

Research Article

Community-Aware DropEdge for Overlapping Community Detection with Graph Attention Networks.

Elaf Adel Abbas

College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq

Article Info

Article history:

Received 12 -2-2026

Received in revised form 12-4-2026

Accepted 5-5-2026

Available online 30 -6-2026

Keywords: Social Network, Graph Attention Networks, Graph Neural Network, Overlapping Community Detection.

Abstract

Community detection plays a central role in analyzing complex real-world networks such as social and collaboration graphs. However, in many practical scenarios, communities are overlapping, meaning that a node may belong to multiple groups simultaneously. Recent deep learning approaches, notably Neural Overlapping Community Detection (NOCD), have addressed this challenge by combining probabilistic modeling with graph neural networks, while Attention-based Overlapping Community Detection (AOCD) further improves representation learning through graph attention mechanisms. Despite these advances, existing methods typically rely on a fixed graph topology during training, which can limit their ability to handle noisy inter-community links in overlapping settings. In this work, a community-aware DropEdge framework has been proposed, built on top of an AOCD-style graph attention encoder. The proposed method adaptively removes edges with low feature coherency, measured using cosine similarity between normalized node features, while preserving full context adjacency for evaluation. This design reduces structural noise during training and strengthens the community-consistent message without changing the probabilistic purpose of the original model. Experimental results on benchmark datasets with overlapping ground truth communities show that the proposed approach consistently outperforms the baseline in all 10 evaluated datasets. On average, this improves retrieval performance by 1.2 NMI points (about 3.35% relative gain), with the largest improvement reaching 4.2 NMI points. These findings indicate that community-aware structural regularization can be effective.

Corresponding Author E-mail: elaf.adil@uokerbala.edu.iq

Peer review under responsibility of Iraqi Academic Scientific Journal and University of Kerbala.

1- Introduction

Complex networks have become an essential model of a wide variety of physical world systems, which include social interactions, citation graphs, biological networks, and information infrastructures [1]. As a natural modeling of such systems, graph modeling involves displaying entities as vertices and defining relational structure as an edge [2]. One of the key tasks in graph mining is the identification of communities, or groups of nodes with dense interrelationships, or both structural and semantic properties. Community structure elucidates insights into the organization, functionality, and dynamical behavior of networks [3], [4]. Moreover, techniques for generating and identifying communities, especially weighted ones, have demonstrated efficacy in maintaining the structural characteristics of the network by ensuring that nodes remain cohesive inside the communities [5].

The classical community detection algorithms are mostly based on the assumption of a strict graph division where all the vertices are only in one community. In real life, this assumption of hard-partition often does not apply. Most real-life situations also include the possibility of overlapping communities, and vertices can be members of more than one community [6]. As an example, in social networks, an individual can be a member of a family, career, and hobby group at the same time. In a collaboration graph, a researcher can be working on several different lines of research. These empirical data encourage research into Overlapping Community Detection (OCD), which aims to approximate multi-membership designations at the expense of the muddying effects of structural overlap and observational stuttering [7].

Recent advances in deep learning on graphs will have a significant impact on the learning of representations of non-Euclidean data. Graph neural networks (GNNs) produce expressive node embeddings by pooling information from nearby nodes, thereby supporting subsequent tasks such

as node classification, link prediction, and community detection [8]. Graph convolutional networks (GCNs) are a different type of GNNs that can be used, in which neighborhood aggregation is done in an equal way [9], [10] i.e., all neighbors have equal weight, and in a graph attention network (GATs) [11], where the neighbors can be chosen unevenly based on their relevance. This heterogeneous neighbor importance ability of GATs makes them an attractive source of OCD, especially in graphs with complex local geometry. One of the most widely used sophisticated frameworks used to identify overlapping communities is Neural Overlapping Community Detection (NOCD). NOCD combines generative models of soft community membership, which uses Bernoulli-Poisson models with GNN encoders and has shown good performance and scalability to large graphs. However, the traditional per-NOCD formulation is quite dependent on GCN-type encoders and standard regularization strategies that might be inadequate to potentially entirely utilize attention-based structural incentives or reduce omnipresent inter-community noise in adjacent situations.

However, existing structural regularization approaches for graph neural networks, including DropEdge, generally operate in a topology-agnostic manner through uniformly dealing with edges randomly, and although this method can alleviate overfitting and oversmoothing, it fails to account for the relevance of semantic or community phases [10]. This dilemma is particularly important in overlapping network identity settings, where inter-community edges often introduce enough structural noise that it can be difficult to discern important group memberships.

To overcome this difficulty, a more community-specific edge reduction approach, distinct from the traditional random edge deletion method, can be proposed. Unlike standard DropEdge, this technique relies on conscious, hierarchical organization by deleting aspects that are conditioned by feature coherence between linked nodes. Specifically, edges connecting nodes with

low task similarity are more likely to be deleted, as they are less indicative of belonging to a shared community. Linking edges are preferentially preserved. This design enables the release to curb cluttered communication between groups and enhance its structure during message propagation.

Moreover, the method fits tightly within the AOCD/NOCD-fashion prospective framework, where a sparsely-done graph is best used at some stage in school to guide representative learning, at the same time as assessment ends consistently in a full graph. They lead to better messaging [9]. As a result, the proposed technique provides a theoretical and enterprise-unique extension of DropEdge that is particularly ideal for overlapping community identity on noisy real-world networks.

