香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE
數據科學學院

深圳市大数据研究院
Shenzhen Research Institute of Big Data

Explainable Graph Representation Learning via Graph Pattern Analysis

Xudong Wang[1] Ziheng Sun[1,2] Chris Ding[1] Jicong Fan[1,2,*]

{xudongwang,zihengsun}@link.cuhk.edu.cn, {chrisding,fanjicong}@cuhk.edu.cn

[1]School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
[2]Shenzhen Research Institute of Big Data, Shenzhen, China

Paper ID: 5835
IJCAI 2025

* Corresponding author: fanjicong@cuhk.edu.cn

Outline

1. Introduction & Motivation
2. Core Contributions
3. Proposed Method 1: PXGL-EGK
4. Proposed Method 2: PXGL-GNN
5. Theoretical Analysis
6. Experimental Results
7. Conclusion Remarks

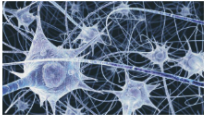
Introduction: The Ubiquitous Nature of Graphs

Modeling Complex Relationships

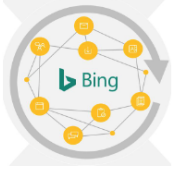
- **Graphs** are a fundamental and versatile data structure for **representing entities** and their **relationships**.
- **Diverse Applications**:

Academic Graph

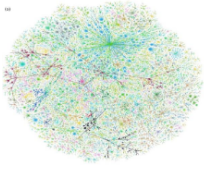
Office/Social Graph



Biological Neural Networks



Knowledge Graph



Internet

Transportation

Introduction of Graph Representation Learning

Graph Representation Learning

Encode graph data (nodes, edges, or entire graphs) into a set of **embeddings**. i.e. **vectorize graph**, transform it into embedding (feature vector).

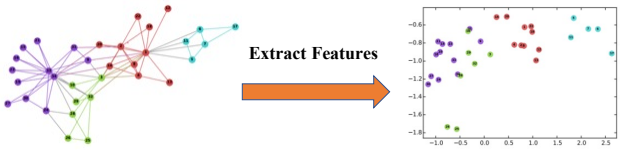


Figure: Input the graph (Karate Club) and visualization of z_v using t-SNE.

Approach	**Methods**
Manual, unsupervised	Graph feature extraction
Semi-automated, typically unsupervised	Graph kernels
Automated, typically supervised	Graph Neural Networks (GNNs)

Motivation & Key Question

The State of Explainable Graph Learning

Current XGL Landscape:

- **Model-level:** Explain GNN architectures, learned model behaviour: e.g XGNN [Yuan et al., 2020], GLG-Explainer [Azzolin et al., 2022], and GCFExplainer [Huang et al., 2023].
- **Instance-level:** Explain individual predictions, identify subgraph structure, ...
- **Gap:** What about **representations**?

Why Representation-level XGL Matters:

- Graph patterns carry domain meaning:
  - Cycles indicate chemical properties
  - Cliques characterize protein complexes
  - Paths reflect information flow
- Need interpretable representations for trustworthy AI
- Bridge structure and learned representations

Fundamental Question

What specific information about a graph is captured in its vector representation $\mathbf{g}$?

Limitation of Existing Methods:

- Graph kernels count patterns but ignore node features and are high-dimensional (Pattern counting can be vast)
- GNNs are powerful but lack transparency at the representation level

Our Contributions

1. **Formal Framework for Representation-level XGL**
   - Address what pattern information is important when graphs embed in representations
   - Graph pattern attributions can be inspected via joint learning
2. **Two Novel Methods**
   - **PXGL-EGK:** Explainable ensemble graph kernels
     - Weighted combination of pattern-counting kernels
     - Learn pattern importance through kernel weights
   - **PXGL-GNN:** Deep pattern-based representations
     - Feature-aware, end-to-end learning
     - Overcome limitations of kernel methods
3. **Theoretical Guarantees**
   - Robustness analysis against perturbations
   - Generalization bounds
   - Complexity analysis for practical deployment
4. **Extensive Experiment Validation and Superior Performance of Proposed Methods**
   - Superior experiment performance on eight widely used benchmark datasets
   - Comprehensive experiments on both supervised and unsupervised tasks

