IJCNN 2023

Interaction-Focused Anomaly Detection
on Bipartite Node-and-Edge-Attributed Graphs

Rizal Fathony*
Grab, Indonesia

Jenn Ng
Grab, Singapore

Jia Chen
Grab, Singapore

*Presenter

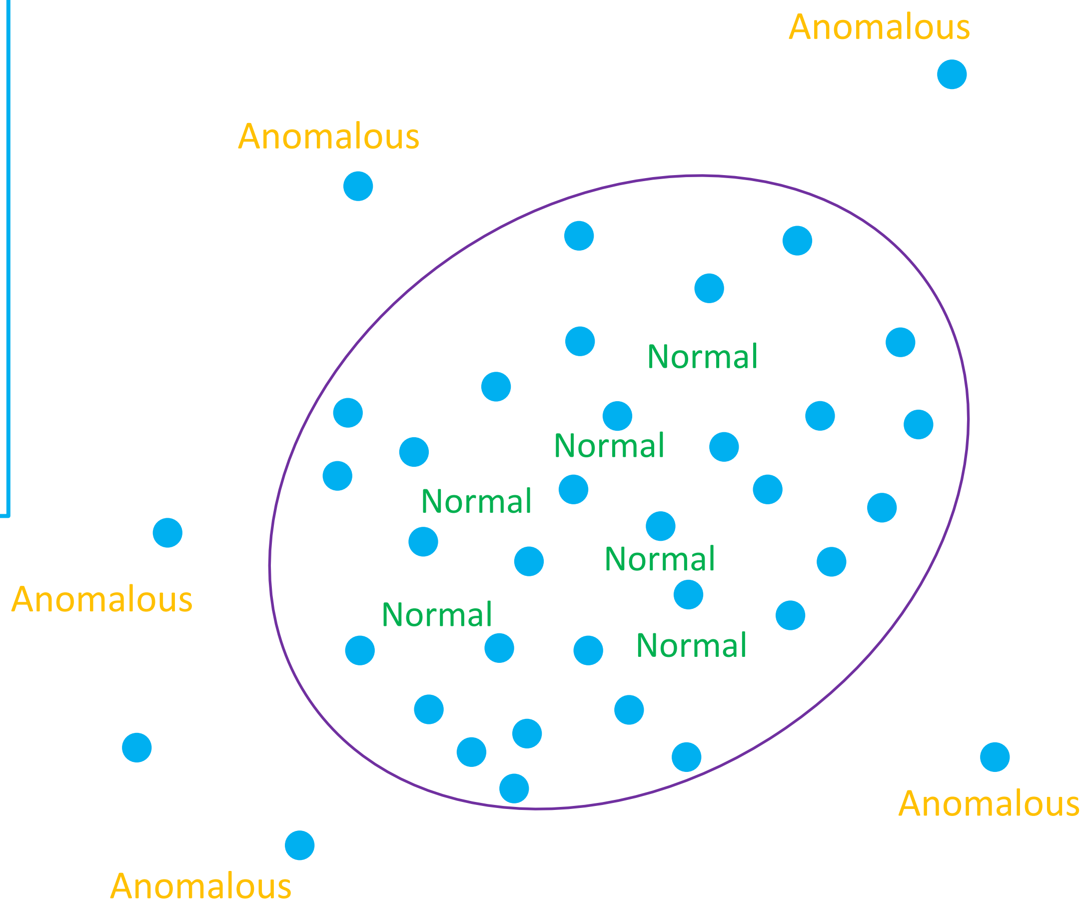
Introduction

Motivations of our study

What is Anomaly Detection?

→ Anomaly Detection

is the process of identifying unexpected observations in datasets, which deviate significantly from the majority of the data.



Why Anomaly Detection is Important?

Anomalous behaviors

Serious implications

Financial Institutions

Anomalous Transactions →

- Stolen Credit Cards
- Money Laundering

Computer Networks

Anomalous Traffic →

- Security Breach
- Network Intrusions

E-Commerce

Anomalous Purchase →

- Fraudulent Transactions
- Fake Reviews

Why **Unsupervised** Anomaly Detection?

Anomaly Detection (AD)

Usually done without label supervision
(unsupervised learning)

Label Availability Issues

- Anomaly events are rare.
- Labeling is sometimes expensive (domain expert are needed).