This paper revisits NOCD using the GAT encoder and the corresponding structurally regular education graph for covered community identification. The fundamental concept is to introduce a community-aware DropEdge mechanism that selectively prunes edges with low attribute coherence, thereby reducing structural noise and improving network static messaging. To ensure honest evaluation with previous AOCD/NOCD-style studies, all experiments in this work for binary community missions use the same old constant-threshold protocol with $p=0.5$.

2-Preliminaries

This section presents the terminology, graphical representation, and learning elements used in this study, adhering to the standard framework for graph neural networks [12], [13] and identifying nested communities [14], [15].

2-1 Graph Notation

Let $G=(V, E)$ represent an undirected graph, where V constitutes the collection of vertices and E comprises the set of edges. Denotes a sparse adjacency matrix graph structure:

$$A \in \{0,1\}^{N \times N} \quad (1)$$

where $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ equation otherwise. In attributed graphs, each node i is linked to a feature vector

$$\mathbf{x}_i \in \mathbb{R}^d \quad (2)$$

The feature matrix is represented by $X \in \mathbb{R}^{N \times d}$. Nodes can be part of more than one community at the same time, a fundamental feature of many real-world datasets.

$$Z \in \{0,1\}^{N \times K} \quad (3)$$

Let $Z_{ik} = 1$ represent the ground truth community membership matrix, indicating that node i is a member of community k . The aim of the overlapping community detection problem is to compute a soft membership matrix.

$$\hat{Z} \in \mathbb{R}_+^{N \times K} \quad (4)$$

Consequently, thresholding is used to determine the final binary membership.

2-2 Normalized Adjacency and Self-Loops

Graph neural networks typically rely on the generalized adjacency matrix to stabilize message passing. Given a degree matrix D , the symmetric generalized adjacency is

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (5)$$

Following common practice in GNNs[16], Self-loops are added when needed:

$$A^{(\text{self})} = A + I \quad (6)$$

Allowing each node to preserve its own representation during aggregation.

2-3 Attention-Based Message Passing

The graph attention encoder has been used (GAT)[11], where the node representation is updated by computing dynamic, edge-specific attention coefficients. For an input layer with node embedding $H^{(l)}$, the attention coefficients between nodes i and j are defined as:

$$e_{ij} = \text{LeakyReLU} (a^i [W H_i^{(l)} \parallel W H_j^{(l)}]) \quad (7)$$

where A is a learnable weight matrix, a is an attention vector, and \parallel denotes organization. Coefficients are normalized between neighbors:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})} \quad (8)$$

The updated representation is then:

$$H_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} WH_j^{(l)} \right) \quad (9)$$

where σ is an activation function like ReLU. Multi-head attention is employed by combining or averaging multiple parallel attention channels, improving model stability and expressivity.

2-4 Overlapping NMI (ONMI)

To evaluate the quality of the estimated communities[17], the Overlapping Normalized Mutual Information (ONMI) has been used, a widely adopted measure for soft and multi-membership segmentation.

Given ground truth Z and predicted communities \hat{Z} , ONMI computes the mutual information between overlapping sets and normalizes it to $[0, 1]$, where:

- ONMI = 1 indicates perfect matching
- Values near 0 indicate no mutual information between partitions.

Since the predictors \hat{Z} are continuous, binarization is necessary:

$$\hat{Z}_{ik} = \begin{cases} 1, & \hat{Z}_{ik} > \tau, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where τ is either a fixed threshold (for example, 0.5) or adaptively tuned. In work, a fixed-threshold $\tau = 0.5$ has been used.

2-5 Data transformations

To study structural robustness and propose new contributions, three variable sets were applied to the graph prior to training:

a) Self-loop increase

Introduces identity edges to preserve local information and stabilize training[18].

b) Random DropEdge

Randomly removes a fraction p of edges:

$$E' = \{(i, j) \in E : u_{ij} > p\}, u_{ij} \sim \text{Uniform}(0,1) \quad (11)$$

acting as a regularize.

c) Community-Aware DropEdge (proposed)

Edges between nodes with low feature similarity are removed more aggressively[19]:

$$p_{ij}^{\text{drop}} = p \cdot (1.5 - s_{ij}) \quad (12)$$

where s_{ij} the normalized cosine similarity between node features. This change reduces noise from cross-community edges and increases separation.

3- Related Work

3-1 Overlapping community detection

Yang and Leskovec presented BIGCLAM, a scalable method for detecting overlapping communities with non-negative matrix factorization (NMF), which represents the membership strength of each node to communities via a non-negative affiliation matrix [20]. Their principal empirical discovery is that overlapping areas in real networks exhibit more density than non-overlapping areas, prompting a generative model in which the likelihood of an edge grows with the quantity of shared communities. This approach determines node-community associations by optimizing a likelihood function

through an efficient gradient-based method that accommodates networks with millions of nodes. Post-training, community membership is evaluated by assessing the acquired association strengths. Experimental findings on extensive social and information networks indicate that BIGCLAM surpasses conventional overlapping approaches, including clique percolation and link clustering, thereby establishing itself as a

formidable non-neural benchmark for overlapping community detection.

Lee et al. suggest LEMON, a local spectral clustering system that is proposed to identify overlapping communities in large-scale networks based on expanding around a small seed set instead of using the entire graph structure [21]. The algorithm builds a local spectral subspace based on short random walks that are approximations of an invariant subspace in the neighborhood of the seeds and then finds communities by solving a constrained minimization task of the form of a sparse indicator vector by means of the (L1)-minimization. LEMON can be described as highly accurate: It uses operation-based sweep cuts as a way of defining the boundaries of a community and reseeding strategies to refine its predictions. Benchmarking LFR synthetic networks and experiments on large real graphs, such as Amazon, DBLP, YouTube and Orkut, shows that LEMON performs better than all baselines, global and local, including Heat Kernel, Page Rank, OSLOM, Daemon and Link Communities, and performs better with the size of the discovered community size, but not with the size of the entire graph.

