LightDiC: A Simple yet Effective Approach for Large-scale Digraph Representation Learning [Technical Report]

1 OUTLINE

The appendix is organized as follows:

1.1 Proof of Lemma.3.1

We represent the adjacency matrix and the diagonal degree matrix of a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ nodes, $|\mathcal{E}| = m$ edges by $\mathbf{A}$, $\mathbf{D}$. Based on this, let $\mathbf{L}_m \in \mathbb{C}^{N\times N}$ be the magnetic Laplacian of a digraph $\mathcal{G}$. The node feature matrix represents $\mathbf{X} \in \mathbb{C}^{N\times 1}$, the smoothness of $\mathbf{X}$ over $\mathcal{G}$ is defined as $\mathbf{X}^\dagger \mathbf{L}_m \mathbf{X}$. Then, the total variation of the graph signal $\mathbf{X}$ is a smoothness measure, quantifying how much the signal $\mathbf{X}$ changes with respect to the graph topology encoded in magnetic Laplacian $\mathbf{L}_m$ as the following quadratic form, which is also known as Dirichlet energy

$$\mathbf{X}^\dagger \mathbf{L}_m \mathbf{X} = \sum_{(u,v)\in\mathcal{E}} |\mathbf{X}_u - e^{i\theta_{uv}}\mathbf{X}_v|^2 = \sum_{(u,v)\in\mathcal{E}} |e^{-i\theta_{uv}}\mathbf{X}_u - \mathbf{X}_v|^2. \quad (1)$$

First, by expanding $\mathbf{L}_m\mathbf{X}$, we have

$$(\mathbf{L}_m\mathbf{X})_k = \sum_{w\in\mathcal{N}^{in}(k)\cup\mathcal{N}^{out}(k)} \left(\mathbf{X}_k - e^{i\theta_{kw}}\mathbf{X}_w\right). \quad (2)$$

Based on this, $\mathbf{X}^\dagger \mathbf{L}_m \mathbf{X}$ can be formulated as:

$$\begin{aligned}
\mathbf{X}^\dagger \mathbf{L}_m \mathbf{X} &= \sum_{k\in\mathcal{V}} \sum_{w\in\mathcal{N}^{in}(k)\cup\mathcal{N}^{out}(k)} \left(\mathbf{X}_k - e^{i\theta_{kw}}\mathbf{X}_w\right) \\
&= \sum_{(u,v)\in\mathcal{E}} \mathbf{X}_u^\dagger\mathbf{X}_u - e^{i\theta_{uv}}\mathbf{X}_u^\dagger\mathbf{X}_v + \mathbf{X}_v^\dagger\mathbf{X}_v - e^{i\theta_{vu}}\mathbf{X}_v^\dagger\mathbf{X}_u \\
&= \sum_{(u,v)\in\mathcal{E}} \mathbf{X}_u^\dagger\mathbf{X}_u + \mathbf{X}_v^\dagger\mathbf{X}_v - \left(e^{i\theta_{uv}}\mathbf{X}_u^\dagger\mathbf{X}_v + e^{i\theta_{vu}}\mathbf{X}_v^\dagger\mathbf{X}_u\right) \\
&= \sum_{(u,v)\in\mathcal{E}} |\mathbf{X}_u|^2 + |\mathbf{X}_v|^2 - \\
&\qquad |\mathbf{X}_u||\mathbf{X}_v| \left(e^{i\left(-\theta_{\mathbf{X}_u}+\theta_{\mathbf{X}_v}+\theta_{uv}\right)} + e^{i\left(-\theta_{\mathbf{X}_v}+\theta_{\mathbf{X}_u}+\theta_{vu}\right)}\right) \\
&= \sum_{(u,v)\in\mathcal{E}} |\mathbf{X}_u|^2 + |\mathbf{X}_v|^2 - 2|\mathbf{X}_u||\mathbf{X}_v| \cos\left(\theta_{\mathbf{X}_u} - \theta_{\mathbf{X}_v} - \theta_{uv}\right),
\end{aligned} \quad (3)$$