PXGL-EGK: Explainable Ensemble Graph Kernels

Pattern Counting Vector:

- Given graph $G = (\mathbf{A}, \mathbf{X})$, define $\mathbf{h} = \phi(G; \mathcal{P})$
- $h^{(i)}$ counts occurrences of pattern P_i in graph G

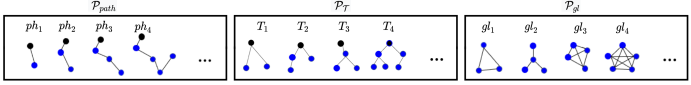


Figure: Examples of graph patterns: $\mathcal{P}_{\text{path}}$, $\mathcal{P}_{\text{T}}$ and $\mathcal{P}_{\text{gl}}$

Definition of Learnable Ensemble Kernel

Given M different kernels $\{K_{\mathcal{P}_1}, K_{\mathcal{P}_2}, \ldots, K_{\mathcal{P}_M}\}$ and weight vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M]^\top$:

$$K_{ij}(\boldsymbol{\lambda}) = \sum_{m=1}^{M} \lambda_m K_{\mathcal{P}_m}(G_i, G_j)$$

Key Insight: After learning with task specific info, the learned weights λ_m directly indicate the importance of pattern $\mathcal{P}_m$ for the task.

PXGL-EGK: Loss Functions & Optimization

Supervised Contrastive Loss:

$$\mathcal{L}_{\text{SCL}} = -\sum_{i \neq j} \mathbb{I}_{[\mathbf{y}_i = \mathbf{y}_j]} \Bigg(\log K_{ij}(\boldsymbol{\lambda}) - \log \Big[\sum_k \mathbb{I}_{[\mathbf{y}_i = \mathbf{y}_k, i \neq k]} K_{ik}(\boldsymbol{\lambda}) + \mu \sum_k \mathbb{I}_{[\mathbf{y}_i \neq \mathbf{y}_k]} K_{ik}(\boldsymbol{\lambda}) \Big] \Bigg)$$

Unsupervised KL Divergence:

$$\mathcal{L}_{\text{KL}} = \mathbb{KL}(\mathbf{K}(\boldsymbol{\lambda}), \mathbf{K}'(\boldsymbol{\lambda}))$$

where

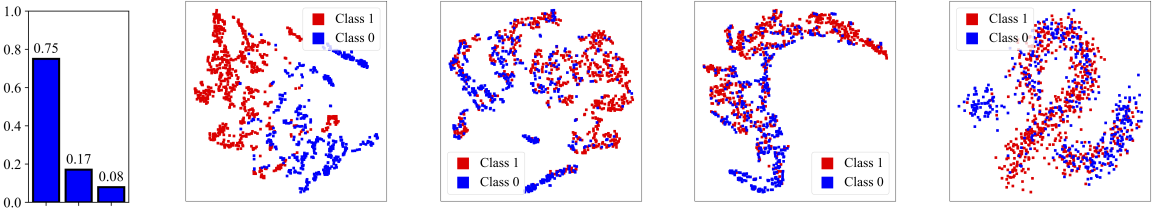
$$K'_{ij}(\boldsymbol{\lambda}) = \frac{K^2_{ij}(\boldsymbol{\lambda})/r_j}{\sum_{j'} K^2_{ij'}(\boldsymbol{\lambda})/r_{j'}}$$

Optimization Problem:

$$\boldsymbol{\lambda}^* = \underset{\mathbf{1}_M^\top \boldsymbol{\lambda} = 1, \boldsymbol{\lambda} \geq 0}{\arg\min} \; \mathcal{L}_{\text{ker}}(\boldsymbol{\lambda})$$

