Uncertainty-Aware Fuzzy Graph Contrastive Learning

Xinlan Xu[1] **Fei Hao**[1]* **Ce Yang**[1] **Jiaxing Shang**[2]
Jinsong Chen[3] **Jianrui Chen**[1] **Hongying Zhang**[4] **Aziz Nasridinov**[5]
[1]Shaanxi Normal University, Xi'an, China
[2]Chongqing University, Chongqing, China
[3]Central China Normal University, Wuhan, China
[4]Xi'an Jiaotong University, Xi'an, China
[5]Chungbuk National University, Cheongju, Korea

Abstract

Graph Contrastive Learning (GCL) is an effective approach for learning node representations. However, the intrinsic uncertainty of real-world graph data introduces significant challenges, as GCL relies on crisp deterministic representations that struggle to capture such ambiguity. To model graph uncertainty, fuzzy logic has attracted increasing attention due to its mathematical rigor. Existing methods, however, simply inject fuzzy modeling into the contrastive framework without tailoring data augmentation to the needs of fuzzy representations, resulting in limited discriminability and suboptimal downstream performance. To address these limitations, we propose a novel Fuzzy Graph Contrastive Learning (FGCL) framework for uncertainty-aware node representation learning. It adopts a vertex entanglement-based augmentation strategy to generate tailored multi-views with controlled perturbations while preserving core graph topology and semantics, mitigating semantic corruption. We also design a Deep Weighted Fuzzy Graph Convolutional Neural Network (DWFGCNN) encoder, which maps crisp features to fuzzy representations via learnable membership functions, explicitly models uncertainty, and resolves the core contradiction that traditional encoders cannot balance discriminability and robustness simultaneously. Extensive experiments on multiple datasets demonstrate that FGCL consistently outperforms baselines in node classification, community detection, and link prediction tasks, validating its effectiveness in handling uncertainty in graph representation learning. Our code is available at: `https://github.com/xxl-1237/FGCL.git`

1 Introduction

In recent years, Graph Contrastive Learning (GCL) has emerged as a research hotspot in the field of graph representation learning due to its outstanding performance in unsupervised scenarios [1, 2]. By constructing positive and negative sample pairs and maximizing the consistency of representations between similar samples, this approach learns robust node or graph embeddings from both graph topology and node attributes [3, 4]. It effectively avoids challenges such as high annotation costs and insufficient generalization ability in supervised learning, demonstrating outstanding performance in many fields including social network analysis [5, 6], molecular structure prediction [7, 8], and recommendation systems [9, 10, 11].

However, most existing GCL methods generate deterministic crisp representations, which often fail to adequately capture the uncertainties prevalent in graph data [12]. For example, in social networks, the intensity of interactions between users may range from intimate to distant, making it difficult

*Corresponding author: Fei Hao ⟨fhao@snnu.edu.cn ⟩

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

to characterize with simplistic 0-1 edge weights. In Biological molecular networks, the functional attributes of protein nodes frequently exhibit fuzzy distributions due to experimental errors, where single deterministic features cannot fully reflect their underlying semantics. Such fuzziness hidden in node connections and attributes constrain the models' ability to perform deep semantic exploration of complex graph structures, thereby reducing the expressiveness and robustness of the learned representations. Although fuzzy Graph Neural Networks (GNNs) have been developed to handle such uncertainties by incorporating fuzzy logic into graph modeling, they primarily operate in supervised or semi-supervised settings [13]. These methods often rely heavily on label information, which limits their applicability in real-world scenarios where annotated data is scarce or expensive to obtain. On the other hand, existing contrastive learning methods, while effective in unsupervised representation learning, typically generate deterministic embeddings that cannot adequately capture the intrinsic uncertainties and fuzzy relationships inherent in graph data [14]. This critical gap underscores the necessity for a unified framework that effectively combines the advantages of both fuzzy logic and GCL. Despite this recognized need, current researches to integrate fuzzy logic with GCL remain limited. Moreover, existing hybrid methods have failed to fully leverage data augmentation strategies from GCL to enhance the quality of fuzzy representations, resulting in limited expressive capability and consequently impacts model performance.

To address the aforementioned challenges, this paper proposes a novel FGCL framework to generate highly expressive fuzzy node representations. The FGCL mainly comprises three core modules: data augmentation module, the Deep Weighted Fuzzy Graph Convolutional Neural Network (DWFGCNN) module, and contrastive learning module. Among them, the data augmentation module adopts a vertex entanglement (VE)-based strategy. VE is a metric measuring node importance by characterizing a node's global topological influence via quantifying single-node perturbation on graph spectral entropy. Guided by VE, the module selectively modify graph structures and node attributes for more discriminative augmented views. The DWFGCNN module, as the core of feature extraction, is composed of multiple stacked WFGCNN layers. Each WFGCNN layer first performs a weighted aggregation of the topological and attribute information of nodes and their neighbors through graph convolution operations, yielding crisp feature representations. These crisp features are then mapped to the fuzzy feature space by fuzzy logic operators to effectively capture the uncertainties in the data. Finally, the final fuzzy node representation is generated by selectively fusing the transformed fuzzy information with the original crisp features. The contrastive learning module drives the optimization process of the entire model by calculating the contrastive loss of fuzzy representations between augmented views, which enhances the model's ability to learn features. The main contributions of this work are summarized as follows:

1. We propose a VE-based data augmentation strategy that quantifies node importance from a global perspective to selectively remove unimportant graph structures and node attributes, thereby generating meaningful augmented views without disrupting the original graph semantics. This method overcomes the limitations of random perturbation methods used in existing approaches combining fuzzy logic with GCL.
2. We propose a novel WFGCNN module to model uncertain relationships in graph data. Its core innovation is a dual fuzzification design: at the structural level, it uses soft K-means clustering for global edge weighting to capture fuzzy associations between nodes. At the feature level, a fuzzy logic-driven aggregation mechanism maps neighborhood information into a fuzzy space for inference and fusion.
3. We conduct extensive experiments on multiple datasets across three downstream tasks: node classification, node clustering, and link prediction. The results demonstrate that FGCL achieves superior performance compared to baselines, validating its effectiveness in handling graph data uncertainties while maintaining strong representation learning capabilities.

2 Related work

In this section, we provide a brief discussion of research on GCL and Fuzzy Deep Learning (FDL).

Graph contrastive learning. GCL, as a self-supervised learning approach, learns node or graph representations by contrasting different augmented views of graphs [15, 16]. It has demonstrated considerable potential in representation learning for graph-structured data and has received extensive attention from researchers in recent years. More recently, GCL has replaced traditional random

perturbations with learnable adaptive augmentations. It generates graph views via adversarial mechanisms [17] or multi-head attention [18], and integrates GNNs and Transformers to capture both local and global graph information [19]. For negative sample generation, E2Neg [20] selects representative negative nodes via spectral clustering and incorporates graph reconstruction for GCL. Neighbor-aware contrastive losses following graph homophily are also introduced to improve sample selection. From the perspective of causal and invariant learning, Mo *et al.* [21] proposed GCIL that leverages a causal perspective to generate invariant representations by perturbing non-causal factors through spectral graph augmentation. In addition, to capture multi-scale and fine-grained information, researchers have proposed various approaches. For example, MSSGCL [22] leverages global and local subgraph views to capture fine-grained semantic information. Beyond general graph representation learning, dedicated recommendation-oriented GCL solutions have been proposed to tackle interaction noise and data sparsity. These methods construct reliable augmented views based on discrete vector quantization [23], and exploit user-item collaborative patterns with customized graph encoders. Despite the above progress, most GCL methods still assume that the data representation is clear and deterministic, which limits their effectiveness in scenarios with inherent uncertainties, such as noisy or incomplete graph data. Our approach aims to address this by integrating fuzzy logic to model uncertainty within the GCL framework.

Fuzzy deep learning. Fuzzy logic demonstrates notable advantages in handling uncertain information, but it faces limitations in processing high-dimensional data and complex patterns, restricting its application in modern artificial intelligence tasks. In contrast, deep neural networks (DNNs) excel at extracting features from complex data but suffer from limited interpretability in their decision-making processes. Consequently, FDL bridges this gap by integrating the interpretability of fuzzy logic with the representational power of DNNs, thereby achieving complementary strengths between the two paradigms. This integration has demonstrated significant success across diverse fields including natural language processing [24], image analysis [25, 26], and medical diagnosis [27, 28, 29]. For example, Xing *et al.* [30] proposed FGAT, which integrates fuzzy rough sets for dynamic negative sampling to select high-quality negative edges and enhances node feature aggregation through FGAT layers. Lin *et al.* [31] proposed FuzzyGCN, which treated model parameters as fuzzy numbers to address uncertainties in graphs such as attribute noise and topology corruption. Wei *et al.* [32] proposed the DFS to optimize neighbor sampling for GNN inductive learning, which dynamically models neighbor-target membership via a Gaussian fuzzy system to enhance post-sampling feature aggregation effectiveness. Although these studies demonstrate the effectiveness of FDL, the integration of fuzzy logic with GCL remains underexplored.