the last equality holds because $\theta_{uv} = -\theta_{vu}$. Since

$$\begin{aligned}
|\mathbf{X}_u - e^{i\theta_{uv}}\mathbf{X}_v|^2 &= ||\mathbf{X}_u|e^{i\theta_{\mathbf{X}_u}} - e^{i\theta_{uv}}|\mathbf{X}_v|e^{i\theta_{\mathbf{X}_v}}|^2 \\
&= ||\mathbf{X}_u|e^{i\theta_{\mathbf{X}_u}} - |\mathbf{X}_v|e^{i(\theta_{\mathbf{X}_v}+\theta_{uv})}|^2 \\
&= ||\mathbf{X}_u|\cos\theta_{\mathbf{X}_u} + i|\mathbf{X}_u|\sin\theta_{\mathbf{X}_u} - \\
&\quad \left(|\mathbf{X}_v|\cos\left(\theta_{\mathbf{X}_v}+\theta_{uv}\right) + i|\mathbf{X}_v|\sin\left(\theta_{\mathbf{X}_v}+\theta_{uv}\right)\right)|^2 \\
&= ||\mathbf{X}_u|\cos\theta_{\mathbf{X}_u} - |\mathbf{X}_v|\cos(\theta_{\mathbf{X}_v}+\theta_{uv}) + \\
&\quad \left(|\mathbf{X}_u|\sin\theta_{\mathbf{X}_u} - |\mathbf{X}_v|\sin\left(\theta_{\mathbf{X}_v}+\theta_{uv}\right)\right)|^2 \\
&= \left(|\mathbf{X}_u|\cos\theta_{\mathbf{X}_u} - |\mathbf{X}_v|\cos(\theta_{\mathbf{X}_v}+\theta_{uv})\right)^2 + \\
&\quad \left(|\mathbf{X}_u|\sin\theta_{\mathbf{X}_u} - |\mathbf{X}_v|\sin\left(\theta_{\mathbf{X}_v}+\theta_{uv}\right)\right)^2 \\
&= |\mathbf{X}_u|^2 + |\mathbf{X}_v|^2 - \\
&\quad 2|\mathbf{X}_u||\mathbf{X}_v|\cos\left(\theta_{\mathbf{X}_u} - \theta_{\mathbf{X}_v} - \theta_{uv}\right) \\
&= |e^{-i\theta_{uv}}\mathbf{X}_u - \mathbf{X}_v|^2,
\end{aligned} \quad (4)$$

the lemma is proved.

1.2 Proof of Lemma.3.2

In LightDiC, we predefine the magnetic graph operator

$$\text{MGO} := \hat{\mathbf{A}}_m = \left(\widetilde{\mathbf{D}}_m^{-1/2}\widetilde{\mathbf{A}}_m\widetilde{\mathbf{D}}_m^{-1/2} \odot \exp\left(i\Theta^{(q)}\right)\right). \quad (5)$$

MGO is a digraph operator defined in terms of the global topology based on the complex domain, which models the directional information through the imaginary part. Based on this, we can define complex domain-based message-passing mechanisms for digraphs. Now we further analyze the digraph operator used in our proposed LightDiC: $\left(\widetilde{\mathbf{D}}_m^{-1/2}\widetilde{\mathbf{A}}_m\widetilde{\mathbf{D}}_m^{-1/2} \odot \exp\left(i\Theta^{(q)}\right)\right)$.

As an analogy of spectral graph convolution on undirected graphs, the directed spectral graph convolution is also based on the convolution theorem. The Fourier transform for a signal $\mathbf{X} : \mathbb{V} \to \mathbb{C}$ is defined as $\widehat{\mathbf{X}} = \mathbf{U}^\dagger\mathbf{X}$, so that $\widehat{\mathbf{X}}(k) = \langle \mathbf{X}, \mathbf{u}_k\rangle$, where $\mathbb{V}$ and $\mathbb{C}$ represents vertex domain an complex field-based Fourier domain respectively. Note that the elements of $\mathbf{u}_1, \cdots, \mathbf{u}_n$ are complex numbers, obtained by eigendecomposition of the magnetic Laplacian $\mathbf{L}_m$, we regard $\mathbf{u}_1, \cdots, \mathbf{u}_n$ as the Fourier basis for digraphs. Since $\mathbf{U}$ is unitary, we have the Fourier inverse formula:

$$\mathbf{X} = \mathbf{U}\widehat{\mathbf{X}} = \sum_{k=1}^{N} \widehat{\mathbf{X}}(k)\mathbf{u}_k. \quad (6)$$

Here, convolution corresponds to point-wise multiplication on the Fourier basis. Thus, the convolution of $\mathbf{X}$ with a filter $\mathbf{g}$ in the Fourier domain can be defined as $\widetilde{\mathbf{g}\star\mathbf{X}}(k) = \widetilde{\mathbf{g}}(k)\widetilde{\mathbf{X}}(k)$. According to Eq.(6), $\mathbf{g}\star\mathbf{X} = \mathbf{U}\text{Diag}(\widehat{\mathbf{g}})\widehat{\mathbf{X}} = (\mathbf{U}\text{Diag}(\widehat{\mathbf{g}})\mathbf{U}^\dagger)\mathbf{X}$, the convolution matrix can be written as:

$$\mathbf{G} = \mathbf{U}\Sigma\mathbf{U}^\dagger, \quad (7)$$

For a diagonal matrix Σ, different filter refers to different choices of Σ. In practice, in order to avoid explicit eigen-decomposition, Σ is often set as polynomials of Λ. Suppose $\widetilde{\Lambda}$ is a normalized eigenvalue matrix, which is defined as $\widetilde{\Lambda} = \frac{2}{\lambda_{\max}}\Lambda - \mathbf{I}$, we can write

$$\Sigma = \sum_{k=0}^{K} \mathbf{w}_k \, \mathrm{T}_k(\widetilde{\Lambda}). \tag{8}$$