Adversarial Fraudsters

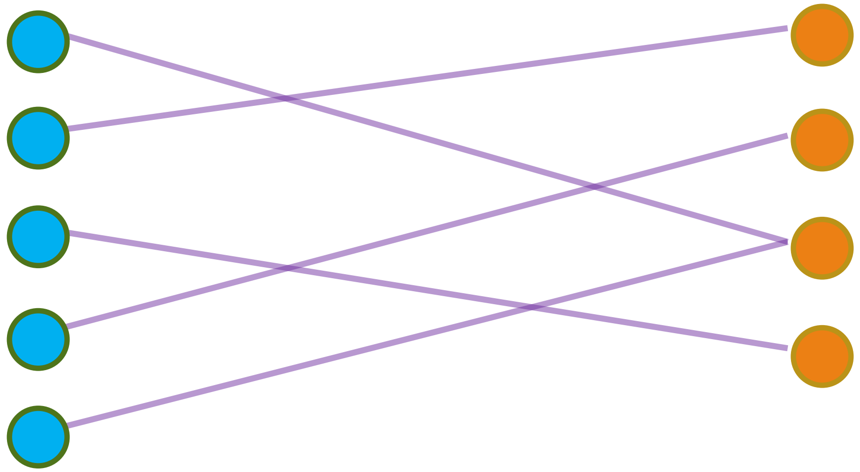
- Fraudsters are incentivized to adversarially innovate their methods of conducting fraud.
- Supervised learning model that rely on historical labels unable to detect new types of fraud.

Why (Bipartite) Graphs?

Bipartite Graphs

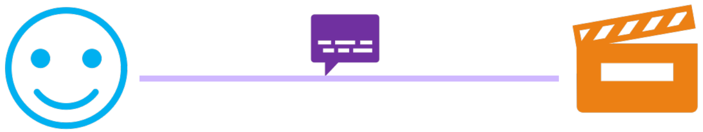
Main abstraction for modeling interactions between two groups.

Often referred as:
user–interaction–item graphs



E-commerce
consumer–purchase–product graph

Movie Streaming Platform
user–review–movie graph



Network Monitoring
ip_address–request–server graph

Why Node-and-Edge Attributed Graphs?

Real World Bipartite Graphs

→ **Rich of information**

in both the entity (**node**)

and the interaction (**edge**)

→ **Example:**

consumer–purchase–product graph

consumer node:
- consumer profile
- historical preference, etc.

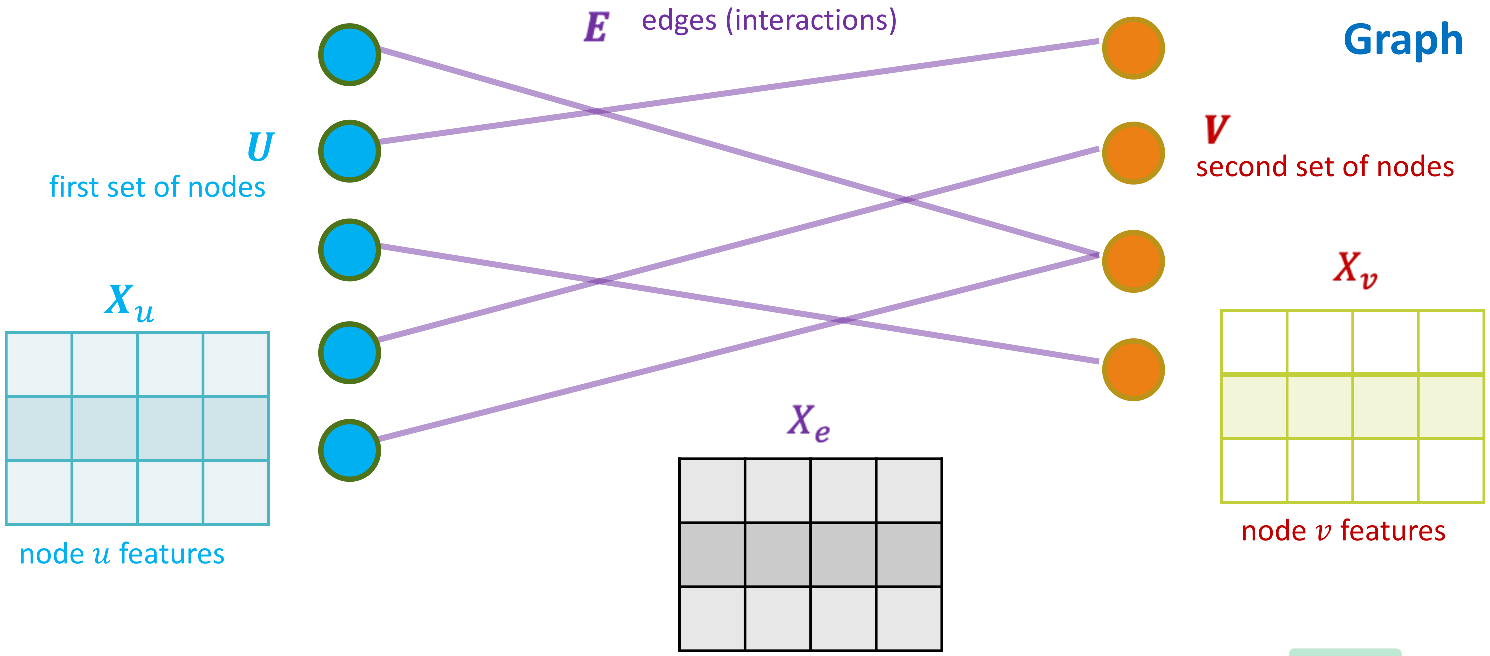
product node:
- product description
- product category, etc.