3 Problem formulation

Given a graph $G = \{V, E\}$, where $V = \{v_1, v_2, ..., v_N\}$ represents the node set, and $E \subseteq V \times V$ represents the edge set. The node features are represented by a matrix $\mathbf{X} = [x_1, x_2, \ldots, x_N]^\top \in \mathbb{R}^{N \times M}$, where each row $x_i \in \mathbb{R}^M$ corresponds to the feature vector of node v_i. The adjacency matrix is denoted as $\mathbf{A} \in \{0, 1\}^{N \times N}$, where the element $\mathbf{A}_{ij}$ indicates the adjacency status between nodes v_i and v_j. $\mathbf{A}_{ij} = 1$ indicates the presence of an edge $(v_i, v_j) \in E$, and $\mathbf{A}_{ij} = 0$ indicates its absence. Our objective is to train a fuzzy graph encoder f by minimizing the contrastive loss function $\mathcal{J}$. This encoder maps the input feature matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ and weighted adjacency matrix $\mathbf{A}_W \in \mathbb{R}^{N \times N}$ to a set of fuzzy node representations $\tilde{\mathbf{H}} \in \mathbb{R}^{N \times d}$, such that $\tilde{\mathbf{H}} = f(\mathbf{X}, \mathbf{A}_W)$. These learned fuzzy embeddings $\tilde{\mathbf{H}}$ can be used to accomplish various downstream tasks, including node classification, community detection, and link prediction.

4 Proposed method

We propose a novel FGCL, with its framework illustrated in Figure 1. The overall architecture consists of three core modules: data augmentation, the DWFGCNN model, and contrastive learning. Specifically, the method first performs data augmentation on the original graph to generate augmented views. These augmented views are then fed into the DWFGCNN model to produce fuzzy representations. Finally, the DWFGCNN model is optimized by minimizing the contrastive loss computed between the fuzzy representations. Figure 1 (a) and (b) illustrate the detailed structure of the data augmentation and the WFGCNN, respectively, where the DWFGCNN is constructed by stacking

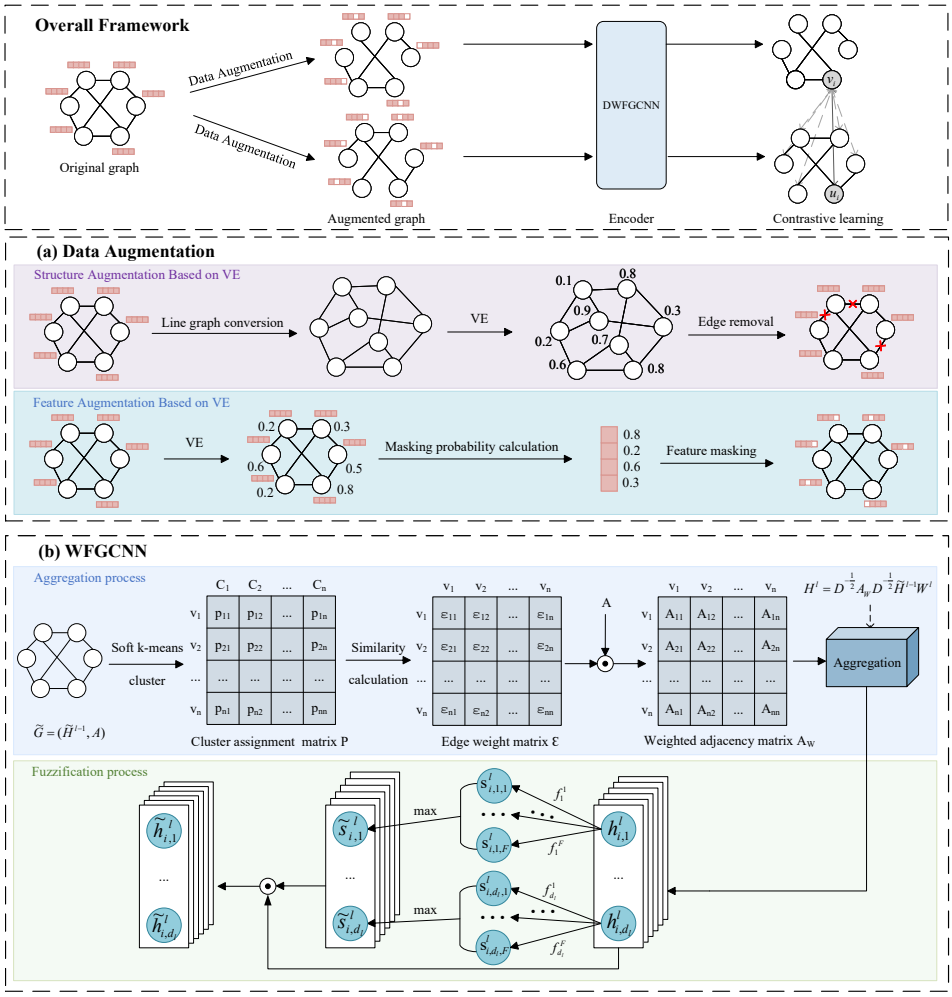


Figure 1: The framework of FGCL.

multiple layers of WFGCNN. In addition, we summarize FGCL with pseudo-code in Appendix A and conduct complexity analysis and comparison in Appendix B.

4.1 Data augmentation

The framework of data augmentation is shown in Figure 1 (a). The proposed approach consists of feature augmentation and structure augmentation, both based on VE.

4.1.1 Feature augmentation based on VE

As shown in Figure 1 (a), the process of the feature augmentation based on VE is as follows. First, the node importance is calculated by leveraging the VE mechanism. On this basis, the importance of each feature dimension is derived, and then the feature dimension importance is normalized to obtain the masking probability. Finally, feature masking is performed according to this probability.

Node importance calculation using VE. VE [33] provides an importance metric based on global topological properties, thereby distinguishing itself from traditional measures that rely on local information. We evaluate the effectiveness of VE in Appendix D.1. The definition of VE is as follows:

$$E(v_i) = \min_{\eta}(S_\eta(G_{v_i}) - S_\eta(G)), \tag{1}$$

where G_{v_i} is a perturbed network constructed by transforming the subgraph of node v_i and its neighbors into a probabilistic complete graph. G is the original network, $S_\eta(\cdot)$ represents the spectral entropy, and η is the diffusion time parameter. Since the value of $E(v_i)$ is negative, we define the importance of node v_i as follows:

$$w_f^i = |E(v_i)|. \tag{2}$$

Masking probability calculation. We define the importance s_j of each feature dimension j as:

$$s_j = \sum_{i=1}^{N} |X_{ij}| \cdot |w_f^i|, \tag{3}$$

where X_{ij} is the feature j of node v_i, and w_f^i is the importance of node v_i. Then, the masking probability for each feature is obtained by normalizing s as $\gamma_f = (s_{\max} - s)/(s_{\max} - s_{\min})$, where $s_{\max}$ represents the maximum value of s, and $s_{\min}$ represents the minimum value of s.

Feature masking. We employ γ_f to adaptively modify the feature perturbation distribution. Specifically, the feature masks m_f are independently sampled from two Bernoulli distributions as follows:

$$\mathbf{m}_f^{(k)} \sim \text{Bernoulli}(1 - \gamma_f p_f^{(k)}), \quad k \in \{1, 2\}, \tag{4}$$

where $p_f^{(1)}$ and $p_f^{(2)}$ are two hyperparameters that control the sampling ranges. Finally, the perturbed feature matrices are defined as follows:

$$\tilde{\mathbf{X}}^{(k)} = \mathbf{m}_f^{(k)} \odot \mathbf{X}, \quad k \in \{1, 2\}, \tag{5}$$

where $\odot$ is the Hadamard product.

4.1.2 Structure augmentation based on VE

As illustrated in Figure 1 (a), the specific process of the structure augmentation based on VE is as follows. First, the original graph is converted into a line graph. Then, VE is applied to the line graph to obtain the node importance in the line graph, which is the edge importance in the original graph. Finally, based on this edge importance, an edge removal operation is performed to decide whether to delete some unimportant edges in the graph.

Line graph transformation. Given an original graph $G = \{V, E\}$, its line graph is $L_G = \{V', E'\}$, where the node set V' is exactly the edge set E of the original graph. For any two edges $e_1, e_2 \in E$, if they share a common vertex in G, then there will be an edge $(e_1, e_2) \in E'$ in the line graph L_G, representing the adjacency relationship of these two edges in the original graph.

Edge importance calculation using VE. Similar to node importance calculation, the importance of edge i is defined as follows:

$$w_e^i = |E'(v_i')| = |\min_\eta (S_\eta(L_{G_{v_i'}}) - S_\eta(L_G))|, \tag{6}$$

where $S_\eta(L_G)$ denotes the spectral entropy of the line graph, and $S_\eta(L_{G_{v_i'}})$ denotes the spectral entropy of the graph after specific perturbation on node v_i' in the line graph. The importance of edges is normalized to the range $[0, 1]$ using $\gamma_e = (w_{e_{\max}} - w_e)/(w_{e_{\max}} - w_{e_{\min}})$, where $w_{e_{\max}}$ represents the maximum value of w_e, and $w_{e_{\min}}$ represents the minimum value of w_e.

