Tsinghua Science and Technology
Doi: 10.26599/TST.2026.9010086

Research/Review Article

SAC-Net: Structure-Aware Collaborative Network for Graph Similarity Computation

LingHang Zeng Yanling Li,(✉), Mingxia Bi, Shilong Lin, Zhijie Luo, Han Wang

Abstract Graph Similarity Computation (GSC) is a core task in graph analysis. However, current mainstream GNN-based similarity models still suffer from two fundamental bottlenecks. First, constrained by the inherent mechanism of recursive local aggregation, namely the 1-Weisfeiler–Lehman (1-wl) test, these models primarily measure similarity by aligning local structures, while struggling to capture long-range dependencies and overall topological configurations. Second, the simplified treatment of edge features prevents them from fully exploiting fine-grained semantic interactions between nodes. To address these challenges, this paper proposes Structure-Aware Collaborative Network (SAC-Net), an end-to-end framework that leverages structural information to unify global contexts with local affinities. Specifically, we design a Dynamic Structural Perception (DSP) backbone to establish a joint evolution paradigm for node, position, and edge features. By treating positional encodings as dynamic states, the model effectively captures long-range dependencies and overall topological configurations to maintain a robust global structural skeleton. Subsequently, this study introduce an Edge-Aware Fusion mechanism that leverages edge features as a bridge to adaptively integrate global and local structural information, thereby effectively addressing the alignment and integration of multi-granularity semantics. Extensive experiments on four real-world datasets demonstrate that SAC-Net effectively integrates global and local information, leading to more accurate graph similarity measurement.

Keywords Graph Similarity Computation, 1-Weisfeiler-Lehman Test, Positional Encodings, Edge-Aware Fusion

• Zeng Linghang, Mingxia Bi, Shilong Lin, Zhijie Luo, and Han Wang are with Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China, and also with Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China. Email: 1213826763@qq.com.
• Yanling Li is with the College of Computer Engineering, Guangxi Normal University, Guangxi Normal University, Guilin 541004, China. Email: liyanling@mailbox.gxnu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2026-03-10; revised: 2026-06-15; accepted: 2026-08-31

1 Introduction

Graph Similarity Computation (GSC) is a fundamental task in graph analysis, serving as a crucial bridge between raw non-Euclidean data and high-level knowledge discovery. Its significance is manifested across various domains: in biomedicine, it accelerates drug discovery by identifying structurally analogous active molecules; in cybersecurity, matching program call graphs enables robust malware detection; and in social networks, it drives personalized recommendations and fraud detection. Furthermore, GSC acts as an indispensable underlying operator for downstream tasks such as graph classification, clustering, and large-scale retrieval. Consequently, accurate and efficient similarity measurement directly dictates the overall performance and viability of these graph intelligence systems.

Traditional graph similarity computation primarily relies on combinatorial optimization algorithms to measure exact distances, such as Graph Edit Distance (GED)[1] or Maximum Common Subgraph (MCS)[2]. Typically, these methods model similarity calculation as a matching problem aimed at minimizing the transformation cost between graph pairs. For instance, A-star beam search[3] employs heuristic search strategies to approximate the optimal edit path, while the Hungarian algorithm[4] and the VJ algorithm[5] focus on solving the bipartite graph matching problem to derive similarity scores. Since these traditional and exact GED and MCS computations are known to be NP-hard[6], these traditional solvers face a strict trade-off between matching accuracy and scalability. Fig. 1 illustrates the detailed computation process of GED.

To bypass the NP-hard computational bottlenecks of traditional exact matching, recent research has shifted toward data-driven Graph Neural Networks (GNNs) for approximate similarity learning. Current methods typically fall into two paradigms. The first is representation-based methods, such as EGSC [7]. Unlike traditional algorithms that rely on computationally expensive combinatorial cross-graph matching, EGSC leverages a knowledge distillation framework to decouple dense cross-graph interactions. By precomputing individual graph embeddings offline and reducing online similarity measurement to a simple vector comparison,

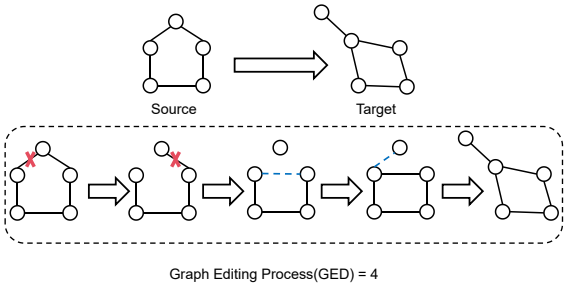


Fig. 1 Illustration of the Graph Edit Distance (GED) between a source graph and a target graph. The red marks indicate edge deletions, while the blue dashed lines denote edge insertions. The transformation from the source graph to the target graph is achieved through four edit operations: (1) delete the first edge (red mark); (2) delete the second edge (red mark); (3) insert a new edge (blue dashed line); (4) insert another new edge (blue dashed line). Since a total of four edit operations are required, the Graph Edit Distance (GED) between the two graphs is 4.

these methods enable fast linear-time inference.The second is interaction-based methods, represented by MGMN [8], which captures fine-grained structural patterns through a multi-level graph matching network framework. While MGMN achieves higher-precision similarity predictions by explicitly modeling cross-graph node alignments, this dense matching mechanism inherently introduces a quadratic time complexity of $O(N^2)$. Although it successfully utilizes GNNs to bypass the exponential overhead of traditional algorithms, this quadratic cost remains a secondary bottleneck, limiting its application in large-scale scenarios. Furthermore, despite both paradigms achieving encouraging results, two fundamental deficiencies persist:

Structural Blindness: A large body of prior work has demonstrated that standard Message Passing Neural Networks (MPNNs) are theoretically constrained by the 1-Weisfeiler-Lehman (1-WL) test [9–12]. Fundamentally, the message-passing paradigm relies on recursively aggregating local features[13] as unordered multisets. This mechanism inherently treats neighbors as independent entities, leading to an inability to encode their relative coordinates within the overall topological configurations[14], which significantly hinders the capture of the overall topological structure[15, 16]. As illustrated in Fig. 2, the two graphs are 1-WL equivalent and thus indistinguishable by MPNN-based methods. Because MPNNs perform permutation-invariant neighborhood aggregation, nodes with identical rooted subtree structures yield identical embeddings[9], resulting in identical graph-level representations.

Fine-grained Semantic Interaction Insufficiency: Most existing models heavily prioritize node attributes while neglecting the independent evolution of edge features[17, 18] during the message passing and readout phases. By treating edges merely as static connectivity indices rather than active semantic carriers, these methods fail to fully exploit fine-grained semantic interactions between nodes[19]. This oversight results in a loss of "structural resolution", preventing the model from detecting subtle topological mismatches that are critical for accurate similarity measurement.

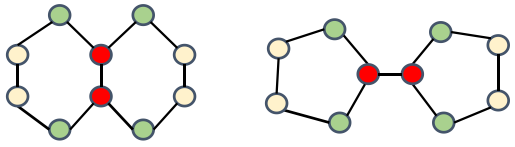


Fig. 2 The left graph and the right graph possess distinct Global Structural Skeletons. However, existing MPNNs rely on local message passing , making them sensitive only to local neighborhoods while remaining "blind" to the overall topological configurations. As shown by the identical node colors representing identical local degrees, the model aggregates identical local features, leading to indistinguishable embeddings.

To address these fundamental challenges, this paper proposes the SAC-Net framework. Unlike conventional methods that treat positional encodings as static inputs[20], SAC-Net adopts a DSP backbone that enables positional features and edge features to evolve dynamically with network depth. This allows the model to capture latent long-range dependencies and overall topological configurations, as well as fine-grained semantic interactions between nodes, thereby achieving a unified modeling of the overall topological context and local structural affinity. Furthermore, to alleviate the potential loss of structural information during the pooling and fusion process[21, 22], we design an Edge-Aware Fusion mechanism. This mechanism leverages edge features as a bridge to fuse node features, transforming simple mean aggregation into dynamically weighted views constrained by the overall topological configurations. In this way, the model significantly enhances its discriminative capability when distinguishing graphs with similar node attributes but distinct connectivity patterns.

- We propose a graph similarity computation framework designed to enhance overall topological perception and fine-grained semantic interactions. Through dynamic positional encodings and dynamic edge features, the framework accurately captures latent long-range dependencies, overall topological configurations, and fine-grained semantic interactions between nodes. In this way, the model effectively alleviates the expressive bot-

tlenecks inherent in traditional GNNs.

- We design an Edge-Aware Fusion mechanism and a Tri-Branch Interaction module, which effectively exploit fine-grained semantic interactions between nodes to significantly enhance the discriminative power of graph-level representations. This ensures that the model achieves high-precision similarity computation while maintaining linear time complexity.
- Extensive experiments on AIDS700, LINUX, IMDB-Multi, and PTC datasets demonstrate that SAC-Net achieves superior or competitive performance in both accuracy and efficiency.

2 Related work

2.1 Graph Similarity Computation

Graph similarity computation is a fundamental task in graph analysis. Recent neural-based models can be broadly categorized into two paradigms based on how they process structural information: methods focusing on global graph-level representations, and those emphasizing fine-grained cross-graph interactions.

Graph-Level Representation Learning: Methods in this category map each graph into a compact vector using a Siamese architecture. For instance, EGSC [7] employs knowledge distillation to transfer matching capabilities into lightweight models, SimGNN [23] combines GCNs with a Neural Tensor Network (NTN) [24] to generate global embeddings, and ERIC [25] utilizes alignment regularization to substitute explicit interaction modules. More recently, GRASP [26] enhances single-graph representation via positional encoding and multi-scale fusion. The primary advantage of these methods is their linear-time inference complexity, which is highly scalable for large-scale retrieval. However, compressing complex graph structures into a single global vector inevitably leads to a "lossy compression" of localized structural details.

Fine-Grained Interaction Learning: To address the lossy compression issue, another line of research captures local structural differences via dense cross-graph matching. GraphSim [27] treats node-pair similarity matrices as images processed by CNNs. MGMN [8] and H2MN [28] leverage multi-level and hypergraph matching networks to align local sub-structures, respectively. NA-GSL [29] introduces cross-graph co-attention, and CLSim [30] integrates cross-level interactions between node and graph embeddings. While these methods achieve superior accuracy by explicitly capturing fine-grained mappings, their quadratic computational complexity severely prohibits their application in real-world, large-scale graph databases.

Existing methods face a dilemma between the efficiency of global representations and the high precision of fine-grained interactions. To bridge this gap, SAC-Net aligns with the representation-based paradigm to maintain linear-time scalability. However, unlike previous global methods constrained by the 1-WL bottleneck and edge feature neglect, SAC-Net introduces a DSP backbone and an Edge-Aware Fusion mechanism. It extracts latent topological dependencies and fine-grained semantic affinities without computationally expensive cross-graph node matching. Consequently, SAC-Net achieves interaction-level high precision and deeper structural perception while maintaining linear-time complexity.

2.2 Positional Information and The 1-WL Limit

Message Passing Neural Networks (MPNNs), such as GCN and GIN, have become the standard paradigm for graph representation learning. However, their expressive power is theoretically upper-bounded by the 1-Weisfeiler-Lehman (1-WL) isomorphism test[9]. This expressive bottleneck stems from the inherent mechanism of recursive neighborhood aggregation.