purchase interaction:
- price paid
- payment method
- rating, review, etc.

E-commerce

consumer–purchase–product graph

Attributed Graph



$G = \{U, V, E, X_u, X_v, X_e\}$

graph

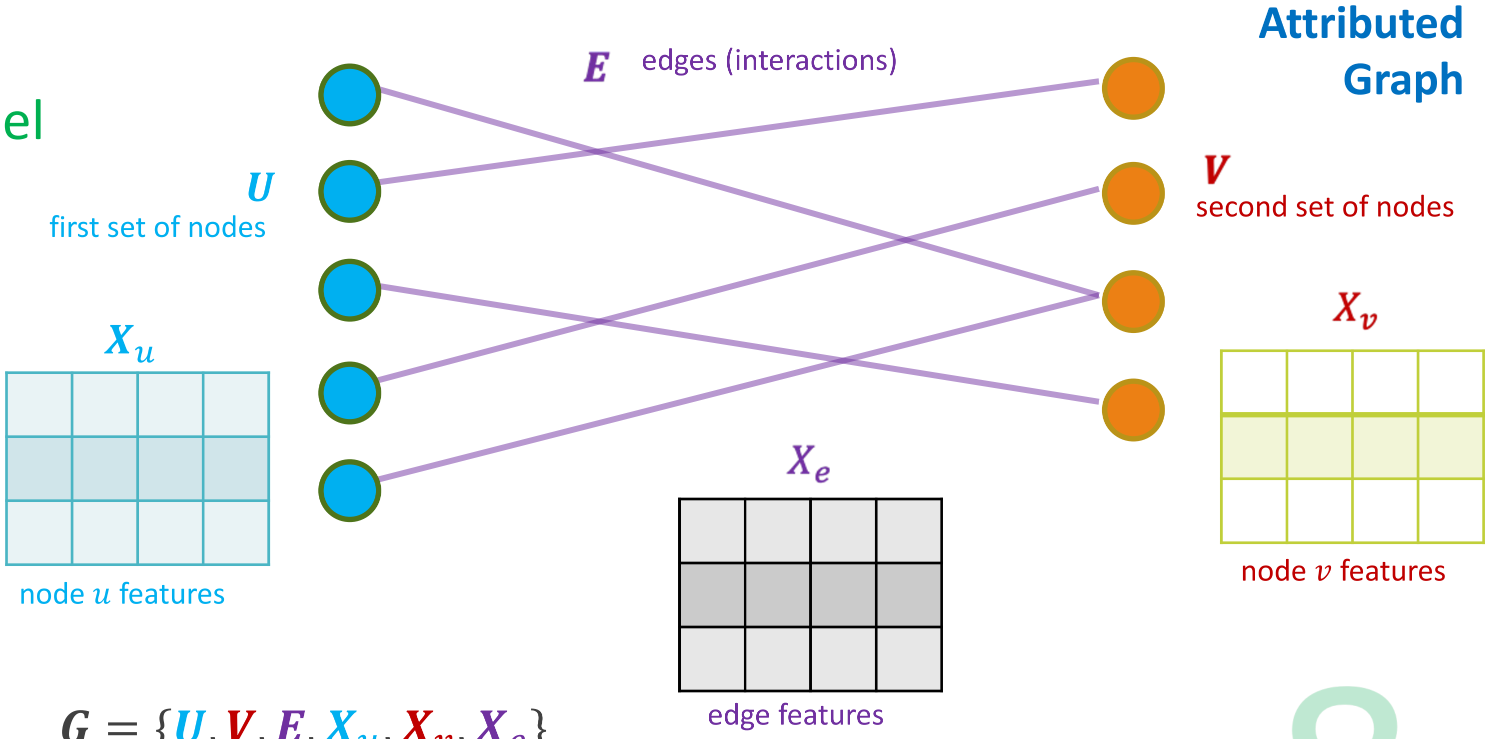
Anomaly Detection on
Bipartite Node-and-Edge-Attributed Graphs