PXGL-EGK: Visualization & Results

(a) $\boldsymbol{\lambda}$ (b) $K(\lambda)$: ensemble (c) $K_{\mathcal{P}_1}$: path (d) $K_{\mathcal{P}_2}$: tree (e) $K_{\mathcal{P}_3}$: graphlet



$\sum_{m=1}^{M} \lambda_m \; K_{\mathcal{P}_m}$ ($\lambda_1 = 0.7502$) ($\lambda_2 = 0.1707$) ($\lambda_3 = 0.07912$)

Figure: t-SNE visualizations of PXGL-EGK's different kernel embeddings for the dataset PROTEINS.

Table: Graph Clustering Performance of PXGL-EGK.

	PROTEINS				MUTAG			
	RW	Sub-tree	Graphlet	**PXGL-EGK**	RW	Sub-tree	Graphlet	**PXGL-EGK**
ACC (↑ %)	71.2 ± 2.1	69.2 ± 2.7	63.6 ± 1.7	**72.1 ± 2.8**	74.3 ± 5.2	72.9 ± 1.3	73.5 ± 2.6	**76.1 ± 2.5**
NMI (↑ %)	26.8 ± 1.6	15.1 ± 2.8	15.4 ± 2.6	**32.1** ± 1.9	23.8 ± 1.6	19.5 ± 4.7	21.4 ± 1.9	**32.8** ± 4.6

Finding: Ensemble kernel outperforms traditional kernel methods via the joint learning of individual kernels and its importance weights $\boldsymbol{\lambda}$.

Limitations of Pattern Counting Methods

Key Limitations

1. **Ignoring Node Features:** Pattern counting vector $\mathbf{h}$ captures topology but ignores node features $\mathbf{X}$
2. **High Dimensionality:** Pattern set $\mathcal{P} = \{P_1, P_2, \ldots\}$ can be vast, making $\mathbf{h}$ high-dimensional and impractical
3. **Time Complexity:** Counting patterns like P_i in G is time-consuming, especially for large patterns
4. **Lacking Implicit Information:** Fixed patterns cannot capture implicit structural information that GNNs learn

Solution: PXGL-GNN

Combine pattern analysis with deep learning to overcome these limitations

PXGL-GNN: Pattern-based Deep Learning Framework

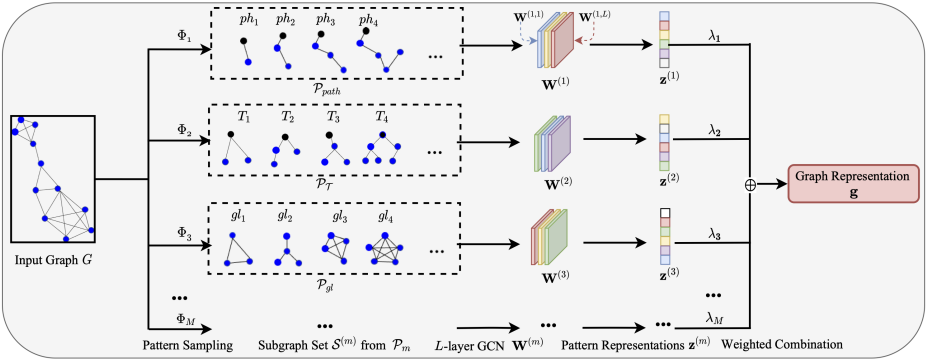


Figure: Framework of proposed PXGL-GNN

Three-Stage Process:

1. **Pattern Sampling:** Extract subgraphs $\mathcal{S}^{(m)}$ matching pattern $\mathcal{P}_m$
2. **Pattern Representation Learning:** GNN $F(\cdot; \mathcal{W}^{(m)})$ learns $\mathbf{z}^{(m)}$ for each pattern
3. **Explainable Weighted Combination:** $\mathbf{g} = \sum_{m=1}^{M} \lambda_m \mathbf{z}^{(m)}$

