta_i)\omega_{11} + n\omega_{12}] = n\theta_i[(1 - \frac{\theta_i}{n})\omega_{11} + \omega_{12}] = \theta_i \bar{\omega} n + o(1).$$

where $\bar{\omega} = \omega_{11} + \omega_{12}$.

A trivial application of the bounded difference inequality shows that the scaled degree $n^{-1} d_i$ concentrates rapidly around its mean (see Wainwright [2019] Chapter 2):

$$\mathbb{P}[\frac{1}{n}|d_i - \mathbb{E}[d_i]| > t] \leq 2e^{-2nt^2}$$

Effect of the convolution under the DCSBM model Let us now focus on the effect of the convolution. We have:

$$
\begin{aligned}
Z_i &= \frac{\beta}{(\beta+d_i)^{2\alpha}}X_i + \sum_{j=1,j\neq i}^{n} \frac{A_{ij}}{(\sum_{k\neq j,i} A_{ki} + A_{ij} + \beta)^{\alpha}(\sum_{k\neq j,i} A_{kj} + A_{ij} + \beta)^{\alpha}}X_j \\
\implies \mathbb{E}[Z_i] &= \mathbb{E}[\frac{\beta}{(\beta+d_i)^{2\alpha}}]\mathbb{E}[X_i] + \sum_{j=1,j\neq i}^{n} \mathbb{E}\big[\mathbb{E}[\frac{A_{ij}}{(z_i+A_{ij})^{\alpha}(z_j+A_{ij})^{\alpha}}\Big|\sum_{k\neq j,i} A_{ki} + \beta = z_i, \sum_{k\neq j,i} A_{ki} + \beta = z_j]\big]\mathbb{E}[X_j]
\end{aligned}
\tag{8}
$$

Consider the term $\mathbb{E}[\frac{A_{ij}}{(z_i+A_{ij})^{\alpha}(z_j+A_{ij})^{\alpha}}\Big|\beta + \sum_{k\neq j,i} A_{ki} = z_i, \beta + \sum_{k\neq j,i} A_{ki} = z_j]$. This is a binary variable, with value $\frac{1}{(z_i+1)^{\alpha}(z_j+1)^{\alpha}}$ with probability $\theta_i\theta_j\omega_{Z_iZ_j}$, and 0 otherwise. Therefore:

$$
\mathbb{E}\Big[\frac{A_{ij}}{(z_i+A_{ij})^{\alpha}(z_j+A_{ij})^{\alpha}}\Big|\beta + \sum_{k\neq j,i} A_{ki} = z_i, \beta + \sum_{k\neq j,i} A_{ki} = z_j\Big] = \frac{\theta_i\theta_j\omega_{Z_iZ_j}}{(z_i+1)^{\alpha}(z_j+1)^{\alpha}}
$$

Thus the trick becomes to characterize the behaviour of the random variable $\tilde{Y} = \frac{1}{(z_i+1)^{\alpha}(z_j+1)^{\alpha}}$. Note that, by construction, z_j and z_i are independent of one another. Since the function $\phi : x \to x^{-\alpha}$ is continuous, by the continuous mapping theorem, we know that $\phi(X_n)$ converges to $\phi(\mathbb{E}[X]) = \frac{1}{\mathbb{E}[\lim_{n\to\infty} X_n]}$. Here, we have shown above that:

$$
\lim_{n\to\infty} \frac{d_i}{n} = \theta_i\bar{\omega}.
$$

Therefore, assuming (without loss of generality) that $Z_i = 1$, so that $\mathbb{E}[X_i] = \mu^{(1)}$:

$$
\begin{aligned}
n^{2\alpha}\mathbb{E}[Z_i] &= \mathbb{E}[\frac{\beta}{(\frac{\beta}{n} + \frac{d_i}{n})^{2\alpha}}]\mathbb{E}[X_i] + \sum_{j=1,j\neq i}^{n} \mathbb{E}\big[\frac{\theta_i\theta_j\omega_{Z_iZ_j}}{(\frac{z_i}{n} + \frac{1}{n})^{\alpha}(\frac{z_j}{n} + \frac{1}{n})^{\alpha}}\big]\mathbb{E}[X_j] \\
n^{2\alpha}\mathbb{E}[Z_i] &= \frac{\beta}{(\bar{\omega}\theta_i)^{2\alpha}}\mathbb{E}[X_i] + \sum_{j=1,j\neq i}^{n} \frac{\theta_i\theta_j\omega_{Z_iZ_j}}{\theta_i^{\alpha}\theta_j^{\alpha}\bar{\omega}^{2\alpha}}\mu^{(Z_j)} + O(1)
\end{aligned}
\tag{9}
$$

Therefore

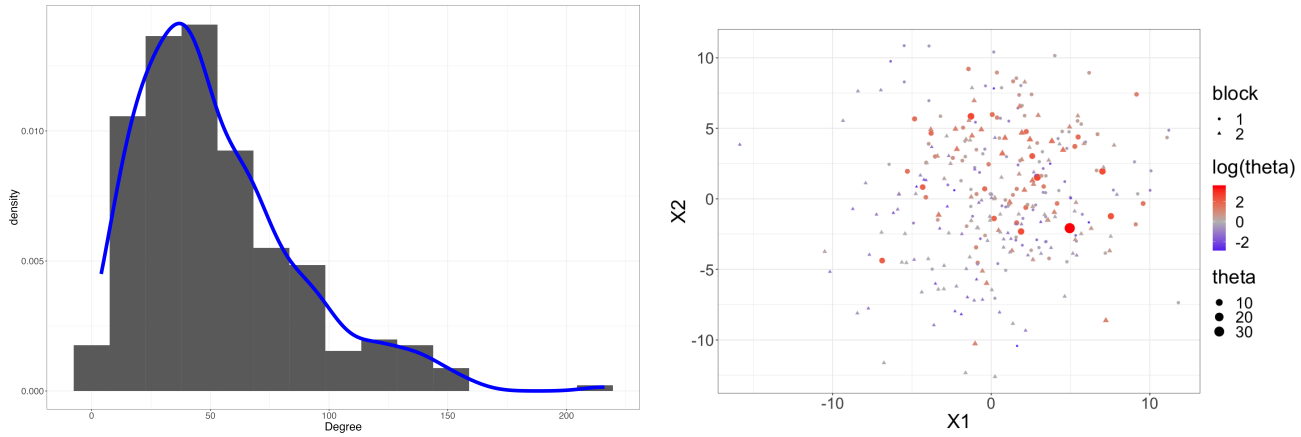
$$
\begin{aligned}
\mathbb{E}[Z_i] &= \frac{\beta}{n^{2\alpha}\bar{\omega}^{2\alpha}\theta_i^{2\alpha}}\mu^{(1)} + n^{-2\alpha}\theta_i^{1-\alpha}\frac{\omega_{11}}{\bar{\omega}^{2\alpha}} \sum_{j\neq i, Z_j = Z_i} \theta_j^{1-\alpha}\mu^{(1)} + +n^{-2\alpha}\theta_i^{1-\alpha}\frac{\omega_{12}}{\bar{\omega}^{2\alpha}} \sum_{j\neq i, Z_j \neq Z_i} \theta_j^{1-\alpha}\mu^{(2)} + o(n^{-2\alpha}) \\
&= \frac{\beta}{n^{2\alpha}\bar{\omega}^{2}\theta_i^{2}}\mu^{(1)} + n^{-2\alpha}\theta_i^{1-\alpha}\frac{\omega_{11}}{\bar{\omega}^{2\alpha}}(S_1 - \theta_i^{1-\alpha}\mu^{(1)})) + n^{-2\alpha}\theta_i^{1-\alpha}\frac{\omega_{12}}{\bar{\omega}^{2\alpha}}(S_2 - \theta_i^{1-\alpha}\mu^{(2)}) + o(n^{-2\alpha})
\end{aligned}
\tag{10}
$$

This shows that the embedding scales as $n^{1-2\alpha}$: for $\alpha = 1$, we see that the embedding will converge to 0, as is observed empirically. Reciprocally, for $\alpha = 0$, the embedding can expand. This expression is interesting as well. As we can see, the embedding is directly proportional to $\theta_i^{1-\alpha}$. Consequently, for $\alpha = 1$, the leading term is independent of θ_i. Reciprocally, for $\alpha = 0$, the embedding is directly proportional to θ_i.

To see this, we provide the following example. Consider a DC-SBM graph on 300 nodes with two classes, with connectivity parameters $\omega_{11} = \omega_{22} = 0.1$ and $\omega_{11} = \omega_{22} = 0.005$. The features here are taken to be multivariate normal with $\mu^{(1)} = 2, \mu^{(2)} = -2$ and standard deviation equal to 4. We generate the θ_i for each group from a lognormal distribution, with mean 0 and standard deviation 3. The histogram of the degree distribution is provided in Figure 2a, along with a plot of the original features $X \in \mathbb{R}^{n\times 2}$ in Figure 2b.

$$
\mathbb{E}[\frac{A_{ij}}{d_i^{\alpha}}] = \mathbb{E}[\frac{\theta_i\theta_j\omega_{Z_iZ_j}}{\mathbb{E}[d_i] + W}] \leq \frac{\theta_i\theta_j\omega_{Z_iZ_j}}{\mathbb{E}[d_i]}(1 - \frac{W}{\mathbb{E}[d_i]} + \frac{W^2}{\mathbb{E}[d_i]^2})
\tag{11}
$$

So as long as the fluctuations around the mean are controlled, the entire expression remains manageable.



(a) Degree distribution

(b) Original node attributes.

Figure 2: Degree distribution and raw attributes in the DC-SBM serving as our example.

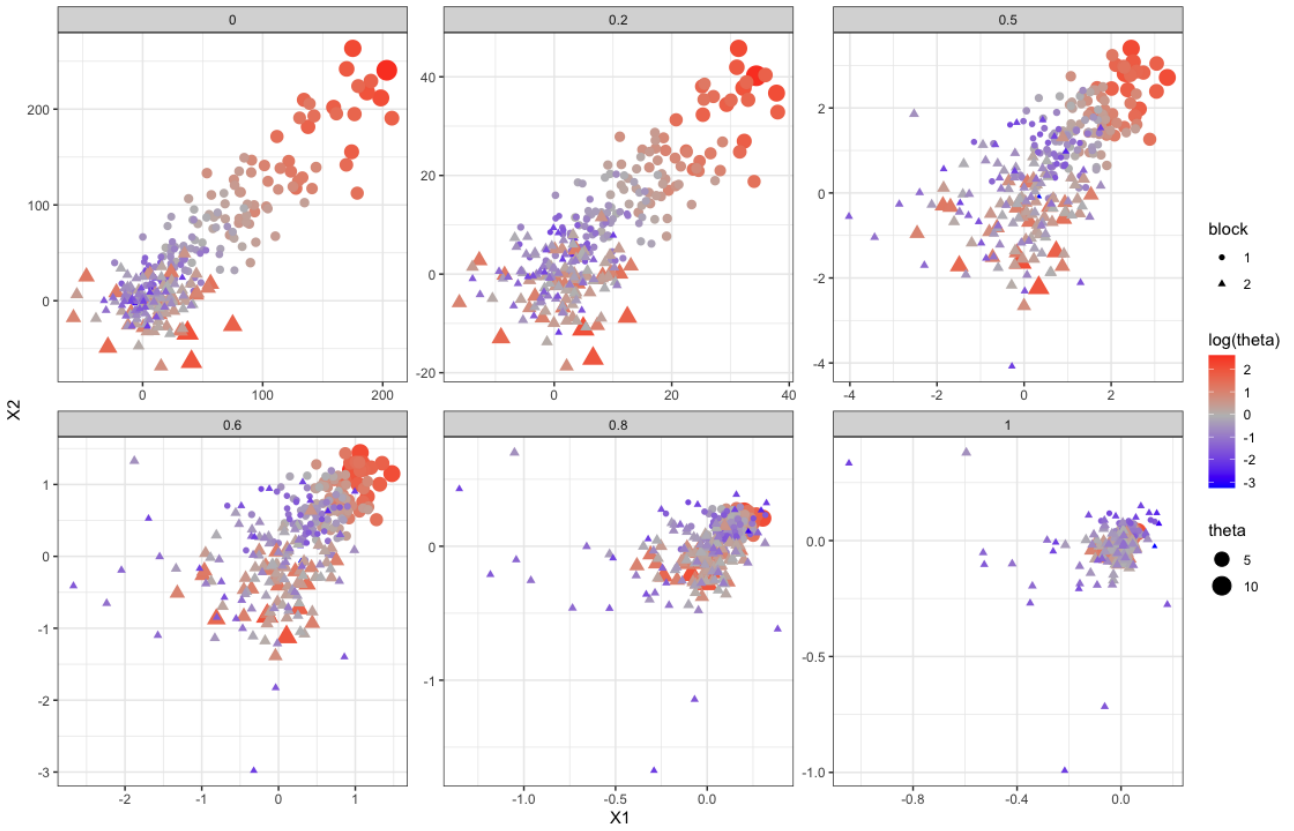


Figure 3: Attributes after convolution for different values of α and $\beta = 1$.