High-performing
anomaly detection system

→ All rich information in the graph need to be considered by the model

→ Capable to detect **entity** (node-level) anomaly, and **interaction** (edge-level) anomaly

→ Our work!



$$G = \{U, V, E, X_u, X_v, X_e\}$$
graph

Related Works

Unsupervised Anomaly Detection
using Graph Neural Networks (GNN)

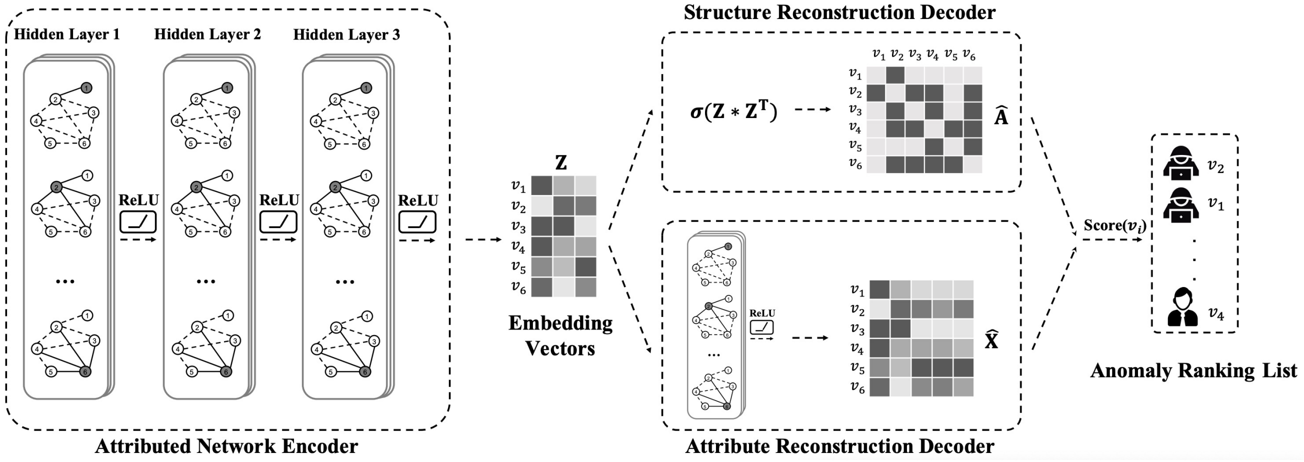
Anomaly Detection using Graph Neural Networks (GNN)

GNN models for unsupervised anomaly detection on attributed graphs

→ Motivated by the success of GNN architectures
for supervised and semi-supervised learning

