Statement of Purpose

Project: Improving Embedding of Attributed Networks using Louvain Algorithm
Team: Amay Dixit (12340220, DSAI), Saurav Gupta (12341940, CSE), Rohit Raghuwanshi (12341820, DSAI), Shashank Yadav (12342010, DSAI)
Supervisor: Prof. Soumajit Pramanik **Course:** UGQ301

Understanding the Problem

We have thoroughly studied the LouvainNE paper and identified a clear research gap. While LouvainNE achieves impressive scalability through hierarchical community detection and multi-level embedding aggregation, it completely ignores node attributes—a critical limitation for real-world networks.

Consider citation networks: two papers may be structurally connected through citations but discuss entirely different topics, while two semantically similar papers may have no direct citation link. Current LouvainNE embeddings capture only the former relationship, missing semantic coherence entirely.

Our insight: The hierarchy construction, meta-graph formulation, and aggregation mechanism in LouvainNE provide natural injection points for attribute information, but the optimal integration strategy remains an open question.

Our Approach: Systematic Investigation

Rather than proposing a single ad-hoc solution, we will conduct a **rigorous comparative study** of four fundamentally different integration paradigms. This systematic approach demonstrates our understanding that research requires controlled experimentation, not just implementation.

Method 1: Structural Reweighting

Key Innovation: Preserve LouvainNE's architecture entirely while encoding attributes into edge weights.

$$A^*_{ij} = \alpha \cdot A_{ij} + (1 - \alpha) \cdot \text{sim}(x_i, x_j)$$

Why this matters: Tests whether attributes can improve embeddings with *zero algorithmic overhead*—maintaining LouvainNE's quasi-linear complexity.

We will implement both edge-preserving (modify only existing edges) and edge-augmenting variants (add k-NN edges in attribute space), analyzing the topology-semantics trade-off.

Method 2: Deep Learning Approach

Key Innovation: Invert the traditional graph embedding paradigm—learn from attributes first, use structure as regularization.

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \cdot \mathcal{L}_{\text{comm}}$$

The community loss uses Louvain partitions as weak supervision, forcing embeddings to respect graph structure without explicitly encoding it.

Why this matters: When attributes are rich (e.g., paper abstracts, user profiles), structure may be secondary. This tests the limits of attribute-first thinking.

We will compare three loss formulations—contrastive loss (pairwise), triplet loss (anchor-positive-negative), and center loss (per-community centroids)—to understand which structural constraint is most effective.

Method 3: Modified Modularity

Key Innovation: Redefine what constitutes a "good" community at the algorithmic core.

$$Q = \lambda \cdot Q_{\text{struct}} + (1 - \lambda) \cdot Q_{\text{attr}}$$

This modifies Louvain's optimization objective itself, ensuring communities are formed based on both edge density and attribute homogeneity from the ground up.

Why this matters: Early fusion at the objective level produces inherently meaningful hierarchies, not post-hoc semantic retrofitting.

We will analyze how joint optimization affects convergence properties and whether semantic communities align with structural communities or create new hierarchical patterns.

Method 4: Late Fusion

Key Innovation: Treat structure and attributes as independent views, combine adaptively.

$$z^{\text{struct}} = \text{LouvainNE}(G), \quad z^{\text{attr}} = f(X)$$
$$z^{\text{final}} = g(z^{\text{struct}}, z^{\text{attr}})$$

Why this matters: Maximum modularity—allows task-specific fusion (classification may need different weights than link prediction).

We will benchmark concatenation, weighted averaging, and learned projection, measuring which fusion strategy generalizes best across tasks.

Evaluation Plan

Datasets: Cora (2.7K nodes, text), Citeseer (3.3K nodes, categorical), BlogCatalog (10K nodes, metadata)—covering different scales and attribute types.

Comprehensive metrics:

- *Quality:* Micro-F1, Macro-F1 (classification); AUC, Precision@K (link prediction)
- *Efficiency:* Wall-clock time, memory footprint
- *Robustness:* Performance under attribute noise, missing attributes, sparse graphs

Ablation studies:

- Vary α, λ, γ systematically
- Test with partial attributes (50%, 75%)
- Compare against non-hierarchical baselines (Node2Vec + attributes, GCN)