Sun et al. come up with OCDID (Overlapping Community Detection based on Information Dynamics) that views community discovery as an information-propagation process on the graph, rather than an explicit optimization problem[22]. Each node has a state vector of its association strengths with the possible communities, and the resulting community structure is the result of two mutually dependent mechanisms: information accumulation, which sums the signals received by the node, and information dissipation, which smooths out the state of nodes before uncontrolled propagation occurs. The repeated updates bring the nodes to a steady state eventually, which models the multi-community patterns of membership. The suggested local and scalable update rule avoids optimization globally and bypass with the overlap communities, after which a post-processing phase of refining soft memberships in crisp overlapping communities through similarity threshold and hierarchical merging is

conducted. Empirical analysis of various benchmark networks has shown OCDID to perform competitively and even better than classical algorithms like COPRA, SLPA, and fuzzy modularity, especially in high-overlap density graphs, highlighting the importance of probabilistic, deep-learning, and dynamical-systems formulations.

3-2 Graph neural networks for community detection

Bruna and Lee have proposed one of the earliest frameworks combining spectral community detection, belief propagation, and graph neural networks (GNNs) into a unified learning-based solution [23]. Their model begins by observing that classical community detection in stochastic block models (SBMs) relies either on spectral relaxation – using operators such as the graph Laplacian or Bethe Hessian – or on probabilistic inference via belief propagation. They show that these iterative spectral or message-passing processes can be unrolled into a deep GNN architecture, where each layer corresponds to a localized graph operator followed by nonlinearity and normalization. This formulation enables GNNs to explicitly learn the appropriate spectral operator, even in sparse graphs where traditional Laplacian methods fail. A consequence is that the trained GNN can attain the information-theoretic detection limit of SBM without access to model parameters and significantly exceed parametric or fixed-operator baselines, demonstrating GNNs as an effective and versatile alternative to manually designed graph operators in sparse or noisy networks.

One of the first GNN-based models explicitly designed to fill the gap in a previously existing deep-learning literature, which aimed at disjoint segmentation, was the NOCD [14], proposed by Schur and Gunnemann and developed specifically to detect overlapping communities. Their method combines the representation learning of GNNs with the Bernoulli-Poisson generative models in order to learn the soft membership of the community, and the method is optimized

end-to-end and has scalable inference on large, sparse graphs.

The research notes that the GNN architecture offers an inductive bias, which promotes local smoothness of the association vectors, which is much more accurate on identification compared to matrix-factorization, probabilistic baselines (BigCLAM, CESNA, EPM), and embedding-based designs (DeepWalk-NEO and Graph2Gauss-NEO). Extensive experiments on Facebook ego-networks and real co-authorship graphs demonstrate that NOCD is always able to achieve the state of the art NMI scores, and ablation experiments indicate that the omission of the GNN component has a devastating impact on performance, illustrating the significance of graph-organized neural messaging to capture overlapping community structure.

Sobolewski and Bailey introduce a GNN-inspired recurrent optimization algorithm (GNNS) that models the maximization of modularity as a continuous quadratic target and optimizes it by training a neural-style update, rather than a discrete target estimation[24]. Their algorithm updates soft community assignments with each node taking the weighted information of its neighbors in the form of modularity, and then by combining the previous state of the node with the weighted information via nonlinear activation and normalization. To avoid poor local optima, GNNS employs multiple random initializations and parameter perturbations in multiple stages, selecting high-modularity candidates for further refinement. Although fundamentally unsupervised, the algorithm inherits useful properties from neural architectures – parallelism, iterative message sending, and constant relaxation – while producing clean separation through discretization upon convergence. Empirically, GNNS achieves competitive or better modularity than state-of-the-art optimizers such as Combo, Louvain, Leiden, and ADVNDS, and its extension to temporal graphs shows that the same recurrent architecture can effectively track community evolution over time.

Recently, Yuan et al. Introduce CDMG[25], a GCN-based framework that detects overlapping

communities by maximizing Markov stability, a dynamical parameter that measures how long random-walk trajectories remain within candidate communities. Unlike NOCD, which optimizes first-order similarity and may degrade on noisy or structurally complex networks, CDMG uses a two-layer GCN to learn community affiliation weights while directly optimizing a discrete Markov-stability objective, leading to more robust extraction of multi-scale structural patterns. The method naturally incorporates node characteristics and scale in large attributed graphs, outperforming classical algorithms (CPM, SimNMF, NNSD) and neural baselines (Node2VEC, LINE, COME, NOCD) in real, synthetic and attributed datasets. A central insight is the role of Markov time, which controls the temporal scale of stability; The authors propose a trichotomy search strategy to estimate the optimal time, which significantly improves the accuracy on large networks. Their analysis further shows that among multiple GNN architectures, GCN outperforms GAT and GIN for this task due to uniform aggregation over edges, which aligns well with the structural nature of community detection.

Taken together, these works demonstrate how GNNs and GNN-inspired architectures can encode spectral, probabilistic, modularity-based, and dynamical principles within a unified learning framework for community detection, and they also provide the neural baselines against which the proposed method has been compared.