For Row-normalized convolutions:

$$
\begin{aligned}
H^{(1)} &= \mathbb{E}[\sum_{i\in G_1}\frac{A_{ij}}{\beta+\sum_k A_{ik}}X + \sum_{i\in G_2}^{n}\frac{A_{ij}}{\beta+\sum_k A_{ik}}X] \\
&= \mathbb{E}[(\frac{\beta}{(\beta+d_i)^{2\alpha}} + \sum_{j\in G_1}\frac{1}{(\beta+d_i)^{\alpha}}\frac{1}{(\beta+d_j)^{\alpha}})\mu^{(1)} \\
&+ \sum_{i\in G_2}^{n}\frac{\theta_i\theta_j q}{\beta+\theta_i(\sum_j\theta_j}\mu^{(2)}]
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
H^{(1)} &= \sum_{i\in G_1}\frac{\theta_i\theta_j p}{\theta_i^{\alpha}(m_1-\theta_i+m_2)^{\alpha}\theta_j^{\alpha}(m_1-\theta_j+m_2)^{\alpha}}\mu^{(1)} \\
&+ \sum_{i\in G_2}^{n}\frac{\theta_i\theta_j q}{\theta_i^{\alpha}(m_1-\theta_i+m_2)^{\alpha}\theta_j^{\alpha}(m_1-\theta_j+m_2)^{\alpha}}\mu^{(2)} \\
&= \theta_i^{1-\alpha}\Big(\sum_{i\in G_1}\frac{\theta_j^{1-\alpha}p}{(m_1-\theta_i+m_2)^{\alpha}(m_1-\theta_j+m_2)^{\alpha}}\mu^{(1)} \\
&+ \sum_{i\in G_2}^{n}\frac{\theta_j^{1-\alpha}q}{(m_1-\theta_i+m_2)^{\alpha}(m_1-\theta_j+m_2)^{\alpha}}\mu^{(2)}\Big)
\end{aligned}
\tag{13}
$$

Several cases:

- When $\alpha = 0$:

$$H^{(1)} = \theta_i\Big(\sum_{i\in G_1}\theta_j p\mu^{(1)} + \sum_{i\in G_2}^{n}\theta_j q\mu^{(2)}\Big)$$

$$H^{(1)} = \theta_i(m_1-\theta_i)p\mu^{(1)} + \theta_i m_2 q\mu^{(2)}$$

- When $\alpha = 1$:

$$H^{(1)} = \Big(\frac{p}{(m_1+m_2)^2}\mu^{(1)} + \frac{q}{(m_1+m_2)^2}\mu^{(1)})$$

We also have:

$$
\begin{aligned}
A &= \sum_{i\in G_1}\frac{\theta_i^{1-\alpha}\theta_j^{1-\alpha}p}{(m_1+m_2)^{2\alpha}}(1-\alpha\frac{\theta_i}{m_1})(1-\alpha\frac{\theta_j}{m_1})\mu^{(1)} \\
&+ \sum_{i\in G_2}\frac{\theta_i^{1-\alpha}\theta_j^{1-\alpha}q}{(m_1+m_2)^{2\alpha}}(1-\alpha\frac{\theta_i}{m_1})(1-\alpha\frac{\theta_j}{m_1})\mu^{(2)}
\end{aligned}
\tag{14}
$$

For Row-normalized convolutions:

$$
\begin{aligned}
A &= \theta_i^{1-\alpha}\Big[\sum_{i\in G_1}\frac{\theta_j^{1-\alpha}p}{(m_1+m_2)^{2\alpha}}(1-\alpha(\frac{\theta_i}{m_1}+\frac{\theta_j}{m_1})) \\
&+ \sum_{i\in G_2}\frac{\theta_j^{1-\alpha}p}{(m_1+m_2)^{2\alpha}}1-\alpha(\frac{\theta_i}{m_1}+\frac{\theta_j}{m_1}))\Big] \\
B/A &= \frac{\theta_j^{1-\alpha}p}{(m_1+m_2)^{2\alpha}}(1-\alpha(\frac{\theta_i}{m_1}+\frac{\theta_j}{m_1}))/\Big[\sum_{i\in G_1}\frac{\theta_j^{1-\alpha}p}{(m_1+m_2)^{2\alpha}}(1-\alpha(\frac{\theta_i}{m_1}+\frac{\theta_j}{m_1})) \\
&+ \sum_{i\in G_2}\frac{\theta_j^{1-\alpha}p}{(m_1+m_2)^{2\alpha}}1-\alpha(\frac{\theta_i}{m_1}+\frac{\theta_j}{m_1}))\Big]
\end{aligned}
\tag{15}
$$

Name	Node	Edge	Features	Class	Avg. Degree	Mean Centrality	$\mathbf{h}_{edges}$	$\mathbf{h}_{nodes}$
Cora	2,708	10,556	1,433	7	3.90	1.65E-03	0.81	0.83
pubMed	19,717	88,648	500	3	4.50	2.71E-04	0.80	0.79
Citeseer	3,327	*9,104*	3,703	6	2.74	1.02E-03	0.74	0.71
Coauthor CS	18,333	163,788	6,805	15	8.93	2.42E-04	0.81	0.83
Amazon Photos	7,650	238,162	745	8	31.13	3.82E-04	0.83	0.84
Actor	7,600	30,019	932	5	3.95	3.18E-04	0.22	0.21
Cornell	183	280	1,703	5	1.53	1.07E-04	0.31	0.21
Wisconsin	251	515	1,703	5	2.05	2.42E-04	0.20	0.13
PATTERN	108	4,884	3	2	45.22	5.45E-03	0.67	0.69
CLUSTER	117	4,104	7	6	35.08	6.07E-03	0.37	0.36
WikiCS	11,701	297,110	300	10	25.39	1.76E-04	0.69	0.64
OGBN-arxiv	169,343	1,166,243	128	40	6.89		0.66	

Table 2. Statistics for datasets used for experiments. Node and edge homophily indices are calculated by the formula suggested in [Pei et al., 2020], [Zhu et al., 2020a] respectively.

$$
\begin{aligned}
A = &\sum_{i \in G_1} \frac{\theta_i^{1-\alpha}\theta_j^{1-\alpha}p}{(m_1+m_2)^{2\alpha}}(1-\alpha\frac{\theta_i}{m_1})(1-\alpha\frac{\theta_j}{m_1})\mu^{(1)} \\
&+ \sum_{i \in G_2} \frac{\theta_i^{1-\alpha}\theta_j^{1-\alpha}q}{(m_1+m_2)^{2\alpha}}(1-\alpha\frac{\theta_i}{m_1})(1-\alpha\frac{\theta_j}{m_1})\mu^{(2)}
\end{aligned}
\tag{16}
$$

Maybe a good way of understanding things is through a sensitivity analysis.

E FURTHER RESULTS AND EXPERIMENTS

We analyzed the impact of the choice of operator across α and β using standard benchmark datasets. In particular, the node classification task has been performed. The performance turns out to be dependent on the choice of an operator as well as the inherent characteristics of the datasets. We use visualizations to further investigate the properties of each embedding space in relation to the choice of operator. The code for the experiments can be found here

E.1 DATASET STATISTICS

Datasets We used twelve datasets for experiments including eight standard benchmark datasets, namely, Cora, Pubmed, Citeseer, and Amazon Photos, Coauthor CS, and four novel benchmarks proposed by [Dwivedi et al., 2020]. These datasets include ,in particular, synthetic graphs (PATTERN, CLUSTER) — which offer a more controlled environment to perform experiments—, as well as social/academic networks such as WikiCS and OGBN-arxiv. We use the processed version provided by PyTorch Geometric[Fey and Lenssen, 2019]. Detailed statistics for the datasets used in the experiments are shown in Table 2.

We further expand on our experimental results by considering the benchmarks proposed by .

Citation networks. Cora, Citeseer, and Pubmed are standard citation network benchmark datasets.[Yang et al., 2016] In these networks, nodes represent scientific publications, and edges denote citation links between publications. Node features are the bag-of-words representation of papers, and node label is the academic topic of a paper.

Coauthor In Coauthor CS[Shchur et al., 2018] network, each node represents the author of the scientific publication, and edge shows whether any of the authors coauthored the paper. Node features are bag-of-word representations of these documents, and node labels denote the field of study.

Amazon In Amazon Photo[Shchur et al., 2018] network, nodes represent goods and edges show whether two goods are frequently bought together. Node features are bag-of-word representation of product reviews.

WebKB. WebKBCra is a webpage dataset collected from computer science departments of various universities by Carnegie

Dataset	Batch Normalized	Num of training nodes	Learning rate	Epoch	Num of exp	Training time(sec)	Dim of hidden layers	Num of GCN layers
Cora	**X**	140	0.02	200	50	9.58	32	2
pubMed	**X**	140	0.001	500	30	100.49	32	2
Citeseer	**X**	1694	0.05	500	30	31.73	32	2
Coauthor CS	**X**	9194	0.05	200	30	150.68	32	2
Amazon Photos	**X**	3844	0.05	200	30	88.04	32	2
Actor	**X**	3804	0.02	200	30	13.02	32	2
Cornell	**X**	87	0.01	500	30	2.95	32	2
Wisconsin	**X**	126	0.001	500	30	3.52	32	2
PATTERN	**O**	42	0.01	300	10	5.39	32	2
	X	42	0.01	200	30	1.38	32	2
CLUSTER	**O**	47	0.005	500	10	8.07	32	2
	X	47	0.005	500	30	4.36	32	2
WikiCS	**O**	5851	0.001	300	5	916.48	120	2
	X	5851	0.1	200	30	124.96	32	2
OGBN-arxiv	**X**	16124	0.005	500	5	7496.92	64	2

Table 3. Hyperparameters and training details for all datasets. Training time(sec) is the training time for the first epoch applying neither normalization nor regularization.