Fundamentally, MPNNs represent the local structure of a node by iteratively aggregating the features of its neighbors. This process essentially captures the feature distribution of local subgraphs, meaning that nodes with highly symmetric or isomorphic local neighborhoods are easily encoded into identical feature representations. However, this paradigm is primarily sensitive to node degrees and local attributes, overlooking the actual topological connectivity among neighbors. Specifically, since the aggregation function is typically permutation-invariant [31, 32], it fails to distinguish between distinct subgraphs that share the same multiset of neighbor features.

A critical consequence of this limitation is the inability to capture non-tree-structured motifs. Since MPNNs essentially aggregate information through a process that can be viewed as tree-structured unfolding, they are inherently incapable of detecting closed-loop substructures[15]. Furthermore, the 1-WL test suffers from structural blindness in highly symmetrical or regular graphs, where nodes may occupy distinct global positions but possess identical local neighborhoods. To surmount this issue, injecting Positional Encodings (PE) has emerged as a prevalent strategy to break GNN symmetry[33, 34]. Recently, advanced models such as GraphGPS[35] and SAN[36] have established state-of-the-art benchmarks by utilizing Laplacian eigenvectors for PE. While

excelling in single-graph representation, adapting them to Graph Similarity Computation (GSC) poses challenges, as GSC requires fine-grained cross-graph matching heuristics lacking in standard readout mechanisms. Building upon these Laplacian-based insights, SAC-Net specifically tailors the use of eigenvectors for the GSC task. Unlike models treating PEs as static initial priors, we utilize dynamic positional encodings as a "Structural GPS". By explicitly tracking structural evolution during graph-pair interaction, we establish a joint evolution paradigm that dismantles the 1-WL ceiling, effectively capturing latent structural dependencies and global spatial configurations invisible to traditional message-passing schemes.

2.3 Representation Learning

The field of representation learning has undergone a transformative evolution, extending its influence beyond traditional graph theory into diverse scientific domains. Recent breakthroughs in volumetric 3D printing have demonstrated the power of advanced optimization and tomographic representation learning to achieve sub-second precision in complex material structuring[37]. In the context of graph-structured data, recent studies have increasingly focused on integrating domain-specific knowledge to enhance feature discriminativeness. For instance, the CircNet framework [38] introduces a dependency attention mechanism at both the node and edge levels, employing a gating strategy to capture topological and semantic features effectively. Similarly, the CKDTA model[39] presents a novel layer-wise attention network capable of learning representations from multi-layer graphs. Specifically, node-level attention aggregates the features of neighboring atoms into layer-specific embeddings, while semantic-level attention adaptively fuses information across layers to generate a comprehensive representation of the drug.

These multi-domain advances in representation learning-ranging from physical object reconstruction to knowledge-dense biological systems-highlight a critical paradigm shift: from static feature extraction to adaptive, context-aware encoding. Inspired by these pioneering concepts of high-dimensional feature optimization and knowledge-integrated learning, our work proposes a DSP mechanism. Unlike traditional methods that rely on static structural cues, our approach treats positional coordinates as evolving states, mirroring the sophisticated adaptive representation strategies observed in these leading studies to break the expressive bottlenecks of standard graph neural networks.

3 Problem Definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be a graph with node set $\mathcal{V}$, edge set $\mathcal{E}$, and node attribute matrix $\mathbf{X}$. Graph Similarity Computation (GSC) aims to learn a mapping $f : \mathcal{G} \times \mathcal{G} \to [0, 1]$ to approximate the structural similarity between graph pairs. To map the unbounded exact Graph Edit Distance (GED) into a normalized regression target, the ground-truth similarity y_{ij} between $\mathcal{G}_i$ and $\mathcal{G}_j$ is formally defined via an exponential decay function:

$$y_{ij} = \exp\left(-\frac{\mathrm{GED}(\mathcal{G}_i, \mathcal{G}_j)}{(|\mathcal{V}_i| + |\mathcal{V}_j|)/2}\right), \tag{1}$$

where $|\mathcal{V}|$ denotes the number of nodes in the respective graph.

MPNNs iteratively update node representations via:

$$h_v^{(l)} = \mathrm{UPD}^{(l)}\Big(h_v^{(l-1)}, \mathrm{AGG}^{(l)}(\{h_u^{(l-1)} : u \in \mathcal{N}(v)\})\Big), \tag{2}$$

where AGG and UPD are differentiable functions. Because AGG operates on unordered multisets, this formulation is theoretically bounded by the 1-Weisfeiler-Lehman (1-WL) test. It inherently ignores relative spatial coordinates and edge features, failing to distinguish graphs with identical local degrees but distinct overall topological configurations, and struggling to exploit fine-grained semantic interactions between nodes.

To break the 1-WL limit and capture overall topological configurations, we introduce global positional coordinates $P \in \mathbb{R}^{|\mathcal{V}| \times k}$ into a novel position-aware aggregation function:

$$h_v^{(l)} = \mathrm{UPD}^{(l)}\left(h_v^{(l-1)}, \mathrm{AGG}_{pos}^{(l)}\big(h_u^{(l-1)}, p_u^{(l-1)}\big)_{u \in \mathcal{N}(v)}\right). \tag{3}$$

where $h_v^{(l)}$ denotes the representation of node v at the l-th layer, $\mathcal{N}(v)$ is the set of its neighbors, and $p_u^{(l-1)}$ represents the global positional coordinates of neighbor u. Here $\mathrm{AGG}_{pos}^{(l)}$ and $\mathrm{UPD}^{(l)}$ are differentiable functions denoting the novel position-aware aggregation and the state update mechanisms.

4 Methods

The proposed SAC-Net is an end-to-end neural framework designed to capture both overall topological configurations and fine-grained semantic interactions between nodes for robust graph similarity learning. As illustrated in Fig. 3, the overall framework consists of four key stages to transform a pair of input graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ into a scalar similarity score.

4.1 Input and Structural Initialization

To enable the model to simultaneously capture semantic content, overall topological configurations, and fine-grained

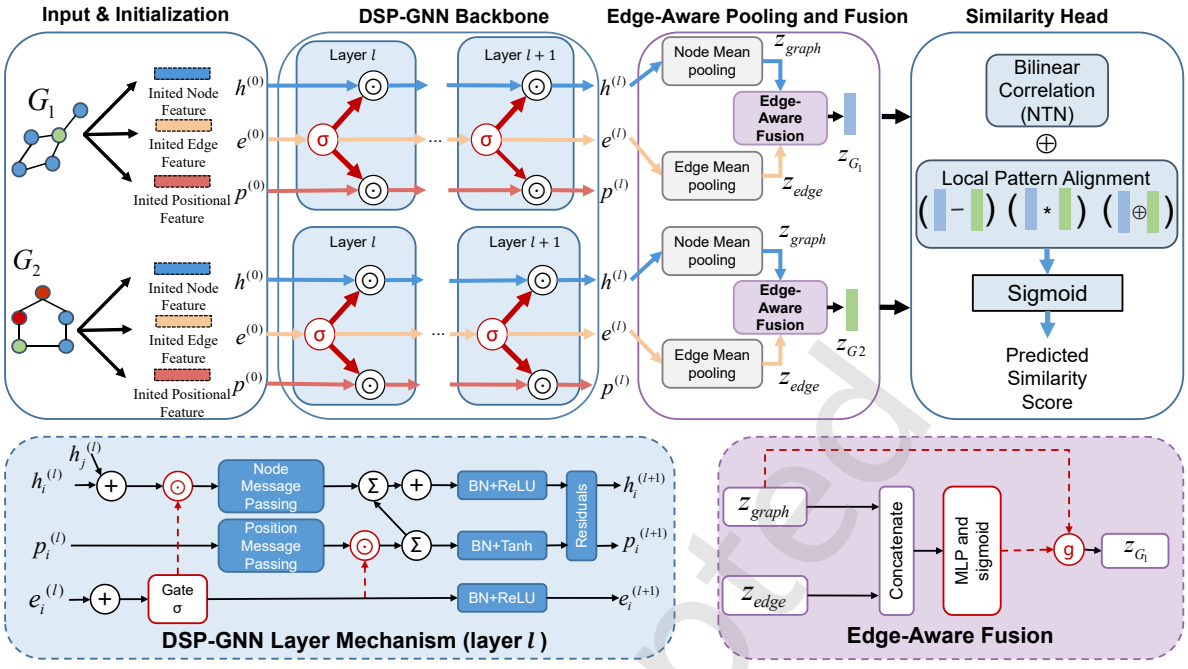


Fig. 3 Overview of SAC-Net. The model begins by initializing node, positional, and edge features. These streams are processed by the Dynamic Structural Perception Graph Neural Networks (DSP-GNN) backbone, which performs dynamic feature evolution to maintain structural sensitivity. Subsequently, the Edge-Aware Fusion mechanism uses global connectivity patterns to gate the pooled node representations. Finally, the Similarity Head employs a Tri-Branch Interaction mechanism to comprehensively compute the similarity score between the graph pair.

semantic interactions between nodes, we initialize three distinct streams of features: node features h^0, positional features p^0, and edge features e^0.

Node Feature Initialization (h^0). For a graph $\mathcal{G}$ with initial node attributes $\mathcal{X} \in \mathbb{R}^{N \times D_{in}}$, where D_{in} denotes the dimension of raw semantic features, we first project them into a d-dimensional hidden space using a learnable linear transformation. To accelerate training convergence and ensure scale invariance across graphs of different sizes, we apply Graph Normalization (GraphNorm) [40]. The initialized node feature h_i^0 for node v_i is computed as:

$$h_i^0 = \text{GraphNorm}(\mathbf{W}_{node} x_i + b_{node}), \tag{4}$$

where $\mathbf{W}_{node} \in \mathbb{R}^{d \times D_{in}}$ and $b_{node} \in \mathbb{R}^d$ are learnable parameters.

Global Positional Initialization (p^0). Standard GNNs rely on local message passing and lack awareness of the global graph structure. To address this, we initialize the positional stream using Laplacian Eigenvectors, which provide a unique global coordinate system for the graph. Specifically, we compute the normalized graph Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{A}$ is the adjacency matrix and $\mathbf{D}$ is the degree matrix. We perform eigendecomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ and select the top-k eigenvectors corresponding to the smallest non-trivial eigenvalues to form the positional matrix $\mathbf{U}_k \in \mathbb{R}^{N \times k}$. To allow the model to adaptively interpret these coordinates, we pass them through a MLP:

$$p_i^0 = \text{MLP}_{pos}(\mathbf{U}_k[i,:]), \tag{5}$$

where $U_k[i,:]$ denotes the i-th row of the eigenvector matrix. This initialization injects explicit overall topological configurations into the model from the very beginning.

Edge Feature Initialization (e^0). Edges represent the local affinity between nodes. Since many graph datasets do not contain explicit edge attributes, or to learn a task-specific structural bias, we initialize edge features using a learnable embedding layer. For every edge $(v_i, v_j) \in \mathcal{E}$, we assign an initial embedding:

$$e_{ij}^0 = \text{Embedding}(1) \in \mathbb{R}^d. \tag{6}$$

This ensures that the edge stream starts with a uniform learnable representation, which will be dynamically refined in subsequent layers based on the interacting nodes.