Edge removal. Similar to feature masking, we use γ_e to adjust the edge perturbation distribution:

$$\mathbf{m}_e^{(k)} \sim \text{Bernoulli}(\gamma_e p_e^{(k)}), \quad k \in \{1, 2\}, \tag{7}$$

where $p_e^{(1)}$ and $p_e^{(2)}$ are two hyperparameters that control the sampling ranges. Therefore, the perturbed adjacency matrices can be represented as follows:

$$\tilde{\mathbf{A}}^{(k)} = \mathbf{m}_e^{(k)} \odot \mathbf{A}, \quad k \in \{1, 2\}, \tag{8}$$

where $\odot$ is the Hadamard product.

4.2 The DWFGCNN model

The DWFGCNN is constructed by stacking multiple WFGCNN layers, which can aggregate the fuzzy information of neighboring nodes layer by layer, thereby further enhancing the expressiveness of the representations. The proposed WFGCNN mainly consists of an aggregation process and a fuzzification process, as illustrated in Figure 1 (b). The aggregation process aggregates the topological and attribute information of nodes and their neighborhoods via graph convolution to extract high-level deterministic features, while the fuzzification process transforms the aggregated deterministic features into fuzzy representations to capture feature-level uncertainty.

4.2.1 Aggregation process

We propose an edge weight calculation method based on soft K-means clustering, this method first uses the K-means algorithm to obtain cluster centroids $C = \{c_1, c_2, \ldots, c_k\}$. A Gaussian kernel is then applied to transform the Euclidean distances between node features and centroids into probability distributions, thereby constructing the node-cluster assignment matrix as follows.

$$p_{i,k} = \frac{\exp\left(-\|x_i - c_k\|^2/(2\sigma^2)\right)}{\sum_{j=1}^{K} \exp\left(-\|x_i - c_j\|^2/(2\sigma^2)\right)}, \tag{9}$$

where $p_{i,k}$ represents the probability that node v_i belongs to cluster k, σ denotes the standard deviation of the Gaussian distribution, and x_i represents the features of node v_i. Then, the edge weight is defined as the cosine similarity between its two end nodes in the cluster probability distribution space, according to the following formula.

$$\mathcal{E}_{ij} = \frac{\mathbf{p}_i^\top \mathbf{p}_j}{\|\mathbf{p}_i\| \cdot \|\mathbf{p}_j\|}, \tag{10}$$

where $\mathbf{p}_i = [p_{i,1}, \ldots, p_{i,k}]^\top$ and $\mathbf{p}_j = [p_{j,1}, \ldots, p_{j,k}]^\top$ are the clustering probability distributions of nodes v_i and v_j, respectively. Finally, the weighted adjacency matrix is obtained as follows.

$$\mathbf{A}_W = \mathbf{A} \odot \mathcal{E}, \tag{11}$$

where $\odot$ is the Hadamard product. Then, the aggregation process is defined as follows:

$$\mathbf{H}^l = \mathbf{D}^{-\frac{1}{2}} \mathbf{A}_W \mathbf{D}^{-\frac{1}{2}} \tilde{\mathbf{H}}^{l-1} \mathbf{W}^l, \tag{12}$$

where $\mathbf{D}$ is the degree matrix, $\tilde{\mathbf{H}}^{l-1} \in \mathbb{R}^{N \times d_{l-1}}$ denotes the output of the $(l-1)$-th layer of the WFGCNN, with the initial input $\tilde{\mathbf{H}}^0 = \mathbf{X}$, $\mathbf{W}^l \in \mathbb{R}^{d_{l-1} \times d_l}$ is the trainable weight matrix for the l-th layer, and $\mathbf{H}^l \in \mathbb{R}^{N \times d_l}$ denotes the aggregated hidden feature matrix of the l-th layer.

4.2.2 Fuzzification process

This process is achieved through Gaussian membership functions and feature fusion. Specifically, for each feature dimension $h_{i,j}^l$, the Gaussian membership function is first used to compute its membership degrees to F fuzzy sets:

$$s_{i,j,k}^l = f_j^k\left(\mathbf{h}_{i,j}^l\right) = \exp\left(-\frac{(\mathbf{h}_{i,j}^l - \mu_j^k)^2}{2(\sigma_j^k)^2}\right), \tag{13}$$

where $\mathbf{h}_{i,j}^l$ represents the j-th dimensional feature of node v_i of the l-th layer, $j \in \{1, 2, \ldots, d_l\}$ denotes the dimension index of the features at the l-th layer, $k \in \{1, 2, \ldots, F\}$ represents the number of fuzzy sets, $s_{i,j,k}^l$ is a membership degree value, and the larger its value, the more likely the feature $\mathbf{h}_{i,j}^l$ belongs to the k-th fuzzy set. μ_j^k and σ_j^k, which serve as trainable parameters initialized via K-means clustering, denote the mean and standard deviation of the k-th fuzzy set for the j-th dimension, respectively. Subsequently, for each feature dimension, the maximum membership degree is selected as the fuzzy information for that dimension to avoid the influence of secondary fuzzy sets:

$$\tilde{s}_{i,j}^l = \max_{k=1,\ldots,F} \left\{s_{i,j,k}^l\right\}, \tag{14}$$

where $\tilde{s}_{i,j}^l$ denotes the most important fuzzy information. Finally, the fuzzy membership degree $\tilde{s}_i^l$ is fused with the original features $\mathbf{h}_i^l$ via Hadamard product to generate the final fuzzy representation:

$$\widetilde{\mathbf{h}}_i^l = \tilde{s}_i^l \odot \mathbf{h}_i^l, \tag{15}$$

where $\odot$ reprsents Hadamard product, $\tilde{h}_i^l$ is the fuzzy representation of node v_i, which retains both the original semantics and fuzzy uncertainty. Moreover, once each hidden feature vector is transformed into the fuzzy feature space, subsequent WFGCNN layers can progressively aggregate more fuzzy information from neighboring nodes. A deep model called DWFGCNN can be constructed through the cascaded stacking of multiple WFGCNN layers. For layers where the index $l \geq 2$, the calculation of the fuzzy representation matrix is defined as follows:

$$\mathbf{H}^{l+1} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_W\mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{H}}^l\mathbf{W}^{l+1}, \tag{16}$$

$$\tilde{\mathbf{H}}^{l+1} = \tilde{\mathbf{S}}^{l+1} \odot \mathbf{H}^{l+1}, \tag{17}$$

where $\mathbf{D}$ is degree matrix, $\mathbf{A}_W$ is the weighted adjacency matrix, $\widetilde{\mathbf{H}}^l$ is the fuzzy feature matrix output by the fuzzification process of the l-th WFGCNN, $\mathbf{W}^{l+1}$ is the trainable weight matrix for the $(l+1)$-th layer, $\mathbf{H}^{l+1}$ is the deterministic hidden feature matrix generated by the aggregation process of the $(l+1)$-th layer, and $\tilde{\mathbf{S}}^{l+1}$ stands for the membership degree matrix of the $(l+1)$-th layer.

4.3 Contrastive loss

In contrastive learning, the core of the loss function lies in optimizing the model by contrasting positive and negative sample pairs. We adopt the widely used InfoNCE loss [34] as the contrastive loss function. Specifically, we define the representations v_i and u_i of the same node i under two different views as a positive pair (v_i, u_i), while treating the combinations of v_i with the representations of all other nodes in both views as negative pairs. The loss function for node i is defined as follows:

$$\mathcal{L}_i = -log\frac{e^{\frac{\mathcal{S}(v_i,u_i)}{\tau}}}{e^{\frac{\mathcal{S}(v_i,u_i)}{\tau}} + \sum_{j\neq i}[e^{\frac{\mathcal{S}(v_i,v_j)}{\tau}} + e^{\frac{\mathcal{S}(v_i,u_j)}{\tau}}]}, \tag{18}$$

where $\mathcal{S}(\cdot,\cdot)$ represents the similarity function, defined as the inner product, and τ is the temperature parameter. The total objective function J of the model is defined as:

$$J = \frac{1}{2N}\sum_{i=1}^{2N}\mathcal{L}_i + \frac{\lambda}{2}\|\theta\|_2^2, \tag{19}$$

where N denotes the number of nodes, the first term represents the mean loss over all node representations, and the second term is the L_2 regularization term applied to model parameters θ, with λ serving as the regularization coefficient to prevent model overfitting.

5 Experiments

In this section, we conduct comprehensive experiments on four datasets to evaluate the performance of FGCL across three downstream tasks: node classification, community detection, and link prediction. In addition, we perform parameter analysis and robustness analysis, with their corresponding results presented in Appendix D.2 and Appendix D.3 respectively.

5.1 Experiment setup

Experimental environment. We implement our model using Python 3.10, along with essential libraries including torch 2.1.2, torch-geometric 2.6.1, networkx 3.3, and numpy 1.26.4. To evaluate the performance of FGCL, we conducted experiments on multiple datasets including Cora, CiteSeer, PubMed, and Coauthor-CS[2]. The detailed statistical information of these datasets are provided in Appendix C.1. Moreover, the detailed hyperparameters is shown in Appendix C.2.

