CentraleSupélec

université PARIS-SACLAY

Graphs Mini Project

Analysis of Social Networks & Email Communication Networks

Lucía Fernandez Sánchez
Alexandra Perruchot-Triboulet Rodríguez

February 2026

Outline

- Datasets
- Project Overview
- Part I: Recap
- Part II: Graph Embeddings & Link Prediction
- Conclusions & Takeaways

Outline

- **Datasets**
- Project Overview
- Part I: Recap
- Part II: Graph Embeddings & Link Prediction
- Conclusions & Takeaways

Datasets

Context: why these 2 datasets?

Contrasting graph types

- Facebook is undirected (mutual friendship).
- Email is directed (asymmetric communication).
- This forces different embedding methods: Node2Vec can’t handle directed edges, GAT can.

Different community strengths

- Facebook modularity = **0.83** → separable communities → Node2Vec provides separable clusters in the embedding space.
- Email modularity = **0.43** → fuzzier boundaries → embeddings will be noisier.

Real ground truth

- Facebook users manually labeled their own friend circles.
- Can test if our approaches match how humans actually group people.

Noisy edges

- Not every email gets a reply.
- GAT’s attention mechanism is designed so that irrelevant connections contribute less to each node’s final embedding.

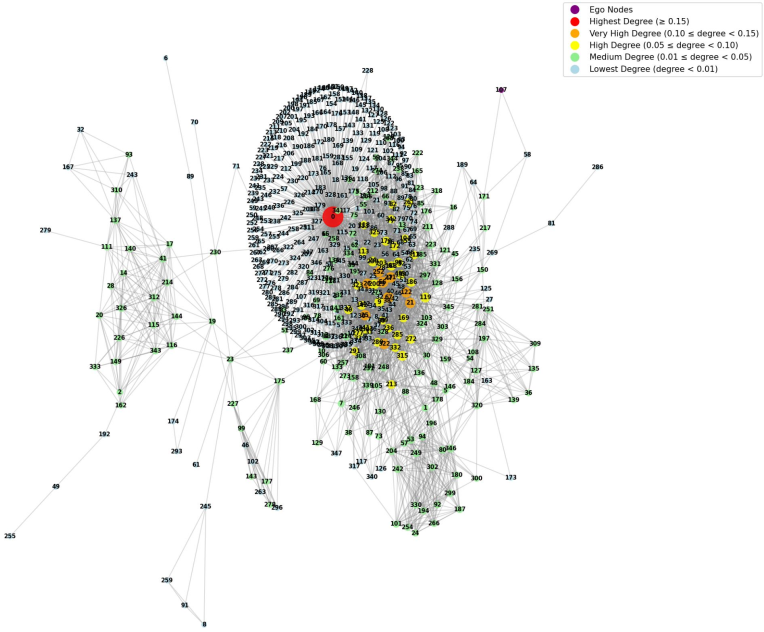
Datasets

Dataset Comparison

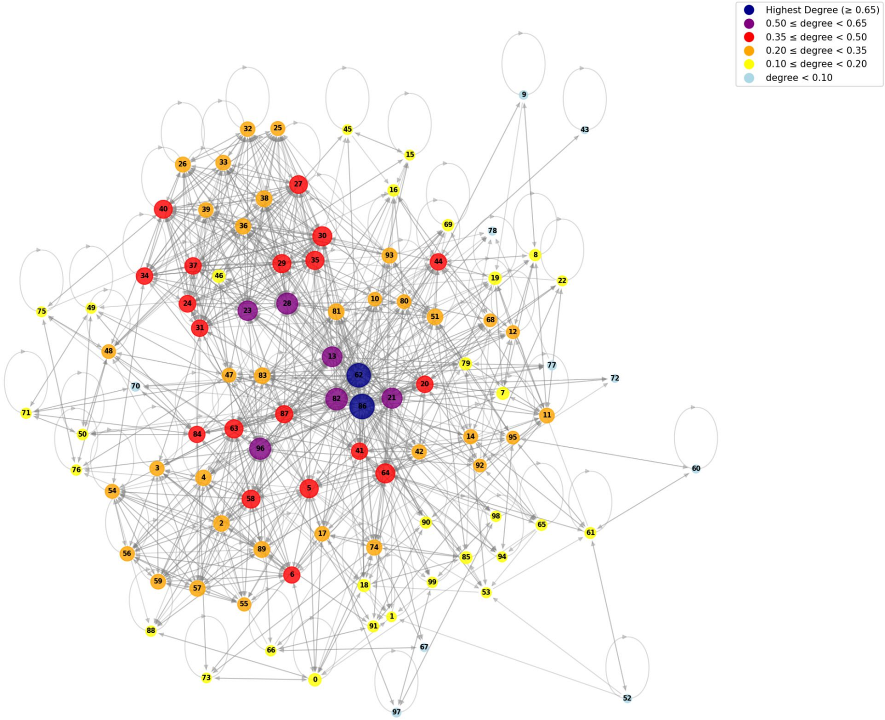
Property	Social Network (Facebook)	Communication Network (Email)
Type	**Undirected**	**Directed**
Nodes	4,039	1,005
Edges	88,234	25,571
Density	0.0108	0.0331
Avg. Clustering	0.606	0.399
Diameter	8	7
Louvain Modularity	**0.83**	**0.43**
Louvain Communities	16	27
Community Intra-Density	**0.96**	**0.62**

Datasets

Dataset Comparison


● Ego Nodes
● Highest Degree (≥ 0.15)
● Very High Degree (0.10 ≤ degree < 0.15)
● High Degree (0.05 ≤ degree < 0.10)
● Medium Degree (0.01 ≤ degree < 0.05)
● Lowest Degree (degree < 0.01)