Note on Sign Invariance: Since Laplacian eigenvectors are defined up to a sign flip (i.e., $\mathbf{u}$ and $-\mathbf{u}$ are

both valid), we apply the absolute value transformation $p_i^0 = MLP_{pos}(|U_k[i,:]|)$ during initialization. This ensures that the structural encoding remains invariant to arbitrary sign assignments, providing a stable Structural GPS for breaking the 1-WL symmetry.

4.2 Dynamic Structural Perception Graph Neural Networks(DSP-GNN)

Standard GNNs typically inject Positional Encodings (PE) solely at the input layer. We identify a critical limitation in this static approach: "Coordinate Washout". As the network deepens, these initial global signals are progressively diluted by the smoothing effect of message passing, causing the model to lose its Structural GPS. To address this, we propose the DSP-GNN backbone, which treats positional coordinates not as fixed inputs, but as evolvable states. By orchestrating the parallel evolution of node contents (h), global positional coordinates (p), and local edge affinities (e), the model continuously recalibrates the overall topological context at each layer, ensuring that deep structural semantics are preserved against the 1-WL bottleneck.

4.2.1 Edge-Gated Affinity and Structural Denoising

The identification of structural noise is pivotal in graph similarity computation. A similar intuition has been explored in the Global Navigation Satellite System (GNSS) domain, where the spatial distribution characteristics of Satellites' Residual Vectors (SRV) are utilized to distinguish authentic signals from spoofing interferences[41]. Real-world graphs often contain "structural noise"–spurious connections that obscure the intrinsic graph structure. Standard neighbor aggregation treats all edges equally, propagating this noise and reducing structural resolution. Inspired by the gating mechanisms in foundation models [42], we employ an edge-gate $\sigma_{ij}^{(l)}$ as a differentiable structural filter. At layer l, the edge feature e_{ij} (representing Local Structural Affinity) is updated by fusing the features of its connecting nodes and its own previous state via linear transformations B_1, B_2, B_3:

$$\hat{\eta}_{ij}^{(l)} = B_1 h_i^{(l-1)} + B_2 h_j^{(l-1)} + B_3 e_{ij}^{(l-1)}, \tag{7}$$

Crucially, $\hat{\eta}_{ij}^{(\ell)}$ determines a gating coefficient $\sigma_{ij}^{(\ell)} \in (0,1)$ via the sigmoid function:

$$\sigma_{ij}^{(\ell)} = \sigma(\hat{\eta}_{ij}^{(\ell)}), \tag{8}$$

This gate acts as a denoising valve that effectively suppresses irrelevant connections while amplifying significant pathways. Consequently, this allows the model to aggregate information over the "effective structure" rather than the noisy raw graph. Although the formulation of $\sigma_{ij}^{(\ell)}$ appears localized to the endpoints, it inherently captures a broader global context. Because the node representation $h_i^{(\ell-1)}$ is recursively updated by concatenating the global positional encoding $p_i^{(\ell-1)}$ in previous layers (as detailed in Eq. (12)), the gate effectively leverages global topological coordinates to filter structural noise.

4.2.2 Position-Aware Message Passing and Dynamic Evolution

Leveraging the learned structural gate $\sigma_{ij}^{(\ell)}$, the model aggregates messages for both semantic content (h) and positional coordinates (p). To ensure the semantic updates are sensitive to the global structure, we explicitly concatenate the neighbor's positional feature $p_j^{(\ell)}$ into the node message flow:

$$m_{ij}^h = \sigma_{ij}^{(\ell)} \odot \mathbf{A}_2 \left([h_j^{(\ell)} | p_j^{(\ell)}] \right), \tag{9}$$

We now formally define the dynamic layer-wise evolution of the positional encodings. Rather than treating positions as static inputs, the structural message flow dictates their coordinate updates. The positional message m_{ij}^p aggregated from the local neighborhood $\mathcal{N}(i)$ is formulated as:

$$m_{ij}^p = \sigma_{ij}^{(\ell)} \odot \mathbf{C}_2 \left(p_j^{(\ell)} \right), \tag{10}$$

where $\mathbf{A}_2$ and $\mathbf{C}_2$ are learnable projection matrices, $\|$ denotes concatenation, and $\odot$ denotes the Hadamard product. This multiplicative interaction introduces high-order nonlinearity, significantly enhancing the model's expressiveness beyond standard additive aggregation [43].

4.2.3 State Update and Residuals

The aggregated messages are combined with the node's previous state to compute the final representations for layer $\ell+1$. To facilitate training and prevent the vanishing gradient problem, we employ residual connections [44] and Batch Normalization (BN).Formally, the dynamic evolution of the positional coordinate p_i from layer ℓ to $\ell+1$ is strictly defined as:

$$p_i^{(\ell+1)} = p_i^{(\ell)} + \mathrm{Tanh}\Big(\mathrm{BN}\big(\mathbf{C}_1(p_i^{(\ell)}) + \sum_{j \in \mathcal{N}(i)} m_{ij}^p\big)\Big), \tag{11}$$

Concurrently, the semantic node representations and edge affinities are updated as follows:

$$h_i^{(\ell+1)} = h_i^{(\ell)} + \mathrm{ReLU}\Big(\mathrm{BN}\big(\mathbf{A}_1\big[h_i^{(\ell)} \,\|\, p_i^{(\ell)}\big] + \sum_{j\in\mathcal{N}(i)} m_{ij}^{h}\big)\Big), \tag{12}$$

$$e_{ij}^{(\ell+1)} = e_{ij}^{(\ell)} + \mathrm{ReLU}\left(\mathrm{BN}\left(\hat{\eta}_{ij}^{(\ell)}\right)\right). \tag{13}$$

It is worth noting that we apply the Tanh activation function exclusively for the positional stream $p_i^{(\ell+1)}$. This mathematically ensures that the evolving positional coordinates remain within a bounded numerical range, stabilizing the continuous recalibration of the global coordinate system. Through stacking K such layers, the DSP-GNN backbone generates the final representations $h^{(K)}$, $e^{(K)}$, and $p^{(K)}$, which inherently encode both deep topological configurations and fine-grained semantic interactions.

4.3 Edge-Aware Pooling and Fusion

While DSP-GNN successfully evolves structural features, a critical risk remains at the readout phase. Previous approaches typically discard edge features, performing pooling solely on nodes. We argue that this node-centric aggregation causes Structural Collapse, where graphs with distinct local structures become indistinguishable if their average node features are similar. To preserve the structural fidelity developed in deep layers, we propose an Edge-Aware Fusion mechanism that utilizes fine-grained semantic interactions between nodes to explicitly modulate node representations.

4.3.1 Dual-Stream Mean Pooling

We aggregate local features into graph-level descriptors. Crucially, we maintain a dedicated stream for edges, as they encode the fine-grained semantic interactions between nodes, which are essential for characterizing the overall topological configurations of the graph. We employ Mean Pooling for both streams to ensure permutation invariance:

$$z_{graph} = \frac{1}{|\mathcal{V}|}\sum_{v_i\in\mathcal{V}} h_i^{(L)}, \tag{14}$$

$$z_{edge} = \frac{1}{|\mathcal{E}|}\sum_{(v_i,v_j)\in\mathcal{E}} e_{ij}^{(L)}, \tag{15}$$

After pooling, node embeddings are aggregated into a graph-level representation, denoted as $z_{graph} \in \mathbb{R}^d$, while edge embeddings are aggregated into a graph-level edge representation, denoted as $z_{edge} \in \mathbb{R}^d$. Here, $z_{graph} \in \mathbb{R}^d$ aggregates the semantic content and positional context ("Where am I?"), while $z_{edge} \in \mathbb{R}^d$ aggregates the intensity of pairwise interactions ("How tight is the connection?"). z_{edge} serves as a summary of the graph's overall connectivity pattern.

4.3.2 Edge-Aware Gating Mechanism

To integrate these two modalities, we design a non-linear gating mechanism. Merely concatenating z_{graph} and z_{edge} would treat structural and semantic information linearly. Recent studies have also demonstrated that adaptive multi-scale feature fusion can effectively integrate node-level and graph-level representations and improve the discriminative capability of graph similarity models [45]. Instead, we use the global edge context to determine the importance of node features. We first concatenate the pooled representations and pass them through a MLP ending with a Sigmoid activation to generate a fusion gate g:

$$g = \sigma\left(\mathbf{W}_2 \cdot \mathrm{ReLU}\left(\mathbf{W}_1 \cdot [z_{graph} \| z_{edge}]\right)\right), \tag{16}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times 2d}$ and $\mathbf{W}_2 \in \mathbb{R}^{d\times d}$ are learnable weights. The gate $g \in (0,1)^d$ acts as a structural filter applied to the node embeddings via element-wise multiplication, The final graph representations $z_{\mathcal{G}}$ is as follows:

$$z_{\mathcal{G}} = g \odot z_{graph}. \tag{17}$$

4.3.3 Mechanism Analysis

Topological Contextualization: By conditioning the gate on z_{edge}, the model incorporates the overall topological configurations. For instance, in graphs with complex connectivity patterns, the accumulated edge features can modulate node representations based on fine-grained semantic interactions between nodes, thereby strengthening the perception of long-range dependencies and suppressing redundant information. The edge stream provides the necessary structural context to interpret the node stream from an overall topological perspective.

Structural Denoising (Soft Attention): The Sigmoid gate functions as a soft attention mechanism for feature selection. If the network deems certain node feature dimensions irrelevant or noisy given the current structural topology, the gate value approaches 0, effectively suppressing these signals. Conversely, critical features aligned with the structural pattern are retained. We employ an edge-aware fusion mechanism to filter structural information, an attention-enhanced strategy that has proven effective in extracting critical features and suppressing noise in heterogeneous knowledge representation[46].

Non-linear Adaptive Modulation: The Fusion Gate transforms the graph representation from a static average of nodes

into a dynamically weighted view constrained by the Global Structural Skeleton. This non-linear mapping allows SAC-Net to distinguish graphs that possess similar node attribute distributions but distinct fine-grained semantic interaction patterns, thereby significantly boosting the discriminative power against the 1-WL bottleneck.

4.4 Tri-Branch Interaction and Prediction

After obtaining the structure-aware graph representations z_{G_1} and z_{G_2}, relying on a single distance metric is insufficient to capture the multifaceted nature of graph differences. To address this, we propose a Tri-Branch Interaction Mechanism. Recent studies have demonstrated that multi-view representation learning can effectively capture complementary structural patterns and enhance semantic consistency across different representation spaces [47]. Motivated by this general principle, we extract comprehensive interaction features through three parallel and complementary branches:

1. Bilinear Correlation (The NTN View): To capture complex, high-order multiplicative interactions across different latent dimensions, we employ a Neural Tensor Network (NTN). The interaction feature for the k-th slice of the tensor is defined as:

$$\mathbf{f}_{ntn}^{(k)} = \sigma\left(\mathbf{h}_1^T \mathbf{W}^{[k]} \mathbf{h}_2 + \mathbf{V}_k^T [\mathbf{h}_1 | \mathbf{h}2] + b_k\right), \tag{18}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d \times K}$ is a learnable core tensor consisting of K slices, $\mathbf{V}_k \in \mathbb{R}^{2d}$ is the weight vector for the concatenated representations, and $b_k \in \mathbb{R}$ is the bias. The outputs from all K slices are concatenated to form the dense bilinear interaction vector $\mathbf{f}_{ntn} \in \mathbb{R}^K$.

