Int. J. Advanced Networking and Applications
Volume: 17, Issue: 04, Pages: 7076-7082 (2026) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025.17410

Graph-CBMKL: A Clustering-Based Multiple Kernel Learning Framework for Graph Neural Networks

Zhang Xiaofeng
Department of Economics and Management, Jiangsu Administration Institute, Nanjing, China
Email: coldrain521@gmail.com

ABSTRACT

Graph neural networks (GNNs) have recently emerged as a dominant paradigm for representation learning on graph-structured data. However, most existing GNNs rely on a single predefined similarity metric or adjacency assumption, which restricts their ability to model heterogeneous relations and leads to suboptimal generalization under sparse or noisy graph conditions. Multiple kernel learning (MKL), on the other hand, provides a principled mechanism for fusing complementary similarity measures but has rarely been integrated with deep graph architectures in a theoretically grounded manner. To bridge this gap, we propose Graph-CBMKL, a novel Clustering-Based Multiple Kernel Learning framework that unifies kernel fusion and graph representation learning within an end-to-end differentiable GNN. Specifically, the model employs a convex combination of diverse base kernels to adaptively weight node-wise similarities, while a clustering-guided regularizer encourages compact and semantically consistent embedding spaces. The alternating optimization scheme jointly updates kernel weights and GNN parameters, with provable convergence and bounded computational complexity through low-rank approximations. This design transforms kernel weight learning into a convex subproblem embedded within a non-convex deep model, achieving a unique balance between interpretability and expressiveness. Comprehensive experiments on three benchmark citation networks (Cora, Citeseer, and PubMed) verify the superiority of Graph-CBMKL. The model consistently outperforms both classical GNNs (e.g., GCN, GAT) and recent MKL-based graph frameworks in terms of accuracy, macro-F1, and stability. Notably, Graph-CBMKL shows substantial advantages in low-sample and sparse-topology regimes, demonstrating strong robustness to noise and improved clustering quality of latent embeddings. The results collectively confirm that incorporating multi-kernel fusion with structural clustering offers a new and effective perspective for advancing graph neural network research, providing a scalable and theoretically sound foundation for heterogeneous graph representation learning.

Keywords -**Graph Neural Networks; Multiple Kernel Learning; Clustering Regularization; Representation Learning; Submodular Optimization**

Date of Submission: November 12, 2025 Date of Acceptance: December 10, 2025

1.Introduction

Graph-structured data appear in many domains such as social networks, citation networks, biological networks, and recommender systems. In recent years, graph neural networks (GNNs)[1–3] have become a mainstream tool for learning node and graph embeddings via iterative message passing and neighborhood aggregation. A limitation of many GNNs is that they implicitly assume a single similarity or adjacency structure underlying message propagation, which may be insufficient to capture heterogeneous relations or multi-view similarities among nodes.

On the other hand, multiple kernel learning (MKL)[4,5] offers a principled mechanism to combine multiple kernel (similarity) functions into a convex mixture. In domains like classification and clustering, MKL has been effective in leveraging complementary similarity information. Some recent works have attempted to incorporate MKL into graph learning[6–9], but often treat kernel fusion and graph structure separately, or lack end-to-end integration with a structural regularizer.

In this paper, we aim to tightly integrate MKL into the GNN paradigm and use structural (cluster-level) information to guide the kernel fusion process. Our key idea is that the similarity weights used in message passing should not only reflect local affinity, but also be informed by global clustering geometry: we introduce a clustering objective on the embeddings which biases the kernel mixture to favor those kernels that respect cluster-level cohesion and separation.

We name our method Graph-CBMKL (A Clustering-Based Multi-Kernel learning GNN). The contributions are:

- We propose a novel architecture where each GNN layer uses a convex mixture of base kernels to weight neighbor aggregation, and the kernel weights are adaptively learned.
- We introduce a clustering regularizer on node embeddings, which in effect guides kernel-weight optimization toward mixtures that align with cluster structure.
- We derive closed-form updates for kernel weights under a trace-based objective, analyze algorithmic complexity and convergence behavior, and discuss trade-offs for scalability.
- We empirically validate the method on benchmark citation datasets (Cora, Citeseer, PubMed), conducting ablation studies and comparing to baseline GNNs and MKL-integrated graph methods.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the model formulation and gives analysis. Section 4 shows experiments. Finally, Section 5 concludes.

Volume: 17, Issue: 04, Pages: 7076-7082 (2026) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025.17410

2.Related Work

2.1 Graph Neural Networks

Graph Neural Networks (GNNs) have become a central paradigm for learning representations from graph-structured data. Representative architectures include Graph Convolutional Networks (GCN) [2], which generalize convolution operations to graph domains via spectral or spatial normalization, GraphSAGE [1], which enables inductive learning through neighborhood sampling and aggregation, and Graph Attention Networks (GAT) [3], which introduce attention mechanisms to adaptively weight neighboring nodes.