Mellon University. We use Cornell, and Wisconsin among them. Nodes represent web pages, and edges are hyperlinks between them. Node features are the bag-of-words representation of web pages. The web pages are manually classified into the five categories: student, project, course, staff, and faculty.

Cooccurrence network Actor dataset is the actor-only induced subgraph of the film-director-actor-writer network[Pei et al., 2020]. Each node corresponds to an actor, and the edge between two nodes denotes co-occurrence on the same Wikipedia page. Node features correspond to corresponding Wikipedia keywords.

PATTERN and CLUSTER We used the processed version provided by PyTorch Geometric[Fey and Lenssen, 2019]. In [Dwivedi et al., 2020], the 10,000 training graphs are used to train the model for node classification task. In our experiments, we only used the first graph from the respective datasets and randomly split the training and test nodes within the graph for each training epoch.

WikiCS. We used the processed version provided by PyTorch Geometric[Fey and Lenssen, 2019].

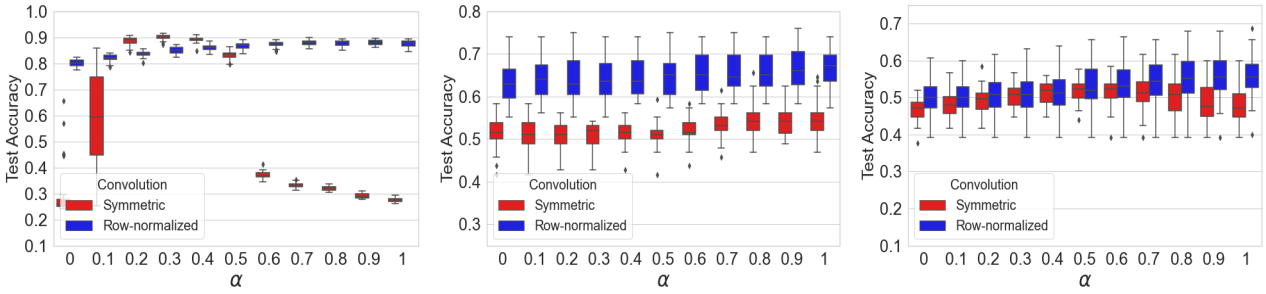
OGBN-arxiv. We used dataset from *Open Graph Benchmark github repository* [Hu et al., 2020].

E.2 EXPERIMENT SETUP

Models We use a two-layer GCN[Kipf and Welling, 2016] model with varying families of spatial convolution operator across the choice of α while keeping $\beta \in \{0, 1\}$. For each experiment, we randomly split the data into training and test sets (using the default number of train and test points in Pytorch geometric). The number of training nodes used are specified in Table 3.

The GCN model in [Dwivedi et al., 2020] uses batch normalization between GCN layers, unlike our experiments for standard benchmark datasets(e.g. Cora, Pubmed, Citeseer). Batch normalization alters the geometry of the embedding space, which is the main focus of this paper. Consequently, to enable the comparison between the experiments presented in this section and these on traditional benchmark datasets in the last, we train the model with and without batch normalization. Further training details including data split, number of experiments and learning rate are also summarized in Table 3.

Hardware and Software Specifications. Our models are implemented with Python 3.8.8, PyTorch Geometric 2.0.5 [Fey and Lenssen, 2019], and PyTorch 1.10.0 [Paszke et al., 2019]. We conduct experiments on a computer equipped with 2.3 GHz Quad-Core Intel Core i7 processor and Intel Iris Plus Graphics 1536 MB.



(a) Amazon Photos ($h_e = 0.83$) (b) Cornell ($h_e = 0.30$) (c) Wisconsin ($h_e = 0.21$)

Figure 4. Effect of α on the performance of the algorithm for our family of convolutions defined in Eq.2 and Eq.3 (30 independent experiments, with random training and test set). Here, h_e denotes the edge homophily in the dataset (defined as the fraction of edges whose vertices share the same label) Note the strong dependency of the results on α. See Appendix E for further details and results.

E.3 EXPERIMENT RESULTS

In this section, we highlight the results of our experiments on the various datasets aforementioned. Additional plots are provided in the folder of supplementary materials associated with this paper.

E.3.1 Node Classification

First, we want to investigate the impact of the choice of operators on the node classification task. We observe that the performance of the node classification task varies by choice of α. We fix $\beta = 1$ — in other words, we add self-loops, consistently with the standard GCN architecture. In general, we observe that performance are highly dependent on the choice of α especially for the symmetrized operator, but the performance of node classification task of row-normalized operator is relatively robust to the choice of α

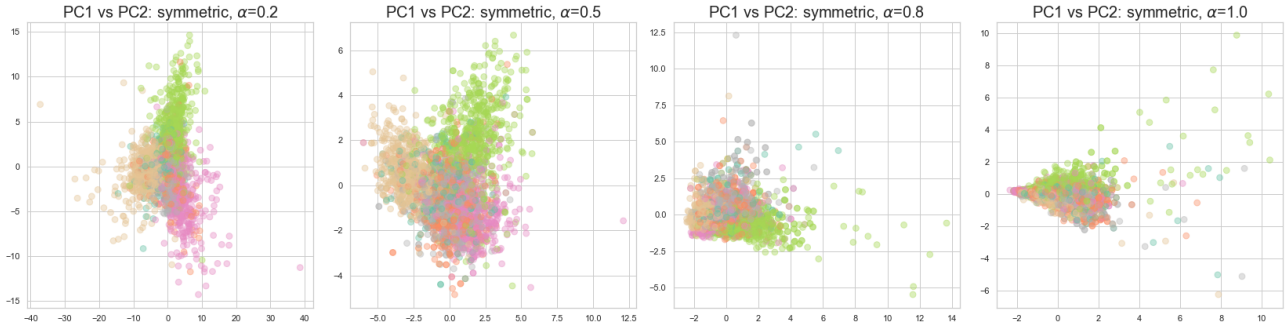
Analysis For standard homophilic datasets such as Citeseer(Figure 5), clustering of each node class has become less identifiable for a symmetric operator when α increases. On the other hand, the node class has been well separated across the alpha for the row-normalized operator– the row-normalized operator is robust to the choice of α when it comes to node classification performance. Conversely, for the datasets with low homophily shown in Table 3, such as Wisconsin(Figure 6), the separation of the node label does not change much depending on the choice of α or the choice of operator. The visual inspections on the embedding space transformed by PCA and UMAP are in line with the numerical result of test accuracy, Figure 7.

Consistently with the results for standard benchmark sets, we observe that the performance of the model also depends on the choice of α for *PATTERN*, *CLUSTER*, *WikiCS*, and*OGBN-arxiv*(see Table-1). Without batch normalization, the row-normalized convolution are quite stable (the variation in α only induces gains in accuracy of 8% for Cluster, and 3% for WikiCS). By contrast, the tuning of α has a more dramatic effect on the performance, yielding increases of up to 81% and 17% for these two datasets. Not only the performance differs by the choice of α, the resulting embedding space is also affected. As in Figure 12, Figure 13 shows the analogous arrangement of embedded nodes by their node degree. Particularly, the high degree nodes are concentrated to the origin as α increases, and the lower degree nodes are located at the margin of the embedding space for both symmetric and row-normalized operators. Overall, these experiments confirm the phenomena observed in the previous section.

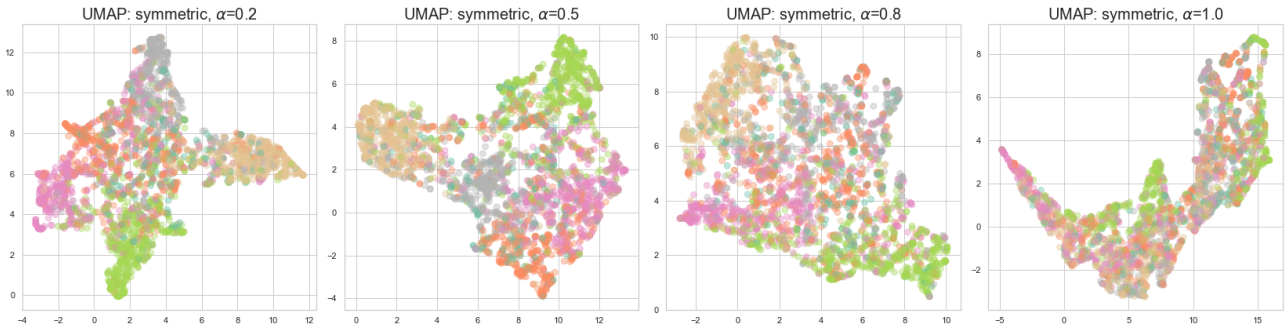
E.3.2 Degree

In this subsection, we propose to investigate how basic topological characteristics (more specifically, here, the node degree) drive the organization of the embedding space. Consequently, to complement the analysis performed in the main text, we conduct visual inspection on the embedding plots of our benchmark datasets colored by node degree.

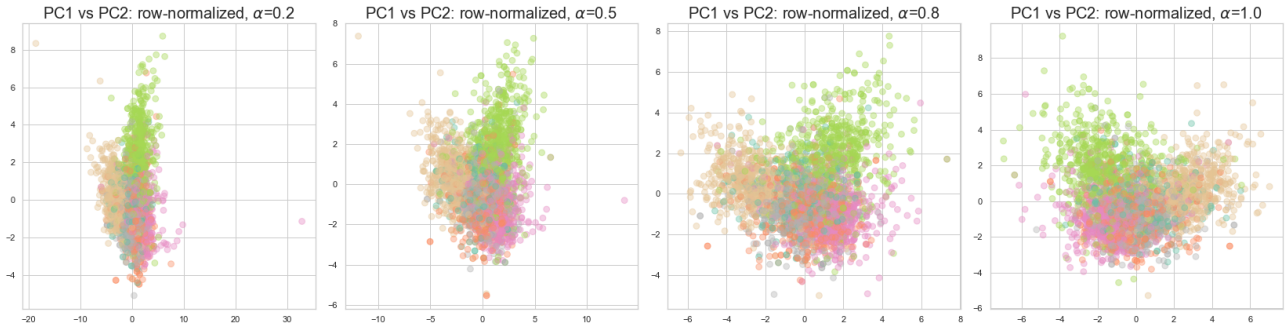
Figures 11 and 12 show the embedding spaces transformed by PCA, and the size and color of points denote the node degree. It is noted that the high degree nodes are marginalized when α close to 0, and lower degree nodes tend to be located at the origin. As α gets closer to 1, this pattern seems to be reverted— the higher degree nodes are located at the origin and the



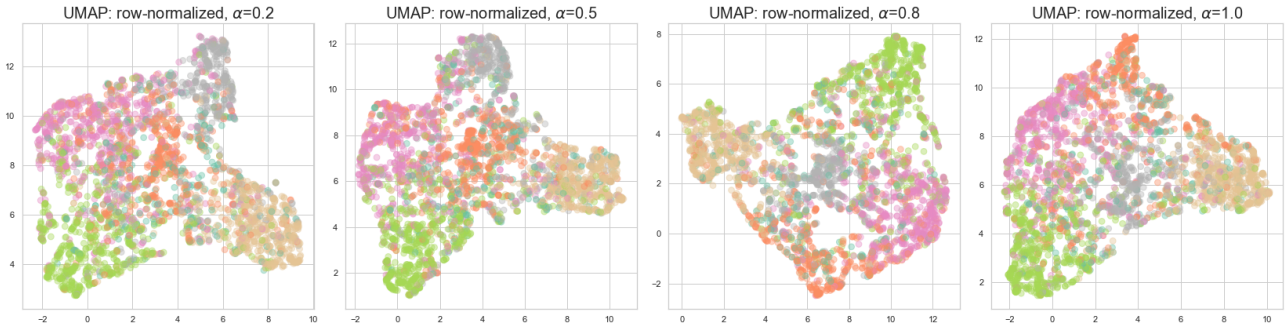
(a) Citeseer, symmetrized, PCA, colored by node label