4- Background and Model Foundation

4-1 Overlapping Community Detection

Let $G=(V,E)$ where $N=|V|$ be an undirected graph with vertices and an adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$.

In the case of overlapping community detection, a node can be a member of one or more communities, with the strength of each affiliation varying. This is conventionally modeled through a non-negative **affiliation** matrix

$$\mathbf{F} \in \mathbb{R}_{\geq 0}^{N \times K} \quad (12)$$

where $F_{u,c}$ represents the strength of membership of node u to community c . Social and collaborative networks are empirical networks, and thus, the overlapping models are more suitable than the non-overlapping models.

4-2 Bernoulli–Poisson Generative Model

Like the Neural Overlapping Community Detection (NOCD) framework [17], a Bernoulli–Poisson generative model is used to model network connectivity[26], which has been widely used in scalable overlapping community detection. In this model, the likelihood of there being an edge between two nodes, u and v , is the inner product of the affiliation vectors of the pair, u and v ;

$$P(A_{uv} = 1) = 1 - \exp(-\mathbf{F}_u^T \mathbf{F}_v) \quad (13)$$

This constitutes a good model of overlapping community structure: node pairs in many communities have greater connection probability, and yet allow sparse overlap. Besides, the model enables the building of a controllable likelihood function that can be tuned over large graphs.

4-3 Neural Overlapping Community Detection (NOCD)

Neural Overlapping Community Detection (NOCD)[14], defines the estimation of the affiliation matrix F as a representation-learning problem. Instead of treating a free variable as an F , NOCD calculates it using a graph neural network encoder:

$$\mathbf{F} = \text{GNN}_\theta(\mathbf{A}, \mathbf{X}) \quad (14)$$

where $\mathbf{X} \in \mathbb{R}^{N \times D}$ represents the attributes of the nodes.

Model parameters are acquired by minimizing a balanced Bernoulli–Poisson negative log-likelihood, wherein positive edges and sampled non-edges contribute equally to the loss function. This design facilitates scalable stochastic optimization while maintaining the model's probabilistic interpretability

4-4 Limitations of Existing Approaches

Regardless of the merits of attention-based encoders, both NOCD and Attention-based Overlapping Community Detection (AOCD) models are trained on the basis of a static-graph paradigm. Edges are all considered equally, irrespective of whether they are indicative of strong intra-community relationships or weak, noisy inter-community relationships. This weakness, in particular, is acute in a viable social and collaborative network, where the presence of extraneous edges or tenuous relationships or sampling artifacts can mask true community boundaries and negatively impact representation learning.

4-5 Motivation for Our Approach

To address this shortcoming, we propose supplementing attention-based message passing with a structured graph that reflects the network's architecture. Instead of simply randomizing the graph, a community-sensitive edge projection has been employed, where the similarity of node characteristics serves as an indicator of community cohesion. This eliminates edges that are unlikely to represent the real-world community structure, allowing us to retain useful communication patterns with greater flexibility and generalizability. This improvement is directly based on the AOCD framework and aligns perfectly with its probabilistic objectives, which form the basis of the methodology presented in the next section.

5- Proposed Method

5-1 Overview

Expanding on the AOCD model[7]. The community-aware DropEdge framework for overlapping community detection has been presented that serves as a structural regularization layer. Even while AOCD uses attention-based message forwarding to enhance representation learning, it trains under the assumption of a fixed graph topology. Without altering the fundamental probabilistic goal, the approach augments the attention mechanism through community-consistent dynamic perturbation of the training graph.

Removing edges that don't accurately represent community membership while keeping very informative linkages between communities is the core notion.

5-2 Model Backbone

The methodology uses the AOCD architecture as the encoder, wherein node-community affiliation strengths are computed thus:

$$\mathbf{F} = \text{ReLU}(\text{GAT}_\theta(\mathbf{A}, \mathbf{X})) \quad (15)$$

The encoder includes two graph attention layers with a hidden dimension of 128, with the initial layer replaced by batch normalization, dropout regularization, and ℓ_2 weight decay. The decoder and loss functions conform to the Bernoulli–Poisson framework, facilitating direct comparison with NOCD and AOCD.

5-3 Motivation: Limitations of Uniform DropEdge

Regardless of the edges' structural or semantic relevance, Uniform DropEdge removes them at

random with a fixed probability. In sparse or noisy networks, this strategy may eliminate critical intra-community edges, even though it is successful as a regularization tool in some conditions.

Experimental results show that random DropEdge frequently causes performance loss or unstable gains, highlighting the necessity for structure-aware replacements.

5-4 Community-Aware Edge Dropout

To decrease this limitation, Community-Aware DropEdge has been developed, a technique for adaptive edge elimination that utilizes the similarity of node properties to indicate underlying community cohesion: commence with the original graph (Figure 1(a)), calculate feature similarity on the edges (Figure 1(b)), identify the edges most susceptible to removal (Figure 1(c)), and then obtain the final sparsified training graph (Figure 1(d)).