Baselines. The baseline models are specifically divided into two categories: one category consists of 4 classic representative models in the field of GCL, namely GCA [35], CSGCL [36], DCMSL [37], and E2Neg [20]; the other category is a newly emerging fuzzy representation learning model, DFGCNN [38]. We describe these methods in detail in Appendix C.3.

[2] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/

Table 1: Experiment results of node classification on four datasets

	Cora		CiteSeer		Coauthor-CS		PubMed	
	ACC	F1-score	ACC	F1-score	ACC	F1-score	ACC	F1-score
GCA	81.37 ± 0.75	80.95 ± 0.92	70.75 ± 0.38	69.10 ± 0.46	92.60 ± 0.18	92.74 ± 0.11	85.03 ± 0.13	85.31 ± 0.22
CSGCL	80.58 ± 0.32	80.17 ± 0.28	71.27 ± 0.36	63.92 ± 0.29	92.69 ± 0.12	93.03 ± 0.08	85.59 ± 0.09	85.48 ± 0.15
DCMSL	77.21 ± 0.42	77.26 ± 0.62	63.99 ± 0.47	64.86 ± 0.34	91.72 ± 0.27	91.96 ± 0.13	82.64 ± 0.92	82.26 ± 0.66
DFGCNN	83.40 ± 0.84	82.02 ± 0.93	71.77 ± 0.68	63.92 ± 1.02	92.00 ± 0.17	89.20 ± 0.19	85.58 ± 0.14	85.27 ± 0.07
E2Neg	77.10 ± 0.66	75.41 ± 0.40	69.88 ± 0.52	63.48 ± 0.24	92.48 ± 0.26	89.38 ± 0.19	85.95 ± 0.31	85.48 ± 0.33
FGCL	$\mathbf{84.41 \pm 0.52}$	$\mathbf{83.95 \pm 0.35}$	$\mathbf{73.30 \pm 0.42}$	$\mathbf{71.05 \pm 0.29}$	$\mathbf{93.05 \pm 0.15}$	$\mathbf{93.07 \pm 0.09}$	$\mathbf{86.09 \pm 0.12}$	$\mathbf{86.23 \pm 0.10}$

Table 2: Community detection results on the ACC and NMI

	Cora		CiteSeer		Coauthor-CS		Pubmed	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
GCA	56.56 ± 0.52	45.06 ± 0.57	53.68 ± 1.09	30.50 ± 0.98	65.88 ± 0.26	71.28 ± 0.35	60.22 ± 0.18	22.18 ± 0.09
CSGCL	62.67 ± 0.53	50.48 ± 0.42	63.53 ± 0.44	39.66 ± 0.35	65.56 ± 0.21	74.18 ± 0.06	57.95 ± 0.15	23.72 ± 0.23
DCMSL	42.61 ± 0.73	24.20 ± 0.55	43.55 ± 0.91	16.91 ± 0.86	37.10 ± 0.68	32.81 ± 0.52	48.49 ± 0.75	13.63 ± 0.80
DFGCNN	59.17 ± 0.43	47.54 ± 0.56	65.94 ± 0.30	41.91 ± 0.26	63.08 ± 0.39	70.10 ± 0.24	39.57 ± 0.24	25.30 ± 0.16
E2Neg	35.99 ± 1.03	19.80 ± 0.83	58.43 ± 1.57	32.87 ± 0.99	66.19 ± 0.56	72.33 ± 0.34	56.85 ± 0.07	22.86 ± 0.16
FGCL	$\mathbf{65.76 \pm 0.81}$	$\mathbf{52.86 \pm 0.50}$	$\mathbf{67.24 \pm 0.37}$	$\mathbf{42.79 \pm 0.24}$	$\mathbf{69.90 \pm 0.29}$	$\mathbf{75.26 \pm 0.10}$	$\mathbf{65.97 \pm 0.19}$	$\mathbf{31.59 \pm 0.18}$

5.2 Overall performance

To comprehensively evaluate the performance of FGCL, we conduct experiments on three downstream tasks: node classification, community detection, and link prediction.

Node classification. In this experiment, similar to [35], we randomly split the dataset into the training set, validation set, and test set in a ratio of 1:1:8. To ensure the reliability of the experimental results, the experiment is repeated 50 times with the data randomly split each time. The classification performance is quantitatively assessed using Accuracy (ACC) and F1-score, and the detailed results are presented in Table 1.

As can be seen from Table 1, FGCL demonstrates significant performance advantages over baseline approaches across all four datasets. Especially on the CiteSeer dataset, the ACC and F1-score of FGCL reach as high as 73.30% and 71.05% respectively, surpassing the second-best method by 1.53% in ACC and 1.95% in F1-score. These results indicate that FGCL can more accurately capture the associations between nodes and category information, effectively improving ACC and F1-score. This fully verifies the generalization and effectiveness of the FGCL method in node classification tasks, and highlights the strong competitiveness of fuzzy representation in node classification scenarios.

Community detection. In this experiment, we utilize the K-means algorithm to cluster the learned node representations, with the number of clusters set to the number of categories of node labels. To ensure the reliability of the results, this experiment is repeated 50 times, with the average ACC, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI) adopted as performance evaluation metrics. The experimental results for ACC and NMI are presented in Table 2, while the results for ARI are shown in Table 3.

We can observe that FGCL outperforms other comparative methods across all datasets, exhibiting particularly superior performance on the Cora, Coauthor-CS, and PubMed datasets. This validates that the fuzzy representations generated by FGCL possess more excellent generalization and invariance properties. Specifically, FGCL achieves performance improvements of 3.09%, 2.38%, and 3.84% in terms of ACC, NMI, and ARI on the Cora dataset, respectively. On the CiteSeer dataset, it realizes improvements of 1.3%, 0.88%, and 1.61% in ACC, NMI, and ARI, respectively. For the Coauthor-CS dataset, the enhancements in ACC, NMI, and ARI reach 3.71%, 1.08%, and 1.79%, respectively. On the PubMed dataset, FGCL yields improvements of 5.75%, 6.29%, and 8.94% in ACC, NMI and

Table 3: Community detection results on the ARI

	ARI			
	Cora	CiteSeer	Coauthor-CS	Pubmed
GCA	33.91 ± 0.81	27.87 ± 0.87	54.09 ± 0.19	20.37 ± 0.13
CSGCL	41.15 ± 0.48	39.24 ± 0.32	59.42 ± 0.13	16.41 ± 0.19
DCMSL	14.49 ± 0.58	14.06 ± 0.74	16.76 ± 0.61	4.99 ± 0.94
DFGCNN	36.02 ± 0.39	41.24 ± 0.35	59.44 ± 0.32	16.53 ± 0.23
E2Neg	9.18 ± 1.10	30.82 ± 1.03	56.10 ± 0.41	19.61 ± 0.14
FGCL	$\mathbf{44.99 \pm 0.59}$	$\mathbf{42.85 \pm 0.28}$	$\mathbf{61.23 \pm 0.21}$	$\mathbf{29.31 \pm 0.15}$

Table 4: Experiment results of link prediction on four datasets

Method	Cora		CiteSeer		Coauthor-CS		PubMed	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP
GCA	93.34±0.15	93.11±0.23	95.22±0.19	95.85±0.25	95.96±0.14	95.98±0.16	94.84±0.07	94.70±0.12
CSGCL	93.81±0.11	93.81±0.17	91.70±0.24	93.13±0.35	96.09±0.27	**96.25±0.31**	95.14±0.15	94.92±0.11
DCMSL	87.80±0.36	89.98±0.30	89.65±0.46	92.15±0.37	94.57±0.29	95.34±0.24	89.50±0.13	89.58±0.10
DFGCNN	93.71±0.12	93.80±0.16	94.12±0.08	93.94±0.13	93.93±0.14	93.46±0.11	94.20±0.09	93.96±0.07
E2Neg	93.98±0.09	93.10±0.13	95.19±0.06	95.15±0.04	93.81±0.15	93.13±0.08	97.05±0.04	96.44±0.03
FGCL	**97.71±0.05**	**97.26±0.14**	**98.24±0.10**	**98.03±0.07**	**96.29±0.12**	95.03±0.13	**97.87±0.06**	**97.59±0.04**

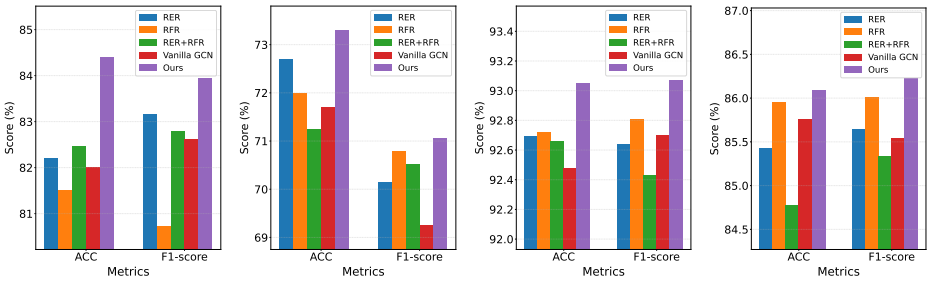
ARI, respectively. The superior performance of FGCL in community detection tasks can be attributed to its capability to capture high-level fuzzy information, which overcomes the limitations of crisp representations in handling noisy or incomplete data. Furthermore, the integration of fuzzy logic principles within the deep architecture significantly enhances the expressiveness and robustness of the learned representations.