PXGL-GNN: Mathematical Formulation

Pattern Sampling:

- For pattern $\mathcal{P}_m$, sample set $\mathcal{S}^{(m)} = \{S_1, S_2, \ldots, S_Q\}$ where $S_q \in \mathcal{P}_m$

Pattern Representation Learning:

$$\mathbf{z}^{(m)} = \frac{1}{|\mathcal{S}^{(m)}|} \sum_{S \in \mathcal{S}^{(m)}} F(\mathbf{A}_S, \mathbf{X}_S; \mathcal{W}^{(m)}), \quad \forall m \in [M]$$

Ensemble Representation:

$$\mathbf{g} = \sum_{m=1}^{M} \lambda_m \mathbf{z}^{(m)}, \quad \text{with } \mathbf{1}^\top \boldsymbol{\lambda} = 1, \boldsymbol{\lambda} \geq 0$$

Advantages over PXGL-EGK:

- Incorporates node features through GNN
- Learns implicit patterns beyond counting vectors

PXGL-GNN: Loss Functions & Optimization

Supervised Classification Loss:

$$\mathcal{L}_{\text{CE}}(\boldsymbol{\lambda}, \mathcal{W}) = -\frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} \sum_{c=1}^{C} y_c \log \hat{y}_c$$

where $\hat{\mathbf{y}} = f_c(\mathbf{g})$ is the predicted label

Unsupervised KL Divergence Loss:

$$\mathcal{L}_{\text{KL}} = \text{KL}(K(\boldsymbol{\lambda}, \mathcal{W}), K'(\boldsymbol{\lambda}, \mathcal{W}))$$

where $K_{ij} = \exp(-\|\mathbf{g}_i - \mathbf{g}_j\|^2/\gamma)$ is Gaussian kernel

Joint Optimization:

$$\boldsymbol{\lambda}^*, \mathcal{W}^* = \underset{\mathcal{W}, \mathbf{1}^\top \boldsymbol{\lambda}=1, \boldsymbol{\lambda} \geq 0}{\arg\min} \mathcal{L}(\boldsymbol{\lambda}, \mathcal{W})$$

Robustness Analysis

Theorem 5.1 (Robustness):
For perturbations Δ_A and Δ_X on graph $\tilde{G} = (A + \Delta_A, X + \Delta_X)$:

$$\|\tilde{\mathbf{g}} - \mathbf{g}\| \leq \frac{\rho^L \beta_W^L}{\sqrt{n}(1+\alpha)^L} \cdot (1 + \beta_A + \|\Delta_A\|_2)^{L-1} \cdot [(1 + \beta_A + 2\|\Delta_A\|_2)\|\Delta_X\|_F + 2L\beta_X(1+\beta_A)\|\Delta_D\|_2]$$

where:

- L: number of GNN layers
- α: minimum node degree
- $\beta_W, \beta_A, \beta_X$: bounds on weights and features

Key Insights:

- Representation change bounded by perturbation magnitude on both structure and features. More connected graphs (larger α) are more robust.
- Robustness sensitive to the structure A when the depth L increases, and it is relatively insensitive to the perturbation on X.

Generalization & Complexity Analysis

Theorem 5.2 (Generalization):
With probability $\geq 1-\delta$:

$$\Pr\left[|\mathbb{E}[\ell_{CE}] - \hat{\mathbb{E}}[\ell_{CE}]| \geq c\left(\eta \log(N)\log\left(\frac{N}{\delta}\right) + \sqrt{\frac{\log(1/\delta)}{N}}\right)\right] \leq \delta$$

where error bound $\eta = \frac{\tau}{\sqrt{n}}\rho^L \hat{\beta}_W^{L-1}\beta_X(1+\beta_A)^L(1+\alpha)^{-L}\left[\hat{\beta}_W\gamma_{\Delta C} + \gamma_C\left(2\hat{\beta}_W + L\hat{\beta}_{\Delta W}\right)\right]$ for $|\ell_{\text{CE}}(\boldsymbol{\lambda}_{\mathcal{D}}, \mathcal{W}_{\mathcal{D}}; G) - \ell_{\text{CE}}(\boldsymbol{\lambda}_{\mathcal{D}^{\setminus i}}, \mathcal{W}_{\mathcal{D}^{\setminus i}}; G)| \leq \eta$