T_k can be set as various orthogonal polynomial bases. When choosing T_k as the k-order Chebyshev polynomial which is defined by $\mathrm{T}_0(x) = 1$, $\mathrm{T}_1(x) = x$, and $\mathrm{T}_k(x) = 2x\,\mathrm{T}_{k-1}(x) + \mathrm{T}_{k-2}(x)$ for $k \geq 2$. Since $(\mathbf{U}\widetilde{\Lambda}\mathbf{U}^\dagger)^k = \mathbf{U}\widetilde{\Lambda}^k\mathbf{U}^\dagger$, we have:

$$\mathbf{GX} = \mathbf{U}\sum_{k=0}^{K} \mathbf{w}_k \, \mathrm{T}_k(\widetilde{\Lambda})\mathbf{U}^\dagger\mathbf{X} = \sum_{k=0}^{K} \mathbf{w}_k \, \mathrm{T}_k(\widetilde{\mathbf{L}}_m)\mathbf{X}. \tag{9}$$

Here, $\widetilde{\mathbf{L}}_m$ is defined as $\widetilde{\mathbf{L}}_m = \frac{2}{\lambda_{\max}}\mathbf{L}_m - \mathbf{I}$. We set $K = 1$, assume that $\mathbf{L}_m = \mathbf{L}_m^{(q)}$, $\lambda_{\max} \approx 2$ and $\mathbf{w} = \mathbf{w}_0 = -\mathbf{w}_1$, we can obtain that:

$$\mathbf{GX} = \mathbf{W}\left(\mathbf{I} + \mathbf{D}_m^{-1/2}\mathbf{A}_m\mathbf{D}_m^{-1/2}) \odot \exp\left(i\Theta^{(q)}\right)\right)\mathbf{X}. \tag{10}$$

By applying further renormalization tricks to avoid instabilities arising from vanishing/exploding gradients, it yields

$$\mathbf{GX} = \mathbf{W}\left(\widetilde{\mathbf{D}}_m^{-1/2}\widetilde{\mathbf{A}}_m\widetilde{\mathbf{D}}_m^{-1/2} \odot \exp\left(i\Theta^{(q)}\right)\right)\mathbf{X}, \tag{11}$$

where $\widetilde{\mathbf{A}}_m = \mathbf{A}_m + \mathbf{I}$ and $\widetilde{\mathbf{D}}_m(i,i) = \sum_j \widetilde{\mathbf{A}}_m(i,j)$.

According to the analysis, we can find that the complex domain feature propagation in our proposed LightDiC is a complex spectral convolution with a low-pass filter.

Since $\hat{\mathbf{A}}_m^K\mathbf{X} = \left(\widetilde{\mathbf{D}}_m^{-1/2}\widetilde{\mathbf{A}}_m\widetilde{\mathbf{D}}_m^{-1/2} \odot \exp\left(i\Theta^{(q)}\right)\right)\mathbf{X}$, we have $\hat{\mathbf{A}}_m = \widetilde{\mathbf{D}}_m^{-1/2}\widetilde{\mathbf{A}}_m\widetilde{\mathbf{D}}_m^{-1/2} \odot \exp\left(i\Theta^{(q)}\right) \approx \mathbf{I} - \mathbf{L}_m$. Then, $\hat{\mathbf{A}}_m^K = (\mathbf{I} - \mathbf{L}_m)^K = \mathbf{U}(\mathbf{I} - \Sigma)^K\mathbf{U}^\dagger$, applying this operation on the digraph signal $\mathbf{X}$ ($\hat{\mathbf{A}}_m\mathbf{X} = \mathbf{U}\,(\mathbf{I} - \Sigma)^K\,\mathbf{U}^\dagger$) is identical to: (i) convert $\mathbf{X}$ to the Fourier domain ($\mathbf{U}^\dagger\mathbf{X}$); (ii) applying a low-pass filter $g(\lambda_i) = (1 - \lambda_i)^K$ for $i = 1, \cdots, n$ on the signal on the Fourier domain ($(\mathbf{I} - \Sigma)^K\mathbf{U}^\dagger\mathbf{X}$); (iii) applying inverse Fourier transform on the signal ($\mathbf{U}(\mathbf{I} - \Sigma)^K\mathbf{U}^\dagger\mathbf{X}$).

The function $(\mathbf{I} - \mathbf{X})^K$ is a monotonically decreasing function. Since each signal $\mathbf{X} \in \mathbb{C}^n$ can be written as $\mathbf{X} = \sum_{i=1}^{n} \omega_i\mathbf{u}_i$. The operation will keep the information of eigenvector components corresponding to the lower eigenvalues and ignore the components corresponding to the higher eigenvalues.

1.3 Proof of Lemma.3.3

Since in real-world graphs, the feature varies smoothly over the graph, an intuition of our feature pre-processing is to smooth the signals. To evaluate the smoothness of $\hat{\mathbf{A}}_m^K\mathbf{x}$, we first analyze the smoothness of the eigenvectors of $\hat{\mathbf{A}}_m$.