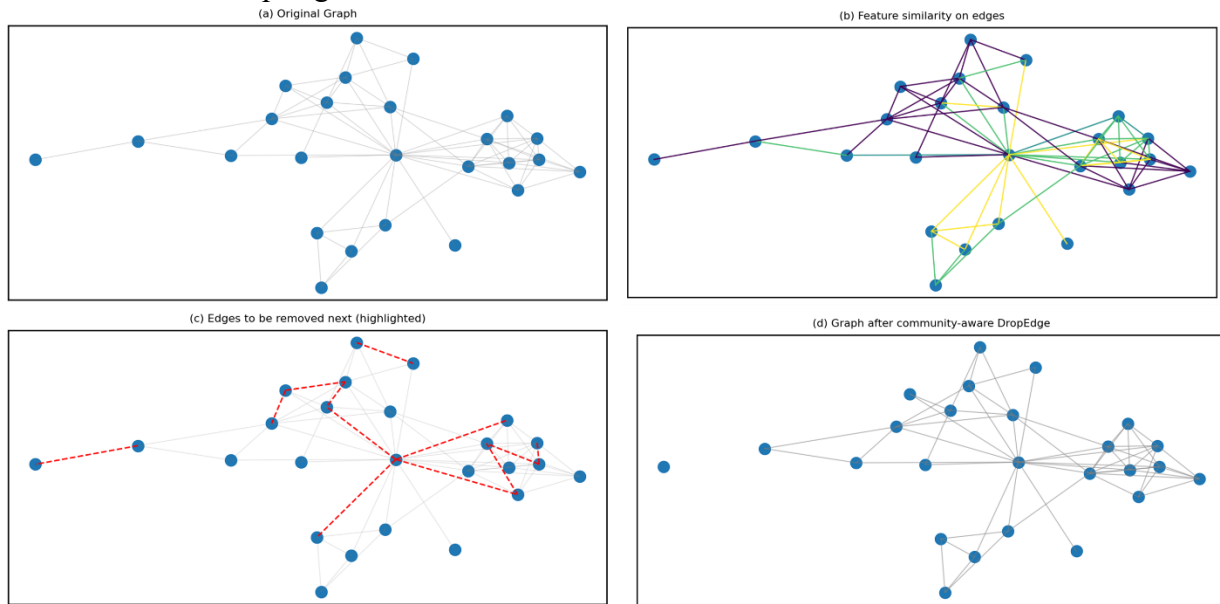


Figure 1: Illustration of the proposed community-aware DropEdge

Feature Normalization

The initial step in normalizing node characteristics is ℓ_2

$$\tilde{\mathbf{X}} = \text{normalize}(\mathbf{X}). \quad (16)$$

Edge Similarity Estimation

It is necessary to compute the cosine similarity for every edge (u, v) that is situated inside E .

$$s_{uv} = \tilde{\mathbf{X}}_u^T \tilde{\mathbf{X}}_v. \quad (17)$$

The similarity scores are min–max normalized to obtain $\hat{s}_{uv} \in [0,1]$.

Adaptive Drop Probability

The definition of an edge-specific dropout rate is different from a uniform dropout probability, which is as follows:

$$p_{uv} = p_{\text{base}} \cdot (1.5 - \hat{s}_{uv}) \quad (18)$$

Here, p_{base} will be the global dropout rate. Therefore,

- Edges that connect nodes with low similarity are often eliminated.
- Most of the edges that connect nodes that are quite similar are kept.

The proposed approach aims to maintain meaningful inter-community connections even as a reduction of noisy edge margins, thereby increasing the excellence of messaging pass. The proposed rarefaction mechanism is intended to complement the ophthalmic mechanism of GAT instead of replacing it. The community-aware DropEdge is applied as a structural regularization step before messaging that modifies the community graph using the removal of less informative edges, and then the GAT encoder learns the attention weight on the size of the closing community. In this sense, sparsening and attention perform at distinct levels of the model and play complementary roles: sparsening reduces structural noise, while attention gives importance to the remaining edges.

5-5 Training Strategy

The training methodology corresponds to the stochastic optimization framework of AOCD. The model is trained using the perturbed adjacency matrix, while the complete original graph is retained for loss assessment to guarantee unbiased findings.

The balanced Bernoulli-Poisson negative log-likelihood has been maximized by mini-batch edge sampling. The preliminary termination verification relies on the sustainability of the loss.

5-6 Community Assignment

The anticipated affiliation matrix F possesses uniform membership powers. A fixed threshold of $\rho=0.5$ is employed to derive binary community allocations.

$$\hat{Z}_{u,c} = \begin{cases} 1, & \text{if } F_{u,c} > \rho, \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

This thresholding method expands upon earlier studies by allowing for direct comparison of retrieval performance[7].

6- Experiments

6-1 Dataset

A collection of graph datasets taken from the actual world was utilized in tests. Facebook is a collection of smaller networks that are ego-networks[27]. The term "Facebook ego network" refers to a network of relationships that exists between the friends of an individual within the Facebook graph. The selected ego-networks have anywhere from fifty to eight hundred nodes.

Chemistry, Computer Science, Medicine, and Engineering are all co-authorship networks that were created from the Microsoft Academic

Graph [28]. These four bigger real-world datasets were employed, and they all had reliable ground truth overlapping community information and node properties. Communities are equivalent to research domains, and the attributes of nodes are determined by the keywords that are listed in the articles written by each author. The results of the tests are shown in Table 1, which contains statistics for all of the datasets that were processed.