Network Dataset



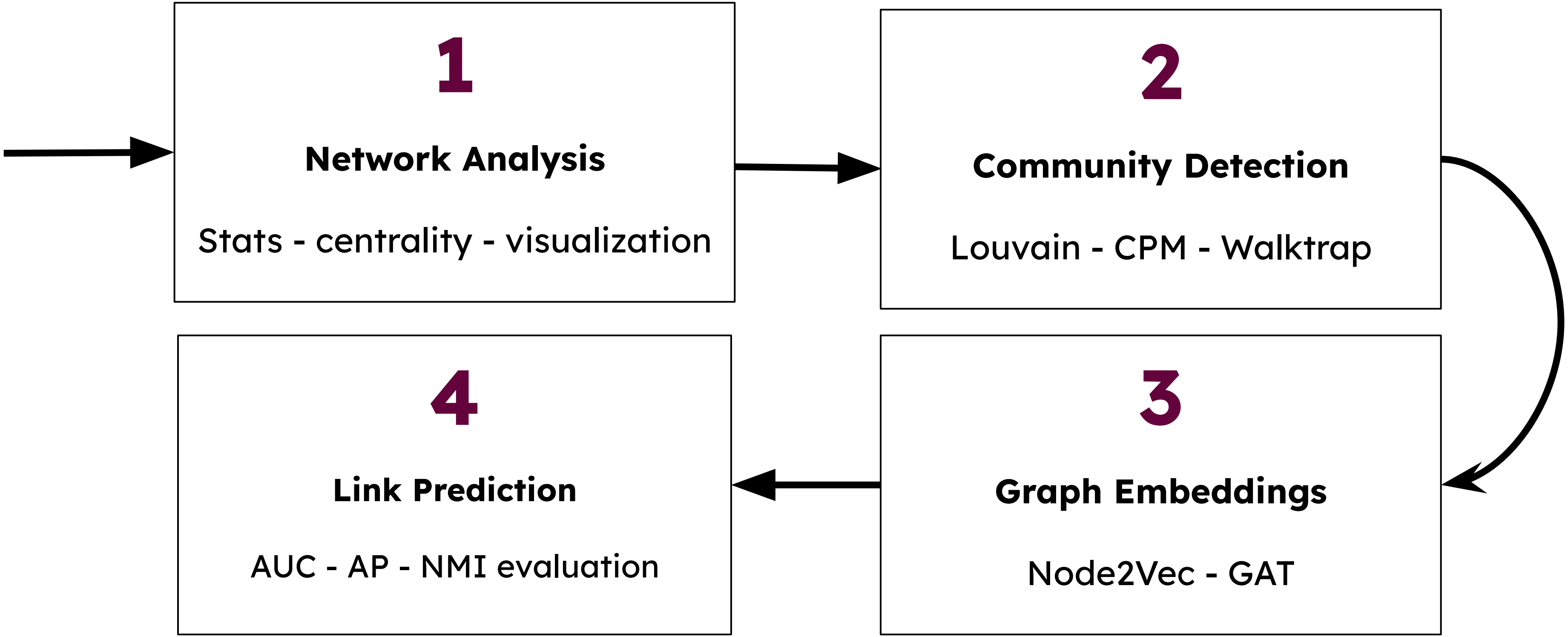
Email Dataset

Outline

- Datasets
- **Project Overview**
- Part I: Recap
- Part II: Graph Embeddings & Link Prediction
- Conclusions & Takeaways

Project Overview

Pipeline



1

Network Analysis

Stats - centrality - visualization

2

Community Detection

Louvain - CPM - Walktrap

3

Graph Embeddings

Node2Vec - GAT

4

Link Prediction

AUC - AP - NMI evaluation

Outline

- Datasets
- Project Overview
- **Part I: Recap**
- Part II: Graph Embeddings & Link Prediction
- Conclusions & Takeaways

Part I: Recap

Community Detection

Louvain

(Modularity optimization)

Facebook:

- **Communities**: 16
- **Modularity**: 0.8349
- **Intra Density**: 0.9609

Email:

- **Communities**: 28
- **Modularity**: 0.4323
- **Intra Density**: 0.596

Walktrap (steps = 2)

(Random walk based)

Facebook:

- **Communities**: 16
- **Modularity**: 0.1049
- **Intra Density**: 0.885

Email:

- **Communities**: 6
- **Modularity**: 0.0081
- **Intra Density**: 0.9874

CPM

(Overlapping communities)

Facebook:

High clustering: CC = 0.61 → overlapping communities → results hard to interpret + hard to compare against the other methods

Email:

- **Communities**: 409
- **Modularity**: 0.0242
- **Intra Density**: 0.7855

High modularity → walks stay within communities → co-occurring nodes are truly similar → similar embeddings → separable communities (in the t-SNE plot).

Note: **Louvain community labels** become the ground truth (NMI reference) for evaluating Part II embeddings.

- **High modularity** in Facebook → Part II **embeddings** will be **clean**.
- **Low modularity** in Email → Part II **embeddings** will be **noisier**.

Outline

- Datasets
- Project Overview
- Part I: Recap
- **Part II: Graph Embeddings & Link Prediction**
- Conclusions & Takeaways

Part II: Graph Embeddings & Link Prediction

Approaches

1. **Node2Vec**

   Facebook Network

2. **GAT (Graph Attention Networks)**

   Email Network

3. **t-SNE + NMI**

   Embedding Evaluation

4. **AUC / AP**

   Link Prediction Metrics

Part II: Graph Embeddings & Link Prediction

Node2Vec - Facebook Network

- Node2Vec handles **undirected** graphs → Facebook Network is chosen.
- Facebook network is a strong fit: well connected + high modularity + high clustering → walks capture friend circles.