According to Courant-Fischer theorem theorem [1], we have

Lemma 1. *Let $\mathbf{L}_m \in \mathbb{C}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$, then we have*

$$\lambda_k = \max_{S\in\mathbb{C}^n,\, dim(S)=k} \min_{\mathbf{X}\in S} \frac{\mathbf{X}^\dagger\mathbf{L}_m\mathbf{X}}{\mathbf{X}^\dagger\mathbf{X}} \tag{12}$$

The smoothness metric $\mathbf{X}^\dagger\mathbf{L}_m\mathbf{X}$ is minimized by the eigenvector $\mathbf{u}_1$ corresponding to the smallest eigenvalue. Furthermore, we can expect that smaller eigenvalues have corresponding eigenvectors with smaller smoothness. We provide an example digraph $\mathcal{G}$ to prove our point, the four eigenvectors of the magnetic Laplacian $\mathbf{L}_m^{(q)} := \mathbf{D}_m - \mathbf{A}_m^{(q)} = \mathbf{D}_m - \mathbf{A}_m \odot \exp\left(i\Theta^{(q)}\right)$ with $q = 0.25$ of $\mathcal{G}$ is illustrated in Fig. 1. It shows that the magnetic Laplaican of $\mathcal{G}$ has 4 eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, $\mathbf{u}_4 \in \mathbb{C}^n$. The elements of these four eigenvectors are all complex numbers, we represent a complex number x by the magnitude $|x|$ and the angle $\theta = \arctan(\mathrm{Imag}(x)/\mathrm{Real}(x))$. Each eigenvector $\mathbf{u}_k$ can be regarded as a complex graph signal on graphs, with $\mathbf{u}_k(i)$ being the complex signal on node $i \in \mathbb{V}$. We depict the value of the four eigenvectors in Figure.1, we also plot the angle of each eigenvector element. Then, the smoothness of each eigenvector can be computed as $\mathbf{u}_k^\dagger\mathbf{L}_m\mathbf{u}_k$ according to our definition. Since $\mathbf{L}_m\mathbf{u}_k = \lambda_k\mathbf{u}_k$, $\lambda_k = \mathbf{u}_k^\dagger\mathbf{L}_m\mathbf{u}_k$, the smoothness value exactly equals the eigenvalue. $\mathbf{u}_1$ corresponding to the smallest eigenvalue has the smallest smoothness value, followed by $\mathbf{u}_2$, $\mathbf{u}_3$ and $\mathbf{u}_4$. It can be seen that the eigenvectors corresponding to smaller eigenvalues vary more smoothly on graphs. As a result, the low eigen-components of the node feature also vary very smoothly over the graph. Thus, the complex domain propagation keeps the low-frequency components of the feature and ignores the high-frequency components. The resulting signal is smooth while keeping the information of the node feature.

1.4 Proof of Lemma.3.4

Given digraph signal $\mathbf{y} \in \mathbb{C}^{N\times 1}$ with noises ϵ, the optimization function is defined as $\min_{\mathbf{X}} f(\mathbf{X}) = \min_{\mathbf{X}\in\mathbb{C}} \|\mathbf{X} - \mathbf{y}\|_2 + \mathbf{X}^\dagger\mathbf{L}_m\mathbf{X}$.

In order to solve a convex optimization problem $\min_{\mathbf{X}} f(\mathbf{X})$, the proximal gradient descent method updates the current solution $\mathbf{X}^{(t)}$ by $\mathbf{X}^{(t+1)} \leftarrow \mathbf{X}^{(t)} + \lambda\nabla f(\mathbf{X}^{(t)})$ where λ is the step size and $\nabla f(\mathbf{X})$ is the gradient of $f(\mathbf{X})$ w.r.t. $\mathbf{X}$. Since $\mathbf{Z}(\mathbf{X}) = \min_{\mathbf{X}\in\mathbb{C}} \|\mathbf{X} - \mathbf{y}\|_2 + \mathbf{X}^\dagger\mathbf{L}_m\mathbf{X}$, we have:

$$\frac{\partial\mathbf{Z}(\mathbf{X})}{\partial\mathbf{X}} = 2\mathbf{L}_m\mathbf{X} + 2\mathbf{X} - 2\mathbf{y}. \tag{13}$$

Then, start from a initial solution $\mathbf{X}^{(0)} = \mathbf{X}$, the $k+1$ step gradient descent can be computed as:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - \alpha[(\mathbf{L}_m + \mathbf{I})\mathbf{X}^{(k)} - \mathbf{y}], \tag{14}$$

where $\lambda = \frac{\alpha}{2}$ is the step size. In practice, the convergence speed of such an update strategy is often slow, pre-conditioning is a well-known technique in numerical optimization to accelerate the convergence. By applying pre-conditioning and re-arranging the terms, we have:

$$\mathbf{X}^{(k+1)} = (1 - \alpha)\mathbf{X}^{(k)} + \alpha\widetilde{\mathbf{D}}_m^{-1}[\mathbf{A}_m\mathbf{X}^{(k)} + \mathbf{y}], \tag{15}$$