Link prediction. In this experiment, we calculate the similarity scores between nodes using the inner product and apply the sigmoid function to convert these scores into probabilities of edge existence. To evaluate link prediction performance, we adopt the Area Under the Receiver Operating Characteristic Curve (AUC) and Average Precision (AP) as evaluation metrics, with all experiments independently repeated for 50 times. The experimental results are shown in Table 4.

It can be seen that the FGCL achieves the optimal performance on most datasets. Specifically, FGCL significantly outperforms other comparison methods in both AUC and AP metrics on the Cora, CiteSeer, and PubMed datasets. Particularly on the CiteSeer dataset, it attains remarkably high scores of 98.24% in AUC and 98.03% in AP, exceeding the second-best method by 3.02% and 2.18%, respectively. This demonstrates its exceptional capability in link prediction. However, on the Coauthor-CS dataset, FGCL outperforms the second-best method by 0.2% in AUC, but its AP is slightly lower than that of CSGCL. This discrepancy is primarily driven by the inherent properties of the dataset and the adaptability of the method. First, Coauthor-CS is a co-author network where edges represent explicit collaborative relationships, featuring much higher semantic certainty than citation networks. This compresses the room for fuzzy modeling to compensate for topological uncertainty, and accordingly narrows the margin of performance improvement. Second, this dataset has a high graph density and strong node homophily, so aggregation based on the raw topology can already provide sufficient neighborhood information. Meanwhile, AP focuses more on the ranking precision of top positive samples; feature fuzzification slightly smooths the fine-grained differences between nodes, exerting a minor negative impact on this metric, whereas AUC, which measures global ranking capability, still maintains its advantage. Overall, the results confirm that FGCL exhibits stronger generalization capability and robustness across most scenarios, validating the effectiveness of its model design.

5.3 Ablation study

We elaborate on the ablation study of key components in the FGCL. Among them, 'RER' indicates the generation of augmented views using random edge removal while keeping other components unchanged. 'RFR' denotes the generation of augmented views using random feature masking while keeping other components unchanged. 'RER+RFR' indicates the generation of augmented views using both random edge removal and random feature masking simultaneously with all other components held constant. 'Vanilla GCN' denotes the replacement of DWFGCNN with a vanilla GCN with all other components held constant. The results of ablation study are depicted in Figure 2. Clearly, the FGCL consistently outperforms all other ablation variants in terms of both ACC and F1-score across all datasets, which fully demonstrates the effectiveness of the proposed model improvement. Specifically, compared with the strategy of using random data augmentation, the VE-based data augmentation method proposed in this paper exhibits significantly better performance. In the vanilla GCN, the output of the encoder is typically fed into a nonlinear projection layer for further processing. In contrast, the proposed FGCL eliminates this projection layer while still achieving superior performance compared to the vanilla GCN. It can be seen from the experimental results that the performance improvement is particularly notable on smaller-scale datasets. We argue that traditional deterministic models are often limited by insufficient samples in small-scale datasets, whereas the fuzzy mechanism introduced in this work effectively mitigates this limitation



(a) Cora (b) CiteSeer (c) Coauthor-CS (d) PubMed

Figure 2: Performance comparison of ablation study.

and enhances the generalization capability of the model. In conclusion, DWFGCNN demonstrates not only higher reliability but also greater potential for practical applications.

6 Conclusion

This paper proposes a novel FGCL framework. Its key innovation lies in a dual strategy: it not only employs VE to guide graph augmentation for preserving critical semantic information, but also explicitly models data uncertainty through fuzzy logic to overcome the limitations of crisp representation in handling noisy or incomplete data. The experimental results across four datasets demonstrate that FGCL significantly outperforms baseline methods in multiple downstream tasks. However, the selection of fuzzy logic operators and the construction of fuzzy rules in this method still rely on empirical design. Furthermore, the current method primarily focuses on static homogeneous graphs, while its capability for processing large-scale, dynamic or heterogeneous graphs remains underexplored. In future work, we plan to investigate adaptive learning mechanisms for automatically optimizing fuzzy membership functions and logic operators, thereby reducing dependence on manual configuration. Moreover, we aim to extend FGCL to support larger-scale, dynamic, and heterogeneous graphs to broaden its applicability and real-world relevance.

Acknowledgments

This work was funded in part by the National Natural Science Foundation of China (Grant No. 62477029,61702317), and the Fundamental Research Funds for the Central Universities (Grant No. GK202602024).

References

[1] Z. Liang, L. Bai, X. Yang, and J. Liang, "Graph contrastive learning for fusion of graph structure and attribute information," *IEEE Transactions on Multimedia*, 2025.

[2] Y. Lin, H. Cai, J.-Y. Hang, H. Wang, Z. Yang, and G. Lyu, "Mitigating local cohesion and global sparseness in graph contrastive learning with fuzzy boundaries," in *Forty-second International Conference on Machine Learning*, 2025.

[3] H. Li, F. Han, and W. Wang, "Cluster-perceptive graph contrastive learning for community detection," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[4] Z. Zhao, Z. Zhu, Y. Liu, J. Guo, and K. Yang, "Heterogeneous graph contrastive learning with augmentation graph," *IEEE Transactions on Artificial Intelligence*, 2024.

[5] Y. Liu, J. Wu, and J. Cao, "Sbp-gca: Social behavior prediction via graph contrastive learning with attention," *IEEE Transactions on Artificial Intelligence*, 2024.

[6] Y. Qi, E. Du, L. Shu, and C. Chen, "Sgca: Signed graph contrastive learning with adaptive augmentation," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–10.

[7] E. W. Vertina, E. Sutherland, N. A. Deskins, and O. Mangoubi, "Mxene property prediction via graph contrastive learning," in *2024 IEEE 14th International Conference Nanomaterials: Applications & Properties (NAP)*. IEEE, 2024, pp. 1–5.

[8] Y. Fan, C. Zhang, X. Hu, Z. Huang, and L. Deng, "Agclnda: Enhancing the prediction of ncrna-drug resistance association using adaptive graph contrastive learning," *IEEE Journal of Biomedical and Health Informatics*, 2025.

[9] G. V. Ramteke, V. Gupta, and P. Singla, "Enhancing recommendation systems with graph contrastive learning: A comprehensive study," in *2025 8th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2025, pp. 903–908.

[10] Z. Yan, W. Weng, H. Shi, and D. Ye, "Multiple augmented graph contrastive learning for recommendation," in *2024 6th International Conference on Electronic Engineering and Informatics (EEI)*. IEEE, 2024, pp. 01–05.

[11] Y. Li, X. Fan, L. Yu, and S. Tian, "Comgcl: Comprehensive graph contrastive learning for recommendation," in *2024 12th International Conference on Information Systems and Computing Technology (ISCTech)*. IEEE, 2024, pp. 1–5.

[12] X. Xiong, X. Wang, S. Yang, F. Shen, and J. Zhao, "Gmni: Achieve good data augmentation in unsupervised graph contrastive learning," *Neural Networks*, vol. 181, p. 106804, 2025.

[13] J. Qiao, Z. Sun, and X. Meng, "Interval type-2 fuzzy neural network based on active semi-supervised learning for non-stationary industrial processes," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 2, pp. 1151–1162, 2023.

[14] B. Tang, X. Chen, S. Wang, Y. Xuan, and Z. Zhao, "Take attention inside: Neighbor pair graph contrastive learning," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[15] J. Chen, K. Guang, H. Yan, Q. Li, and T. Hao, "Graph contrastive learning with adversarial structure refinement (gcl-asr)," in *2024 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2024, pp. 41–50.

[16] M. Yu, J. Liu, and J. Ji, "Causal invariance-aware augmentation for brain graph contrastive learning," in *Forty-second International Conference on Machine Learning*, 2025.

[17] C. Wu, C. Wang, J. Xu, Z. Liu, K. Zheng, X. Wang, Y. Song, and K. Gai, "Graph contrastive learning with generative adversarial network," in *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 2023, pp. 2721–2730.

[18] X. Shen, D. Sun, S. Pan, X. Zhou, and L. T. Yang, "Neighbor contrastive learning on learnable graph augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 8, 2023, pp. 9782–9791.

[19] J. Liang, X. Wei, M. Chen, Z. Wang, and J. Liang, "Gnn-transformer cooperative architecture for trustworthy graph contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, 2025, pp. 18 667–18 675.

[20] Y. Huang, J. Zhao, D. He, D. Jin, Y. Huang, and Z. Wang, "Does gcl need a large number of negative samples? enhancing graph contrastive learning with effective and efficient negative sampling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 16, 2025, pp. 17 511–17 518.

[21] Y. Mo, X. Wang, S. Fan, and C. Shi, "Graph contrastive invariant learning from the causal perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, 2024, pp. 8904–8912.