1) DOMINANT (Ding, et.al; 2019)**:**

- One of the first GNN anomaly model
- Autoencoder-like architecture



2) AnomalyDAE (Fan, et.al; 2020)**:**

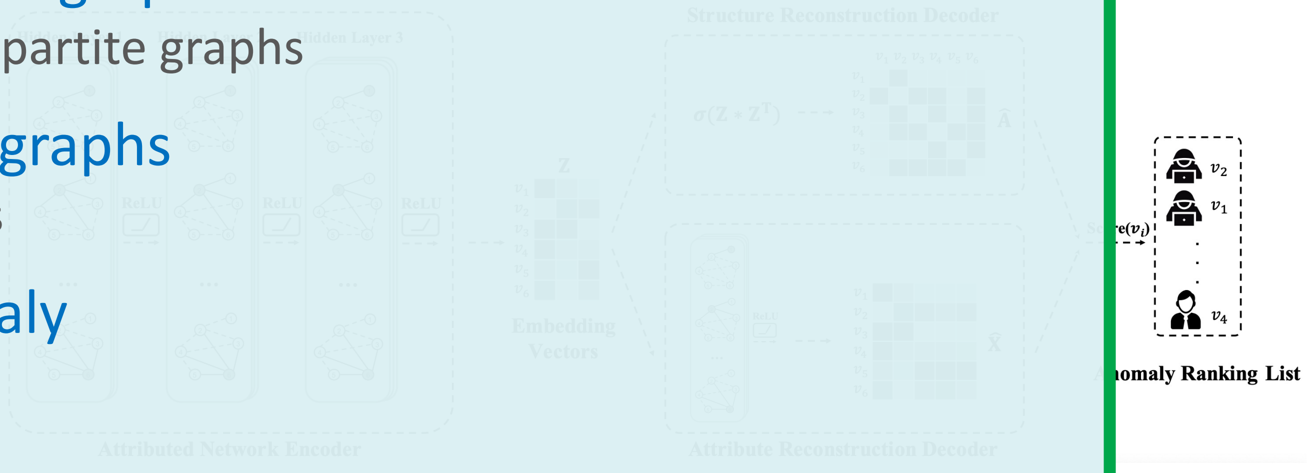
- Extend the architecture with two decoders:
  Structure Decoder and Attribute Decoder

3) Other models:

AdOne (Bandyopadhya, et.al; 2020)
CONAD (Xu, et.al; 2022)
etc…

Anomaly Detection using Graph Neural Networks (GNN)

GNN models for unsupervised anomaly detection on attributed graphs



Shortcomings of the previous works:

- Can only handle homogeneous graphs
  No clear guide on extending it to bipartite graphs
- Only handle node-attributed graphs
  Cannot incorporate edge attributes
- Only detect node-level anomaly
  No edge-level anomaly detection
- Scalability issues
  Cannot handle large-sized graphs