2. Global Structural Deviation (The Distance View): To quantify the spatial displacement between graph skeletons in the latent space, we compute the absolute difference vector:

$$\mathbf{f}_{dist} = |\mathbf{h}_1 - \mathbf{h}2|, \tag{19}$$

where $|\cdot|$ denotes the element-wise absolute value. Since our embeddings explicitly encode dynamically evolved positional states, $\mathbf{f}dist \in \mathbb{R}^d$ serves as a direct proxy for global structural deviation.

3. Local Pattern Alignment (The Alignment View): To measure the fine-grained semantic compatibility and highlight shared local substructures between the graph pair, we compute the element-wise (Hadamard) product:

$$\mathbf{f}_{align} = \mathbf{h}_1 \odot \mathbf{h}2, \tag{20}$$

where $\mathbf{f}align \in \mathbb{R}^d$ effectively captures the degree of alignment between the corresponding semantic attributes of the two graphs.

Similarity Prediction: Finally, we aggregate these multi-view interaction features to yield the final similarity score. First, we concatenate the original embeddings with the linear interaction views and process them through a Multi-Layer Perceptron (MLP) to fuse the basic features:

$$\mathbf{h}cmp = \text{MLP}cmp\left([\mathbf{h}1 | \mathbf{h}2 | \mathbf{f}dist | \mathbf{f}align]\right), \tag{21}$$

where $\mathbf{h}cmp \in \mathbb{R}^{d_{out}}$. This fused linear representation is then concatenated with the high-order bilinear features $\mathbf{f}_{ntn}$ to compute the final graph similarity score $s(\mathcal{G}_1, \mathcal{G}_2) \in (0, 1)$:

$$s(\mathcal{G}1, \mathcal{G}2) = \sigma\left(\mathbf{W}final^T [\mathbf{f}ntn | \mathbf{h}cmp] + \mathbf{b}final\right). \tag{22}$$

where $\mathbf{W}_{final}$ and $\mathbf{b}_{final}$ are the trainable weights and bias of the final prediction layer, and σ denotes the sigmoid activation function. This explicit architectural design ensures that the model's similarity decisions are rigorously grounded in a holistic aggregation of linear structural deviations, local semantic alignment, and high-order tensor correlations.

4.5 Loss Function

To train SAC-Net in an end-to-end manner, we design a hybrid objective function that simultaneously optimizes for accurate similarity prediction and structural representation fidelity. The total loss $\mathcal{L}$ is a weighted combination of the main regression loss and an auxiliary positional regularization term.

4.5.1 Similarity Regression Loss

The primary goal of SAC-Net is to approximate the ground-truth similarity score $y \in (0, 1]$. Since we formulate GSC as a regression problem, we employ the Mean Squared Error (MSE) loss to minimize the discrepancy between the predicted score $\hat{s}(\mathcal{G}_i, \mathcal{G}_j)$ and the ground-truth y_{ij}. For a mini-batch $\mathcal{B}$ of graph pairs, the regression loss is defined as:

$$\mathcal{L}_{reg} = \frac{1}{|\mathcal{B}|} \sum_{(\mathcal{G}_i, \mathcal{G}_j) \in \mathcal{B}} \left(\hat{s}(\mathcal{G}_i, \mathcal{G}_j) - y_{ij}\right)^2, \tag{23}$$

where y_{ij} is the normalized GED score as defined in Section 3.

4.5.2 Auxiliary Positional Regularization

While the dynamic evolution of positional features is a key innovation of SAC-Net, deep message passing can cause these embeddings to drift significantly from their geometric origins. To prevent such "Coordinate Collapse" and preserve structural fidelity before the readout phase, we introduce an auxiliary positional loss $\mathcal{L}_{pos}$.

Instead of penalizing every intermediate layer, $\mathcal{L}_{pos}$ applies a targeted constraint strictly on the final layer's positional

embeddings to maintain consistency with the graph's intrinsic spectral properties. For a graph $\mathcal{G}$ with pre-computed Laplacian eigenvectors $U \in \mathbb{R}^{|\mathcal{V}|\times k}$, we penalize the deviation of the terminal layer's positional features $\mathbf{p}^{(L)}$ across a mini-batch $\mathcal{B}$:

$$\mathcal{L}_{pos} = \frac{1}{|\mathcal{B}|}\sum_{\mathcal{G}\in\mathcal{B}}\frac{1}{|\mathcal{V}_{\mathcal{G}}|}\sum_{v\in\mathcal{V}_{\mathcal{G}}}\left\|\phi_{proj}(\mathbf{p}_v^{(L)}) - u_v\right\|_2^2, \quad (24)$$

where u_v is the exact spectral coordinate of node v, and ϕ_{proj} is a projection head for dimension alignment.

By exclusively regularizing the final layer (L), we establish a robust structural anchor for the ultimate representations while preserving the flexibility for intermediate layers to undergo task-specific structural adaptations.

4.5.3 *Total Objective*

The final objective function is a weighted sum of the regression loss and the positional regularization:

$$\mathcal{L}_{total} = \mathcal{L}_{reg} + \alpha \cdot \mathcal{L}_{pos}.$$

where α is a hyperparameter balancing the trade-off between similarity fitting and structural preservation. Crucially, instead of acting as a rigid hard constraint that freezes the coordinates to their initial eigenvectors, $\mathcal{L}_{pos}$ functions as a soft regularization. Guided by α, it provides a structural "anchor" to prevent Coordinate Collapse while preserving sufficient flexibility for the positional features to undergo task-specific structural adaptation during similarity regression. By jointly optimizing these two objectives, SAC-Net ensures that the predicted similarity is based on robust, structure-aware node representations.

The overall learning algorithm for SAC-Net is presented in Algorithm 1. It outlines the forward propagation from graph pair inputs to similarity score prediction, alongside the computation of the structure-constrained loss function.

Algorithm 1 Similarity Computation of SAC-Net

Require: Graph pair $\mathcal{G}_1 = (\mathbf{X}_1, \mathbf{P}_1, \mathbf{E}_1), \mathcal{G}_2 = (\mathbf{X}_2, \mathbf{P}_2, \mathbf{E}_2)$, Ground-truth similarity y_{gt}
Ensure: Predicted similarity score $\hat{y}$
Function DSP-Encoder($\mathcal{G}$)
Initialize node $\mathbf{h}^{(0)}$, positional $\mathbf{p}^{(0)}$, and edge $\mathbf{e}^{(0)}$ representations
for $l = 1$ **to** L **do**
  // 1. Edge-Gated Affinity
  $\sigma_{ij}^{(l)} \leftarrow \textsc{Sigmoid}\big(\textsc{Linear}(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}, \mathbf{e}_{ij}^{(l-1)})\big)$
  // 2. Position-Aware Message Passing & Update
  $\mathbf{h}_i^{(l)} \leftarrow \mathbf{h}_i^{(l-1)} + \textsc{UpdateNode}\big(\sum_{j\in\mathcal{N}(i)} \sigma_{ij}^{(l)} \odot \mathbf{m}_{ij}^{h}\big)$
  $\mathbf{p}_i^{(l)} \leftarrow \mathbf{p}_i^{(l-1)} + \textsc{UpdatePos}\big(\sum_{j\in\mathcal{N}(i)} \sigma_{ij}^{(l)} \odot \mathbf{m}_{ij}^{p}\big)$
  $\mathbf{e}_{ij}^{(l)} \leftarrow \mathbf{e}_{ij}^{(l-1)} + \textsc{UpdateEdge}\big(\hat{\eta}_{ij}^{(l)}\big)$
end for
// 3. Edge-Aware Pooling and Fusion
$\mathbf{h}_{pool} \leftarrow \textsc{MeanPool}(\mathbf{h}^{(L)})$; $\mathbf{e}_{pool} \leftarrow \textsc{MeanPool}(\mathbf{e}^{(L)})$
$\mathbf{g} \leftarrow \textsc{Sigmoid}\big(\textsc{Linear}([\mathbf{h}_{pool}||\mathbf{e}_{pool}])\big)$
return $\mathbf{z}_{\mathcal{G}} = \mathbf{g} \odot \mathbf{h}_{pool}, \quad \mathbf{P}^{(L)}$

Main Procedure:
$\mathbf{z}_{\mathcal{G}_1}, \mathbf{P}_1^{(L)} \leftarrow$ DSP-Encoder($\mathcal{G}_1$)
$\mathbf{z}_{\mathcal{G}_2}, \mathbf{P}_2^{(L)} \leftarrow$ DSP-Encoder($\mathcal{G}_2$)
// Tri-Branch Interaction & Prediction
$\mathbf{h}_1 \leftarrow \phi_{post}(\mathbf{z}_{\mathcal{G}_1})$
$\mathbf{h}_2 \leftarrow \phi_{post}(\mathbf{z}_{\mathcal{G}_2})$
$\mathbf{s}_{ntn} \leftarrow \textsc{BilinearCorrelation}(\mathbf{h}_1, \mathbf{h}_2)$
$\mathbf{f}_{joint} \leftarrow [\mathbf{s}_{ntn}||\mathbf{h}_1||\mathbf{h}_2|||\mathbf{h}_1 - \mathbf{h}_2|||\mathbf{h}_1 \odot \mathbf{h}_2]$
$\hat{y} \leftarrow \textsc{Sigmoid}\big(\text{MLP}(\mathbf{f}_{joint})\big)$ *// Final Similarity Score*
// Optimization Objective (Briefly Noted)
$\mathcal{L}_{total} \leftarrow \text{MSE}(\hat{y}, y_{gt}) + \alpha \sum_{k=1}^{2} \mathcal{L}_{pos}^{(k)}(\mathbf{P}_k^{(L)}, \mathbf{U}_k)$
return $\hat{y}$

5 Experiments

In this section, we conduct extensive experiments on four standard real-world graph datasets to evaluate the effectiveness of SAC-Net.

5.1 Datasets

Table 1 STATISTICS OF AIDS700nef, LINUX, IMDBMulti, AND PTC.

	# $\|G\|$	Avg. $(\|V\|, \|E\|)$	# Labels	# Graph Pairs
AIDS700nef	700	(8.9, 6.9)	29	78400
LINUX	1000	(7.58, 6.94)	1	360000
IMDBMulti	1,500	(13.00, 65.91)	1	160000
PTC	344	(14.29, 14.69)	19	18975

For all datasets, the graph pairs are randomly partitioned into training, validation, and test sets with a ratio of 60%, 20%, and 20%, respectively. We utilize four benchmark datasets covering diverse domains and scales to ensure a comprehensive evaluation.

"AIDS700": A collection of 700 antiviral screen chemical compounds. It contains rich node and edge labels, serving as a standard testbed for structure-sensitive similarity.