(b) Citeseer, symmetrized, UMAP, colored by node label

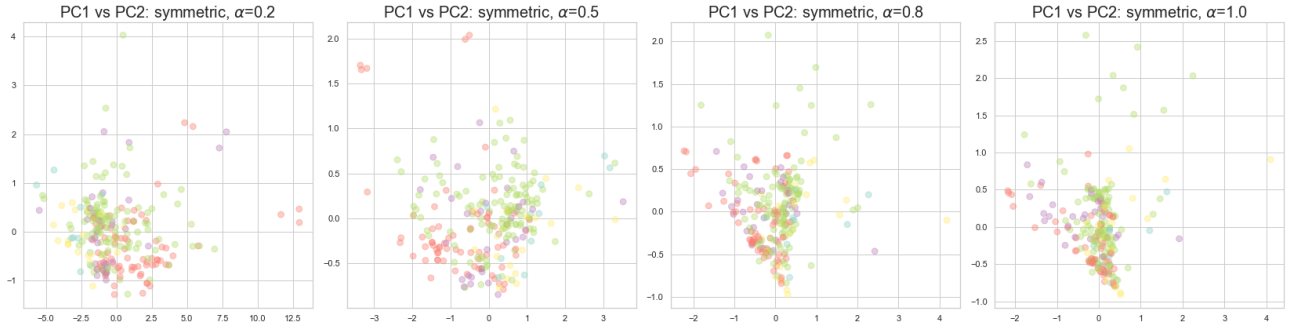


(c) Citeseer, row-normalized, PCA, colored by node label

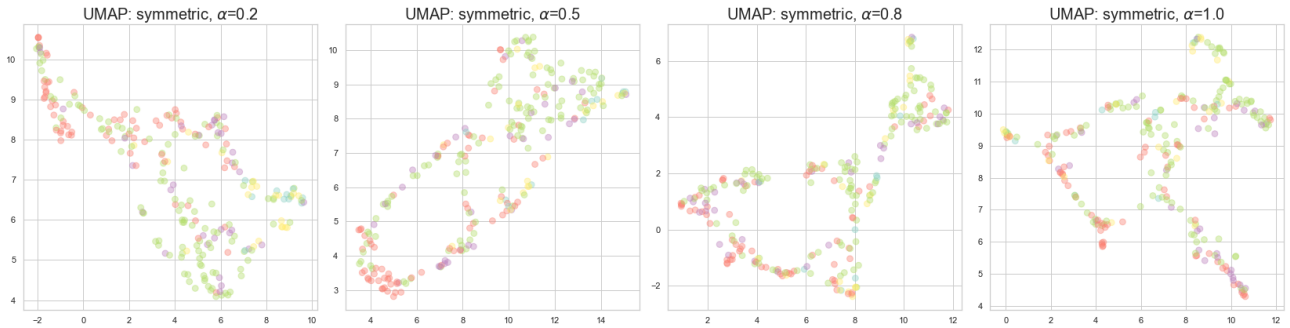


(d) Citeseer, row-normalized, UMAP, colored by node label

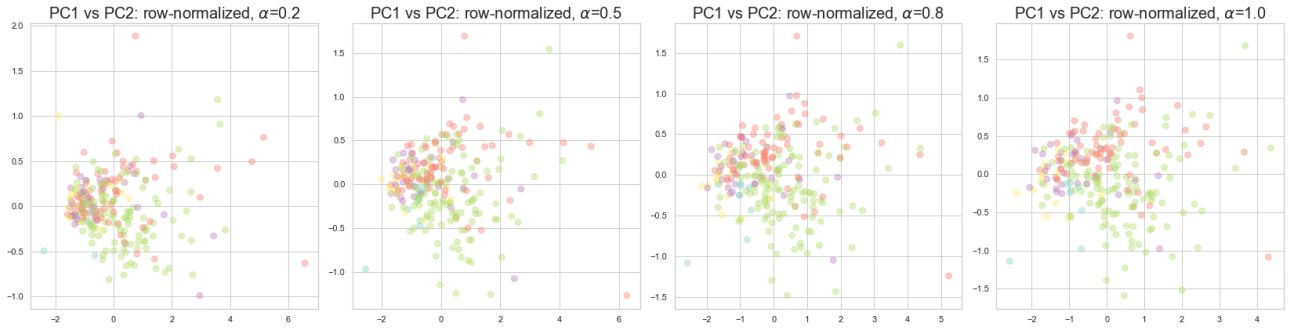
Figure 5. Citeseer. The plots are colored by node labels(product categories). Embedding spaces generated by symmetric operator, (a), (b), as α increases the level of distinction between the cluster of different node labels decreases. Embedding space generated by row-normalized operator seems to be robust to the choice of α– it gives relatively constant level of clustering regardless of α.



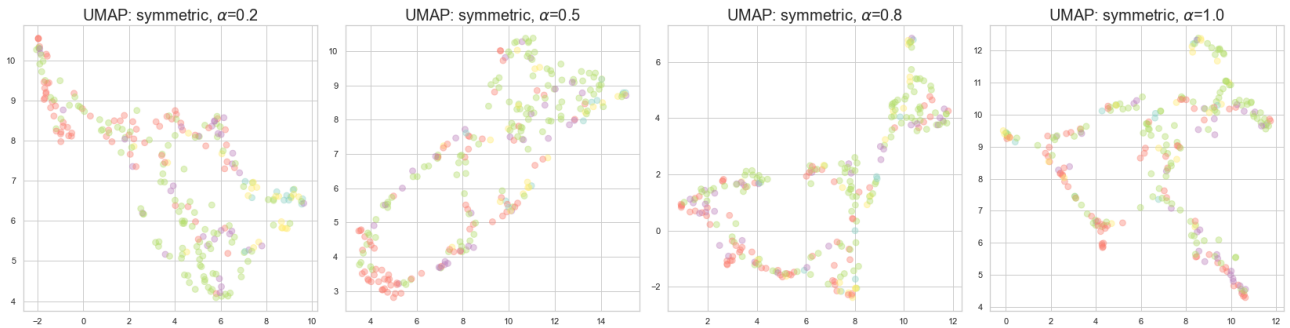
(a) Wisconsin, symmetrized, PCA, colored by node label



(b) Wisconsin, symmetrized, UMAP, colored by node label

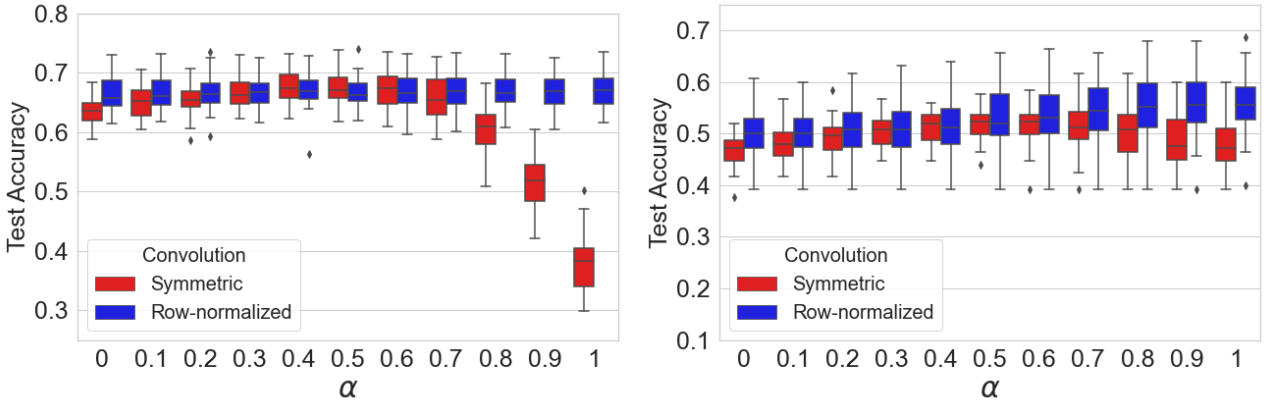


(c) Wisconsin, row-normalized, PCA, colored by node label



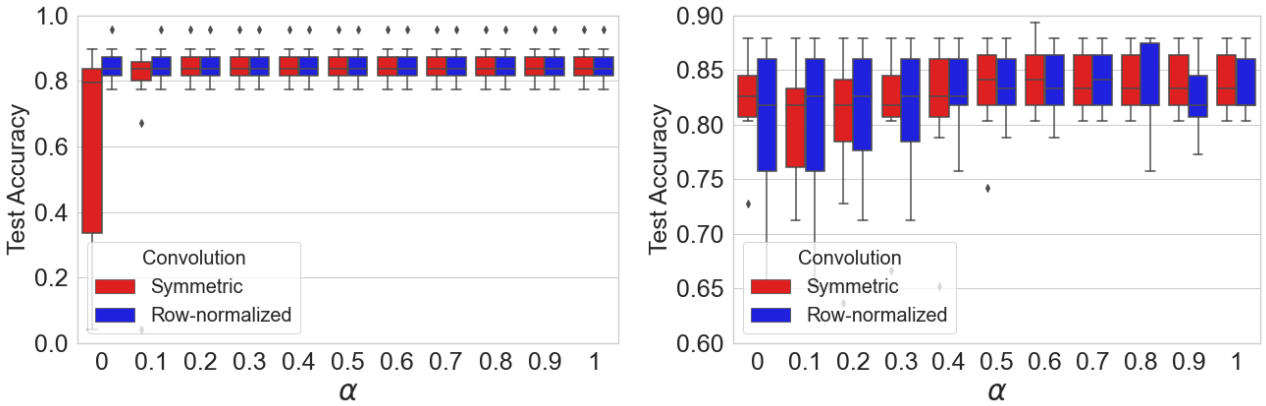
(d) Wisconsin, row-normalized, UMAP, colored by node label

Figure 6. Wisconsin.The plots are colored by node label(categories for the webpage). Unlike the graph in Figure 8, it is hard to detect the change in the level of clustering or separation of each node class as α varies. PCA transformed embedding plot for row-normalized operator (c) even shows that the clustering of node label improves as α gets closer to 1.