Our Approach

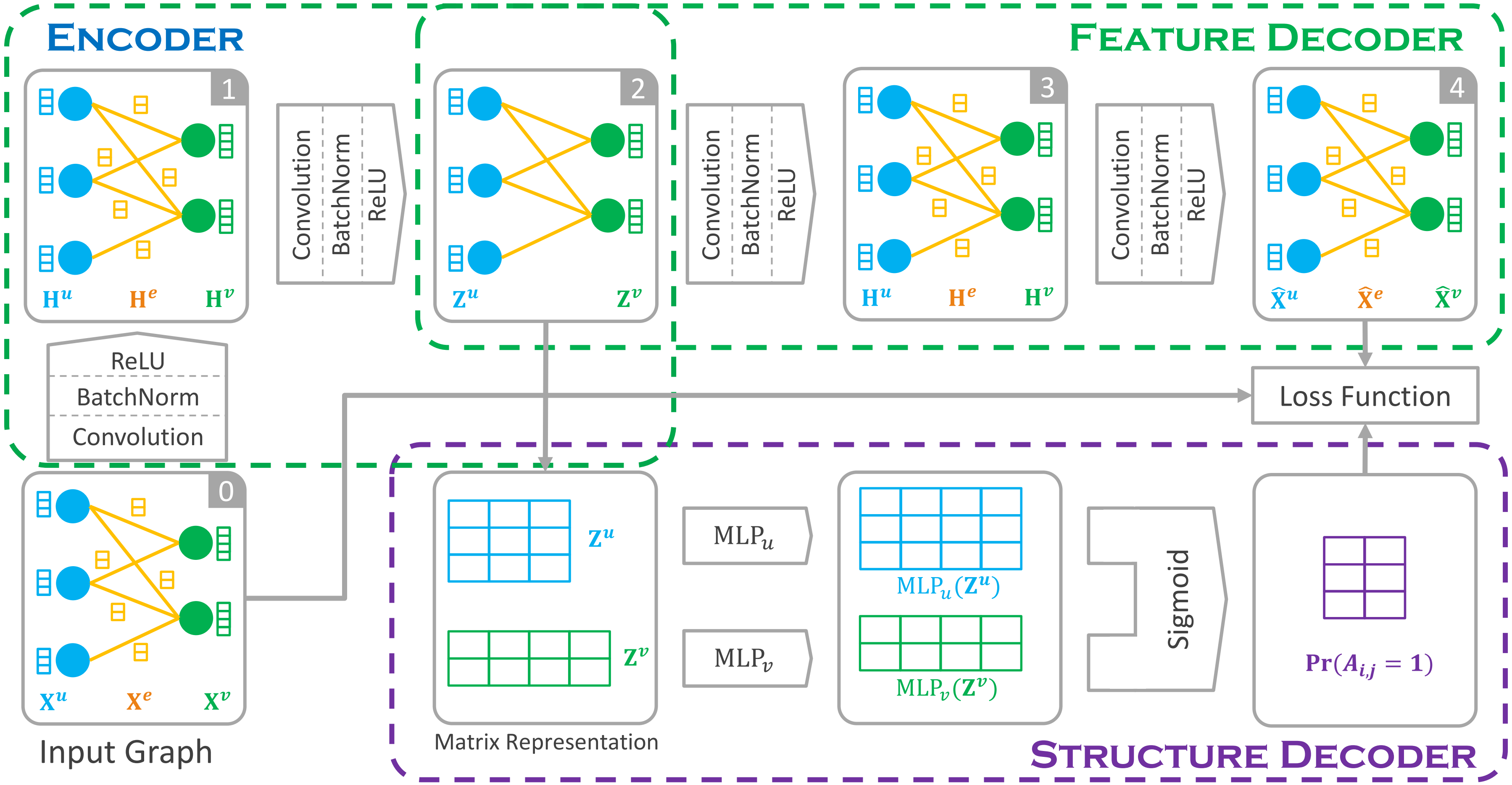
Our GNN architecture

Architecture Overview

GraphBEAN

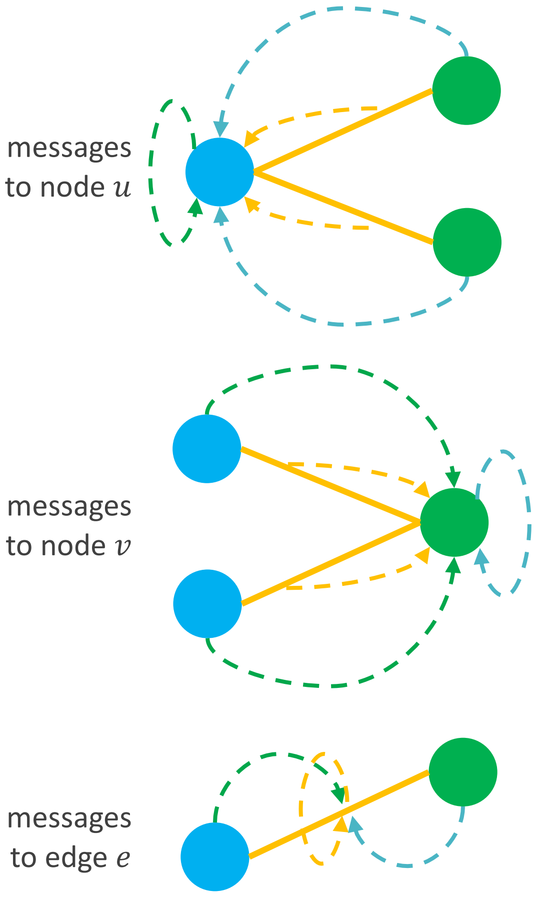
Bipartite
Node-and-**E**dge-
Attributed
Networks