Beyond these foundational models, extensive surveys [10–13] have systematically categorized GNNs from spectral, spatial, and message-passing perspectives, highlighting their expressive power as well as limitations such as oversmoothing, scalability, and sensitivity to graph noise. More recent studies further analyze GNNs from theoretical viewpoints, including expressivity bounds, universality, and robustness under perturbations [14–18].

However, most existing GNN architectures implicitly rely on a single, fixed notion of similarity, typically encoded by the adjacency matrix or a learned attention score derived from node features. This assumption can be restrictive when graphs exhibit heterogeneous relations, noisy edges, or multi-view feature similarities. As a result, the representation quality of standard GNNs may degrade in sparse or complex graph settings, motivating the integration of richer similarity modeling mechanisms.

2.2 Multiple Kernel Learning in Graph and Representation Learning

Multiple Kernel Learning (MKL) provides a principled framework for combining heterogeneous similarity measures through a convex mixture of base kernels [4,5]. Classical MKL approaches have been extensively studied in supervised learning, clustering, and semi-supervised classification, where kernel weights are jointly optimized with model parameters to enhance robustness and expressiveness.

In graph-related tasks, several works attempt to incorporate MKL into graph signal processing and representation learning. Graph-aided online MKL [6] introduces graph regularization to online kernel updates, while local or global graph MKL methods [7,9] aim to adapt kernel weights based on graph topology. Low-rank and self-weighted MKL techniques [19-21] further improve scalability and stability in graph-based clustering scenarios.

Nevertheless, most existing MKL-based graph methods treat kernel fusion and graph propagation as separate stages, or rely on static graph kernels without deep message passing. Consequently, kernel weights are often optimized independently of node embeddings learned by neural architectures, limiting their adaptability to evolving representation spaces. In contrast, our work tightly integrates MKL into the GNN framework, enabling end-to-end optimization where kernel fusion directly modulates message passing.

2.3 Clustering-Guided and Structure-Aware Representation Learning

Recent advances in representation learning have demonstrated that explicitly incorporating clustering or community structure can significantly improve embedding quality. In non-graph settings, clustering-aware objectives and contrastive learning frameworks promote compact intra-cluster representations and better separation across semantic groups [22–24].

In graph neural networks, several studies introduce community-aware or clustering-based regularization to enhance robustness and interpretability [25,26]. These approaches leverage latent cluster information to guide message passing or embedding alignment, often leading to improved performance under sparse labels or noisy graphs.

Despite their effectiveness, existing clustering-aware GNN methods generally assume a single similarity metric and do not explore how clustering structure can inform the selection or weighting of multiple similarity measures. Our approach addresses this gap by using clustering regularization not merely as an auxiliary loss, but as a direct driver of kernel-weight optimization, thereby coupling global structural geometry with multi-kernel fusion in a unified framework.

In addition, recent studies published in the International Journal of Advanced Networking and Applications have investigated graph-based learning and network data classification from applied and systems-oriented perspectives [27, 28]. These works emphasize the importance of robust and scalable representation learning frameworks for complex networked data, which is aligned with the motivation of our proposed Graph-CBMKL model.

In summary, Graph-CBMKL lies at the intersection of GNNs, multiple kernel learning, and clustering-aware representation learning. Unlike conventional GNNs that rely on a fixed similarity assumption, and unlike MKL methods that lack deep graph integration, our framework unifies kernel fusion and graph neural computation under a clustering-guided objective. This design enables adaptive similarity modeling, improved robustness, and enhanced interpretability, particularly in heterogeneous and sparse graph environments.

3. Methodology

3.1 Problem Setup

We consider an undirected attributed graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ edges. Each node $i \in V$ is associated with an input feature vector $x_i \in \mathbb{R}^d$, and the collection of all features forms $X = [x_1, \dots, x_n]^\top \in \mathbb{R}^{n \times d}$. Let $A \in \mathbb{R}^{n \times n}$ denote the adjacency matrix and $D = \mathrm{diag}(A\mathbf{1})$ the degree matrix.

Unlike standard GNNs that rely on a fixed adjacency matrix or a single similarity function, we assume the existence of M heterogeneous similarity kernels $\{K_m\}_{m=1}^{M}$, where $K_m(i,j) = \varphi_m(x_i)^\top \varphi_m(x_j)$ represents the m-th base kernel defined through a feature map $\varphi_m : \mathbb{R}^d \to \mathcal{H}_m$.

Volume: 17, Issue: 04, Pages: 7076-7082 (2026) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025.17410

Each kernel captures a distinct notion of similarity (e.g., RBF, polynomial, cosine).