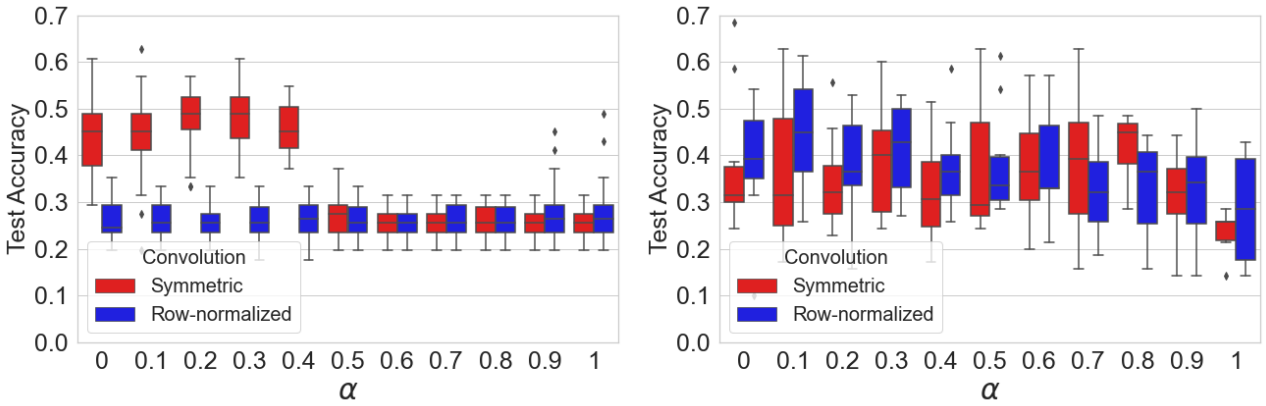
(a) Citeseer (b) Wisconsin

Figure 7: Test accuracy for node classification task on two datasets: Citeseer and Wisconsin.



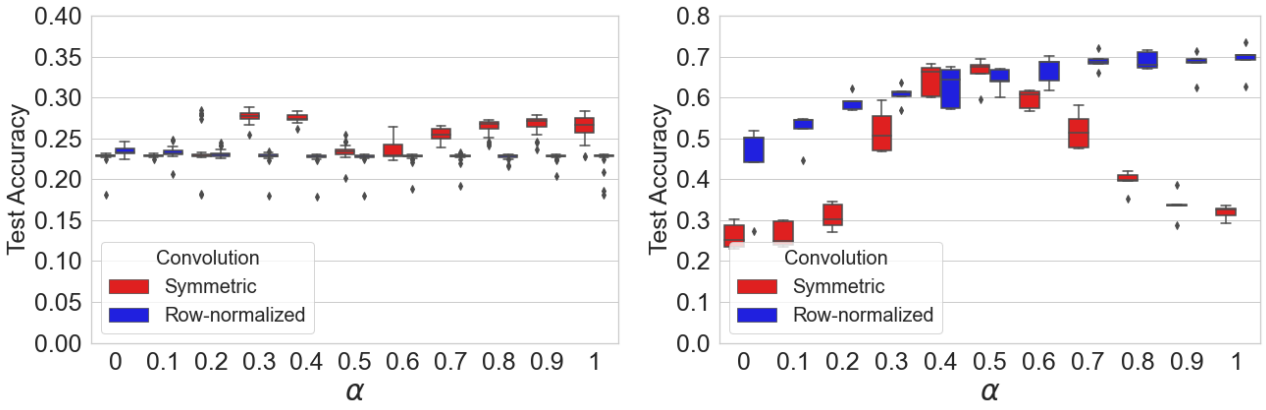
(a) PATTERN without Batch Normalization (b) PATTERN with Batch Normalization

Figure 8. Test accuracy for node classification task. Only the first graph of PATTERN dataset is used, and the nodes within the graph are randomly split into training and test data per each training epoch.



(a) CLUSTER without Batch Normalization (b) CLUSTER with Batch Normalization

Figure 9. Test accuracy for node classification task. Only the first graph of CLUSTER dataset is used, and the nodes within the graph are randomly split into training and test data per each training epoch.



(a) WikiCS without Batch Normalization

(b) WikiCS with Batch Normalization

Figure 10: Test accuracy for node classification task on WikiCS

lower degree nodes are pushed out to be at the periphery.

E.3.3 Distance to the Original Space

In this subsection, we investigate the link between the relative distances between embedding points, and that of the original data.

Distance Calculation The original dataset provides two separate views of the data, for which we can define two separate notions of distance: (1) a distance based on the graph structure (e.g the adjacency matrix), and (2) a distance based on the node features. For the distance in the graph, we choose to consider a distance in the graph space based on the diffusion distance [Coifman and Lafon, 2006] using Gaussian kernel with $\epsilon = 0.5$.

$$K(u, v) = exp\big(- \frac{d_{\text{shortest path}}(\text{node u}, \text{node v})^2_\alpha}{\epsilon}\big)$$

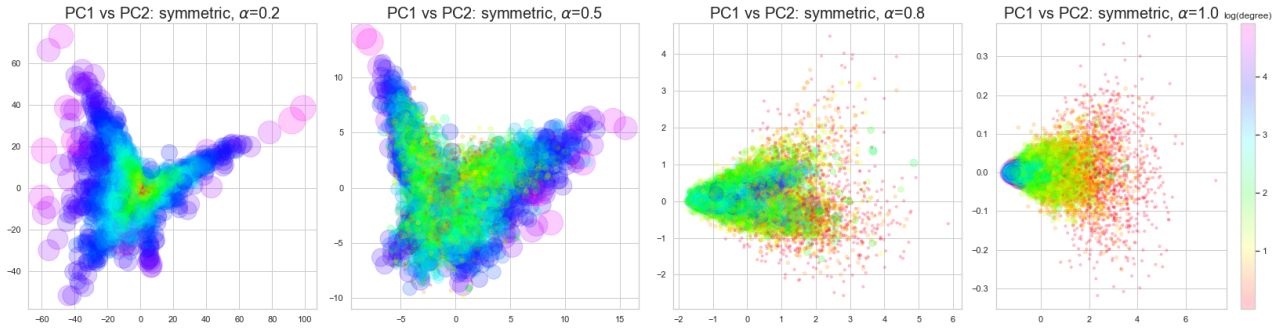
The shortest path distance is computed by build-in function in [Hagberg et al., 2008]. Distance in the feature space is measured by the pairwise euclidean distance of node features space. Finally, the distance in the embedding space is all based on the pairwise ℓ_2 Euclidean distance.

Correlation Analysis It is a natural question to ask how embedding space closely resembles the original graph space or the feature space. The notion of closeness can be defined in several ways, but in this experiment, we first see the correlation between the distance in the original space and in the embedding space. We will use Spearman's rank correlation, which measures the monotonic relationship between the two.

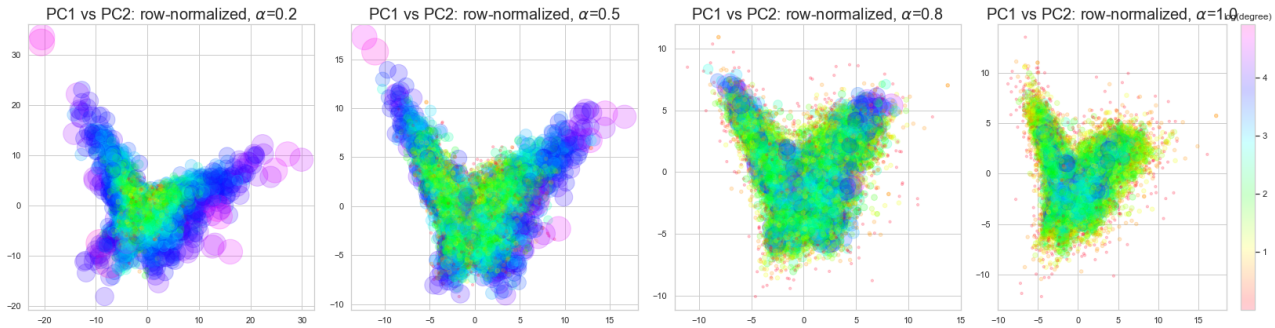
Higher correlation could be interpreted as the amount of information that is retained in the embedding space regarding graph structure or node features. From Figure 14, both dataset show decreasing correlation as alpha increases; however, the correlation itself is actually close to $0 (< 0.05)$. It is be reasonable to suspect no information regarding graph structure has been preserved in the embedding space, so we will come back to this question in E.3.4

Figure 15 shows the trend in consonance with the test accuracy for the node classification task, Figure 7. That is, for row-normalized operator the amount of node feature information is relatively constant for both standard homophilic dataset and heterophilic dataset such as Wisconsin. On the other hand, when it comes to the symmetrized operator, for Cora dataset, the correlation between the embedding spaces and the original feature space drops from 0.15(when $\alpha \approx 0.6$), to 0.005(when $\alpha \approx 1$). For Wisconsin dataset, the correlation still increases for the symmetrized operator, but the absolute value of correlation for the symmetrized operator is lower than that of the row-normalized operator.

Gromov-Wasserstein Distance To this extent, we use Gromov-Wasserstein distance[Mémoli, 2011] which allows to measure the distance between two probability spaces of different dimensions, by comparing the within distance of probability spaces. By estimating Gromov-Wasserstein distance we can evaluate how close our embedding space is to the original space upon the choice of operators.



(a) Coauthor CS, symmetrized, colored by degree



(b) Coauthor CS, row-normalized, colored by degree

Figure 11. Coauthor CS. The point size and color denote the node degree. For both symmetrized and row-normalized operator, high degree nodes are located farther from the origin when $\alpha \approx 0$. As α increases, high degree nodes are concentrated on the origin, and low degree nodes are spread out instead.

Based on the within distance calculated as described earlier, Gromov-wasserstein distance is calculated using python implemented `ot.gromov.gromov.wasserstein` function in `ot` package.[Flamary et al., 2021] *https://pythonot.github.io*. The detailed values from computations are shown in Figure 16 and Figure 17.

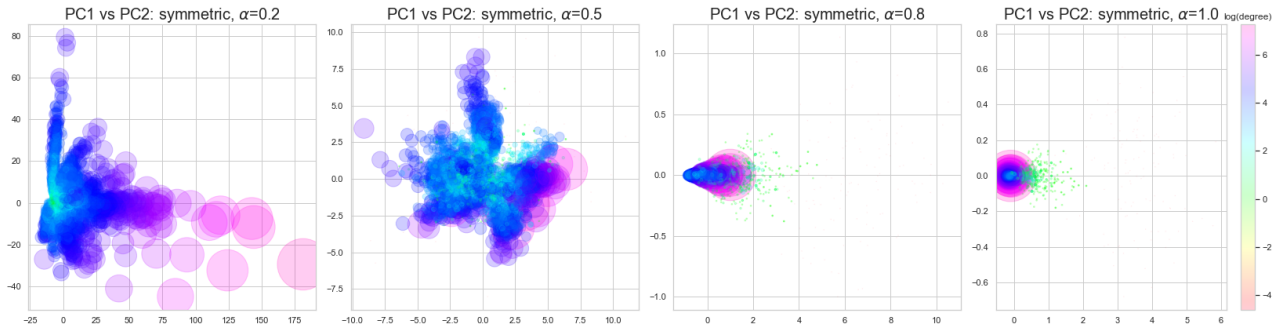
Analysis For the distance, we need to interpret in the opposite way we comprehend the correlation from earlier subsection. The lower the distance, the more the information regarding graph structure of feature has preserved in the embedding spaces. First, Cora shows the opposite pattern of distance with graph space and feature space. It can be viewed as for Cora, feature information has maximally preserved when $\alpha \approx 0.5$ for both symmetrized and row-normalized operator, while the information regarding graph structure has minimally estimated. When $\alpha \approx 0$ or $\alpha \approx 1$, the distance to the graph space is close to 0, while the distance to the feature space is close to the highest value.

On the contrary, Wisconsin seems to have similar pattern of distance for both graph and feature spaces. Embedding space recovered by the symmetrized operator, the information for both graph structure and node features are minimally retained when $\alpha \approx 0.5$. With the row-normalized operator, the distances to the original spaces increase as α increases.