Complexity Analysis:
(n:Max Node Num., e:Max Edge Num., d:hidden dim, N:# graphs, B:batch size, Q:Sample Num., M:Pattern Num.,L:GNN layers Num.)

Method	Space Complexity	Time Complexity
PXGL-EGK	$\mathcal{O}(MN^2)$	$\mathcal{O}(N^2\sum_{m=1}^{M}\psi_m)$
PXGL-GNN (sup.)	$\mathcal{O}(BMQ(e+nd)+MLd^2)$	$\mathcal{O}(BMQL(ed+nd^2))$
PXGL-GNN (unsup.)	$\mathcal{O}(BMQ(e+nd)+MLd^2+B^2)$	$\mathcal{O}(BMQL(ed+nd^2)+B^2)$

Key Point: PXGL-GNN scales linearly with batch size B, making it practical for large datasets

Experimental Results

Please see the full results in our paper. Due to the space limitation of this slide, we only list several representative benchmark methods:

Supervised Graph Classification (ACC↑ %):

Method	MUTAG	PROTEINS	DD	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M5K
GIN	84.53±2.38	73.38±2.16	76.38±1.58	73.36±1.78	75.83±1.29	72.52±1.62	83.27±1.30	52.48±1.57
SAGNN	93.24±2.51	75.61±2.28	84.12±1.73	81.29±1.22	79.94±1.83	74.53±2.57	89.57±2.13	54.11±1.22
ICL	91.34±2.19	75.44±1.26	82.77±1.42	83.45±1.78	81.45±1.21	73.29±1.46	90.13±1.40	56.21±1.35
PXGL-GNN	**94.87±2.26**	**78.23±2.46**	**86.54±1.95**	**85.78±2.07**	**83.96±1.59**	**77.35±2.32**	**91.84±1.69**	**57.36±2.14**

Unsupervised Graph Clustering Performance (ACC↑ %, NMI↑ %):

Method	Metric	MUTAG	PROTEINS	DD	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M5K
InfoGraph	ACC	72.9±2.1	71.6±1.9	54.9±3.5	53.5±1.2	59.7±2.0	62.4±1.6	58.2±2.3	59.7±1.9
	NMI	23.6±0.5	23.1±0.3	26.6±0.4	26.3±0.5	31.1±0.8	19.8±0.5	20.6±0.6	28.6±0.6
GraphACL	ACC	74.2±2.3	73.1±2.7	57.2±2.7	52.2±1.3	55.4±1.3	67.9±1.3	59.4±1.4	56.7±2.3
	NMI	34.7±0.7	27.4±0.8	31.2±0.3	26.0±0.7	23.6±0.6	31.5±0.7	21.5±0.6	23.8±0.9
PXGL-GNN	**ACC**	**77.8±2.9**	**74.6±1.9**	**57.6±3.5**	**56.4±1.3**	**61.2±1.4**	**68.6±2.7**	**61.6±1.7**	**60.8±2.3**
	NMI	**35.2±0.6**	**29.2±1.0**	**31.7±0.3**	**32.7±0.8**	**37.2±0.7**	**32.4±1.1**	**22.4±0.9**	**29.5±1.2**

Our proposed PXGL-GNN consistently outperformed baselines in both supervised and unsupervised settings on eight benchmark datasets.