An autoencoder-like model

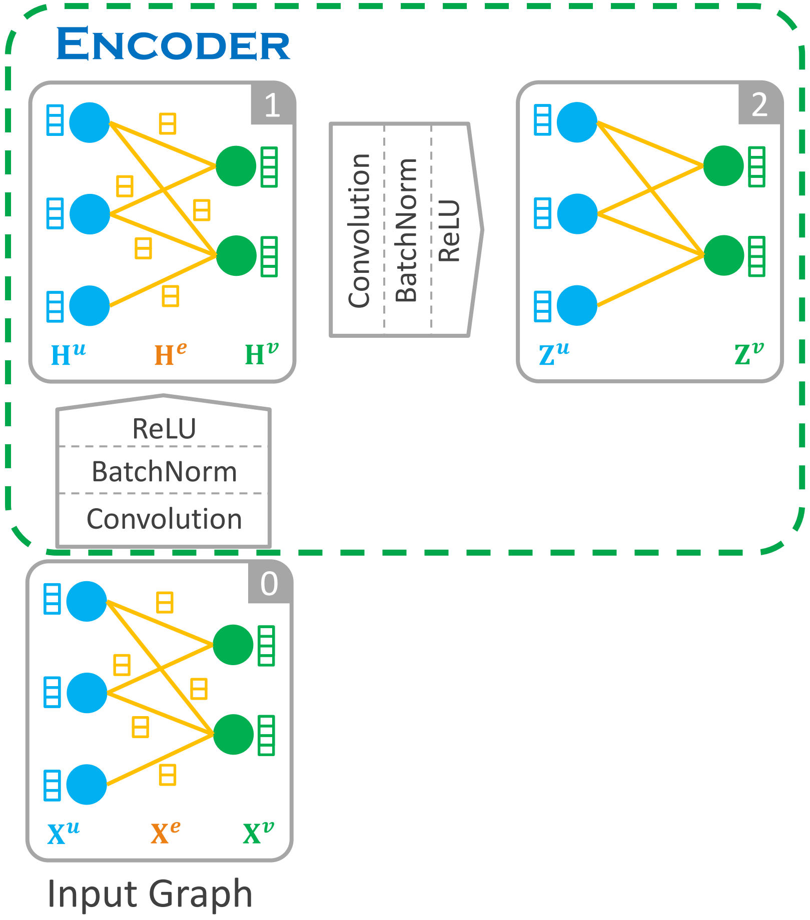


GraphBEAN with 4 layers

Architecture Overview



Message passing flow



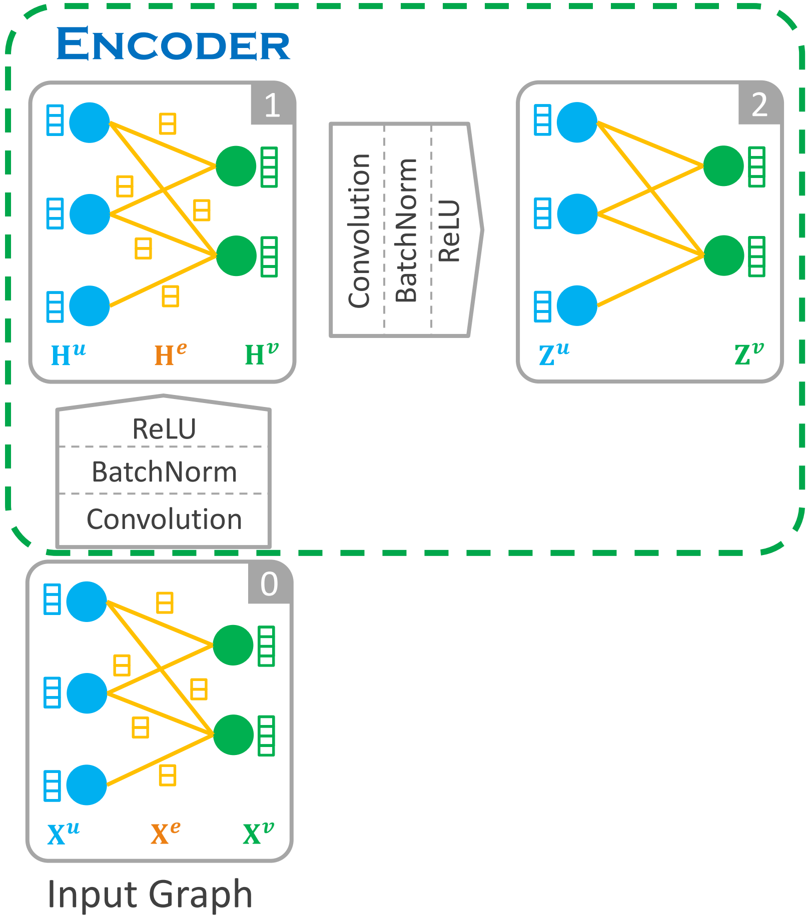
GraphBEAN with 4 layers

Architecture Overview

GraphBEAN

Bipartite
Node-and-**E**dge-
Attributed
Networks

An autoencoder-like model



Latent representation (node only)