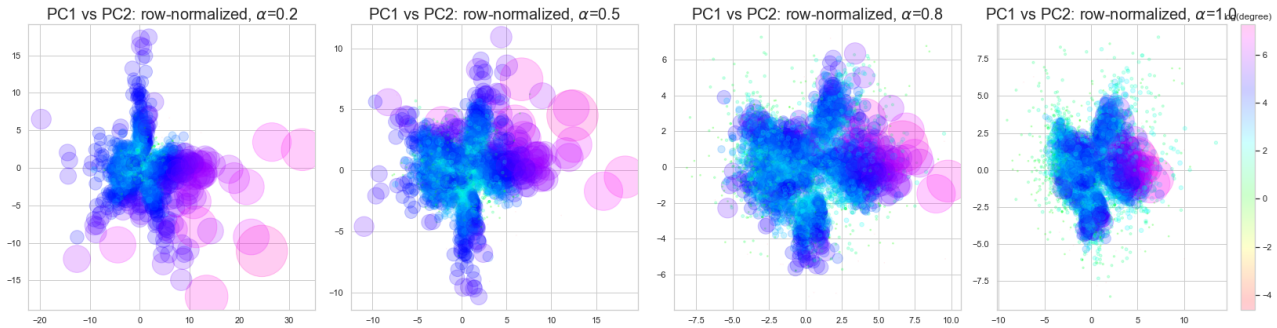
E.3.4 Curvature

In this section, the embedding space is compared to the original space with regard to the geometry of the original space. We first narrow down the notion of geometry to a graph curvature. Graph curvature could explain the structural properties of the data that cannot be fully captured by the node degree. One might reasonably wonder how this structural information or geometry of the graph could be preserved from original space to the embedding space. We use augmented Forman curvature for the graph defined in [Di Giovanni et al., 2022].

To calculate the graph curvature on the embedding space, we have to reconstruct the graph on the embedding space. First, based on the euclidean distance of each node in the embedding space, we connect the same number of edges as the original graph. With this "reconstructed graph" on the embedding space, we calculate the graph curvature. Finally, we compare how much curvature has been preserved upon varying operators by Spearman's rank correlation.



(a) Amazon Photo, symmetrized, colored by degree

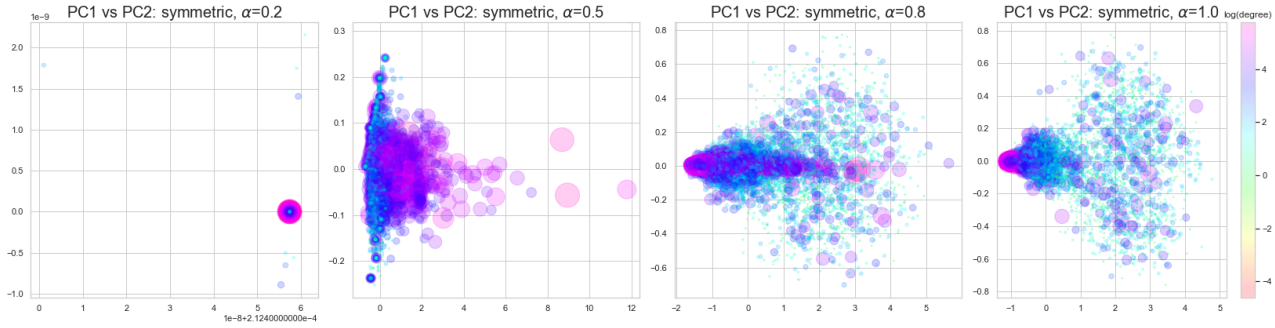


(b) Amazon Photo, row-normalized, colored by degree

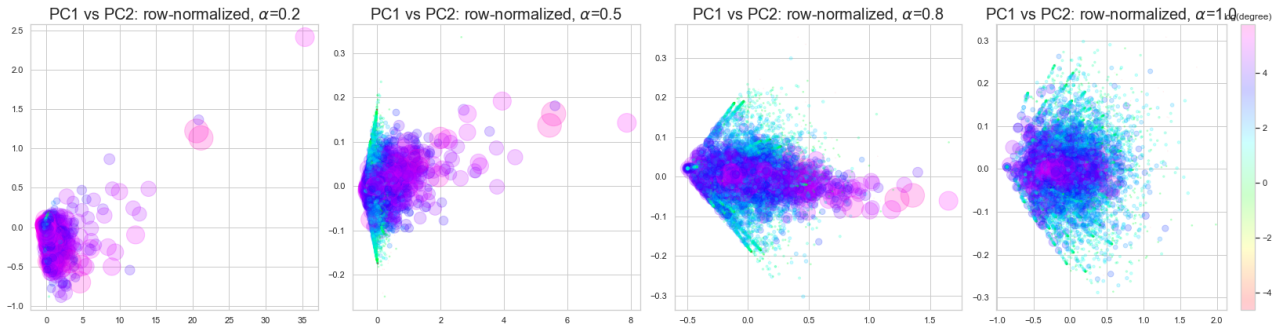
Figure 12. Amazon Photo. The point size and color denote the node degree. Amazon Photo networks have a few nodes with extremely high degree(> 500). Even for such nodes, as α increases the effect of high degree vanishes and all points are clustered near the origin.

	Cora	**Pubmed**	**Citeseer**	**Coauthor CS**	**Amazon Photo**	**Actor**	**Cornell**	**Wisconsin**
Mean	-9.6178	-18.9898	-3.2427	-14.4801	-99.8552	-8.2625	-41.9855	-46.2206
SD	16.0352	15.9152	8.5414	11.7537	110.3011	21.2629	43.2669	52.3953

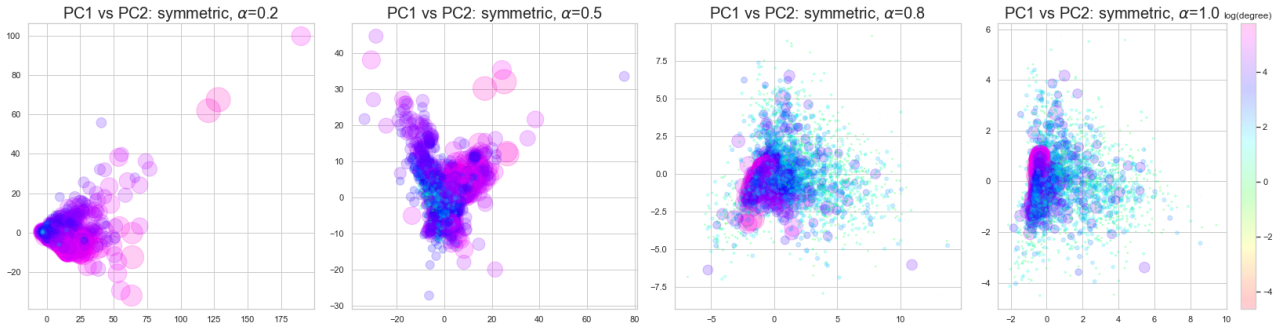
Table 4: Mean and Standard deviation of augmented forman curvature[Di Giovanni et al., 2022] for 8 Datasets



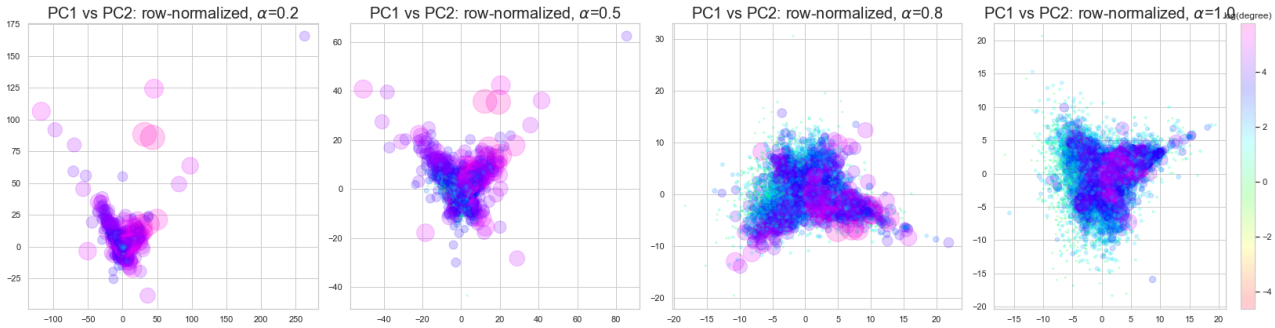
(a) WikiCS, symmetrized, colored by degree, without Batch Normalization



(b) WikiCS, row-normalized, colored by degree, without Batch Normalization

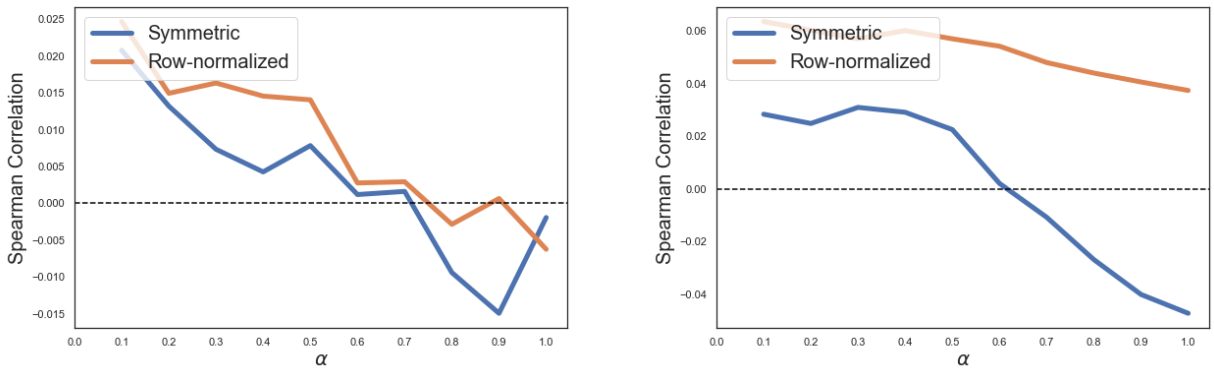


(c) WikiCS, symmetrized, colored by degree, with Batch Normalization



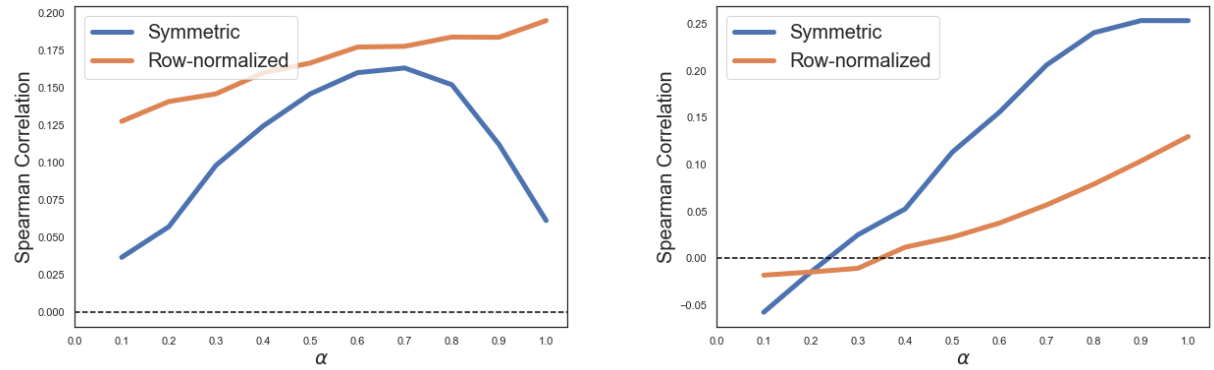
(d) WikiCS, row-normalized, colored by degree, with Batch Normalization

Figure 13. WikiCS. The point size and color denote the node degree. As we have observed from the experiment with Cora or Citeseer, with and without Batch Normalization, as α gets close to 1, the high degree nodes are concentrated near the origin and the low degree nodes are spread out in the embedding space.