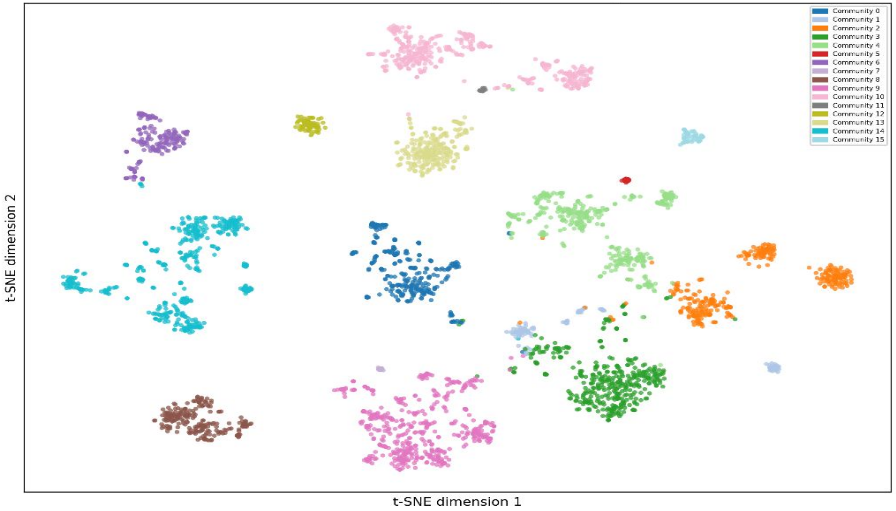
Node2Vec Pipeline

1. **Biased Random Walks on Facebook graph**: simulate walks of fixed length across friendship edges.
2. **Configuration 1: Homophily (p=1, q=0.5) DFS-like walks:** explore far from source (within friend groups).
3. **Configuration 2**: **Structural (p=1, q=2) BFS-like walks:** stay local, capturing structural roles.
4. **Word2Vec:** learns embeddings (32-dim) so nodes appearing together are placed close.
5. **Evaluate embeddings:**
   a. t-SNE visualization.
   b. NMI vs Louvain communities (k = 16).
   c. NMI vs real friend groups (k = 150).
   d. Link prediction.

Node2Vec - 2 Configurations

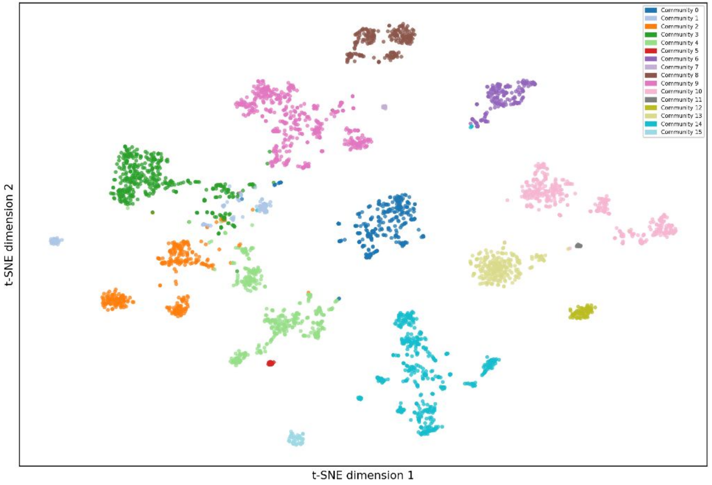
Configuration 1: p = 1, q = 0.5

- **DFS-Like**: explores far from source node.
- **Low q** → walk keeps moving outward, less likely to return.
- Nodes co-occurring in long walks → similar embeddings → larger, more spread blobs.
- t-SNE clusters mirror Louvain communities.
- **NMI vs Louvain: ~0.87**



Configuration 2: p = 1, q = 2

- **BFS-Like**: explores local neighbourhood.
- **High q** → walk often returns close to starting node.
- Nodes with similar local structure→ similar embeddings → compact blobs.
- t-SNE clusters reflect structural patterns, not community labels.
- **NMI vs Louvain: ~0.85**



Node2Vec - NMI Results & Louvain Community Alignment

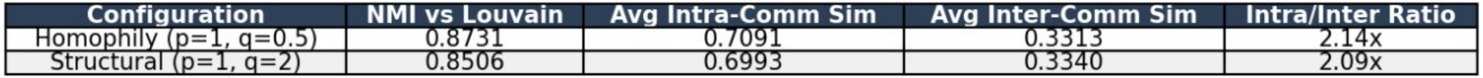
NMI: measures alignment between two different partitions.

1 = perfect match, **0** = no agreement.

We do **3 comparisons:**

1. **Node2Vec vs Louvain:** agreement between 16-Means clusters of Node2Vec embeddings vs Louvain clusters.
2. **Node2Vec vs circles:** agreement between 150-Means clusters on Node2Vec embeddings vs user-defined friend circles.
3. **Louvain vs circles:** pure graph partition vs user-defined friend circles.