Table 1: The Information of the Dataset

Dataset	Nodes	edges	number of node attributes	number of ground-truth communities
fb_348	224	3.2k	21	14
fb_414	150	1.7k	16	7
fb_686	168	1.6k	9	14
fb_698	61	270	6	13
fb_1684	786	14.0k	15	17
fb_1912	747	30.0k	29	46
mag_chem	35.4k	157.4k	4.9k	14
mag_cs	22.0k	96.8k	7.8k	18
mag_eng	14.9k	49.3k	4.8k	16
mag_med	63.3k	810.3k	5.5k	17

6-2 GAT Configuration (AOCD Backbone)

The AOCD setting has been followed, where the node-community affiliation matrix is calculated by the graph attention network[14]:

$$\mathbf{F} = \text{ReLU}(\text{GAT}_\theta(\mathbf{A}, \mathbf{X})) \quad (20)$$

And the model parameters are optimized by minimizing the balanced negative log-likelihood of the Bernoulli–Poisson formulation, exactly like NOCD, but with GCN replaced with GAT.

The remaining network hyperparameters were kept aligned with the reported values for AOCD/NOCD. Batch normalization is applied after the first graph layer, dropout with probability 0.5 is used before each layer, and weight decay is applied with regularization power $\lambda=10^{-2}$. The feature matrix \mathbf{X} is ℓ_2 normalized before the input.

6-3 DropEdge-Based Ablation Protocol

To evaluate whether edge regularization can improve the AOCD encoder, a set of graph transformations was introduced before training. In all cases, the distinguish between:

- $\mathbf{A}_{\text{train}}$: the adjacency used inside sending messages during training, and
- \mathbf{A}_{full} : Reference adjacency is used to calculate completion probability at evaluation checkpoints and for final reporting.

This design follows the "fair comparison" principle: the training neighborhood is affected by change through aggregation, while evaluation is performed against a consistent target proximity. ground line.

Baseline: $\mathbf{A}_{\text{train}} = \mathbf{A}_{\text{full}} = \mathbf{A}$.

Self-loop: Self-loops are added by setting the diagonal of \mathbf{A} to 1, forming $\tilde{\mathbf{A}}$. Then $\mathbf{A}_{\text{full}} = \tilde{\mathbf{A}}$ and $\mathbf{A}_{\text{train}} = \tilde{\mathbf{A}}$. This corresponds to the common normalization step used in GNNs with self-loops.

Random DropEdge: A proportional p of edges are removed uniformly at random from the $\mathbf{A}_{\text{train}}$ (here $p=0.1$). This transformation is used as a

regularizer that reduces dependence on a specific set of edges.

Community-aware DropEdge (proposed):

The proposed community-consistent DropEdge algorithm calculates the feature consistency score for each edge (u,v) using the cosine similarity between the features of the calibrated nodes. It then biases the deletion probabilities so that edges with less similarity are more likely to be deleted. Specifically, the base probability p is modified based on a decreasing function of the calibrated similarity, allowing edges with low similarity to be deleted more effectively. The resulting proximity is used as A_ "train." The rationale for this is that edges connecting dissimilar nodes are more likely to represent community-based or nebulous links that could compro-

mise message aggregation in nested environments. Additionally, the combined variable (the community-consistent DropEdge/self-loop plus randomness) is tested.

6-4 Training

The training process follows AOCD/NOCD implementation. The Adam optimizer is used. Every 50 epochs, the absolute training loss is calculated, and if there is no improvement for the last 500 iterations or after 5000 epochs, training stops. In experiments, training is repeated 50 times with different random seeds for each dataset and setting, and the results are averaged. To improve the reproducibility and clarity of the experimental setup, the primary hyperparameter settings used in all experiments are summarized in Table 2.

Table 2: Hyperparameter settings used within the experiments.	
Hyperparameter	Value
Weight decay	1×10^{-2}
Learning rate	1×10^{-3}
Number of GAT layers	2
Number of attention heads	2
Dropout rate	0.5
Maximum epochs	5000
Early stopping patience	500 iterations
Loss evaluation interval	Every 50 epochs
Batch size	20000
DropEdge probability	0.1
Community assignment threshold (ρ)	0.5

6-5 Assigning nodes to communities

To compare the detected community partitions to the ground truth community structure, the estimated continuous community affiliation **F** is

transformed into a binary community assignment. Each node u is assigned to a community C if its affiliation strength F is above a certain

threshold ρ . The fixed-threshold protocol with $\rho=0.5$.

7- Results and discussion

7-1 Metrics

Normalized Mutual Information (NMI) is chosen to evaluate partitioning performance due to its broad meaning and ability to compare two partitions of different sizes. It is more robust than metrics such as the Jaccard index and F1 score, and has been adopted in NOCD-style evaluation.

7-2 Recovery Performance

Table 3 summarizes the recovery of ground truth communities, measured by NMI (in %), when node features are used as input, and the results are averaged over multiple initializations

(report the mean and standard deviation). Baseline corresponds to AOCD/NOCD-GAT training without a gain decline. Random DropEdge removes edges uniformly at random, while community-aware DropEdge removes edges based on feature similarity. Overall, the proposed community-aware DropEdge variant improves over the AOCD-GAT2 reference on most of the evaluated datasets, with particularly noticeable gains on fb_698, fb_1684, mag_cs, and mag_med. These results suggest that adding community-aware structural regularization can provide a consistent benefit beyond the original attention-based AOCD-GAT2 backbone[7].

Table 3: NMI ground-truth community recovery (%). Node characteristics are input. Average results from 50 initializations. Bold results are best. DNF signifies memory exhausted.