"LINUX": Consists of 1,000 program dependency graphs generated from the Linux kernel. This dataset features unla-

beled graphs, challenging the model's ability to capture pure graph structures.

"IMDB-Multi": Contains 1,500 ego-networks derived from movie actor collaborations. These graphs are denser than chemical graphs and lack node features, relying on degree information for initialization.

"PTC": A toxicology dataset consisting of 344 chemical graphs. Labeled with 19 types of toxicity or chemical properties, PTC serves as a rigorous benchmark for processing complex molecular graphs with limited training samples.

5.2 Baselines and Experimental Setup

To comprehensively assess the performance of SAC-Net, we conduct a comparative study against nine established graph similarity computation methods. These baselines are broadly categorized into two paradigms: representation-based methods, which emphasize computational efficiency by mapping graphs into independent vector representations, and interaction-based methods, which emphasize predictive accuracy through fine-grained cross-graph matching. For a fair and objective comparison, all baseline models are implemented using the optimal hyperparameter configurations and architectures reported in their respective original papers. Furthermore, to ensure the stability and reproducibility of the results, all experiments for both SAC-Net and the baseline methods are executed 10 times independently, and the average values are reported as the final performance metrics.

Representation-based Methods: These methods explicitly focus on extracting global structural information by independently encoding each graph into a fixed-length vector and computing similarity using a metric function or a neural network. While they typically achieve highly efficient linear time complexity, by comparing our model against these baselines, our primary objective is to rigorously evaluate which framework can capture a more comprehensive and precise global structural topology under the same efficiency constraints.

- SimGNN [23]: Combines GCNs with a Neural Tensor Network and utilizes histogram-based pooling to capture the global distribution of node embeddings, providing an efficient coarse-grained structural comparison. Although categorized as an representation-based method, computing the cross-graph pairwise node similarity matrix for the histogram results in a quadratic $O(N^2)$ complexity.
- EGSC [7]: Adopts a Knowledge Distillation framework where a computation-heavy interaction model serves as the "Teacher" to guide a lightweight embedding "Student", balancing fast inference with matching precision.
- ERIC [25]: Introduces alignment regularization to enforce node-level correspondences during training, thereby avoiding expensive node-matching operations during the inference phase.
- GRASP [26]: Simplifies the architecture by removing complex interaction modules, focusing instead on enhancing representation via positional encodings and multi-scale fusion to achieve efficient performance.

Interaction-based Methods: These methods explicitly focus on capturing fine-grained structural alignments to effectively extract local structural information by performing intricate node-level or subgraph-level comparisons between graph pairs. While they typically achieve higher predictive accuracy at the cost of computationally expensive quadratic time complexity, by comparing our model against these baselines, our primary objective is to verify whether our framework can match or exceed this discriminative power for local structural information without relying on their computational overhead.

- GraphSim [27]: Transforms the node-node similarity matrix into a 2D image and applies CNNs to capture local matching patterns, effectively treating graph matching as an image classification task.
- MGMN [8]: Proposes a Multi-level Graph Matching Network that aligns interactions at both node and graph levels simultaneously, allowing for granular similarity assessment across multiple scales.
- H2MN [28]: Introduces hypergraph learning to capture higher-order structural dependencies, determining similarity by matching complex motifs that standard pairwise methods might miss.
- NA-GSL [29]: Integrates a Neighbor-Aware attention mechanism to weigh the importance of different node correspondences, ensuring the score is derived from the most structurally relevant pairs.
- CLSim [30]: Bridges the gap between node-level and graph-level comparisons by explicitly modeling cross-hierarchy interactions, ensuring that local structural nuances effectively influence the global prediction.

Evaluation Metrics. We treat similarity computation as a regression task and use the following metrics:

MSE (10^{-3}): Mean Squared Error measures the divergence between predicted and ground-truth similarity scores (lower is better).

ρ (Spearman's Rank Correlation [48] and τ (Kendall's Rank Correlation) [49]: Assess the consistency of the predicted ranking (higher is better).

p@10 and p@20 (Precision at k): Measure the intersection ratio between the ground-truth top-k similar graphs and the

Table 2 Comparison results on AIDS700nef, LINUX, IMDBMulti, and PTC datasets. The best results are indicated in **bold**, and the second-best results are underlined. OOM mean out of GPU memory.

Method	AIDS700nef					LINUX				
	mse(10^{-3})	ρ	τ	p@10	p@20	mse(10^{-3})	ρ	τ	p@10	p@20
SimGNN	2.171	0.862	0.690	0.455	0.532	0.430	0.981	0.886	0.973	0.950
GraphSim	1.235	0.371	0.467	0.817	0.656	1.207	0.936	0.825	0.856	0.847
MGMN	2.439	0.901	0.744	0.474	0.543	1.807	0.969	0.870	0.952	0.920
H2MN	1.337	0.827	0.669	0.416	0.494	0.478	0.976	0.896	0.948	0.923
EGSC	1.584	0.897	0.734	0.644	0.722	0.310	0.985	0.901	0.978	0.978
ERIC	2.94	0.890	0.726	0.644	0.719	0.450	0.987	0.906	0.980	**0.987**
NA-GSL	3.058	0.842	0.687	0.394	0.500	0.273	0.99	0.955	**0.988**	0.976
Grasp	1.121	0.920	0.789	0.660	0.714	2.816	0.921	0.776	0.402	0.510
CLSim	1.999	0.875	0.705	0.522	0.590	0.159	0.985	0.898	0.981	0.966
SAC-Net(ours)	**1.068**	**0.927**	**0.798**	**0.682**	**0.725**	**0.121**	**0.994**	**0.967**	0.987	0.981

Method	IMDBMulti					PTC				
	mse(10^{-3})	ρ	τ	p@10	p@20	mse(10^{-3})	ρ	τ	p@10	p@20
SimGNN	0.700	0.919	0.799	0.840	0.837	2.281	0.928	0.792	0.520	0.628
GraphSim	3.213	0.801	0.682	0.624	0.664	2.770	0.925	0.780	0.407	0.523
MGMN	13.75	0.788	0.596	0.472	0.496	2.215	0.931	0.772	0.435	0.542
H2MN	**0.312**	0.899	0.811	0.877	0.877	**1.228**	0.901	0.749	0.503	0.582
EGSC	0.640	0.933	0.824	0.850	0.871	2.384	0.923	0.776	0.548	0.657
ERIC	0.955	0.889	0.780	0.860	0.877	-	-	-	-	-
NA-GSL	0.887	0.861	0.788	0.819	0.831			**OOM**		
Grasp	0.761	0.912	0.831	0.841	0.854	1.850	0.933	0.798	0.565	0.669
CLSim	0.563	0.932	0.813	0.857	0.857	1.701	0.927	0.784	0.528	0.604
SAC-Net(ours)	0.493	**0.958**	**0.885**	**0.879**	**0.894**	1.688	**0.937**	**0.805**	**0.570**	**0.674**

predicted top-k results. These are critical metrics for retrieval tasks.

Implementation Details: We set the hidden dimension to $d = 64$ and the dimensionality of positional encodings to 16. The number of GNN layers K is configured as 4 for AIDS700nef and IMDBMulti, and 8 for LINUX and PTC. For model optimization, we employ the Adam optimizer with an initial learning rate of 10^{-3}, a weight decay of 5×10^{-4}, and a batch size of 256. The positional Loss Weight α, which controls the positional loss to prevent Structural Collapse of the Global Structural Skeleton, is set to 0.01. The maximum training epochs are set to 30,000 for AIDS700nef, LINUX, and IMDBMulti, and 5,000 for PTC, respectively. All experiments are conducted on an NVIDIA Tesla V100 GPU.

5.3 Comparative Performance Analysis

Table 2 demonstrates SAC-Net's superiority, driven by two core mechanisms. First, the DSP-GNN evolves positional coordinates dynamically to prevent Coordinate Washout, effectively overcoming the 1-WL limit to distinguish locally isomorphic structures. Second, the Edge-Aware Fusion gates node embeddings using dynamic edge representations, filtering structural noise while preserving the critical connectivity skeleton.

Although trailing H2MN in MSE on PTC and IMDB-Multi, SAC-Net avoids H2MN's computationally intensive quadratic hypergraph matching, achieving a vastly superior accuracy-efficiency balance with linear complexity. Similarly, on the LINUX dataset, SAC-Net ranks behind NA-GSL and ERIC in top-k metrics—which enforce rigid, expensive node-level alignments—but significantly outperforms them in overall MSE. This validates that our dynamic structural perception captures global topology accurately without the heavy overhead of cross-graph matching.

SAC-Net exhibits strong robustness across diverse structural patterns. On the unlabeled LINUX dataset, our Dynamic Positional Encoding acts as a Structural GPS to differentiate isomorphic topologies without semantic cues. Conversely, on dense (IMDBMulti) and symmetric bio-chemical graphs (AIDS700, PTC), the Edge-Aware Fusion leverages fine-grained semantic interactions to modulate node representations. Mirroring heterogeneous graph attention [50], this selective integration enables the model to identify subtle