where $\widetilde{\mathbf{D}}_m^{-1} = (\mathbf{D}_m + \mathbf{I})^{-1}$. Therefore, start from an initial solution $\mathbf{X}^{(0)} = \mathbf{D}_m^{1/2}\mathbf{X}$, after the first gradient step, $\mathbf{X}^{(1)}$ is equivalent to the one layer model $\widetilde{\mathbf{D}}_m^{-1/2}\widetilde{\mathbf{A}}_m\widetilde{\mathbf{D}}_m^{-1/2}\mathbf{X}$ up to a simple re-parameterization by $\widetilde{\mathbf{D}}_m^{-1/2}$. Thus, the K-step propagated features $\hat{\mathbf{A}}_m^K\mathbf{X} = \left(\widetilde{\mathbf{D}}_m^{-1/2}\widetilde{\mathbf{A}}_m\widetilde{\mathbf{D}}_m^{-1/2} \odot \exp\left(i\Theta^{(q)}\right)\right)\mathbf{X}$ is equivalent to applying such a proximal gradient update for K times.

Up to this point, we have presented the theoretical extension of spectral GNNs to digraphs using the magnetic Laplacian. This integration offers a unified theoretical framework encompassing smoothness, Dirichlet energy, and spectral analysis.

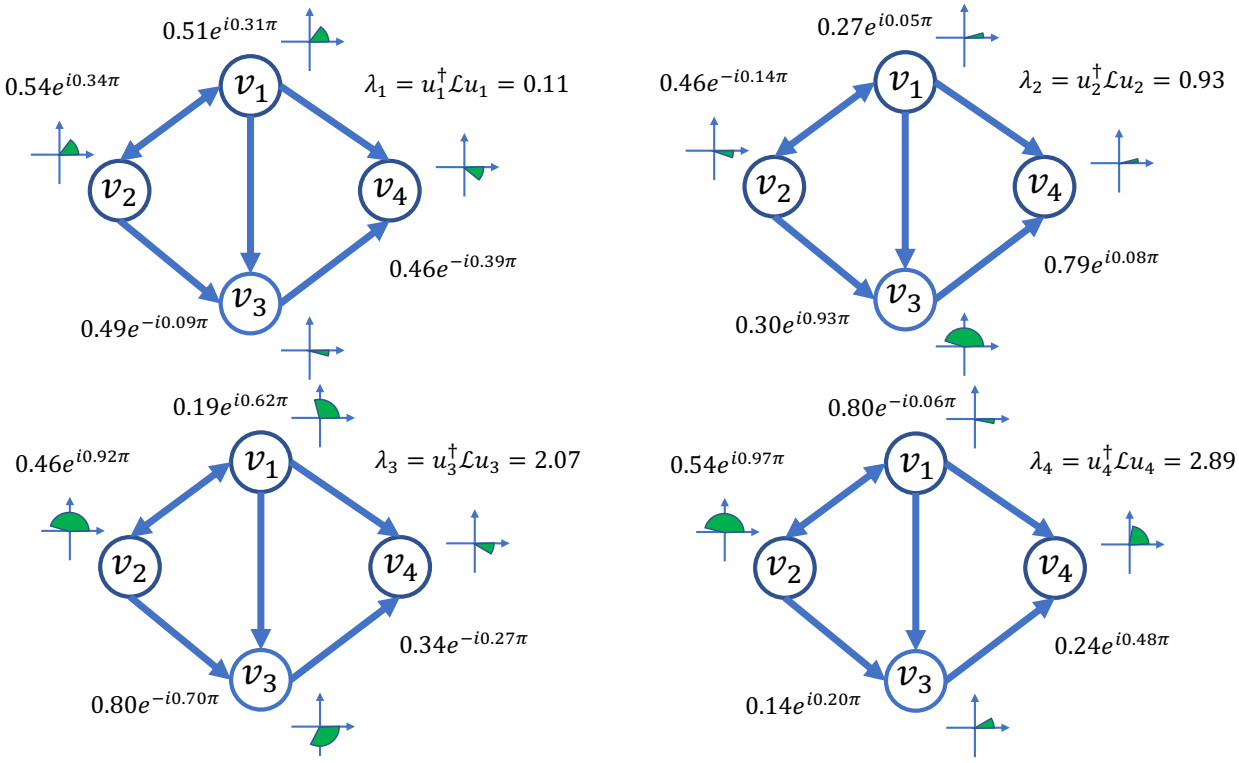

Figure 1: Illustration of all the eigenvectors of a magnetic Laplacian $\mathbf{L}_m^{(q)} = \mathbf{D}_m - \mathbf{A}_m \odot \exp\left(i\Theta^{(q)}\right)$ with $q = 0.25$ of an example graph $\mathcal{G}$. The eigenvectors corresponding to smaller eigenvalues vary more smoothly on graphs. When evaluating the smoothness of a signal $\mathbf{X}$ as $\mathbf{X}^\dagger \mathcal{L} \mathbf{X}$, the smoothness of $\mathbf{u}_1 = [0.51e^{i0.31\pi}, 0.54e^{i0.34\pi}, 0.49e^{-i0.09\pi}, 0.46e^{-i0.39\pi}]^\mathrm{T}$ is $\lambda_1 = 0.11$, followed by the smoothness of $\mathbf{u}_2$ (0.93), $\mathbf{u}_3$ (2.07) and $\mathbf{u}_4$ (2.89).