Dataset	AOCD (published GAT2)	Random DropEdge	Community-aware DropEdge (Ours)
fb_348	32.1 ± 2.8	30.7 ± 2.6	32.2 ± 2.1
fb_414	50.4 ± 4.0	50.6 ± 3.8	51.1 ± 3.5
fb_686	17.2 ± 1.3	17.1 ± 1.2	17.7 ± 1.1
fb_698	31.6 ± 4.4	28.7 ± 4.5	35.8 ± 4.2
fb_1912	35.7 ± 2.8	34.4 ± 3.3	36.4 ± 2.6
fb_1684	29.9 ± 2.3	27.9 ± 1.8	31.7 ± 1.8
mag_chem	42.9 ± 2.5	42.7 ± 1.7	43.5 ± 1.9
mag_cs	44.9 ± 3.0	44.2 ± 3.3	46.1 ± 3.4
mag_eng	37.1 ± 3.0	36.8 ± 3.4	38.4 ± 2.9
mag_med	36.0 ± 2.5	35.9 ± 2.6	36.9 ± 2.0

Overall, the proposed community-aware DropEdge variant improves most of the evaluated datasets over the AOCD-GAT2 reference, with fb_698, fb_1684, mag_cs, and mag_med particularly gaining considerably. These effects recommend that incorporating network-aware structural regularization, fully unique interest-based AOCD-GAT2, may provide a regular benefit beyond the spine. Although the enhancements are modest in some data sets, they are constant in the maximum evaluated parameters and are performed without increasing model complexity, highlighting the effectiveness of the proposed lightweight structural regularization.

In addition, the impact of segment accuracy on the basic structural properties of the graph was analyzed to understand the effect of the proposed mitigation method. Specifically, the number of edges has been recorded, graph density, and clustering coefficient before and after applying the random DropEdge method and the proposed DropEdge method that considers community characteristics to two datasets. These records help clarify whether the proposed approach preserves adjacent graph structure well over random side removal.

Table 4:Structural properties of graphs before and after sparsification

Dataset	Method	Edges	Density	Clustering Coefficient
fb_348	Original	3192	0.124440	0.537088
	Random DropEdge	2855	0.111302	0.529125
	Community-aware DropEdge	2895	0.112861	0.530432
fb_414	Original	1693	0.134782	0.632351
	Random DropEdge	1517	0.120771	0.620207
	Community-aware DropEdge	1548	0.123239	0.628614
fb_686	Original	1656	0.115280	0.527511
	Random DropEdge	1486	0.103446	0.519953
	Community-aware DropEdge	1469	0.102262	0.522551
fb_698	Original	270	0.125874	0.677657
	Random DropEdge	245	0.114219	0.670033

	Community-aware DropEdge	246	0.114685	0.674501
fb_1912	Original	30025	0.105486	0.628672
	Random DropEdge	26978	0.094781	0.620479
	Community-aware DropEdge	27288	0.095870	0.621462
mag_chem	Original	157358	0.000251	0.410019
	Random DropEdge	141516	0.000226	0.404704
	Community-aware DropEdge	140591	0.000224	0.405562
mag_cs	Original	96750	0.000401	0.339812
	Random DropEdge	86981	0.000361	0.335732
	Community-aware DropEdge	85953	0.000357	0.335765
mag_eng	Original	49305	0.000443	0.402998
	Random DropEdge	44292	0.000398	0.397273
	Community-aware DropEdge	44005	0.000395	0.397888
mag_med	Original	810314	0.000405	0.411775
	Random DropEdge	729193	0.000364	0.407340
	Community-aware DropEdge	723882	0.000362	0.407652

As is proved in Table 4, each random and mesh-aware sparseness reduces the size of edges and general graph density in all datasets, and the proposed community-aware DropEdge generally keeps slightly better clustering coefficients than random part removal. This indicates that the proposed approach retains higher local community structure even by casting much less informative

edges, which is regulated with the aim of reducing noisy Internet connections. A challenge of the suggested method is its dependence on the informativeness of node properties, while task similarity is employed for manual partitioning. The efficacy of the proposed methodology may diminish in graphs characterized by sensitive trait-community correlations, noisy node characteristics, or absent attributes.

7-3 Scalability and Training Behavior

The training process calculates the absolute loss every 50 epochs and stops early if no improvement is seen for 500 iterations, up to a maximum of 5000 epochs. The protocol is in line with the AOCD/NOCD training approaches. Communicatively, the given change will change the neighborhoods around which GAT will focus during training. Since the coefficients of attention are based on the interactions within such neighborhoods, removing low-similarity edges has the potential to reduce the effect of cross-community links, thus resulting in more uniform coefficients of association before thresholding. This is in line with the assumption that attention attaches weight to nodes in a neighborhood, even as it also proves useful in the event that the neighborhood is also less noisy.

8-Conclusion

The findings support the fact that the model can identify overlapping communities, and the suggested community-aware version of the DropEdge can increase the retrieval performance in the tested datasets. The improvement

is particularly significant in some Facebook ego-networks, meaning that the deletion of low-similarity edges is especially beneficial in smaller graphs where a few noisy linkages can have a significant effect on the aggregation of neighborhoods. Future investigations can be done to understand how to effectively apply edge regularization to the attention-based overlapping community detection frameworks. It would be fascinating to see if attention coefficients and feature-consistency scores can be used to figure out which linkages connect communities. Also, adaptive thresholding can be looked into as a different way to conduct things instead of a fixed $\rho=0.5$ protocol, as long as a consistent validation setting is used.

Several enhancements to the graph attention mechanism have been suggested, and future testing with these variations may yield further advantages. Another way to make things more scalable is to make sparse operations more efficient and do calculations better on more than one attention head.