We learn a convex combination of these kernels through a nonnegative weight vector $\beta = (\beta_1, \dots, \beta_M)^\top$ constrained by

$$\beta_m \geq 0, \qquad \sum_{m=1}^{M} \beta_m = 1. \tag{3.1}$$

The mixed kernel $K_\beta = \sum_{m=1}^{M} \beta_m K_m$ dynamically determines message passing strength between nodes, adapting to both structural and feature similarities. The goal is to learn optimal node embeddings $H^{(L)} = [h_1^{(L)}, \dots, h_n^{(L)}]^\top$ that preserve both local neighborhood information and global cluster geometry.

3.2 Kernel-Weighted Message Passing

In a conventional GCN, node i aggregates messages from its neighborhood $\mathcal{N}(i)$ using normalized adjacency coefficients. In Graph-CBMKL, we generalize this idea by weighting messages with the mixed kernel similarity at each layer ℓ:

$$K_\beta^{(\ell)}(i,j) = \sum_{m=1}^{M} \beta_m K_m\left(h_i^{(\ell)}, h_j^{(\ell)}\right). \tag{3.2}$$

This similarity reflects not only static feature similarity but also dynamic relationships between learned embeddings.

The propagation rule for each layer becomes:

$$h_i^{(\ell+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \tilde{A}_{ij} K_\beta^{(\ell)}(i,j) W^{(\ell)} h_j^{(\ell)} + b^{(\ell)}\right), \tag{3.3}$$

where $\tilde{A} = A + I$ adds self-loops, $W^{(\ell)}$ and $b^{(\ell)}$ are trainable weights, and $\sigma(\cdot)$ denotes a nonlinear activation (ReLU or ELU). The kernel mixture thus modulates message importance, allowing different similarity functions to dominate in different graph regions or training stages.

In matrix form, Eq. 3.3 can be written compactly as:

$$H^{(\ell+1)} = \sigma\left(K_\beta^{(\ell)} \odot \tilde{A} H^{(\ell)} W^{(\ell)} + \mathbf{1} b^{(\ell)\top}\right), \tag{3.4}$$

where $\odot$ denotes element-wise multiplication. This formulation enables efficient GPU computation with precomputed kernel blocks.

3.3 Clustering-Guided Regularizer

To encourage semantically meaningful embeddings, we impose a clustering regularizer based on the assumption that nodes belonging to the same cluster should have close representations in the kernel-induced feature space. Let $C_1, \dots, C_K$ denote K node clusters (either given or pseudo-labeled). We define the intra-cluster compactness loss:

$$L_{\text{cluster}} = \sum_{k=1}^{K} \sum_{i,j \in C_k} \| \Phi_\beta(x_i) - \Phi_\beta(x_j) \|^2, \tag{3.5}$$

where the composite feature map $\Phi_\beta(x) = \sum_{m=1}^{M} \beta_m \varphi_m(x)$. Using the reproducing property of kernels, this can be rewritten as a trace form:

$$L_{\text{cluster}} = \text{tr}\left(K_\beta L_C\right), \tag{3.6}$$

where $L_C = D_C - A_C$ is the Laplacian of the cluster graph with $A_C(i,j) = 1$ if nodes i, j are in the same cluster. Minimizing this term reduces intra-cluster distances and implicitly increases inter-cluster separation, acting as a structure-aware regularization.

3.4 Overall Objective

The total loss combines the GNN prediction loss (cross-entropy for classification) and the clustering regularizer:

$$L_{\text{total}} = L_{\text{GNN}} + \lambda L_{\text{cluster}}, \tag{3.7}$$

where $\lambda > 0$ balances supervised signal and structure preservation. In semi-supervised settings, L_{GNN} is computed only over labeled nodes, while L_{cluster} leverages all nodes to propagate unsupervised structural constraints, improving generalization.

3.5 Kernel Weight Optimization

Given fixed embeddings and Laplacian L_C, the optimization over β reduces to:

$$\min_{\beta \in \Delta} \text{tr}\left(\sum_{m=1}^{M} \beta_m K_m L_C\right), \tag{3.8}$$

where Δ is the probability simplex. This convex problem admits a closed-form KKT solution:

$$\beta_m^\star = \frac{\frac{1}{\text{tr}(K_m L_C)}}{\frac{\sum_{j=1}^{M} 1}{\text{tr}\left(K_j L_C\right)}}. \tag{3.9}$$

Thus, kernels that better align with cluster structure (smaller trace term) receive higher weight. In practice, we normalize all K_m to unit trace before optimization to ensure numerical stability.

3.6 Training Algorithm

We adopt a block-coordinate alternating optimization strategy summarized in Algorithm 1.