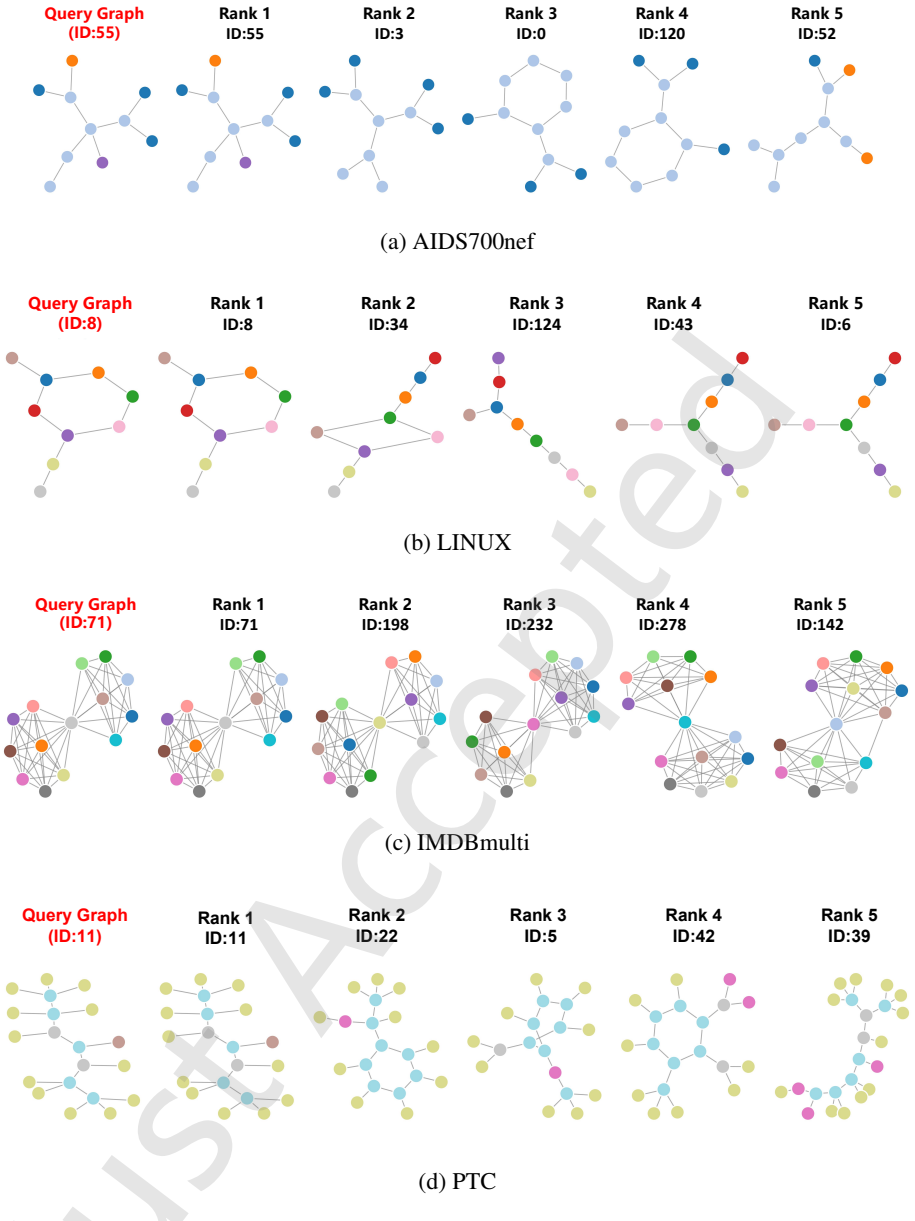


Fig. 4 Case study of graph similarity retrieval based on SAC-Net.The leftmost graph (red border) represents the input query graph, followed by the Top-K candidate graphs retrieved by the model from the dataset, ranked by their predicted similarity scores. Different node colors encode specific feature categories.

motifs like ring structures.

5.4 How Does SAC-Net surpass 1-WL Test

To empirically verify that SAC-Net possesses expressive power beyond the 1-WL test, we evaluated it on the Circular Skip Links (CSL) dataset [51]. The CSL dataset features highly symmetrical graphs, making it an ideal benchmark for testing a model's ability to distinguish isomorphic structures. We demonstrate the superiority of our DSP backbone through both quantitative metric analysis and qualitative clustering visualization. Consistent with our core experiments, all evaluations on the CSL dataset are conducted over 10 independent runs, and the reported accuracy and Macro F1 scores are the average results across these trials. This ensures that SAC-Net's perfect classification performance is robust and not a result of favorable random initialization.

MPNNs rely on recursive local neighborhood aggregation, which makes them theoretically bounded by the 1-WL limit and unable to distinguish the isomorphic graphs present in the CSL dataset.

As shown in Table 3, standard GNN models yield performance that is essentially equivalent to random guessing.

Table 3 Expressive Power Analysis on CSL Dataset.

Model	Accuracy (%)	Macro F1
GCN	8.00	0.02
GIN	14.67	0.03
GAT	10.00	0.10
RGGC	15.33	0.15
DSP-GNN	**100.00**	**1.00**

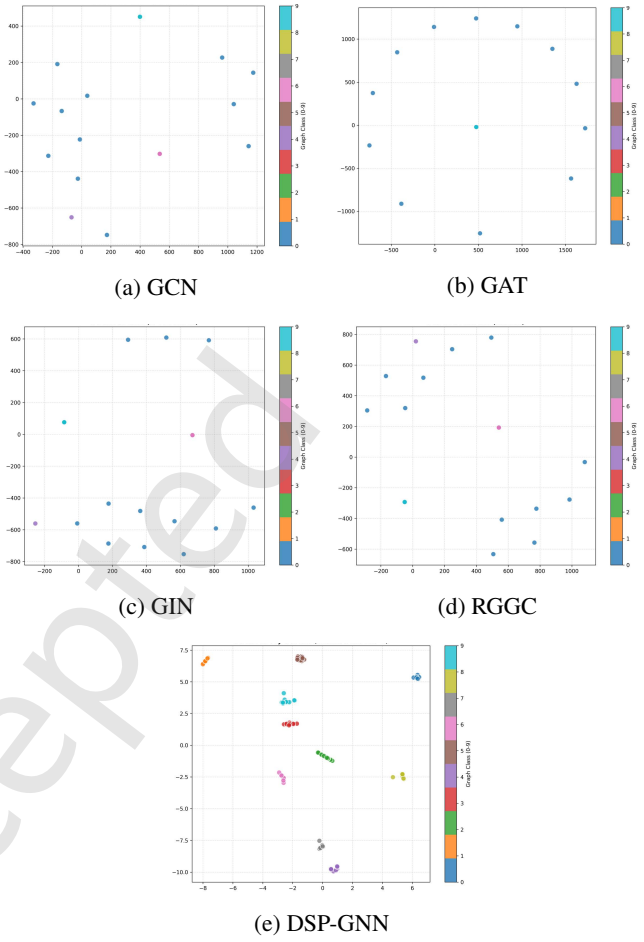
Specifically, GCN and GAT achieve accuracies of 8.00% and 10.00%, respectively. Even slightly more advanced or expressive local aggregation methods struggle, with GIN achieving an accuracy of 14.67% (Macro F1 of 0.03) and RGGC achieving 15.33% (Macro F1 of 0.15).In stark contrast, our SAC-Net framework achieves perfect classification, reaching an accuracy of 100% and a Macro F1 score of 1.00. This significant performance gap empirically demonstrates that the incorporation of DSP allows our model to break the symmetry limitations of standard MPNNs and effectively capture latent structural information.

To provide further interpretability into how our model represents these complex spatial configurations, we visualize the learned graph embeddings using t-SNE. As illustrated in Fig. 5, because traditional GNNs are "blind" to the Global Structural Skeleton, they project graphs with distinct spatial skeletons but identical local neighborhoods into overlapping, indistinguishable regions in the latent space. However, our proposed DSP-GNN successfully forms distinct, well-separated clusters for structurally diverse graphs. This clear spatial separation qualitatively confirms that establishing a joint evolution paradigm for positional coordinates and node attributes acts as an effective Structural GPS, allowing the model to accurately differentiate complex topologies that bypass standard local aggregation.

5.5 Theoretical Analysis on Expressive Power

To formally demonstrate why the proposed DSP-GNN backbone strictly exceeds the expressive power of the 1-Weisfeiler-Lehman (1-WL) test, we provide a theoretical analysis focused on Circular Skip Link (CSL) graphs.

Lemma 1. MPNNs are bounded by the 1-WL test and fail to distinguish non-isomorphic CSL graphs.Proof Sketch: CSL graphs are r-regular graphs, meaning every node possesses the exact same local degree structure. As theoretically established by Xu et al. [9], standard MPNNs are at most as powerful as the 1-WL test. The 1-WL test initializes all nodes with identical features and iteratively updates them based on neighborhood multisets. Because the local topologies in regular graphs are entirely symmetric, the neighborhood multisets remain



(a) GCN (b) GAT (c) GIN (d) RGGC (e) DSP-GNN

Fig. 5 The t-SNE visualization of different models. The five sub-images are compactly arranged within the column.

indistinguishable across iterations. Consequently, standard MPNNs assign identical representations to non-isomorphic CSL graphs, inherently failing this structural task [51].

Lemma 2. Non-isomorphic CSL graphs possess distinct structural graph Laplacian spectra.Proof Sketch: Let $L = D - A$ denote the unnormalized graph Laplacian. The structural topology of a graph is intrinsically encoded in its eigenvalues and eigenvectors. As highlighted by Dwivedi et al. [52], Laplacian eigenvectors can serve as powerful positional encodings to break initial symmetries in highly regular structures. Mathematically, non-isomorphic CSL graphs, despite being regular, exhibit distinct Laplacian eigenspaces.

Theorem 1. The proposed DSP-GNN exceeds the expressive power of the 1-WL test.Proof: The strict superiority of DSP-GNN over 1-WL lies in its structural initialization and dynamic coordinate evolution. Before message passing, DSP-GNN initializes the positional coordinate $p_i^{(0)}$ using the top-k Laplacian eigenvectors. By Lemma 2, for two non-isomorphic CSL graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, their initial positional sets $\{p_i^{(0)}\}_{i\in\mathcal{G}_1}$ and $\{p_j^{(0)}\}_{j\in\mathcal{G}_2}$ are strictly distinct, effectively breaking the

Table 4 Ablation Study Comparison. The best results are indicated in **bold**, and the second-best results are underlined.

Model	AIDS700nef					LINUX				
	mse(10^{-3})	ρ	τ	p@10	p@20	mse(10^{-3})	ρ	τ	p@10	p@20
(GCN)	1.281	0.912	0.776	0.608	0.669	0.130	0.992	0.964	0.987	0.977
(GIN)	1.200	0.920	0.788	0.666	0.706	0.144	0.990	0.964	0.979	0.980
(RGGC)	1.109	0.924	0.794	0.671	0.717	0.114	0.993	0.964	0.985	0.978
(w/o DSP)	1.126	0.926	0.797	0.664	0.718	0.205	0.992	0.966	0.981	0.980
(w/o EAF)	1.083	0.925	0.796	0.677	0.720	0.189	0.993	0.964	0.982	0.980
w/o Mean(use Sum)	1.095	0.926	0.796	0.677	0.720	0.113	0.991	0.963	0.985	0.976
w/o Mean(use Attn)	1.110	0.925	0.795	0.680	0.724	0.136	0.992	0.963	0.986	0.980
(w/o TBIP)	1.306	0.912	0.775	0.629	0.690	0.167	0.992	0.956	0.973	0.967
SAC-Net)	**1.068**	**0.927**	**0.798**	**0.682**	**0.725**	**0.111**	**0.994**	**0.967**	**0.988**	**0.982**
Model	**IMDBMulti**					**PTC**				
	mse(10^{-3})	ρ	τ	p@10	p@20	mse(10^{-3})	ρ	τ	p@10	p@20
(GCN)	0.497	0.953	0.880	**0.883**	0.882	1.769	0.927	0.791	0.563	0.664
(GIN)	0.500	0.955	0.878	0.876	0.892	1.770	0.934	0.801	**0.572**	0.657
(RGGC)	0.524	0.957	0.881	0.867	0.891	1.719	0.933	0.800	0.560	0.668
(w/o DSP)	0.496	0.954	0.882	0.890	0.892	1.694	0.931	0.795	0.552	0.668
(w/o EAF)	0.502	0.954	0.879	0.878	**0.893**	1.729	0.933	0.799	0.556	0.670
w/o Mean(use Sum)	**0.487**	0.952	0.875	0.869	0.890	1.738	0.935	0.802	0.563	0.665
w/o Mean(use Attn)	0.501	0.951	0.873	0.873	0.885	1.804	0.935	0.802	0.557	0.664
(w/o TBIP)	0.613	0.948	0.869	0.872	0.883	2.135	0.923	0.782	0.536	0.651
SAC-Net	0.493	**0.958**	**0.885**	0.879	**0.893**	**1.688**	**0.937**	**0.805**	0.570	**0.674**

initial symmetry that blinds standard MPNNs.During the dynamic evolution phase, DSP-GNN updates these coordinates via a structurally-gated message passing mechanism defined as:

$$p_i^{(\ell+1)} = p_i^{(\ell)} + \mathrm{Tanh}\Big(\mathrm{BN}\big(\mathbf{C}_1(p_i^{(\ell)}) + \sum_{j\in\mathcal{N}(i)} \sigma_{ij}^{(\ell)} \odot \big(\mathbf{W}_{C2} p_j^{(\ell)}\big)\big)\Big). \tag{25}$$

Since the initial states $p^{(0)}$ are already discriminative, and the multi-layer perception matrices with the gating mechanism ($\sigma_{ij}^{(\ell)}$) operate as continuous injective functions over multisets [9], the structural distinctiveness is mathematically preserved across layers. Therefore, DSP-GNN generates distinct final embeddings $p^{(K)}$ for $\mathcal{G}_1$ and $\mathcal{G}_2$, successfully distinguishing them where 1-WL inevitably fails. Thus, DSP-GNN $>$ 1-WL.