REFERENCES

- [1] H. Gul, F. Al-Obeidat, A. Amin, and F. Moreira, "Analyzing complex networks: Extracting key characteristics and measuring structural similarities," *Expert Systems*, vol.41(1), p.e13470, 2024.
- [2] A. S. da Mata, "Complex Networks: a Mini-reviewBrazilian Journal of Physics, vol.50(5), pp.658-672, 2020.
- [3] S. Fortunato, "Community detection in graphs," *Physics reports*, 486(3-5), pp.75-174,2010.
- [4] E. A. Abbas, R. Alubady, A. Sahi, M. Diykh, and S. Abdulla, "The Influence Maximization in Complex Networks: Significant Trends, Leading Contributors, and Prospective Directions," *Complexity*, vol. 2025(1), p.7605463, 2025.
- [5] D. Papp, Z. Knoll, and G. Szucs, "Graph construction with condition-based weights for spectral clustering of hierarchical datasets," *Infocommunications Journal*, vol. 12, no. 2, pp. 34–40, Aug. 2020,
- [6] Y. Zhao, K. Deng, X. Liu, and J. Yao, "Overlapping Community Detection Algorithm Based on High-Quality Subgraph Extension in Local Core Regions of Network," *Wireless Communications and Mobile Computing*, 2023(1), p.4988601, 2023.
- [7] K. Sismanis, P. Potikas, D. Souliou, and A. Pagourtzis, "Overlapping community detection using graph attention networks". *Future Generation Computer Systems*, vol.163, p.107529,2025.

- [8] H. Zhang, Z. Yu, G. Dai, G. Huang, Y. Ding, Y. Xie, and Y. Wang. “Understanding gnn computational graph: A coordinated computation”, io, and memory perspective. *Proceedings of Machine Learning and Systems*, 4, pp.467-484, 2022.
- [9] H. Dong, J. Chen, F. Feng, X. He, S. Bi, Z. Ding, and P. Cui, “On the equivalence of decoupled graph convolution network and label propagation”. In *Proceedings of the Web Conference 2021*, pp. 3651-3662, 2021.
- [10] J. Chen, B. Li, and K. He, “Neighborhood convolutional graph neural network”, *Knowledge-Based Systems*, vol.295, p.111861, 2024.
- [11] S. Brody, U. Alon, and E. Yahav, “How Attentive are Graph Attention Networks?,” *arXiv preprint arXiv*, vol.2105.14491, 2021.
- [12] R. van den Berg, T. N. Kipf, and M. Welling, “Graph Convolutional Matrix Completion,” *arXiv preprint arXiv*:1706.02263, 2017.
- [13] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep Graph Infomax,” *arXiv preprint arXiv*, vol.1809.10341, 2018.
- [14] O. Shchur and S. Günnemann, “Overlapping Community Detection with Graph Neural Networks,” *arXiv preprint arXiv*:1909.12201, 2019.
- [15] J. Yang, J. McAuley, and J. Leskovec, “Community Detection in Networks with Node Attributes,” In *2013 IEEE 13th International Conference on Data Mining*, pp. 1151-1156, 2013.
- [16] Y. Rong, W. Huang, T. Xu, and J. Huang, “DropEdge: Towards Deep Graph Convolutional Networks on Node Classification”, *arXiv preprint arXiv*:1907.10903, 2019.
- [17] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol.11(3), p.033015, 2009.
- [18] F. Wu, T. Zhang, A. Holanda De Souza, C. Fifty, T. Yu, and K. Q. Weinberger, “Simplifying Graph Convolutional Networks.” In *International Conference on Machine Learning*, pp. 6861-6871, 2019.
- [19] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Graph contrastive learning with adaptive augmentation,” in *The Web Conference 2021 - Proceedings of the World Wide Web Conference*, pp. 2069–2080, 2021.
- [20] Yang, J. and Leskovec, J., “Overlapping community detection at scale: a nonnegative matrix factorization approach”. In *Proceedings of the sixth ACM international conference on Web search and data mining* ,pp. 587-596, 2013.
- [21] Y. Li, K. He, K. Kloster, D. Bindel, and J. Hopcroft, “Local Spectral Clustering for Overlapping Community Detection,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2), pp.1-27, 2018.
- [22] Z. Sun, B. Wang, J. Sheng, Z. Yu, and J. Shao, “Overlapping community detection based on information dynamics,” *IEEE Access*, vol. 6, pp. 70919–70934, 2018.
- [23] Z. Chen, X. Li, and J. Bruna, “Supervised Community Detection with Line Graph Neural Networks,” *arXiv preprint arXiv*:1705.08415, 2017.
- [24] S. Sobolevsky and A. Belyi, “Graph neural network inspired algorithm for unsupervised network community detection,” *Applied Network Science*, 7(1), p.63, 2022.
- [25] S. Yuan, H. Zeng, Z. Zuo, and C. Wang, “Overlapping community detection on complex networks with Graph Convolutional Networks,” *Computer Communications*, vol. 199, pp. 62–71, 2023.
- [26] A. Todeschini, X. Miscouridou, and F. Carron, “Exchangeable random measures for sparse and modular graphs with overlapping communities,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 82, no. 2, pp. 487–520, 2020.

- [27] J. Mcauley and J. Leskovec, “Discovering social circles in ego networks,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, pp. 1–28, 2014.
- [28] D. Herrmannova and P. Knoth, “An analysis of the Microsoft academic graph,” *D-lib Magazine*, vol. 22, no. 9/10, p. 37, 2016.