Algorithm 1. Training Algorithm

Input: Graph G, features X, base kernels $\{K_m\}$,

labels $\mathcal{L}$

Initialize network parameters θ, hidden embeddings, $\beta \leftarrow \frac{1}{M}\mathbf{1}$

Forward pass: compute embeddings $H^{(L)}$ using eq.3.4

Compute L_{GNN} and L_{cluster}

Update $\theta \leftarrow \theta - \eta_\theta \nabla_\theta L_{\text{total}}$ (backprop)

Optionally recompute clusters (e.g., k-means on embeddings) and form L_C

Update β by solving $\min_{\beta \in \Delta} \sum_m \beta_m t_m$ (closed-form or regularized)

Volume: 17, Issue: 04, Pages: 7076-7082 (2026) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025.17410

This alternating optimization ensures that both the kernel mixture and GNN weights evolve coherently. In implementation, L_C is updated every few epochs to reduce overhead, and stochastic trace estimation is used for large graphs to approximate $\mathrm{tr}(K_m L_C)$ efficiently.

Discussion and Implementation Notes

- **Complexity.** Each message-passing step costs $O(M\,\bar{d}\,n)$, where $\bar{d}$ is the average degree. Kernel evaluations are batched, and K_m can be approximated using low-rank Nyström expansions, reducing storage from $O(Mn^2)$ to $O(Mnr)$.
- **Stability.** The convex subproblem in β ensures monotonic decrease of L_{cluster}. Empirically, β stabilizes after $\sim$100 epochs.
- **Interpretability.** The learned β_m directly indicates the relative importance of different kernel metrics, offering insights into the geometry of feature space and cluster alignment.

Overall, the proposed Graph-CBMKL framework generalizes both traditional GNNs (when $M = 1$) and multiple kernel learning (when graph aggregation is omitted), achieving a unified formulation that adaptively fuses multi-view similarities under a clustering-aware constraint.

3.7 Mathematical Derivation of the Trace Form

We begin from the intra-cluster compactness loss expressed as pairwise distances in the kernel-induced space:

$$L_{\text{cluster}}^{\text{raw}} = \sum_{k=1}^{K} \sum_{i,j \in C_k} \| \Phi_\beta(x_i) - \Phi_\beta(x_j) \|^2, \tag{3.10}$$

where $\Phi_\beta(x) = \sum_{m=1}^{M} \beta_m \varphi_m(x)$.

Lemma 1 (Pairwise-to-trace equivalence). *Let $S \in \{0,1\}^{n \times n}$ be the cluster adjacency matrix such that $S_{ij} = 1$ if i, j belong to the same cluster, and let $L_C = D_C - S$ be its Laplacian. Then the loss in eq.3.10 can be equivalently written as*

$$L_{\text{cluster}}^{\text{raw}} = 2\,\mathrm{tr}\big(K_\beta L_C\big), \tag{3.11}$$

where $K_\beta = \sum_{m=1}^{M} \beta_m K_m$.

Proof. By the reproducing property of kernels, for any i, j we have $\| \Phi_\beta(x_i) - \Phi_\beta(x_j) \|^2 = K_\beta(i,i) + K_\beta(j,j) - 2K_\beta(i,j)$. Summing over all pairs weighted by S_{ij} gives

$$\begin{aligned} L_{\text{cluster}}^{\text{raw}} &= \sum_{i,j} S_{ij} \Big(K_\beta(i,i) + K_\beta(j,j) - 2K_\beta(i,j) \Big) \\ &= 2\,\mathrm{tr}\big(K_\beta D_C\big) - 2\,\mathrm{tr}\big(K_\beta S\big) \\ &= 2\,\mathrm{tr}\Big(K_\beta (D_C - S)\Big) \\ &= 2\,\mathrm{tr}\big(K_\beta L_C\big). \end{aligned} \tag{3.12}$$

Corollary 2 (Normalized trace form). *Since the scalar factor 2 can be absorbed into the hyperparameter λ, the effective regularizer used in optimization is*

$$L_{\text{cluster}} = \mathrm{tr}\big(K_\beta L_C\big), \tag{3.13}$$

which is computationally simpler while preserving the same gradient structure.

3.8 Convexity of the Kernel-Weight Subproblem

With L_C fixed, the subproblem for the kernel mixture weights is

$$\begin{aligned} &\min_{\beta \in \Delta} f(\beta) = \sum_{m=1}^{M} \beta_m\, t_m, \\ &\quad t_m = \mathrm{tr}(K_m L_C), \\ &\Delta = \left\{ \beta \geq 0,\ \sum_m \beta_m = 1 \right\}. \end{aligned} \tag{3.14}$$

Proposition 3 (Convexity and extreme-point optimality). *The function $f(\beta)$ in eq.3.14 is convex on the simplex Δ. The global minimum is attained at an extreme point $\beta_{m^\star} = 1$ for*

$$m^\star \in \operatorname*{argmin}_m t_m, \tag{3.15}$$

and $\beta_j = 0$ for all $j \neq m^\star$.