[22] Y. Liu, Y. Zhao, X. Wang, L. Geng, and Z. Xiao, "Multi-scale subgraph contrastive learning," *arXiv preprint arXiv:2403.02719*, 2024.

[23] B. Zheng, J. Zhang, H. Lu, Y. Chen, M. Chen, W. X. Zhao, and J.-R. Wen, "Enhancing graph contrastive learning with reliable and informative augmentation for recommendation," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 2025, pp. 2101–2112.

[24] C. Xu and M.-T. Kechadi, "An enhanced fake news detection system with fuzzy deep-learning," *IEEE Access*, 2024.

[25] A. Igali and P. Shamoi, "Image classification using fuzzy pooling in convolutional kolmogorov-arnold networks," in *2024 Joint 13th International Conference on Soft Computing and Intelligent Systems and 25th International Symposium on Advanced Intelligent Systems (SCIS&ISIS)*. IEEE, 2024, pp. 1–6.

[26] A. K. Koshariya, M. S. Ashraf, R. Nigam, T. S. Kumar, D. K. Awasthi, and G. Elavarasan, "Implementation of deep convolution neuro-fuzzy network to plant disease detection, risk assessment, and classification," in *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. IEEE, 2023, pp. 575–581.

[27] M. Tanveer, M. Sajid, M. Akhtar, A. Quadir, T. Goel, A. Aimen, S. Mitra, Y.-D. Zhang, C.-T. Lin, and J. Del Ser, "Fuzzy deep learning for the diagnosis of alzheimer's disease: Approaches and challenges," *IEEE Transactions on Fuzzy Systems*, 2024.

[28] Z. Yao, W. Mao, Y. Yuan, Z. Shi, G. Zhu, W. Zhang, Z. Wang, and G. Zhang, "Fuzzy-vgg: A fast deep learning method for predicting the staging of alzheimer's disease based on brain mri," *Information Sciences*, vol. 642, p. 119129, 2023.

[29] Y. Zheng, Z. Xu, T. Wu, and Z. Yi, "A systematic survey of fuzzy deep learning for uncertain medical data," *Artificial Intelligence Review*, vol. 57, no. 9, p. 230, 2024.

[30] J. Xing, R. Xing, C. Xue, and D. Luo, "Enhancing link prediction with fuzzy graph attention networks and dynamic negative sampling," *arXiv preprint arXiv:2411.07482*, 2024.

[31] Y.-N. Lin, H.-C. Cai, C.-Y. Zhang, H.-Y. Yao, and C. P. Chen, "Fuzzy neural network for representation learning on uncertain graphs," *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 9, pp. 5259–5271, 2024.

[32] J. Wei, X. Zhang, W. Pedrycz, and W. Ding, "Dynamic fuzzy sampler for graph neural networks," *IEEE Transactions on Fuzzy Systems*, 2025.

[33] Y. Huang, H. Wang, X.-L. Ren, and L. Lü, "Identifying key players in complex networks via network entanglement," *Communications Physics*, vol. 7, no. 1, p. 19, 2024.

[34] Y. Wang, X. Meng, M. Zou, and S. Leng, "Multi-hop self-augmented graph contrastive learning for node classification," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[35] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proceedings of the web conference 2021*, 2021, pp. 2069–2080.

[36] H. Chen, Z. Zhao, Y. Li, Y. Zou, R. Li, and R. Zhang, "Csgcl: Community-strength-enhanced graph contrastive learning," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. International Joint Conferences on Artificial Intelligence Organization, 2023, pp. 2059–2067.

[37] H. Chen, Y. Li, S. Y. Philip, Y. Zou, and R. Li, "Dcmsl: Dual influenced community strength-boosted multi-scale graph contrastive learning," *Knowledge-Based Systems*, vol. 304, p. 112472, 2024.

[38] C.-Y. Zhang, Y.-N. Lin, C. P. Chen, H.-Y. Yao, H.-C. Cai, and W.-P. Fang, "Fuzzy representation learning on graph," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 10, pp. 3358–3370, 2023.

[39] J.-T. Mo, F. Liu, and X.-X. Deng, "A group decision making model with non-reciprocal fuzzy preference relations based on pearson correlation coefficient," *Expert Systems with Applications*, vol. 276, p. 127095, 2025.

Algorithm 1 The training algorithm for FGCL

Input: Graph $G = \{V, E\}$, epoch number $\mathcal{N}$
Output: Fuzzy node embeddings $\tilde{H}$
Initialize fuzzy graph encoder parameters θ
for $epoch = 1$ **to** $\mathcal{N}$ **do**
  Generate two augmented views $\tilde{G}^{(1)}$ and $\tilde{G}^{(2)}$ by performing data augmentation on graph G
  Compute the corresponding weighted adjacency matrices $A_W^{(1)}$ and $A_W^{(2)}$ for each view
  Obtain the fuzzy node embeddings $\tilde{H}^{(1)}$ of graph $\tilde{G}^{(1)}$ using the fuzzy graph encoder f
  Obtain the fuzzy node embeddings $\tilde{H}^{(2)}$ of graph $\tilde{G}^{(2)}$ using the fuzzy graph encoder f
  Calculate the contrastive loss $\mathcal{J}$ according to Eq. (19)
  Update parameters θ to minimize the contrastive loss $\mathcal{J}$ using stochastic gradient descent
end for
return Fuzzy node embeddings $\tilde{H}$

A Algorithm of FGCL

The proposed FGCL training algorithm is presented in Algorithm 1. The algorithm begins by initializing parameters θ of the fuzzy graph encoder, followed by N iterative training epochs. During each epoch, data augmentation is first performed on the original graph G to generate two augmented views $\tilde{G}^{(1)}$ and $\tilde{G}^{(2)}$. The weighted adjacency matrices $A_W^{(1)}$ and $A_W^{(2)}$ are computed for these views respectively. Next, the fuzzy graph encoder f is used to process these two augmented views, obtaining the corresponding fuzzy node embeddings $\tilde{H}^{(1)}$ and $\tilde{H}^{(2)}$. After that, the contrastive loss $\mathcal{J}$ is calculated according to Eq. (19), and the parameters θ are updated via stochastic gradient descent to minimize this loss. Upon completion of training, the algorithm outputs the fuzzy node embedding $\tilde{H}$, which is essentially the node representation learned by the encoder f that can capture the uncertainty of the graph structure.

B Complexity analysis and comparison

Complexity analysis of FGCL. The complexity of FGCL primarily stems from two components: VE and DWFGCNN. Specifically, the computation of VE involves three core steps. First, eigenvalue decomposition of the Laplacian matrix is performed, with a complexity of $O(N^3)$, where N denotes the number of nodes. Second, the calculation of spectral entropy incurs a cost of $O(N)$. Finally, node perturbation computation entails a complexity of $O(N^2\bar{a}^2)$, where $\bar{a}$ represents the average degree of the network. Consequently, the overall complexity of the VE can be summarized as $O(N^3 + N^2\bar{a}^2)$. The complexity of DWFGCNN primarily arises from its aggregation and fuzzification processes. The complexity of the aggregation process mainly stems from the soft K-means clustering with a complexity of $O(NKd)$, the edge weight calculation with $O(EK)$, and the graph convolution calculation with $O(L(Ed + Nd^2))$, where E represents the number of edges, d denotes the node feature dimension, K is the number of clusters, and L is the number of network layers. Therefore, the complexity of the aggregation process is $O(NKd + EK + L(Ed + Nd^2)$. The complexity of the fuzzification process mainly stems from the Gaussian membership degree calculation with a complexity of $O(NdF)$, the maximum membership degree selection with $O(Nd)$, and the Hadamard product fusion with $O(Nd)$, where F represents the number of fuzzy sets. Thereby, the complexity of the fuzzification process is $O(L(NdF + Nd)$. Therefore, the overall complexity of WFGCNN is $O(NKd + EK + L(Ed + Nd^2 + NdF + Nd))$, which can be further simplified to $O(NKd + EK + L(Ed + Nd^2 + NdF))$. In summary, the overall complexity of the FGCL algorithm is $O(N^3+N^2\bar{a}^2)+O(NKd+EK+L(Ed+Nd^2+NdF)) \Rightarrow O(L(Ed+Nd^2+NdF)+N^3+N^2\bar{a}^2)$.

