Progress Report for Final Course Project

Differentiable Influence Minimization via GNN-Based Surrogates

Mihil Sreenilayam

November 20, 2025

1 Note:

Generative AI tools (including large language models) were used in this project to assist with code development, debugging, documentation, and the preparation of this written progress report. All conceptual work, experimental design, implementation decisions, and analyses were performed by the author, and all generated material was reviewed, verified, and edited to ensure correctness and academic integrity.

2 Project Overview

The goal of my project is to study differentiable methods for Influence Minimization (IM) on graphs, inspired by the DiffIM framework. The classical IM problem requires estimating the spread of influence under models such as the Independent Cascade (IC) model, which is computationally expensive due to the use of repeated Monte Carlo simulations.

My project focuses on two major components:

1. **Surrogate Modeling:** Training a Graph Neural Network (GNN) to approximate the influence function on graphs.
2. **Continuous Relaxation of Edge Removal:** Introducing differentiable edge masks to enable gradient-based optimization of which edges to prune.

3 Progress So Far

3.1 1. Dataset Construction

I implemented a full data-generation pipeline that includes:

- Synthetic graphs: Erdős–Rényi, Barabási–Albert, and Watts–Strogatz models.
- Real-world graphs: Automatically downloaded and parsed from SNAP datasets (Email-Eu-core, Facebook, Wiki-Vote).
- Random perturbations on graphs to diversify structural patterns.
- Monte Carlo estimation of IC influence (with 400–800 simulations per graph).
- Computation of node features (degree, clustering coefficient, and average neighbor degree), normalized per-graph.

This resulted in a dataset of approximately 1400 graphs, with an additional 300 graphs held out for validation.

3.2 2. GNN Surrogate Model

I implemented a 2-layer GCN with a small MLP head to predict normalized influence:

- Trained on the mixed real + synthetic dataset.
- Achieved very strong correlation ($r \approx 0.99$) on held-out synthetic/perturbed graphs.
- Added cross-validation on unseen synthetic graphs to verify generalization.

Performance on real-world graphs (e.g., Email-Eu-core) is weaker, but the overall predictions remain in the correct range (e.g., predicted influence ≈ 0.74 vs. true ≈ 0.81).

3.3 3. Continuous Relaxation Framework

I implemented a differentiable edge-weight relaxation:

- Introduced trainable mask variables $w_e \in (0, 1)$ for each edge.
- Recomputed effective edge weights for the GNN forward pass.
- Optimized w via gradient descent to *reduce predicted influence.*
- Added regularization:
$$\lambda \cdot \mathbb{E}[|w|] \quad + \quad \alpha \cdot \mathbb{E}[(1-w)^2]$$
  which balances sparsity and edge preservation.

Initial experiments show aggressive pruning (sometimes too aggressive), and the effects of λ and α on sparsity are now well-understood.

4 Remaining Work

- **Stabilize the continuous relaxation.** Tune the tradeoff between sparsity and fidelity.
- **Improve generalization to real graphs.** Consider using additional structural features or graph spectral embeddings.
- **Implement discrete edge-removal baselines:**
  - High-degree pruning
  - Betweenness-based removal
  - Random edge deletion

  for comparison.
- **Integrate both modules into a full differentiable IM pipeline:** surrogate → continuous relaxation → discrete pruning.
- **Run large-scale evaluation on SNAP graphs.**

5 Expected Deliverables by End of Semester

My realistic expected accomplishments include:

- A functioning GNN surrogate that significantly reduces the number of required Monte Carlo samples.
- A working continuous relaxation method that identifies influential edges.
- A quantitative comparison to discrete baselines on both synthetic and small real-world graphs.
- A polished final report and a runnable demonstration notebook.

6 Possible Extensions (If Time Allows)

If substantial progress is made earlier than expected, I would like to explore:

- **Alternative relaxations:** Gumbel-softmax, entropy-based masks, or stochastic pruning.
- **Diffusion-aware GNN architectures:** integrating IC parameters directly into message passing.
- **Fairness- or trust-aware influence minimization:** modifying objectives to incorporate weighted diffusion.

7 Summary

So far I have built a complete data pipeline, a strong GNN surrogate model, and an initial implementation of continuous relaxation. The next steps involve refining the relaxation method, running large-scale experiments, and completing the full differentiable influence minimization pipeline.