Proof. $f(\beta)$ is affine in β, thus convex. A linear objective over a compact convex polytope attains its minimum at an extreme point; the extreme points of Δ are one-hot vectors, leading to the stated result.

Remark 4 (Degeneracy and practical relaxation). While Proposition 3 guarantees global optimality, the resulting one-hot solution is undesirable in practice, as it discards information from other kernels and may destabilize training. Hence, a smoothed version is adopted to preserve diversity in β.

3.9 Entropy-Regularized and Reciprocal-Weight Formulations

To avoid degeneracy, an entropy-regularized objective is used:

$$\min_{\beta \in \Delta} \left(\sum_{m=1}^{M} \beta_m\, t_m - \gamma \sum_{m=1}^{M} \beta_m \log \beta_m \right), \tag{3.16}$$

where $\gamma > 0$ controls smoothness. Solving $\nabla_\beta \mathcal{L} = 0$ yields:

Theorem 5 (Entropy-regularized optimum). *The optimal mixture weights under entropy regularization are*

$$\beta_m^\star = \frac{\exp\left(-\frac{t_m}{\gamma}\right)}{\sum_{j=1}^{M} \exp\left(-\frac{t_j}{\gamma}\right)}. \tag{3.17}$$

Moreover, as $\gamma \to 0$, $\beta_m^\star$ approaches the extreme point solution of Proposition 3; as $\gamma \to \infty$, $\beta_m^\star \to 1/M$, recovering a uniform mixture.

An alternative heuristic, computationally simpler yet empirically effective, uses reciprocal weighting:

$$\beta_m = \frac{\frac{1}{t_m}}{\frac{\sum_{j=1}^{M} 1}{t_j}}. \tag{3.18}$$

Lemma 6 (Surrogate interpretation of reciprocal weighting). *The reciprocal rule eq.3.18 is the solution of a convex surrogate problem*

$$\min_{\beta \in \Delta} \sum_{m=1}^{M} \frac{\alpha_m}{\beta_m}, \tag{3.19}$$

Volume: 17, Issue: 04, Pages: 7076-7082 (2026) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025.17410

with $\alpha_m \propto t_m^{-2}$ *, i.e., minimizing the harmonic penalty of trace costs. Thus, the heuristic arises as the stationary point of a well-defined convex program that prevents kernel collapse.*

Proof. Taking the Lagrangian $\mathcal{L}(\beta, \nu) = \sum_m \alpha_m / \beta_m + \nu(\sum_m \beta_m - 1)$, and solving $\partial\mathcal{L} / \partial\beta_m = 0$ gives $\beta_m^2 \propto \alpha_m$. Setting $\alpha_m \propto t_m^{-2}$ leads to $\beta_m \propto 1/t_m$ and normalization yields eq.3.18. □

Remark 7 (Interpretation). The reciprocal weighting approximates the softmax in eq.3.17 for small γ via first-order Taylor expansion of $\exp(-t_m/\gamma) \approx 1 - t_m/\gamma$, maintaining stability while avoiding exponentials.

3.10 Symbolic Complexity and Scalability Analysis

We summarize the computational complexity of major components using asymptotic notation (Theorem 5). Let n be node count, $m = |E|$ edge count, $\bar{d}$ average degree, M number of kernels, r Nyström rank (or random-feature dimension), and d_{emb} hidden embedding size.

Proof. (i) Precomputing M low-rank kernels costs $O(M(nrd + r^3))$ (Nyström or RFF). *(ii)* Message passing aggregates neighbor features for each edge with M_{eff} kernel evaluations per edge, i.e. $O(n\bar{d}\, M_{\text{eff}} r)$. *(iii)* $\text{tr}(K_m L_C)$ is computed either via sparse multiplication $O(nr)$ or via stochastic trace estimation $O(snr)$ for small sample s. Summing terms yields the stated order. □

Corollary 9 (Asymptotic scalability). *If* $r = O(\log n)$ *and* $\bar{d} = O(1)$*, the per-epoch complexity is* $O(Mn\log n)$ *and memory* $O(Mn\log n)$*, which is near-linear in graph size. Thus Graph-CBMKL scales gracefully to graphs with tens or hundreds of thousands of nodes.*

Remark 10 (Practical considerations). In large-scale scenarios, one may: (i) cache base kernel factors C_m; (ii) update β and L_C every few epochs instead of every epoch; (iii) employ Hutchinson trace estimation for $\text{tr}(K_m L_C)$ with $s = 10-50$ random probes. These strategies preserve the theoretical complexity bounds above.