1.5 Comparison with SGC and MagNet.

Comparison with SGC [18]. LightDiC differs from the related work SGC in: (i) Application-guided generalization: SGC is limited to handling undirected graphs, and our experiments have revealed significant performance degradation in large-scale digraph scenarios. In contrast, LightDiC focuses more on digraphs with complex topologies and aims to provide a universal solution for both directed and undirected graphs through MGO. (ii) Paradigm-guided Scalability: SGC is recognized for introducing the decoupling design paradigm to address scalability challenges posed by large-scale graph data. Inspired by this, subsequent undirected methods like SIGN [5], S^2GC [22], GBP [3], GAMLP [20], and GRAND+ [4] have further improved performance through well-designed learnable mechanisms. However, LightDiC is the first to successfully apply this decoupling paradigm to large-scale digraphs, offering a novel attempt to address real-world data science challenges. (iii) Technical-guided Innovation: SGC relies on symmetrically normalized adjacency matrices for smoothing signals on undirected graphs and utilizes the propagated features from the final step for downstream task training. Compared to SGC, LightDiC employs the magnetic Laplacian to smooth signals on digraphs in the complex number domain. We theoretically demonstrate that both the real and imaginary parts contain rich topological insights. Consequently, LightDiC proposes a weight-free message aggregation function that combines information from these two different encoding perspectives.

Comparison with MagNet [21]. LightDiC differs from the related work MagNet in: (i) Scalability: While MagNet achieves efficient convolutions on digraphs using the magnetic Laplacian, it inherits unnecessary computational redundancy from deep learning (complex computations and more learnable parameters). Additionally, MagNet's model depth and applicability are limited to shallow design and toy-sized datasets due to the recursive computations in the complex number domain, lacking scalability. In contrast, LightDiC minimizes precomputation overhead through a decoupling design paradigm, making it solely dependent on efficient matrix multiplication based on sparse matrices. LightDiC also compresses complex learning processes into simple linear mappings to maximize training and inference efficiency. (ii) Encoding Strategy: Unlike MagNet, which employs recursive GCN as the encoding strategy, LightDiC introduces a weight-free message aggregation function based on appropriate theoretical extensions on digraphs. This function encodes multi-scale deep structural information, enhancing predictive performance. (iii) Supplemental Interpretability: MagNet theoretically proves that the symmetric conjugate magnetic Laplacian shares properties similar to symmetrically normalized undirected graph Laplacians. It also provides empirical analysis of its eigenvalues and eigenvectors. Building upon this, LightDiC extends theoretical spectral analysis from undirected to digraphs from the perspective of graph signal denoising. It uses this perspective to guide the design of suitable encoding mechanisms.

1.6 Dataset Description

The description of all digraph benchmark datasets is listed below:

CoraML, **CiteSeer** [2], and **ogbn-papers100M** [8] are three citation network datasets. In these three networks, papers from different topics are considered nodes, and the edges are citations among the papers. The node attributes are binary word vectors, and class labels are the topics the papers belong to.

WikiCS [13] is a Wikipedia-based dataset for benchmarking GNNs. The dataset consists of nodes corresponding to computer science articles, with edges based on hyperlinks and 10 classes representing different branches of the field. The node features are derived from the text of the corresponding articles. They were calculated as the average of pre-trained GloVe word embeddings [**?**], resulting in 300-dimensional node features.

Slashdot [14] is from a technology-related news website with user communities. The website introduced Slashdot Zoo features that allow users to tag each other as friends or foes. The dataset is a common signed social network with friends and enemies labels. In our experiments, we consider only friendships.

Epinions [12] is a who-trust-whom online social network. Members of the site can indicate their trust or distrust of the reviews of others. The network reflects people's opinions of others. In our experiments, we consider only trust relationships.

WikiTalk [11] contains all users and discussions from the inception of Wikipedia until Jan. 2008. Nodes in the network represent Wikipedia users and a directed edge from node v_i to node v_j denotes that user i edited at least once a talk page of user j.

1.7 Compared Baselines

The main characteristics of all baselines are listed below

DGCN [17]: DGCN proposes the first and second-order proximity of neighbors to design a new message-passing mechanism, which in turn learns aggregators based on incoming and outgoing edges using two sets of independent learnable parameters.

DIMPA [7]: DIMPA represents source and target nodes separately. However, DIMPA aggregates the neighborhood information within K hops in each layer to further increase the receptive field (RF), and it performs a weighted average of the multi-hop neighborhood information to capture the local network information.

NSTE [10]: NSTE is inspired by the 1-WL graph isomorphism test, which uses two sets of trainable weights to encode source and target nodes separately. Then, the information aggregation weights are tuned based on the parameterized feature propagation process to generate node representations.