5.6 Ablation Study

To evaluate the contribution of each component, we perform ablation studies across four datasets, as reported in Table 4.

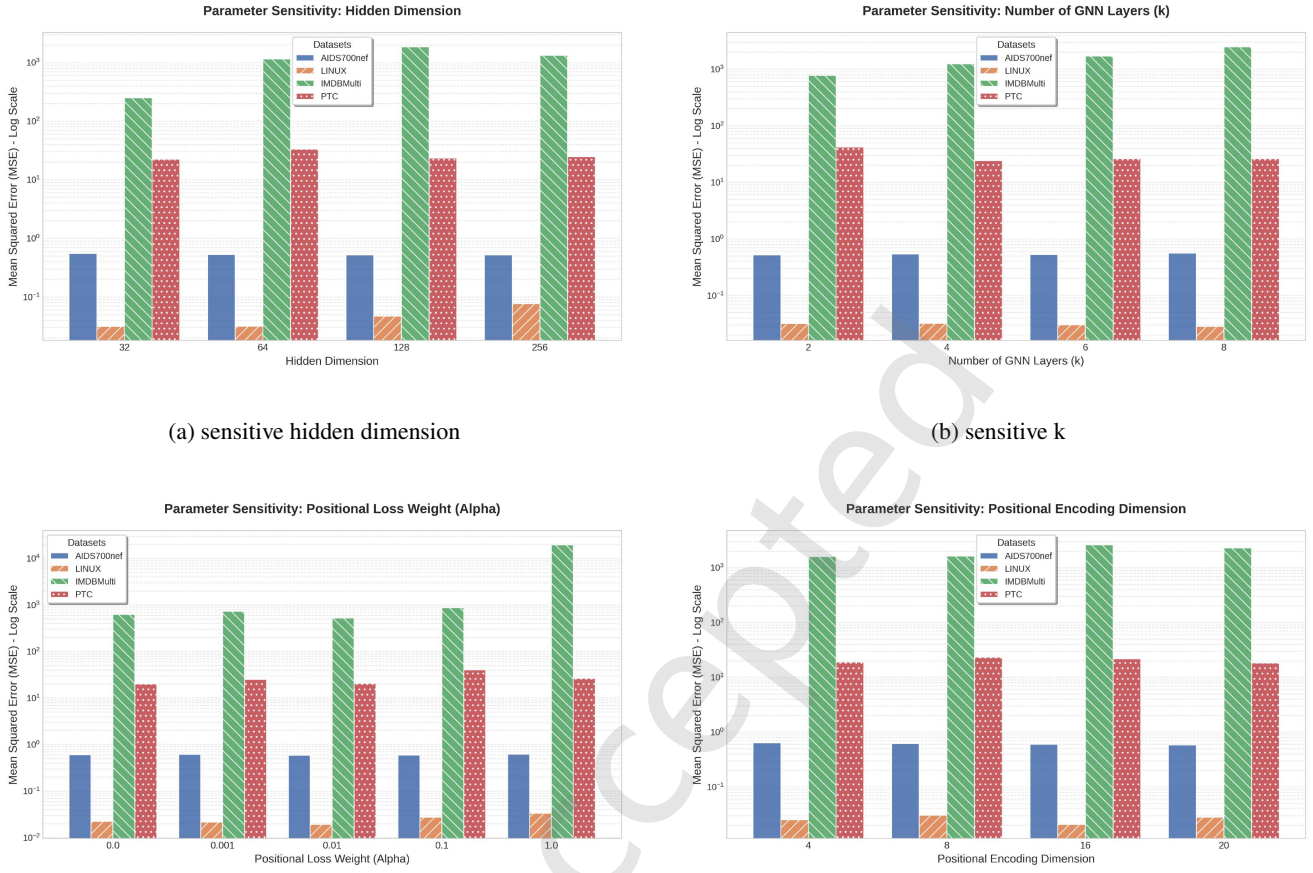
Backbone Variants (GCN, GIN, and RGGC): Replacing our DSP-GNN backbone with standard message-passing architectures (GCN, GIN, and RGGC). As shown in Table 4, these variants exhibit consistent performance degradation across the datasets, particularly on AIDS700nef and PTC. This demonstrates that relying solely on traditional local neighborhood aggregation or basic relational convolutions is insufficient, highlighting the superiority of our framework in jointly capturing the overall topological configurations and fine-grained semantic interactions.

w/o DSP: Replacing the DSP-GNN with a standard GatedGCN [53]. A consistent performance drop is observed across the datasets, confirming that the dynamic evolution of positional features is beneficial for preventing Coordinate Washout in deep layers, thereby more effectively capturing long-range dependencies and overall topological configurations.

w/o EAF: Removing the edge pooling stream and the Edge-Aware Gating mechanism. The slight but consistent degradation in performance supports our hypothesis that transforming fine-grained semantic interactions between nodes into an explicit structural filtering signal via this gating mechanism helps prevent Structural Collapse during graph pooling.

w/o TBIP: Using a simple MLP concatenation instead of the Tri-Branch module. The relatively more noticeable performance drops indicate that bilinear correlation, global structural deviations, and local pattern alignment are complementary and essential for robust fine-grained comparison.

w/o Mean(use Sum and Attn): Replacing mean pooling with sum and attention pooling. Both alternatives generally



(a) sensitive hidden dimension

(b) sensitive k

(c) sensitive Positional Loss Weight

(d) sensitive Positional Encoding Dimension

Fig. 6 Sensitivity analysis of hyperparameters.

lead to less stable or slightly inferior performance across most datasets, indicating that mean pooling offers a more robust and scale-invariant graph-level aggregation in our framework.

5.7 Complexity Analysis

To establish a clear comparative baseline, we first analyze the theoretical complexities of existing paradigms. Representation-based methods encode each graph independently, typically achieving an efficient linear time complexity of $O(|V| + |E|)$ and a space complexity of $O(|V| \cdot d)$, where d is the hidden feature dimension. A notable exception within this category is SimGNN, which incurs a quadratic $O(|V|_1 \cdot |V|_2)$ time and space overhead due to its node-level pairwise similarity histogram computation. In stark contrast, interaction-based methods rely on exhaustive cross-graph node matching. This alignment fundamentally dictates both a computationally prohibitive quadratic time complexity of $O(|V|_1 \cdot |V|_2)$ and a heavy $O(N^2)$ space complexity required to store pairwise similarity matrices or attention maps.

The computational complexity of our proposed SAC-Net framework is primarily governed by three components: structural initialization, the DSP-GNN backbone, and the similarity prediction head. During the initialization phase, extracting the top-k Laplacian eigenvectors using a sparse solver incurs a temporal cost of approximately $O(k \cdot |E|)$, while the L-layer message passing within the backbone network maintains a time complexity of $O(L \cdot (|V| + |E|))$. Given that the number of layers L, the feature dimension d, and the number of eigenvectors k are fixed constants, the overall time complexity of SAC-Net scales strictly linearly with the graph size, denoted as $O(|V| + |E|)$. Similarly, the space complexity of SAC-Net is efficiently bounded by $O(|V| \cdot d + |E| \cdot d)$ for storing node and edge embeddings, ensuring a strictly linear space footprint.

5.8 Visualization and Case Study

To intuitively evaluate the retrieval performance of SAC-Net in real-world scenarios, we conduct a visualization case

study across four datasets. In these datasets, accurate graph similarity search typically plays an essential role in identifying analogous structural motifs.

Fig. 4 illustrates the graph ranking results under the GED metric. For each of the four datasets, the leftmost column presents a specific Query Graph, while the subsequent columns (Rank 1 to Rank 5) display the top-5 most similar graphs retrieved from the dataset based on the GED scores predicted by SAC-Net.SAC-Net demonstrates exceptional precision. For all queries, the Rank 1 result successfully retrieves the query graph itself, successfully recognizing structurally identical graphs. As the rank progresses from 2 to 5, the retrieved graphs exhibit highly analogous topological configurations and semantic features, with only subtle, incremental deviations—such as a small number of edge insertions/deletions or node label substitutions.

5.9 Parameter Sensitivity Analysis

To empirically validate our theoretical claims and evaluate model robustness, we conduct a sensitivity analysis on four hyperparameters. This design strictly echoes our earlier statements: varying the number of GNN layers (k) and positional encoding dimension verifies that the DSP-GNN resists Coordinate Washout and breaks the 1-WL limit even in deep or compact configurations. Concurrently, analyzing the hidden dimension and loss weight confirms that our Edge-Aware Fusion and auxiliary regularization effectively filter structural noise and prevent Coordinate Collapse without degrading task accuracy. Performance is measured via Mean Squared Error (MSE) on a logarithmic scale.

Concurrently, the analysis of the hidden dimension and positional loss weight in Fig. 6 (a) and (c) confirms the efficacy of our noise filtering and regularization mechanisms. Although demonstrating robust stability across the other three datasets, denser and attribute-less datasets like IMDBMulti exhibit slight performance fluctuations under varying parameter ranges. We attribute this minor volatility to highly dense hub structures and a complex loss landscape, both of which easily amplify local topological noise.Regarding the positional loss weight, performance drops at $\alpha = 0.0$ due to the occurrence of Coordinate Collapse. Conversely, when the weight is excessively high ($\alpha = 1.0$), performance also degrades significantly. We attribute this degradation to the fact that an overpowering positional loss acts as a rigid constraint, forcing the evolving positional features to remain overly close to the initial Laplacian eigenvectors. This rigid behavior severely stifles the network's capacity and contradicts the need for task-specific structural adaptation during similarity regression.Nevertheless, by employing a moderate hidden dimension and an optimally calibrated auxiliary loss at $\alpha = 0.01$, SAC-Net successfully restrains extreme error spikes. This precise calibration avoids both the Coordinate Collapse and the rigid suppression of the primary similarity regression objective. These findings collectively demonstrate SAC-Net's strong structural robustness and its optimal balance between expressive capacity and generalization.

6 Conclusions

In this paper, we proposed SAC-Net, an efficient framework that addresses the 1-WL limitation and the neglect of edge features by unifying global positional context with local edge affinity. By leveraging a SAC-Net backbone for dynamic feature evolution and an Edge-Aware Fusion mechanism for structural denoising, our model effectively overcomes the bottlenecks of static injection. Coupled with a Tri-Branch Interaction module and an auxiliary positional loss, experimental results demonstrate that our method yields significant performance advantages across multiple datasets. It significantly improves discriminative power regarding structural isomorphism while maintaining linear time complexity.