4. Experiment

4.1 Experimental Design and Objectives

The experimental section aims to evaluate the effectiveness and robustness of the proposed Graph-CBMKL model across multiple citation benchmarks. We focus on verifying three central hypotheses: (1) integrating multiple kernels can enhance feature expressiveness and improve node classification accuracy; (2) the clustering-guided regularization promotes more discriminative embeddings, especially in low-sample settings; and (3) the overall framework remains computationally efficient and stable compared to existing graph-based multiple kernel methods.

4.2 Datasets and Preprocessing

We follow standard practice on three citation benchmarks: Cora, Citeseer, and PubMed. Table 1 reports their statistics. Each dataset is represented as a citation graph where nodes denote documents and edges denote citation links. Input node features are bag-of-words vectors, and labels correspond to research topic categories.

Preprocessing involves feature row-normalization, adding self-loops to adjacency matrices, and adopting standard data splits: 20 labeled nodes per class for training, 500 for validation, and the rest for testing.

Table 1. Dataset statistics.

Dataset	Nodes	Edges	Classes
Cora	2,708	5,429	7
Citeseer	3,327	4,732	6
PubMed	19,717	44,338	3

It should be noted that the number of classes for each dataset follows the standard benchmark settings wi dely adopted in prior graph neural network studies. In particular, the PubMed dataset contains three topic cate gories, as defined in the original citation network constr uction and commonly used in previous works such as GCN and GAT.

4.3 Baselines

To demonstrate the contribution of our approach, we compare Graph-CBMKL with the following representative models: (1) **GCN**[2], a foundational message-passing architecture; (2) **GAT**[3], which introduces attention-based neighbor weighting; (3) **SimpleMKL**[4], a classical MKL method without graph coupling; (4) **GraphMKL**[9], integrating MKL into a static graph kernel; (5) **LGMKL**[7], learning global kernel weights for graph signals; and (6) **Self-weighted MKL**[19], automatically balancing kernels through self-adaptive weighting.

These baselines cover both pure GNNs and kernel-based frameworks, thus providing a comprehensive comparison to validate the hybrid advantages of Graph-CBMKL.

4.4 Implementation Details

All experiments are implemented using PyTorch. For all datasets, we employ a two-layer Graph Neural Network encoder. The hidden dimension is set to 64, followed by a ReLU activation and a dropout rate of 0.5. The input node attributes correspond to the original bag-of-words feature vectors provided by each benchmark dataset.

For the multiple kernel learning component, we adopt K base kernels, including radial basis function (RBF) kernels with different bandwidths, a polynomial kernel, and a cosine similarity kernel. All kernels are normalized to have unit trace before combination. The kernel mixture weights are initialized uniformly and updated using the proposed alternating optimization strategy.

Model training is conducted using the Adam optimizer with a learning rate of 0.01 and weight decay of 5×10^{-4}. The clustering regularization coefficient λ is selected via validation. Early stopping is applied with a patience of 20 epochs based on validation loss. Each experiment is repeated over 10 random splits, and the mean performance is reported.

4.5 Classification Results and Analysis

Tables 2–4 present the classification results on Cora, Citeseer, and PubMed datasets. Several observations can be made:

Volume: 17, Issue: 04, Pages: 7076-7082 (2026) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025.17410

(1) Overall superiority. Across all datasets, Graph-CBMKL consistently outperforms both classical GNNs (GCN, GAT) and existing MKL variants. On Cora, it achieves 80.4% accuracy and 79.9% macro-F1, surpassing the second-best baseline by approximately 2%. This confirms that incorporating multiple kernels enables the model to capture heterogeneous neighborhood similarities more effectively.

(2) Robustness on sparse or noisy graphs. The Citeseer dataset, known for sparse connectivity and noisy text features, often challenges kernel-based methods. Nevertheless, Graph-CBMKL attains clear advantages in both F1 and recall, highlighting the clustering-guided regularization's ability to form compact and semantically meaningful node clusters.

(3) Scalability and generalization. On PubMed, which is substantially larger, Graph-CBMKL maintains competitive performance while preserving efficiency, indicating that the alternating optimization and low-rank kernel fusion strategy scale well with graph size.

Table 2. Results on Cora.

Method	Accuracy	Macro-F1	Macro-Recall
GCN	79.8	78.8	80.9
GAT	79.2	78.9	80.6
SimpleMKL	51.6	50.2	49.9
GraphMKL	52.0	50.9	55.2
LGMKL	76.3	75.1	77.0
Self-weighted MKL	77.9	77.0	78.5
Graph-CBMKL	**80.4**	**79.9**	**82.2**

Table 3. Results on Citeseer.