Complexity comparison with baselines. To comprehensively evaluate the computational efficiency of the proposed FGCL framework, we conduct a systematic complexity comparison between FGCL and all baselines, along with a quantitative evaluation of the actual per-epoch runtime on the standard Cora dataset. The detailed comparison results are shown in Table 5 and Figure 3. In Figure 3, 'VE-node' and 'VE-edge' represent calculating node importance and edge importance using VE in FGCL, respectively. Experimental results show that FGCL has a higher per-epoch runtime than some baseline methods, with its additional computational overhead mainly stemming from the VE-related

Table 5: Complexity comparison with baselines

Method	Complexity
GCA	$O(L(Ed + Nd^2) + NF + N^2d)$
CSGCL	$O(L(Ed + Nd^2) + E \log N + NFM + N^2d)$
DCMSL	$O(L(Ed + Nd^2) + E \log N + NF + N^2d + M^2d)$
DFGCNN	$O(L(Nd^2 + Ed + NFd) + N^2d)$
E2Neg	$O(L(Ed + Nd^2) + NMd + N^2)$
FGCL	$O(L(Ed + Nd^2 + NdF) + N^3 + N^2\bar{a}^2)$

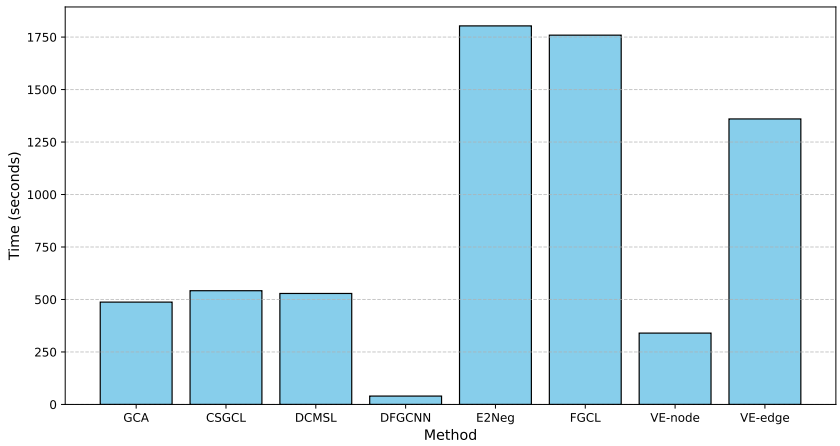


Figure 3: Comparison of wall-clock runtime.

calculation. Nevertheless, FGCL achieves substantial and consistent performance gains across all downstream tasks with an acceptable additional time cost, which demonstrates a favorable trade-off between computational efficiency and model effectiveness.

C Experiment details

C.1 Datasets details

We use Cora, CiteSeer, PubMed, and Coauthor-CS to conduct our experiments. Table 6 presents the detailed statistical information of four datasets, including the type, number of nodes, number of edges, feature dimensionality, and number of classes for each dataset.

Table 6: The details of datasets.

	Cora	**CiteSeer**	**Coauthor-CS**	**PubMed**
Type	citation	citation	co-author	citation
Nodes	2,708	3,327	18,333	19,717
Edges	10,556	9,104	613,788	88,648
Features	1,433	3,703	6,805	500
Classes	7	6	15	3

C.2 Detailed hyperparameters

The detailed hyperparameters in the experiments are shown in Table 7. Among them, $p_e^{(1)}$ and $p_e^{(2)}$ represent the edge removal rates, $p_f^{(1)}$ and $p_f^{(2)}$ represent the feature masking rates, τ is the temperature coefficient in the loss function, F represents the number of fuzzy rules, C denotes the number of clusters, and d_1 and d_2 denote the output dimensions of the first layer of WFGCNN and the second layer of WFGCNN, respectively.

Table 7: Detailed hyperparameters.

	Cora	**CiteSeer**	**Coauthor-CS**	**PubMed**
Learning rate	0.0001	0.0001	0.0001	0.0005
Weight decay	1e-5	1e-5	1e-5	1e-5
$p_f^{(1)}$	0.4	0.8	0.4	0.2
$p_f^{(2)}$	0.8	0.4	0.7	0.2
$p_e^{(1)}$	0.7	0.8	0.7	0.3
$p_e^{(2)}$	0.3	0.7	0.6	0.8
τ	0.7	0.8	0.7	0.4
d_1	256	256	256	512
d_2	256	256	256	256
F	5	5	5	5
C	7	6	15	3

C.3 Detailed baselines

We select GCA [35], CSGCL [36], DCMSL [37], E2Neg [20] and DFGCNN [38] as baseline methods.

- **GCA** utilizes adaptive graph augmentation strategies guided by node importance to improve the quality of representations in contrastive learning.
- **CSGCL** introduces the concept of community strength and employs community strength-based data augmentation methods alongside a dynamic contrastive learning scheme to enhance the quality of node representations.
- **DCMSL** defines dual influenced community strength and node cruciality, proposes node cruciality-guided graph augmentation methods, and employs a joint multi-scale contrastive loss scheme to learn effective graph representations.
- **E2Neg** employs spectral clustering to select representative negative samples, which replace traditional negative sampling strategies.
- **DFGCNN** integrates fuzzy logic with graph convolutional networks to generate fuzzy node representations that effectively capture uncertainties inherent in graph-structured data.

D Experimental results

D.1 The effectiveness of VE

To evaluate the effectiveness of VE, we conduct correlation analyses comparing it with traditional centrality measures such as degree centrality, betweenness centrality, and closeness centrality. This aims to verify whether VE can provide information distinct from existing centrality metrics, thereby more accurately assessing node importance. We employ the Pearson Correlation Coefficient (PCC) [39] to measure the correlations between these metrics. A negative value indicates a negative correlation between variables, a positive value signifies a positive correlation, and a value close to zero suggests no significant linear dependence between the compared measures. Figure 4 shows the results of the correlation analysis for each centrality measure across the four datasets.

Overall, VE generally demonstrates low correlations with traditional centrality metrics. This indicates that the global perspective based on spectral entropy perturbations can reveal information that is overlooked by conventional centrality measures focused on local topological perspectives. However, this correlation exhibits variations across different datasets. In networks with high clustering coefficients, such as Cora, VE has minimal correlations with other centrality metrics. Here, nodes with high VE values are mainly situated at bridging positions between communities, rather than among highly connected nodes with dense local connections. Conversely, in networks like CiteSeer where degree distributions more closely follow a power law, VE exhibits strong correlations with degree and betweenness centrality, while remaining weakly correlated with closeness centrality. This variation is largely attributable to differences in network topological heterogeneity. The core of VE lies in evaluating how nodes influence global spectral properties, rather than merely measuring local topological characteristics. Thus, VE distinctly differs from traditional metrics in highly modular,

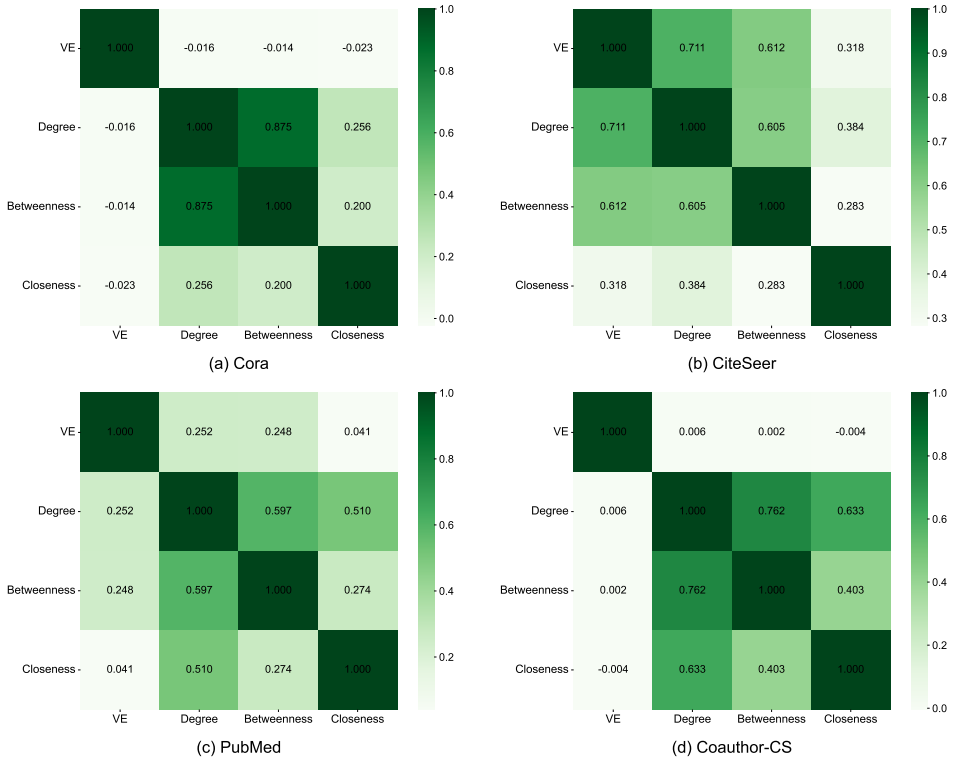


Figure 4: The PCC heat map of four centrality metrics across four datasets.

structurally consistent networks. In networks exhibiting power-law heterogeneity, while VE overlaps to some extent with degree and betweenness centrality, it still preserves its distinct analytical perspective, offering a unique complement to existing node importance evaluation methods. This finding is intuitively supported by node spatial distribution visualizations. Using the Cora dataset as an example for illustration, Figure 5 shows the node importance distributions for degree centrality, betweenness centrality, closeness centrality, and VE, respectively. As seen in Figure 5 (a)–(c), critical nodes identified by traditional centrality metrics tend to be densely interconnected, forming clusters in specific areas of the network. In contrast, the critical nodes identified by VE in Figure 5 (d) are not concentrated locally but are distributed more uniformly across the entire network. This significant spatial distribution difference further demonstrates that VE can effectively identify structural bridge nodes that maintain global network connectivity, rather than being limited to core nodes within locally dense areas. The visualization analysis strongly supports the following key conclusion: VE provides a unique analytical perspective that is independent of conventional centrality measures.