DiGCN [16]: DiGCN notices the inherent connections between graph Laplacian and stationary distributions of PageRank, it theoretically extends personalized PageRank to construct real symmetric Digraph Laplacian. Meanwhile, DiGCN uses first-order and second-order neighbor proximity to further increase RF.

DiGCN-Appr [16]: DiGCN with fast personalized PageRank approximation, without inception blocks, which can be viewed as the generalization of GCN based on the digraph Laplacian.

DiGCN-IB [16]: DiGCN with inception blocks, without fast personalized PageRank approximation, which can be viewed as an optimized message-passing mechanism.

MagNet [21]: MagNet utilizes complex numbers to model directed information, it proposes a spectral GNN for digraphs based on a complex Hermitian matrix known as the magnetic Laplacian. Meanwhile, MagNet uses additional trainable parameters to combine the real and imaginary filter signals separately to achieve better prediction performance.

MGC [19]: MGC introduces the magnetic Laplacian, a discrete operator with the magnetic field, which preserves edge directionality by encoding it into a complex phase with an electric charge parameter. By adopting a truncated variant of PageRank named Linear-Rank, it design and build a low-pass filter for homogeneous graphs and a high-pass filter for heterogeneous graphs.

GCN [9]: GCN is guided by a localized first-order approximation of spectral graph convolutions. This model's scalability is directly proportional to the number of graph edges, and it learns intermediate representations in hidden layers that capture both the local graph arrangement and node-specific features.

GraphSAGE [6]: GraphSAGE is a scalable and flexible approach for node embedding generation in large graphs, which leverages a sampling-based neighborhood aggregation scheme and a multilayer perceptron to generate embeddings that capture both local and global structural information.

UniMP [15]: UniMP adopts the Graph Transformer model as the base model and then combines label embedding to feed node features and part of node labels into the model in parallel, which enables the propagation of node features and labels.

SGC [18]: SGC is a simple and efficient graph convolutional network that applies a fixed graph filter to the input features. It simplifies GCN by removing nonlinearities and collapsing weight matrices between consecutive layers, which achieves competitive performance on various graph-related tasks with significantly reduced computational cost compared to other GCN models.

SIGN [5]: SIGN is a scalable GNN that uses an inception module to learn hierarchical representations of nodes in large-scale graphs. It achieves state-of-the-art performance on various graph classification tasks and is scalable to graphs with millions of nodes.

GBP [3]: GBP lies in its bidirectional propagation mechanism, a process that computes a versatile Generalized PageRank matrix to capture a wide spectrum of graph convolutions, thereby expanding the horizon of expressive power within graph-based operations.

S^2GC [22]: S^2GC uses a modified Markov Diffusion Kernel to generalize GCN, and it can be used as a trade-off of low and high-pass filter which captures the global and local contexts of nodes.

GAMLP [20]: GAMLP introduces a duo of novel attention mechanisms: recursive attention and JK attention, revolutionizing the way representations are learned over RF of varying sizes in a dynamic, node-specific fashion.

1.8 Message Aggregation Functions

In this section, we extend our analysis of LightDiC by conducting experiments with different message aggregation functions. This expansion serves to further validate the viewpoints presented in Sec 3.1 of our main text: (1) Utilizing a weight-based method inspired by SIGN [5] to merge real and imaginary features could negatively impact predictive performance. This is attributed to the distinct physical meanings of the real and imaginary components.

Table 1: Performance on different message aggregation functions.

Datasets	Tasks	Last	Mean	Sum	Concat
CoraML	Node-C	**84.2±0.7**	83.6±0.5	84.0±0.6	83.9±0.5
	Existence	78.9±0.2	79.6±0.1	**80.3±0.2**	80.1±0.2
	Direction	90.3±0.2	90.5±0.3	**90.7±0.3**	90.6±0.2
	Link-C	74.1±0.2	74.3±0.1	74.6±0.1	**74.8±0.3**
WikiCS	Node-C	**79.8±0.4**	79.2±0.2	79.6±0.3	79.5±0.2
	Existence	87.8±0.0	**85.6±0.1**	85.5±0.0	85.3±0.1
	Direction	88.4±0.1	88.7±0.1	**89.2±0.1**	89.0±0.1
	Link-C	79.5±0.2	80.2±0.1	80.1±0.1	**80.4±0.2**
ogbn papers100M	Node-C	65.2±0.2	64.6±0.1	64.8±0.1	**65.4±0.2**
	Existence	90.7±0.0	91.3±0.0	**91.6±0.1**	91.2±0.1
	Direction	93.0±0.0	93.2±0.1	**93.8±0.1**	93.4±0.0
	Link-C	89.4±0.1	89.8±0.1	90.0±0.1	**90.3±0.1**