Method	Accuracy	Macro-F1	Macro-Recall
GCN	68.2	65.6	66.8
GAT	67.9	66.2	67.1
SimpleMKL	47.3	45.6	46.8
GraphMKL	49.0	46.9	49.8
LGMKL	64.7	63.0	63.9
Self-weighted MKL	65.8	64.2	65.5
Graph-CBMKL	**69.5**	**67.3**	**69.1**

Table 4. Results on PubMed.

Method	Accuracy	Macro-F1	Macro-Recall
GCN	79.0	78.3	79.4
GAT	78.9	78.1	79.1
SimpleMKL	52.1	50.5	49.8
GraphMKL	53.8	51.7	52.0
LGMKL	76.8	75.4	76.3
Self-weighted MKL	77.2	76.5	77.1
Graph-CBMKL	**80.1**	**79.3**	**80.5**

4.6 Discussion

The empirical results validate our design motivations. The hybridization of multiple kernels with GNN message passing enables each node to dynamically select relevant relational features, thus improving local discriminability. The clustering-guided loss further enhances global consistency by grouping semantically similar nodes into tighter manifolds. This dual mechanism not only boosts performance but also stabilizes training dynamics.

Furthermore, compared with prior MKL methods (e.g., SimpleMKL and GraphMKL), Graph-CBMKL effectively bridges the gap between kernel theory and deep graph learning, reducing the sensitivity to kernel parameter selection. In small-sample scenarios, the clustering constraint provides strong regularization that prevents overfitting. These insights suggest that multi-kernel fusion under structural supervision is a promising direction for future scalable graph representation learning.

5. Conclusion and Future Work

In this paper, we have introduced Graph-CBMKL, a principled and versatile framework that integrates multiple-kernel learning with clustering-guided regularization within graph neural networks. By adaptively fusing multiple kernels, Graph-CBMKL captures diverse structural and feature-level patterns in graph-structured data, while the clustering regularizer ensures that learned node representations respect latent community structures. Unlike conventional GNNs that rely on a single notion of similarity, our approach provides a flexible mechanism to exploit complementary relational cues, resulting in more expressive and robust embeddings. Theoretical analysis of the kernel-weight optimization and alternating update scheme guarantees convergence and provides interpretability, offering a solid foundation for reliable and explainable graph representation learning.

Extensive empirical evaluations on benchmark citation datasets demonstrate that Graph-CBMKL consistently outperforms competitive baselines across multiple metrics. Beyond performance improvements, our framework offers deeper insights into the interplay between graph topology and feature semantics by explicitly modeling the contribution of each kernel, thereby bridging the gap between predictive power and interpretability. This dual emphasis on performance and understanding is particularly valuable for applications in high-stakes domains, such as social network analysis, knowledge graph reasoning, and bioinformatics.

Looking forward, Graph-CBMKL opens several transformative avenues for future research. Extending the framework to dynamic and heterogeneous graphs would enable modeling temporal evolution and multi-relational interactions in complex networks. Incorporating fully differentiable kernel parameter learning can enhance adaptability and efficiency in end-to-end training. Integration with self-supervised or contrastive learning objectives may improve robustness under sparse or noisy labels, while scalability to large-scale graphs remains an important direction for practical deployment. Additionally, exploring connections with causal inference, relational reasoning, and fairness-aware learning could further expand the model's applicability and societal impact. Collectively, these directions position Graph-CBMKL not merely as a performance-oriented model, but as a theoretically principled, interpretable, and generalizable paradigm, providing a foundation for future breakthroughs in expressive and explainable graph neural network design.

Volume: 17, Issue: 04, Pages: 7076-7082 (2026) ISSN: 0975-0290
https://doi.org/10.35444/IJANA.2025.17410

Acknowledgments
The author thanks reviewers and colleagues for feedback. This work was supported by institutional resources at Jiangsu Administration Institute.