Node2Vec vs Louvain

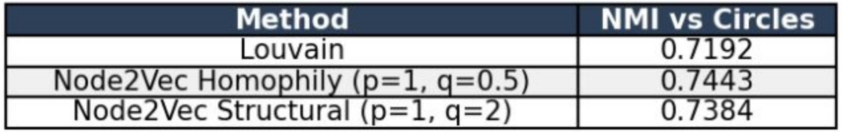


Configuration	NMI vs Louvain	Avg Intra-Comm Sim	Avg Inter-Comm Sim	Intra/Inter Ratio
Homophily (p=1, q=0.5)	0.8731	0.7091	0.3313	2.14x
Structural (p=1, q=2)	0.8506	0.6993	0.3340	2.09x

- (p = 1, q = 0.5) configuration performs slightly better.
- **Nodes within the same community** are **more similar** to each other **in the DFS-Like configuration**.
- **DFS** walks tend to **stay within dense connected regions**, same regions **Louvain identifies as communities** → higher NMI alignment.

Node2Vec and Louvain vs Circles

Node2Vec and Louvain vs Circles



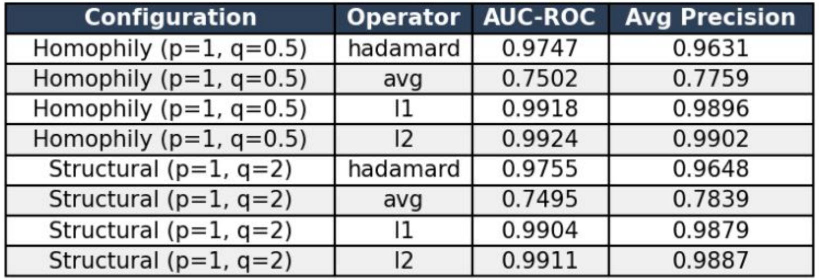
Method	NMI vs Circles
Louvain	0.7192
Node2Vec Homophily (p=1, q=0.5)	0.7443
Node2Vec Structural (p=1, q=2)	0.7384

- There is a total of 150 friend circles in the network vs 16 Louvain communities → algorithms must capture more specific, detailed groupings.
- **Louvain scores lowest** since 16 broad groups cannot capture 150 specific groups.
- No method reaches 1.0 since **real circles overlap** (same person can be in multiple circles).
- The **3 methods produce non-overlapping assignments.**

Node2Vec - Link Prediction

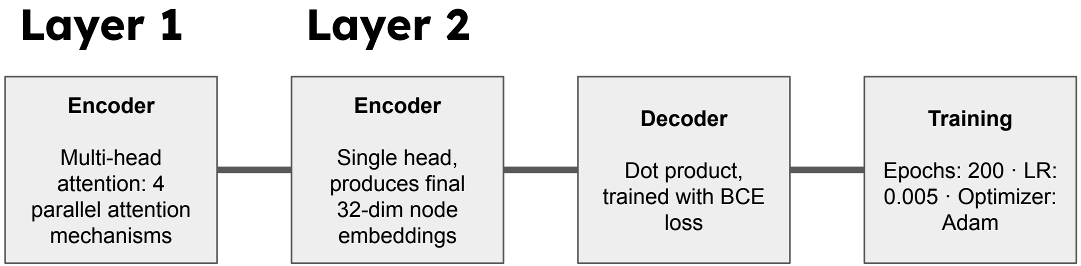
Link Prediction Pipeline

1. **Train/Test Edge Split**: hide 20% of edges are test set.
2. **Negative Sampling:** sample equal number of non-existing pairs: same number of friends vs non-friends.
3. **Edge Feature Construction:** combining each node embedding pair (u,v) into one vector using 4 operators: Hadamard (dot-product), average, L1-dist, and L2-dist.
4. **Logistic Regression:** linear classifier trained on edge features. Evaluated with **AUC-ROC: 1.0** = perfect, **0.5** = random.