Acknowledgment

The work was supported partly by the Guangxi Natural Science Foundation under Grant Nos. 2024JJB180051, 2025GXNSFFA069015, 2026GXNSFDA00640006; the National Natural Science Foundation of China under Grant Nos. 62372119, 62166003; the Guangxi Bagui Young Talent Training Program; the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education (Guangxi Normal University, Guilin 541004, China) under Grant No. EBME24-03; the Innovation Project of Guangxi Graduate Education under Grant Nos. JGY2024080, JGY2022046; and the Guangxi Collaborative Innovation Center of Multi-Source Information Integration, China.

[1] Neuhaus M, Riesen K, Bunke H. Fast suboptimal algorithms for the computation of graph edit distance[C]//Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 163-172.

[2] Bunke H, Shearer K. A graph distance metric based on the maximal common subgraph[J]. Pattern recognition letters, 1998, 19(3-4): 255-259.

[3] Neuhaus M, Riesen K, Bunke H. Fast suboptimal algorithms for the computation of graph edit distance[C]//Joint IAPR International Workshops on Statistical Techniques in Pattern

Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 163-172.

[4] Kuhn H W. The Hungarian method for the assignment problem[J]. Naval research logistics quarterly, 1955, 2(1-2): 83-97.

[5] Fankhauser S, Riesen K, Bunke H. Speeding up graph edit distance computation through fast bipartite matching[C]//International Workshop on Graph-Based Representations in Pattern Recognition. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011: 102-111.

[6] Zeng Z, Tung A K H, Wang J, et al. Comparing stars: On approximating graph edit distance[J]. Proceedings of the VLDB Endowment, 2009, 2(1): 25-36.

[7] Qin C, Zhao H, Wang L, et al. Slow learning and fast inference: Efficient graph similarity computation via knowledge distillation[J]. Advances in Neural Information Processing Systems, 2021, 34: 14110-14121.

[8] Ling X, Wu L, Wang S, et al. Multilevel graph matching networks for deep graph similarity learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 34(2): 799-813.

[9] Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks?[J]. arXiv preprint arXiv:1810.00826, 2018.

[10] Verma S, Zhang Z L. Stability and generalization of graph convolutional neural networks[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 1539-1548.

[11] Tang H, Liu Y. Towards understanding generalization of graph neural networks[C]//International Conference on Machine Learning. PMLR, 2023: 33674-33719.

[12] Chen Z, Chen L, Villar S, et al. Can graph neural networks count substructures?[J]. Advances in neural information processing systems, 2020, 33: 10383-10395.

[13] Zhao L, Jin W, Akoglu L, et al. From stars to subgraphs: Uplifting any GNN with local structure awareness[J]. arXiv preprint arXiv:2110.03753, 2021.

[14] Pei H, Wei B, Chang K C C, et al. Geom-gcn: Geometric graph convolutional networks[J]. arXiv preprint arXiv:2002.05287, 2020.

[15] Morris C, Ritzert M, Fey M, et al. Weisfeiler and leman go neural: Higher-order graph neural networks[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 4602-4609.

[16] Maron H, Ben-Hamu H, Serviansky H, et al. Provably powerful graph networks[J]. Advances in neural information processing systems, 2019, 32.

[17] Gong L, Cheng Q. Exploiting edge features for graph neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 9211-9219.

[18] Jo J, Baek J, Lee S, et al. Edge representation learning with hypergraphs[J]. Advances in neural information processing systems, 2021, 34: 7534-7546.

[19] Hussain M S, Zaki M J, Subramanian D. Global self-attention as a replacement for graph convolution[C]//Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. 2022: 655-665.

[20] Li P, Wang Y, Wang H, et al. Distance encoding: Design provably more powerful neural networks for graph representation learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 4465-4478.

[21] Alon U, Yahav E. On the bottleneck of graph neural networks and its practical implications[J]. arXiv preprint arXiv:2006.05205, 2020.

[22] Yu H, Yuan J, Yao Y, et al. Not all edges are peers: Accurate structure-aware graph pooling networks[J]. Neural Networks, 2022, 156: 58-66.

[23] Bai Y, Ding H, Bian S, et al. Simgnn: A neural network approach to fast graph similarity computation[C]//Proceedings of the twelfth ACM international conference on web search and data mining. 2019: 384-392.

[24] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion[J]. Advances in neural information processing systems, 2013, 26.

[25] Zhuo W, Tan G. Efficient graph similarity computation with alignment regularization[J]. Advances in Neural Information Processing Systems, 2022, 35: 30181-30193.

[26] Zheng H, Shi J, Yang R. Grasp: Simple yet effective graph similarity predictions[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2025, 39(21): 22884-22892.

[27] Bai Y, Ding H, Gu K, et al. Learning-based efficient graph similarity computation via multi-scale convolutional set matching[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(04): 3219-3226.

[28] Zhang Z, Bu J, Ester M, et al. H2mn: Graph similarity learning with hierarchical hypergraph matching networks[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 2274-2284.

[29] Tan W, Gao X, Li Y, et al. Exploring attention mechanism for graph similarity learning[J]. Knowledge-Based Systems, 2023, 276: 110739.

[30] Zou C, Lu G, Du L, et al. Graph similarity learning for cross-level interactions[J]. Information Processing & Management, 2025, 62(1): 103932.

[31] Gilmer J, Schoenholz S S, Riley P F, et al. Neural message passing for quantum chemistry[C]//International conference on machine learning. Pmlr, 2017: 1263-1272.

[32] You J, Ying Z, Leskovec J. Design space for graph neural networks[J]. Advances in Neural Information Processing Systems, 2020, 33: 17009-17021.

[33] Dwivedi V P, Luu A T, Laurent T, et al. Graph neural networks with learnable structural and positional representations[J]. arXiv preprint arXiv:2110.07875, 2021.

[34] Rampášek L, Galkin M, Dwivedi V P, et al. Recipe for a general, powerful, scalable graph transformer[J]. Advances in Neural Information Processing Systems, 2022, 35: 14501-14515.

[35] Rampášek L, Galkin M, Dwivedi V P, et al. Recipe for a general, powerful, scalable graph transformer[J]. Advances in Neural Information Processing Systems, 2022, 35: 14501-14515.
[36] Kreuzer D, Beaini D, Hamilton W, et al. Rethinking graph transformers with spectral attention[J]. Advances in neural information processing systems, 2021, 34: 21618-21629.
[37] Wang X, Ma Y, Niu Y, et al. Sub-second volumetric 3D printing by synthesis of holographic light fields[J]. Nature, 2026: 1-9.
[38] Guo Y, Zhu H, Cao J, et al. Learning heterogeneous biological interactions via meta-relation-guided dual-channel graph transformer for circular ribonucleic acid function prediction[J]. Knowledge-Based Systems, 2026: 115929.
[39] Zhao X, Guo Y, Wang B, et al. CKDTA: A chemical knowledge-enhanced framework for drug-target affinity prediction[J]. Journal of Computational Science, 2025: 102706.
[40] Chen Y, Tang X, Qi X, et al. Learning graph normalization for graph neural networks[J]. Neurocomputing, 2022, 493: 613-625.
[41] Wu Q, Cui X, Lu M, et al. A GNSS anti-spoofing technique based on the spatial distribution characteristic of the residual vectors[J]. Tsinghua Science and Technology, 2023, 29(2): 457-468.
[42] Qiu Z, Wang Z, Zheng B, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free[J]. arXiv preprint arXiv:2505.06708, 2025.
[43] Frazzetto P, Pasa L, Navarin N, et al. Beyond the Additive Nodes' Convolutions: a Study on High-Order Multiplicative Integration[C]//Proceedings of the 39th acm/sigapp symposium on applied computing. 2024: 474-481.
[44] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[45] Zou C, Lu G, Zhang W, et al. Enhanced Graph Similarity Learning via Adaptive Multi-scale Feature Fusion[C]//IJCAI. 2025: 7309-7317.
[46] Hua Q, Zhou J, Zhang F, et al. Attention-Enhanced and Knowledge-Fused Dual Item Representations Network for Recommendation[J]. Tsinghua Science and Technology, 2024, 30(2): 585-599.
[47] Du L, Huang Z, Li J, et al. A multi-view graph neural network with subgraph variational autoencoder for class-Imbalanced node classification[J]. Knowledge-Based Systems, 2025: 115081.
[48] Spearman C. The proof and measurement of association between two things[J]. 1961.
[49] Kendall M G. A new measure of rank correlation[J]. Biometrika, 1938, 30(1-2): 81-93.
[50] Guo Y, Qiao L, Yang Z, et al. Fake News Detection: Extendable to Global Heterogeneous Graph Attention Network with External Knowledge[J]. Tsinghua Science and Technology, 2024, 30(3): 1125-1138.
[51] Murphy R, Srinivasan B, Rao V, et al. Relational pooling for graph representations[C]//International Conference on Machine Learning. PMLR, 2019: 4663-4673.
[52] Dwivedi V P, Joshi C K, Luu A T, et al. Benchmarking graph neural networks[J]. Journal of Machine Learning Research, 2023, 24(43): 1-48.
[53] Bresson X, Laurent T. Residual gated graph convnets[J]. arXiv preprint arXiv:1711.07553, 2017.

Author biography

LingHang Zeng is pursuing the Master's degree in Software Engineering at the College of Computer Engineering, Guangxi Normal University, Guilin, China. His research interests include graph representation learning and graph neural networks.

Yanling Li is a Lecturer at the School of Computer Science and Engineering, Guangxi Normal University. Her research interests focus on applied artificial intelligence, particularly natural language processing, automated negotiation, human–computer dialogue systems, and game theory. She is committed to advancing intelligent decision-making and interactive systems and actively participates in related teaching and research.

MingXia Bi is pursuing her Doctor of Engineering in Computer Engineering at Guangxi Normal University. Meanwhile, she serves as the Lead Expert of the Qingdao Master Teacher Studio in Information Technology. Her main research interests include talent training and curriculum design for computer majors, with a particular focus on the applications of artificial intelligence and pattern recognition.

Shilong Lin received the B.E. degree in software engineering from the Institute of Technology, East China JiaoTong University, NanChang, China, in 2023. He is currently pursuing the master's degree with the Guangxi Normal University, Guilin, China. His research interests include deep learning and graph representation learning.

Zhijie Luo was born in Nanchang, Jiangxi Province, China, in 2002. He received his B.Eng. degree from the Science and Technology College of Nanchang Hangkong University (China) in 2024. He is now a master student majoring in Computer Science and Technology at the School of Computer Science and Engineering, Guangxi Normal University (China), with research focus on composed image retrieval.

Han Wang is currently pursuing the master's degree in Software Engineering with the School of Computer Science, Guangxi Normal University, Guilin, China,in 2026. His research direction focuses on graph representation learning and graph contrastive learning, which aims to improve the representation quality and generalization ability of graph-structured data by using self-supervised learning and contrastive paradigms, and explore its applications in node classification, graph classification, recommendation systems and other scenarios.