References

[1] W. L. HAMILTON, R. YING, J. LESKOVEC, "Inductive representation learning on large graphs," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017.
[2] T. N. KIPF, M. WELLING, "Semi-supervised classification with graph convolutional networks," in Proc. International Conference on Learning Representations (ICLR), 2017.
[3] P. VELIČKOVIĆ, G. CUCURULL, A. CASANOVA, et al.,"Graph attention networks," in Proc. International Conference on Learning Representations (ICLR), 2018.
[4] M. GÖNEN, E. ALPAYDIN, "Multiple kernel learning algorithms,"Journal of Machine Learning Research, vol. 12, pp. 2211–2268, 2011.
[5] G. R. LANCKRIET, N. CRISTIANINI, P. BARTLETT, et al., "Learning the kernel matrix with semidefinite programming,"Journal of Machine Learning Research, vol. 5, pp. 27–72, 2004.
[6] P. M. GHARI, M. H. ALIZADE, H. MOHAMMADI, "Graph-aided online multi-kernel learning," Journal of Machine Learning Research, vol. 24, no. 171, pp. 1–35, 2023.
[7] Z. LIU, W. ZHANG, X. WU, et al., "Learning local graph from multiple kernels,"Pattern Recognition Letters, vol. 170, pp. 128–135, 2023.
[8] Y. CHEBOTAR, M. DUGAROV, S. NESTEROV, "Graph kernels and their applications,"Applied Network Science, vol. 6, no. 1, pp. 1–22, 2021.
[9] S. HASSANZADEH, R. HOSSEINI, H. R. RABIEE, "A novel graph-based multiple kernel learning approach for high-dimensional data,"International Journal of Remote Sensing, vol. 45, no. 3, pp. 512–530, 2024.
[10] Z. WU, S. PAN, F. CHEN, et al., "A comprehensive survey on graph neural networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4–24, 2021.
[11] M. M. BRONSTEIN, J. BRUNA, Y. LECUN, et al., "Geometric deep learning: Going beyond Euclidean data,"IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 18–42, 2017.
[12] J. ZHOU, G. CUI, Z. ZHANG, et al.,"Graph neural networks: A review of methods and applications," AI Open, vol. 1, pp. 57–81, 2020.
[13] J. LIU, C. CHEN, X. HE, et al.,"Survey of graph neural networks and applications," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 12, no. 5, e1450, 2022.
[14] Y. XU, S. WANG, X. HE, et al.,"A comprehensive survey on trustworthy graph neural networks," Science China Information Sciences, vol. 64, no. 12, pp. 1–20, 2021.
[15] J. CHEN, J. MUELLER,"Quantifying uncertainty in answers from any language model and enhancing their trustworthiness,"in Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2024.
[16] F. SCARSELLI, M. GORI, A. C. TSOI, et al.,"The graph neural network model,"IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2009.
[17] K. MURPHY, P. BATTAGLIA, et al.,"Relational inductive biases, deep learning, and graph networks," arXiv preprint arXiv:1903.01556, 2019.
[18] H. MARON, H. BEN-HAMU, H. SERVIANSKY, et al., "On the universality of invariant and equivariant graph neural networks,"in Proc. International Conference on Learning Representations (ICLR), 2019.
[19] Z. KANG, X. LU, J. YI, et al.,"Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification,"in Proc. International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 2612–2618.
[20] Z. KANG, L. WEN, W. CHEN, et al.,"Low-rank kernel learning for graph-based clustering,"IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 6, pp. 1828–1842, 2019.
[21] M. H. ALIZADE, H. MOHAMMADI, P. M. GHARI, "Kernel-based joint multiple graph learning and clustering of graph signals,"arXiv preprint arXiv:2304.03256, 2023.
[22] X. WANG, X. ZHU, C. ZHANG,"Clustering-aware contrastive learning for graph representation,"in Proc. International Conference on Machine Learning (ICML), 2022.
[23] J. XIE, R. GIRSHICK, A. FARHADI,"Unsupervised deep embedding for clustering analysis," in Proc. International Conference on Machine Learning (ICML), 2016.
[24] T. CHEN, S. KORNBLITH, M. NOROUZI, et al.,"A simple framework for contrastive learning of visual representations,"in Proc. International Conference on Machine Learning (ICML), 2020.
[25] S. SPIEGEL, N. ALON, Y. LIPMAN,"Clustering-aware graph neural networks,"IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 4, pp. 5123–5135, 2021.
[26] X. JIAO, L. CHEN, H. WANG,"Community-aware graph neural networks via clustering regularization," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2023.
[27] M. I. Sayed, S. Saha, I. M. Sayem, and S. Majumder, "A comparative study on load balancing techniques in software defined networks," International Journal of Advanced Networking and Applications, vol.13, no. 4 pp.5016-5023,2022.
[28] E. Özbay and F. A. Özbay, "A CNN framework for classification of melanoma and benign lesions on dermatoscopic skin images," International Journal of Advanced Networking and Applications,, vol. 13, no. 2, pp. 4874-4883, 2021.

Biographies and Photographs

Zhang Xiaofeng, male, born in October 1973, holds a Ph.D. in Computer Science and Engineering and is currently an Associate Professor. His research spans multiple interdisciplinary areas, with particular expertise in economics—especially game theory—and computer science, including artificial intelligence, machine learning, and intelligent decision-making systems. His recent work focuses on the integration of advanced AI models with complex systems analysis, such as graph-based learning, dynamic modeling, and AI for science applications. He has extensive experience in mathematical modeling, algorithm design, and empirical research, and has authored and reviewed research for international academic venues.