Configuration	Operator	AUC-ROC	Avg Precision
Homophily (p=1, q=0.5)	hadamard	0.9747	0.9631
Homophily (p=1, q=0.5)	avg	0.7502	0.7759
Homophily (p=1, q=0.5)	l1	0.9918	0.9896
Homophily (p=1, q=0.5)	l2	0.9924	0.9902
Structural (p=1, q=2)	hadamard	0.9755	0.9648
Structural (p=1, q=2)	avg	0.7495	0.7839
Structural (p=1, q=2)	l1	0.9904	0.9879
Structural (p=1, q=2)	l2	0.9911	0.9887

GAT - Email Network



Layer 1

Layer 2

Encoder

Multi-head attention: 4 parallel attention mechanisms

Encoder

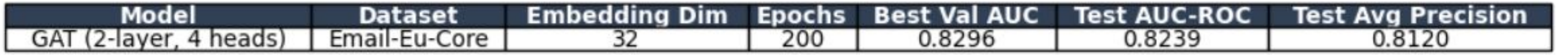
Single head, produces final 32-dim node embeddings

Decoder

Dot product, trained with BCE loss

Training

Epochs: 200 · LR: 0.005 · Optimizer: Adam



Model	Dataset	Embedding Dim	Epochs	Best Val AUC	Test AUC-ROC	Test Avg Precision
GAT (2-layer, 4 heads)	Email-Eu-Core	32	200	0.8296	0.8239	0.8120

Directed edges: GCN uses symmetric normalisation (ignores direction), GAT computes per-edge attention naturally preserving asymmetry.

Noisy Connections: Not all emails are equally meaningful, GAT learns attention weights that down-weight less relevant connections. GCN treats all neighbours equally.

Weaker Communities: (Email modularity around 0.43) subtler boundaries require selective attention to identify which neighbours carry community signal. Uniform aggregation would blur these boundaries further.

Part II: Graph Embeddings & Link Prediction

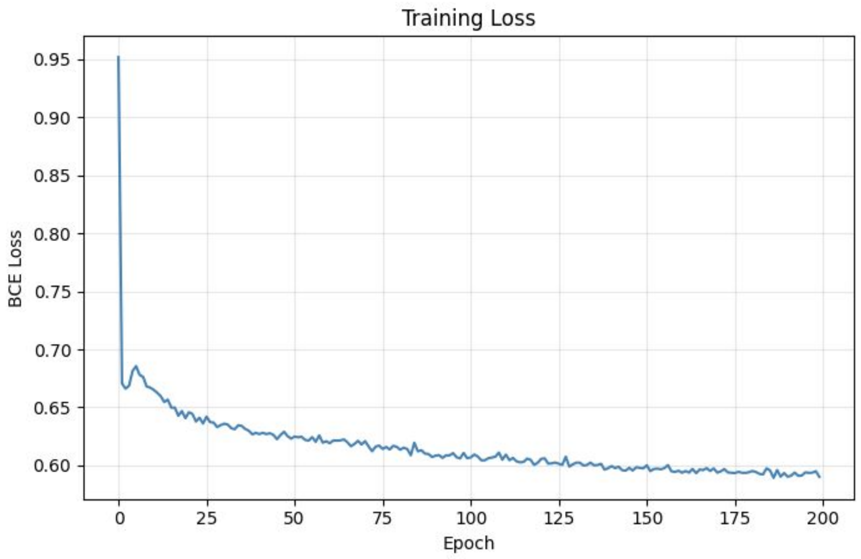
GAT - Email Network

Training Curves (200 epochs)

Training Loss (BCE):

- Decreases steadily from ~0.7 (converges).
- Confirms model is learning to distinguish real edges from non-edges.