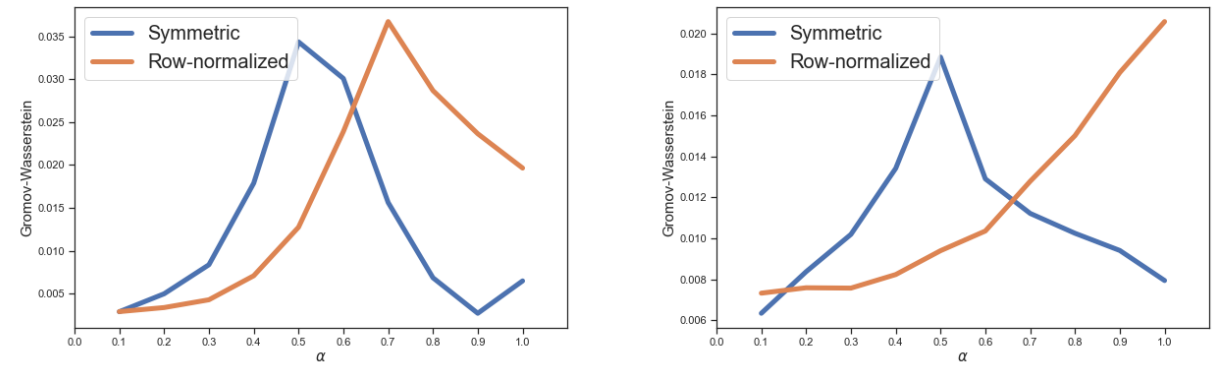
(a) Cora (b) Wisconsin

Figure 14. Spearman's correlation between the pairwise distances in the graph space and pairwise distance in the embedding space.



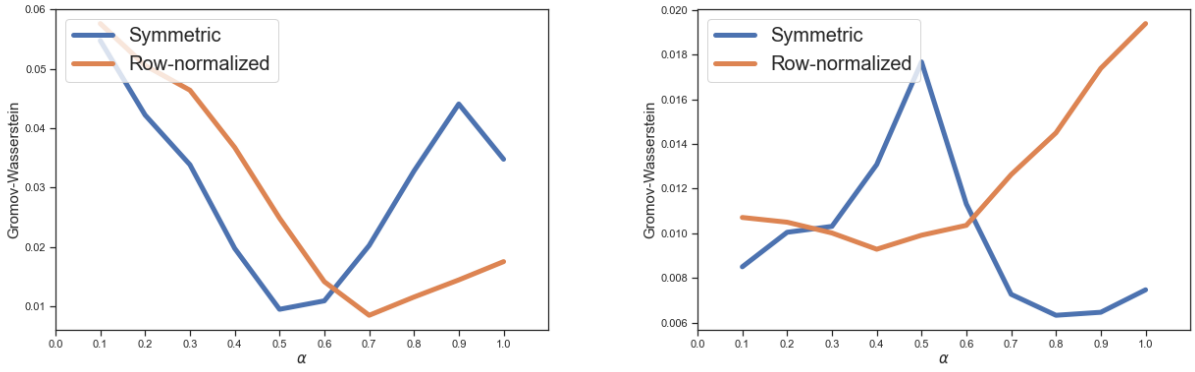
(a) Cora (b) Wisconsin

Figure 15. Spearman's correlation between the pairwise distances in the feature space and pairwise distance in the embedding space



(a) Cora (b) Wisconsin

Figure 16. Gromov-Wasserstein distance between the graph space and embedding space. For both datasets, when α is close to 0 or 1, the distance between two spaces is small for symmetrized operator.



(a) Cora (b) Wisconsin

Figure 17. Gromov-Wasserstein distance between the feature space and embedding space. Note that the distance variation across α is very similar to the accuracy of node classification task along α on Figure 7

Figure 18 shows (a) Cora and (b) Amazon Photos, standard datasets with high homophily as shown in Table 3, Spearman's correlation between the original curvature and embedding curvature are relatively constant around 0.3 with row-normalized operator. On the other hand, symmetrized operator has stronger positive correlation when $\alpha \approx 1$. For the dataset with low homophily, denoted as heterophilic graph dataset, such as (c) Cornell or (d) Wisconsin, not only the absolute value of the Spearman's correlation is much lower than that of results from homophilic dataset, but also there is a decreasing trend across the α for both symmetrized and row-normalized operator.

Based on these observations, the geometry in terms of curvature seems to be better preserved when $\alpha \approx 1$ for the dataset with high homophily. When the graph is of low homophily, symmetrized operator works slightly better preserving the curvature, but the absolute value of the correlation itself is fairly low compared with the result of high homophily dataset, such as Figure 18 (a) or (b).

E.4 NODE HOMOPHILY

In this section, we focus on the effect of β on the prediction performance. Graph Neural Network implicitly assumes that the neighboring node will share similar properties.McPherson et al. [2001] To overcome this shortcoming, there are several attemptsZhu et al. [2020a], Pei et al. [2020],Jin et al. [2021] to improve the performance on the dataset with low node homophily. From our experiments, we showed that by simply adjusting the β, we could gain comparable empirical performance on low-homophilic graphs, without employing any architectural adjustment.

Analysis We have used both the synthetic dataset(Synthetic-Cora) provided by Zhu et al. [2020a] with varying levels of node homophily, and the actual datasets(Actor, Cornell). We observed the competitive level of node classification accuracy both on synthetic and actual datasets compared to the model with architectural adjustment.

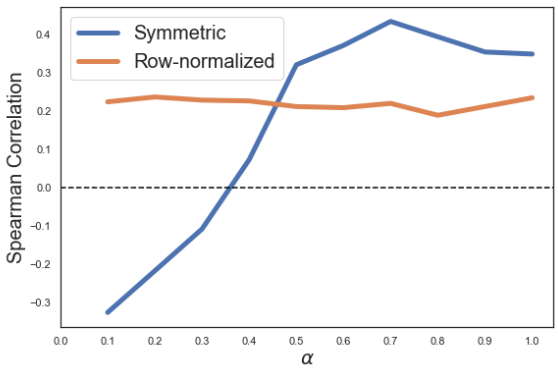
For *synthetic-Cora*, we observed that both for symmetric and row-normalized operators, the node prediction accuracy increases as β increases. However, the performance sharply drops if we increase β too much (β = 50).

For *Actor* data, we observed that the prediction accuracy using both symmetric and row-normalized operators monotonically increases as β increases. Compared to the literature, 35.86%(H_2GCN), 31.63%(Geom-GCN), our experiments showed reasonable performance up to 36.3% for the symmetric and 36.3% for the row-normalized operator.

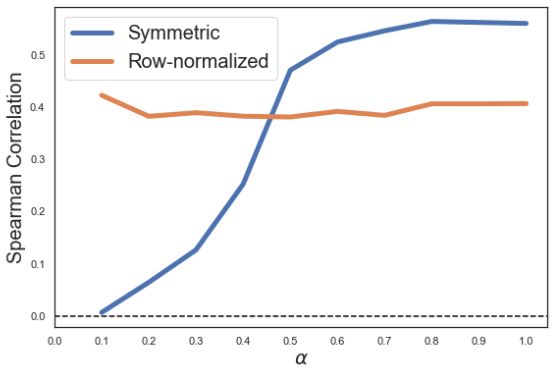
For *Cornell* data, we observed that the prediction accuracy using both symmetric and row-normalized operators monotonically increases as β increases. Compared to the literature, 82.16%(H_2GCN), 60.81%(Geom-GCN), 69.77%(UGCN), our experiments showed reasonable performance up to 69.2% for the symmetric operator and 70.6% for the row-normalized operator.

References

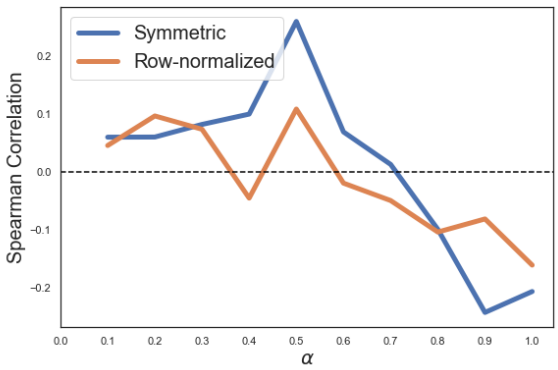
Learning to extract symbolic knowledge from the world wide web.



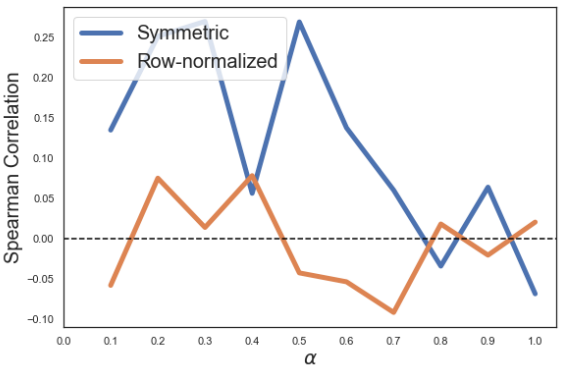
(a) Cora



(b) Citeseer

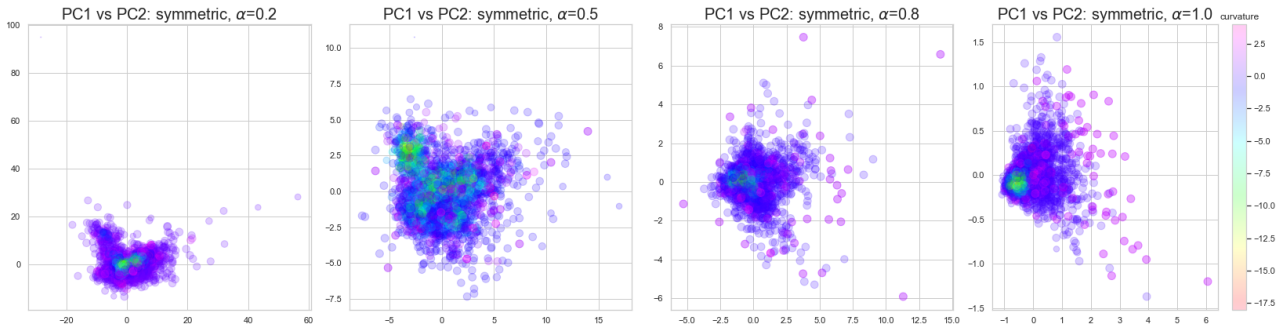


(c) Cornell

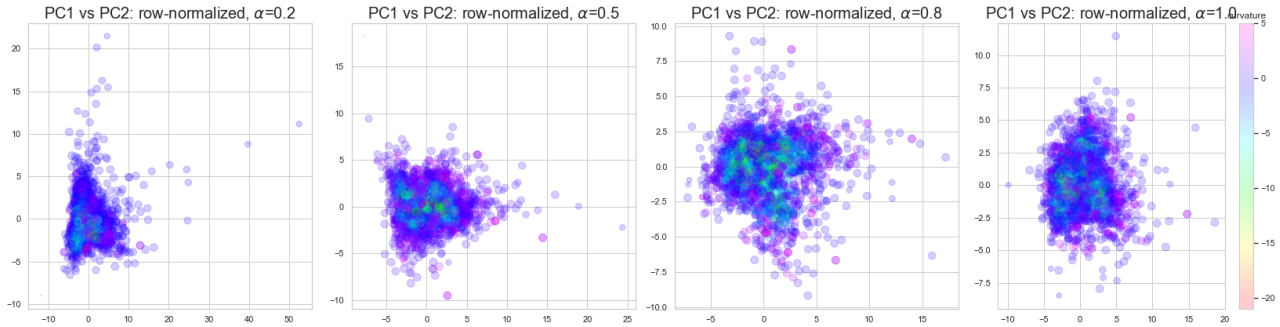


(d) Wisconsin

Figure 18: Spearman's correlation between the original and embedding curvature.