D.2 Parameter analysis

To evaluate the sensitivity of the proposed model to key parameters, we conduct experiments on the Cora dataset, with the results presented in Figure 6. As can be observed, the model achieves optimal performance when $F = 5$. An insufficiently small F fails to provide enough fuzzy sets to fully characterize the fuzzy distribution of the features, thereby limiting the model's capacity for uncertainty modeling. Conversely, an excessively large F leads to overly fragmented fuzzy partitions and introduces redundant information, which not only increases computational overhead but also degrades the discriminability of the learned representations. For C, the model attains its best performance when $C = 7$. Notably, even when C deviates from this optimal value, the performance degradation remains modest, demonstrating that FGCL is robust to the choice of C. The learning rate exerts the most drastic impact on performance: as the learning rate increases from 0.0001 to 0.05, both ACC and F1-score drop significantly, reflecting the high sensitivity of the optimization process to the learning rate. The τ reaches its peak performance at $\tau = 0.7$; both lower and higher values cause noticeable performance degradation. For $p_e^{(1)}$, $p_e^{(2)}$, $p_f^{(1)}$, and $p_f^{(2)}$, their optimal values

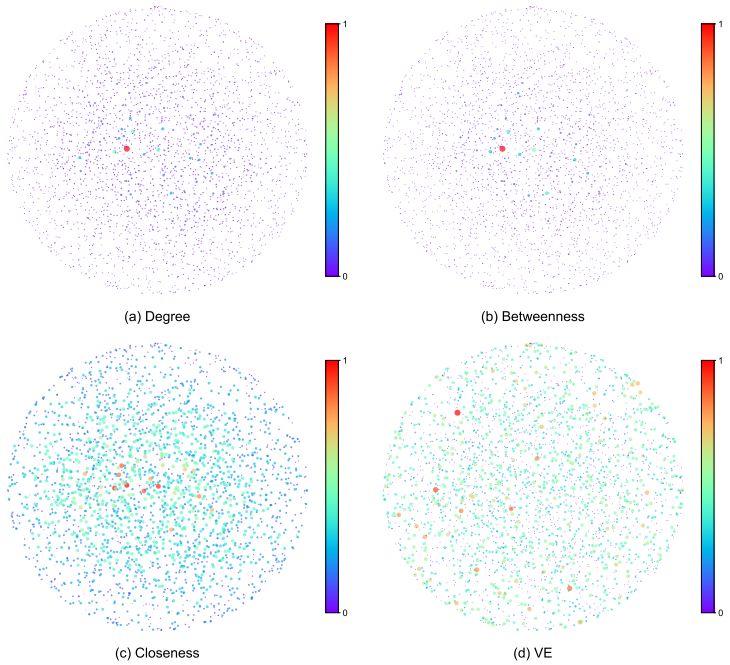


Figure 5: Visualization of four centrality metrics on the Cora dataset.

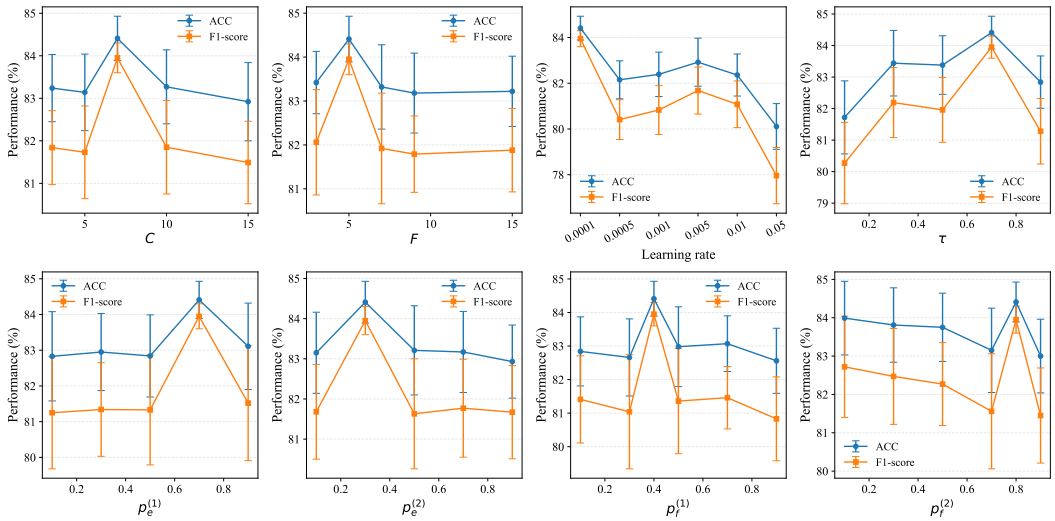


Figure 6: Parameter analysis.

are 0.7, 0.3, 0.4, and 0.8, respectively. Values that are too high or too low both lead to decreased model performance.

D.3 Robustness analysis

To further verify the robustness of FGCL under uncertain graph environments, we inject 10% and 20% feature and structural noise into the Cora dataset for node classification. The experimental results are presented in Figure 7. The results demonstrate that FGCL consistently surpasses all baselines across all noise settings. Baselines suffer severe performance degradation with increasing noise, especially under feature corruption, while FGCL exhibits moderate decay and retains competitive performance. FGCL also maintains strong performance under structural noise. Across all models, structural noise

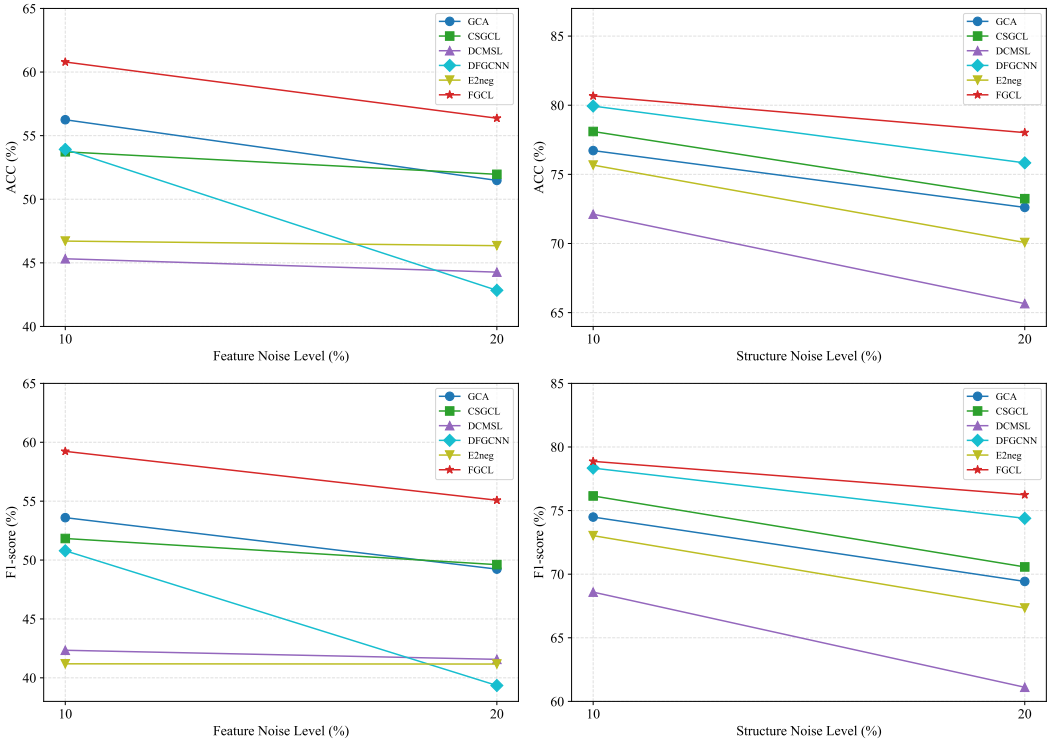


Figure 7: Performance comparison under different noise levels.

leads to milder performance drops than feature noise, consistent with graph learning characteristics: node features carry fine-grained semantics and are more vulnerable to noise, while graph topology provides redundant neighborhood information with inherent fault tolerance. Overall, FGCL learns robust representations against both feature and topological perturbations and outperforms existing GCL baselines, verifying its robustness for noisy graphs.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction delineate the key challenges addressed in this work, our main contributions, along with an overview of the experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include the limitation of our work in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper is experimental instead of theoretical. It includes no theorems or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided a thorough description of the entire experimental process, and have properly cited all open-source data needed for reproduction of the experiments. The code will be publicly accessible upon acceptance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our experiments utilized publicly available datasets, and we intend to release the code upon acceptance of the paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide experimental details in Section 5.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide error bars in Table 1, 2, 4, and 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources in Section 5.1 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper is compliant with the NeurIPS code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The proposed algorithm has no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly identifies the creators or original owners of the assets used (such as code, data, and models). In addition, we have correctly cited the relevant literature.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We will disclose the code after the confirmation of the status of the paper, but currently no new assets is available.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.