Employing a single learnable weight to combine them indiscriminately is ill-advised due to the lack of consideration for their individual characteristics. Conversely, a weight-free approach, while simple, can have a positive impact on prediction by intuitively encoding deep structural information. (2) Building upon the aforementioned point, in our proposed LightDiC, we opt for a weight-free message aggregation function to maintain simplicity, avoiding additional computational overhead caused by separately considering real and imaginary components, as well as the intricacies of a model architecture that combines both. This choice facilitates the handling of propagated features obtained from performing K-step propagation based on the MGO. For the weight-free message aggregation functions based on the K-step graph propagation, we consider the following: (i) Last$(\cdot)$ directly selects the propagated feature matrix obtained at the K-th step as the output. (ii) Mean$(\cdot)$ computes the element-wise average of propagated features; (iii) Sum$(\cdot)$ calculates the element-wise sum of propagated features; (iiii) Concat$(\cdot)$ concatenates K propagated feature matrices;

Building upon the experimental findings in Table 1, the following conclusions can be drawn: (1) For node-level tasks, the choice of optimal aggregation functions is contingent upon the dimensional of node features provided by the original dataset. For instance, in the case of feature-rich datasets such as CoraML and WikiCS, node prediction performance relies on smoothed features. Consequently, selecting Last$(\cdot)$ yields the best performance by attaining maximal smoothing signals. Conversely, for ogbn-papers100M with only 128 dimensions for node features, Concat$(\cdot)$ is necessary to enhance predictive performance through concatenating multi-scale node smoothing representations. (2)For link-level tasks, the enhancement of predictive performance necessitates capturing the intricate topological structure of the digraph. Thus, operators such as Mean$(\cdot)$, Sum$(\cdot)$, and Concat$(\cdot)$, which encode multi-dimensional deep structural information, tend to outperform Last$(\cdot)$ significantly. Additionally, due to Mean$(\cdot)$ mitigating difference between propagated features, resulting in less distinguishable final node representations, its effectiveness falls short compared to Sum$(\cdot)$. It is worth noting that as Link-C demands more fine-grained classification requirements, the enriched node representations generated by Concat$(\cdot)$ offer substantial benefits for its performance.

REFERENCES

[1] Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15, 6 (2003), 1373–1396.

[2] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*. https://openreview.net/forum?id=r1ZdKJ-0W

[3] Ming Chen, Zhewei Wei, Bolin Ding, Yaliang Li, Ye Yuan, Xiaoyong Du, and Ji-Rong Wen. 2020. Scalable graph neural networks via bidirectional propagation. *Advances in neural information processing systems, NeurIPS* 33 (2020), 14556–14566.

[4] Wenzheng Feng, Yuxiao Dong, Tinglin Huang, Ziqi Yin, Xu Cheng, Evgeny Kharlamov, and Jie Tang. 2022. Grand+: Scalable graph random neural networks. In *Proceedings of the ACM Web Conference 2022*. 3248–3258.

[5] Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. 2020. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198* (2020).

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems, NeurIPS* (2017).

[7] Yixuan He, Gesine Reinert, and Mihai Cucuringu. 2022. DIGRAC: Digraph Clustering Based on Flow Imbalance. In *Learning on Graphs Conference, LOG*. PMLR, 21–1.

[8] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems, NeurIPS* 33 (2020), 22118–22133.

[9] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations, ICLR*.

[10] Georgios Kollias, Vasileios Kalantzis, Tsuyoshi Idé, Aurélie Lozano, and Naoki Abe. 2022. Directed Graph Auto-Encoders. In *Proceedings of the Conference on Artificial Intelligence, AAAI*, Vol. 36. 7211–7219.

[11] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1361–1370.

[12] Paolo Massa and Paolo Avesani. 2005. Controversial users demand local trust metrics: An experimental study on epinions. com community. In *AAAI*, Vol. 1. 121–126.

[13] Péter Mernyei and Cătălina Cangea. 2020. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks. *arXiv preprint arXiv:2007.02901* (2020).

[14] Bruno Ordozgoiti, Antonis Matakos, and Aristides Gionis. 2020. Finding large balanced subgraphs in signed networks. In *Proceedings of The Web Conference, WWW*. 1378–1388.

[15] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2021. Masked label prediction: Unified message passing model for semi-supervised classification. *International Joint Conference on Artificial Intelligence, IJCAI* (2021).

[16] Zekun Tong, Yuxuan Liang, Changsheng Sun, Xinke Li, David Rosenblum, and Andrew Lim. 2020. Digraph inception convolutional networks. *Advances in neural information processing systems, NeurIPS* 33 (2020), 17907–17918.

[17] Zekun Tong, Yuxuan Liang, Changsheng Sun, David S Rosenblum, and Andrew Lim. 2020. Directed graph convolutional network. *arXiv preprint arXiv:2004.13970* (2020).

[18] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning, ICML*.

[19] Jie Zhang, Bo Hui, Po-Wei Harn, Min-Te Sun, and Wei-Shinn Ku. 2021. MGC: A complex-valued graph convolutional network for directed graphs. *arXiv e-prints* (2021), arXiv–2110.

[20] Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin Cui. 2022. Graph Attention Multi-Layer Perceptron. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD* (2022).

[21] Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. 2021. Magnet: A neural network for directed graphs. *Advances in neural information processing systems, NeurIPS* 34 (2021), 27003–27015.

[22] Hao Zhu and Piotr Koniusz. 2021. Simple spectral graph convolution. In *International conference on learning representations, ICLR*.