Data Description	Configuration
Training Edges	80% of real edges + equal neg samples
Validation Edges	10% (monitored during training)
Test Edges	10% (strictly held-out, used once)
Node Features	3D: in-degree, out-degree, total degree (normalised)



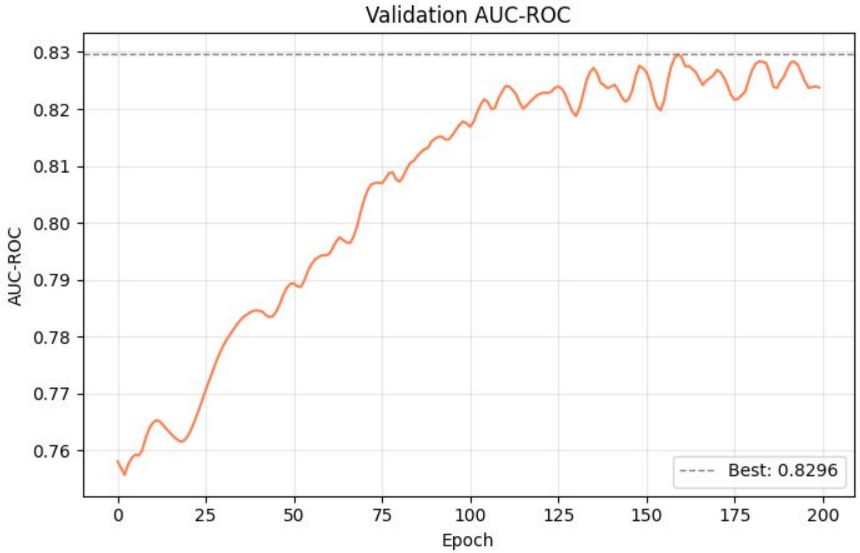
Part II: Graph Embeddings & Link Prediction

GAT - Email Network

Training Curves (200 epochs)

- Rises from 0.5 and converges near best value.
- Dashed line marks best checkpoint.
- Model is restored to that point for final evaluation.

- High precision on edges



AUC >> 0.5

GAT learned genuine graph patterns, not random guessing

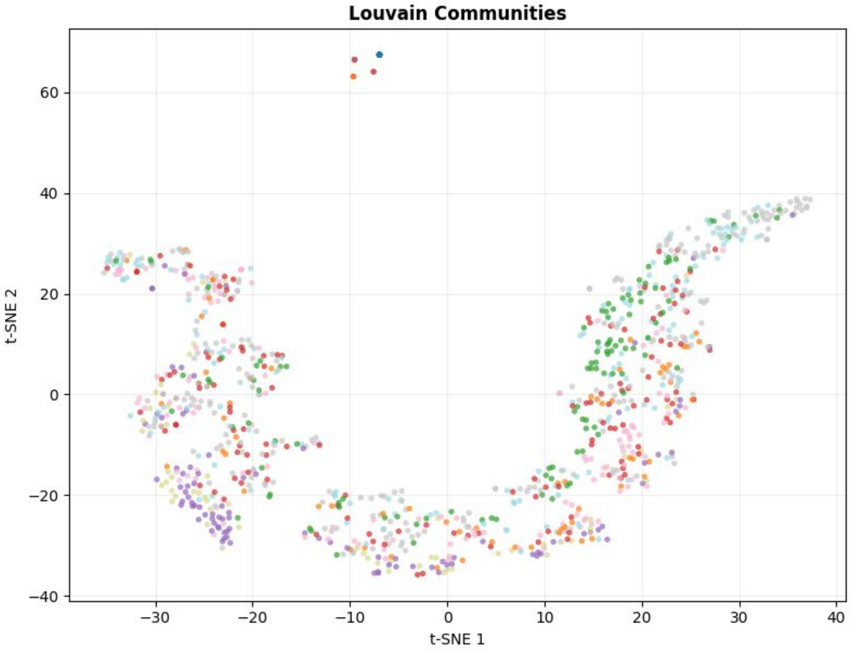
Part II: Graph Embeddings & Link Prediction

GAT - Email Network

t-SNE Plot (Louvain)

- The GAT embeddings are projected to 2D with t-SNE and nodes are coloured by their Louvain community label.

- If nodes of the same colour cluster together, it means GAT has implicitly learned the community structure.