Pattern Importance Analysis

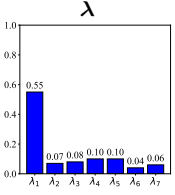
The learned explainable pattern weights λ of PXGL-GNN (supervised)

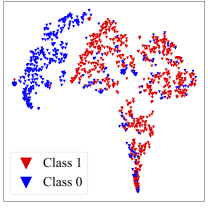
Pattern	MUTAG	PROTEINS	DD	NCI1
paths	0.095 ± 0.014	$\mathbf{0.550 \pm 0.070}$	0.093 ± 0.012	0.022 ± 0.002
trees	0.046 ± 0.005	0.074 ± 0.009	0.054 ± 0.006	0.063 ± 0.008
graphlets	0.062 ± 0.008	0.081 ± 0.011	0.125 ± 0.015	0.101 ± 0.013
cycles	$\mathbf{0.654 \pm 0.085}$	0.099 ± 0.013	0.094 ± 0.012	0.176 ± 0.022
cliques	0.082 ± 0.011	0.098 ± 0.012	$\mathbf{0.572 \pm 0.073}$	$\mathbf{0.574 \pm 0.075}$
wheels	0.026 ± 0.003	0.039 ± 0.005	0.051 ± 0.007	0.012 ± 0.002
stars	0.035 ± 0.005	0.056 ± 0.007	0.011 ± 0.002	0.052 ± 0.007

Learned weights align with domain knowledge:

- **MUTAG:** Cycles (0.65) - mutagenicity
- **PROTEINS:** Paths (0.55) - protein folding
- **DD/NCI1:** Cliques (0.57) - molecular structure

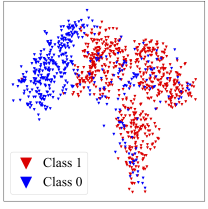
Example t-SNE visualizations of PXGL-GNN's pattern representations (supervised) for PROTEINS.





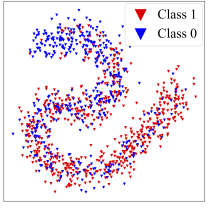
$\boldsymbol{g}$: ensemble

$\sum_{m=1}^{M} \lambda_m \boldsymbol{z}^{(m)}$



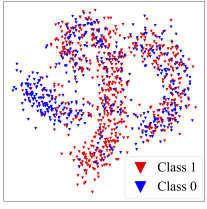
$\boldsymbol{z}^{(1)}$: path

$(\lambda_1 = 0.5504)$



$\boldsymbol{z}^{(2)}$: tree

$(\lambda_2 = 0.0746)$



$\boldsymbol{z}^{(3)}$: graphlet

$(\lambda_3 = 0.08103)$

Conclusion Remarks

Takeaways

- Graph representations can be made explainable through pattern analysis without sacrificing performance.
- Graph Pattern weights provide interpretable insights aligned with domain knowledge.

Two Novel Proposed Approaches:

- **PXGL-EGK:** Explainable ensemble graph kernels
  - Weighted combination of pattern-counting kernels
  - Learn pattern importance through kernel weights
- **PXGL-GNN:** Deep pattern-based representations
  - Feature-aware, end-to-end learning
  - Overcome limitations of kernel methods

Potential Future Work: Extend to dynamic graphs, etc.

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE
數據科學學院

深圳市大数据研究院
Shenzhen Research Institute of Big Data

Thanks for Listening!

Explainable Graph Representation Learning via Graph Pattern Analysis

Paper ID: 5835

Xudong Wang[1], Ziheng Sun[1,2], Chris Ding[1], Jicong Fan[1,2,*]

{xudongwang,zihengsun}@link.cuhk.edu.cn, {chrisding,fanjicong}@cuhk.edu.cn

[1]School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China

[2]Shenzhen Research Institute of Big Data, Shenzhen, China

IJCAI 2025

* Corresponding author: fanjicong@cuhk.edu.cn