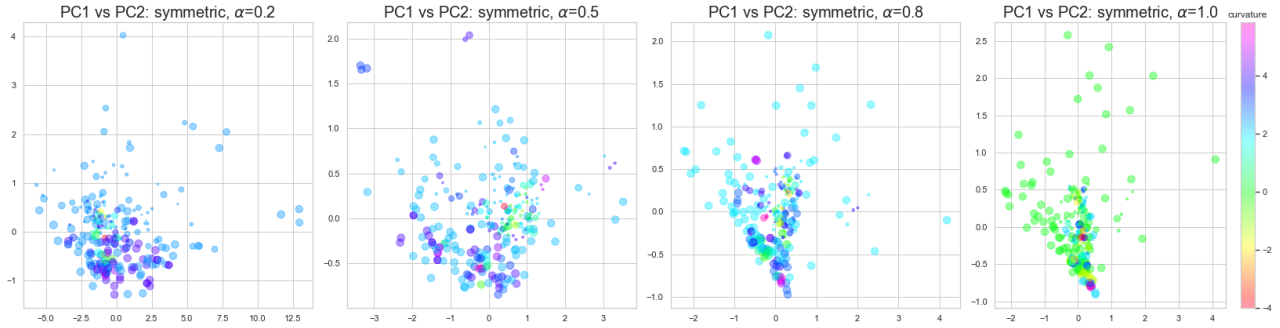


(a) Cora, symmetrized, colored by embedding curvature

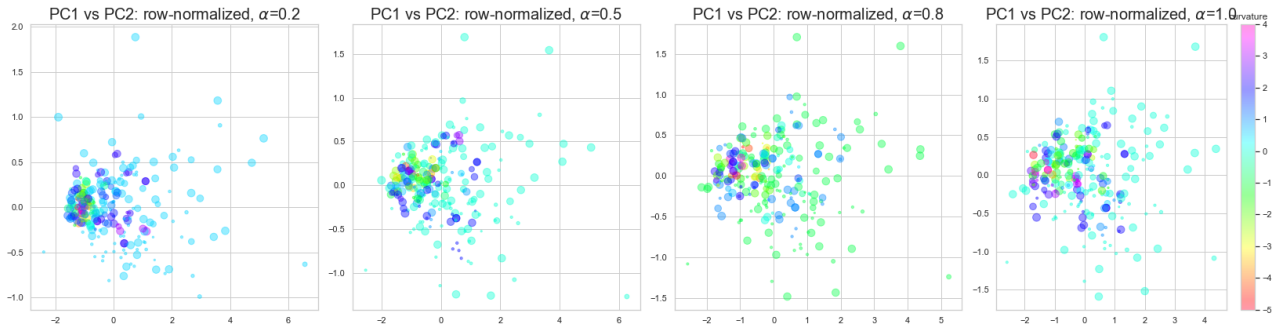


(b) Cora, row-normalized, colored by embedding curvature

Figure 19: Cora. The points are colored by embedding curvature and the size is proportional to the original curvature.

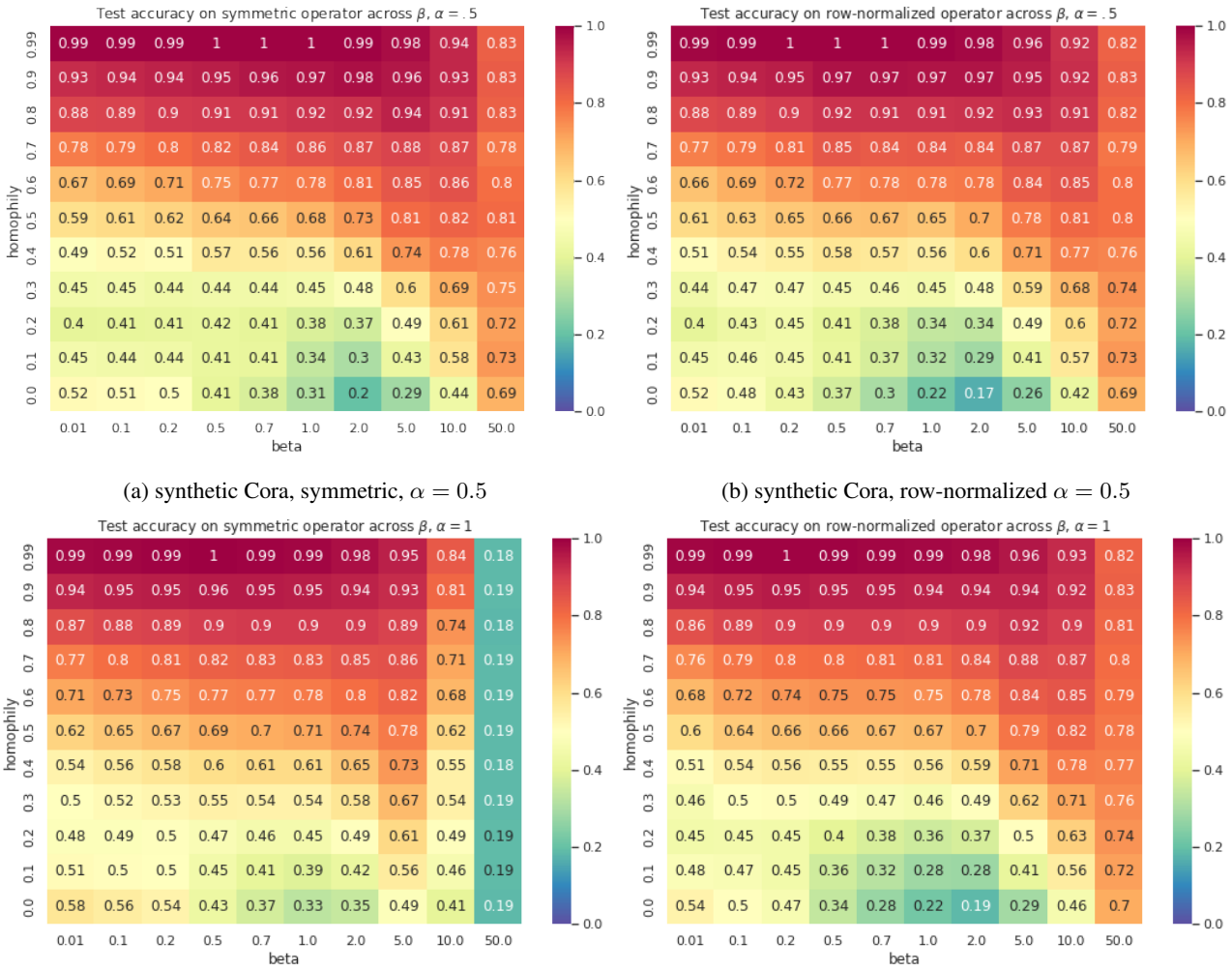


(a) Wisconsin, symmetrized, colored by embedding curvature



(b) Wisconsin, row-normalized, colored by embedding curvature

Figure 20: Wisconsin. The points are colored by embedding curvature and the size is proportional to the original curvature.



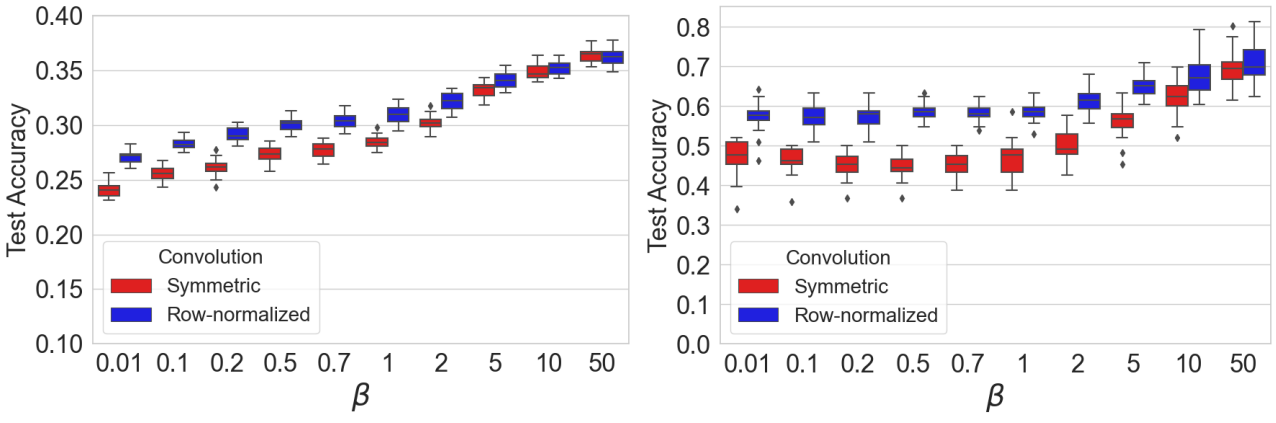
(a) synthetic Cora, symmetric, $\alpha = 0.5$

(b) synthetic Cora, row-normalized $\alpha = 0.5$

(c) synthetic Cora, row-normalized $\alpha = 1.0$

(d) synthetic Cora, row-normalized $\alpha = 1.0$

Figure 21. Synthetic Cora dataset provided in Zhu et al. [2020b]. Node homphily index ranges from 0.1 to 1.0. α value is fixed to see the effect of varying β. Node classification accuracy is given across the different level of β.



(a) Actor

(b) Cornell

Figure 22. Experiments on actual datasets with low node homophily. Node classification accuracy is given across the different levels of β and fixed $\alpha = 0.5$.

Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016.

Francesco Di Giovanni, Giulia Luise, and Michael Bronstein. Heterogeneous manifolds for curvature-aware graph embedding. 2022. doi: 10.48550/ARXIV.2202.01185. URL https://arxiv.org/abs/2202.01185.

Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*, 2017.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

Di Jin, Zhizhi Yu, Cuiying Huo, Rui Wang, Xiao Wang, Dongxiao He, and Jiawei Han. Universal graph convolutional networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10654–10664. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/5857d68cd9280bc98d079fa912fd6740-Paper.pdf.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83 (1):016107, 2011.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001. doi: 10.1146/annurev.soc.27.1.415. URL https://doi.org/10.1146/annurev.soc.27.1.415.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, pages 1–71, 2011. ISSN 1615-3375. URL http://dx.doi.org/10.1007/s10208-011-9093-5. 10.1007/s10208-011-9093-5.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. 2020. URL https://arxiv.org/abs/2002.05287.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. 2018.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735*, 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. 2016. URL https://arxiv.org/abs/1603.08861.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. 2020a. URL https://arxiv.org/abs/2006.11468.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs, 2020b. URL https://arxiv.org/abs/2006.11468.