Less clean than Facebook's Node2Vec clusters:

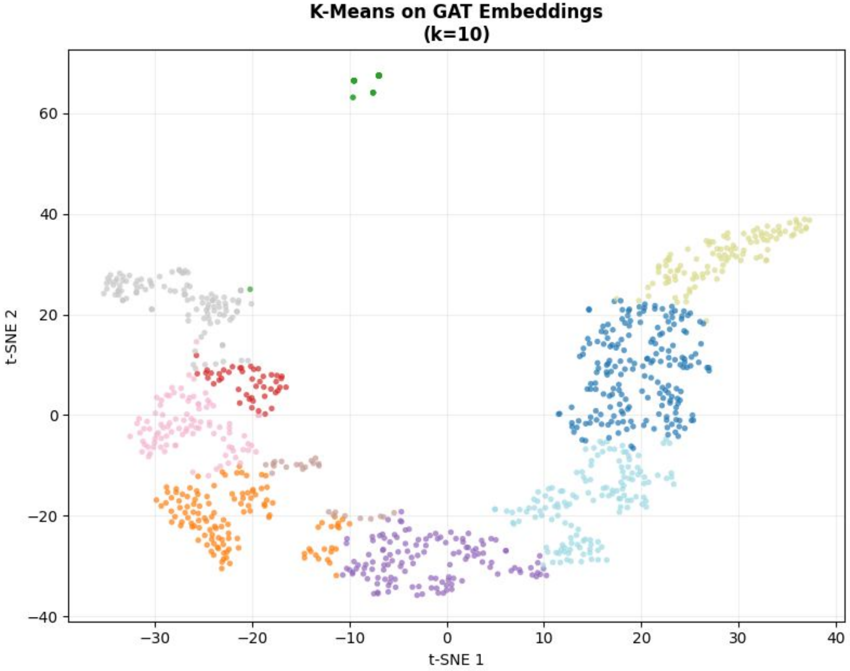
1. email graph is directed and noisier
2. lower modularity means community boundaries are fuzzier
3. many one-way communications with no reply.

Part II: Graph Embeddings & Link Prediction

GAT - Email Network

Normalized Mutual Information (NMI): K-Means

- NMI review whether the geometry of GAT embeddings agrees with community structure.
- A high score means the model implicitly learned to encode community membership



1. Extract 32D GAT embeddings after training
2. Run K-Means with k = number of Louvain communities
3. Compute NMI between K-Means cluster assignments and Louvain community labels

Part II: Graph Embeddings & Link Prediction

Embedding Evaluation

Property	Node2Vec	GAT
Graph Type	Undirected	Directed
Embedding Type	General-purpose (reusable)	Task-specific (link prediction)
Walk Strategy	Biased random walks (p, q)	Attention over neighbours
Direction handled	No (treats all edges equally)	Yes (per-edge attention weights)
Noisy handled	No (uniform walk probability)	Yes (learns to down-weight noise)
Link Prediction	0.97–0.99 (Logistic Reg)	> 0.5 (end-to-end trained)
NMI vs Louvain	~0.86–0.89 (High)	Moderate (weaker graph structure)
t-SNE clusters	Tight, well-separated (mod 0.83)	More diffuse (mod 0.43, noisy)
Best Edge Operator	L1 / L2 (distance-based)	Dot product
Chosen Strategy	All 6 suitability criteria met	Directed + noisy + small graph

Outline

- Datasets
- Project Overview
- Part I: Recap
- Part II: Graph Embeddings & Link Prediction
- **Conclusions & Takeaways**

Conclusions & Takeaways

Conclusions

Conclusions:

- Facebook → Node2Vec: good fit for undirected graph with strong community structure.
- Both Node2Vec configurations align well with Louvain.
- Link prediction: L1/L2 operators outperform Hadamard → embedding distance is more informative than element-wise product.
- Email → GAT: handles directed edges, learns which neighbours actually matter, no sampling needed.
- GAT trained end-to-end for link prediction.

Key Takeaways:

- **No single best method:** choice depends on graph properties.
- **Node2Vec:** general-purpose embeddings, best for undirected + strong community structure.
- **GAT:** task-specific, best for directed + noisy edges.
- **High modularity:** cleaner embeddings (Facebook > Email).
- Node2Vec embeddings are reusable across tasks; GAT optimises directly for the downstream task.

THANK YOU

Questions ?