No edge latent

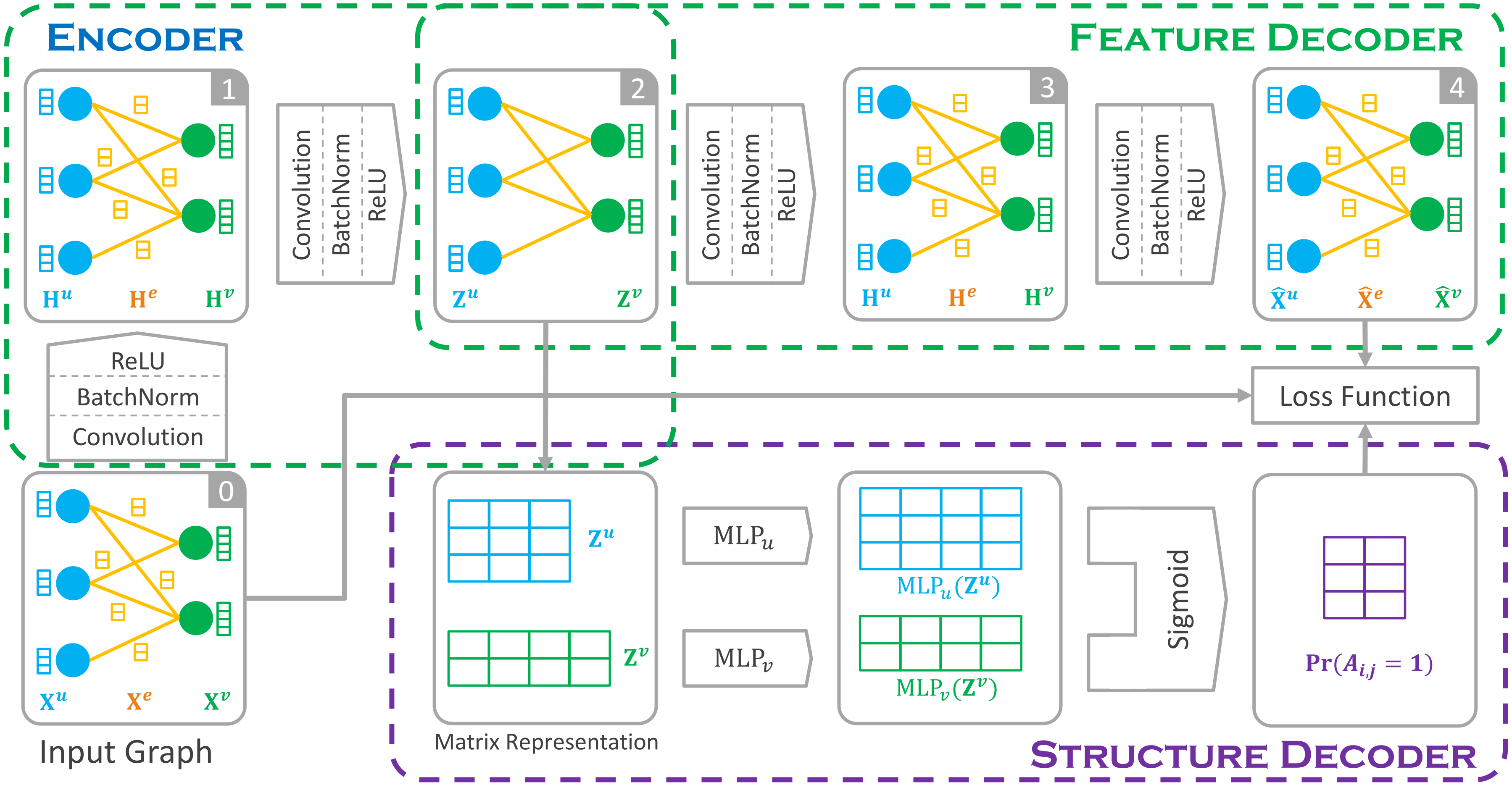
GraphBEAN with 4 layers

Architecture Overview

GraphBEAN

Bipartite
Node-and-**E**dge-
Attributed
Networks

An autoencoder-like model



GraphBEAN with 4 layers

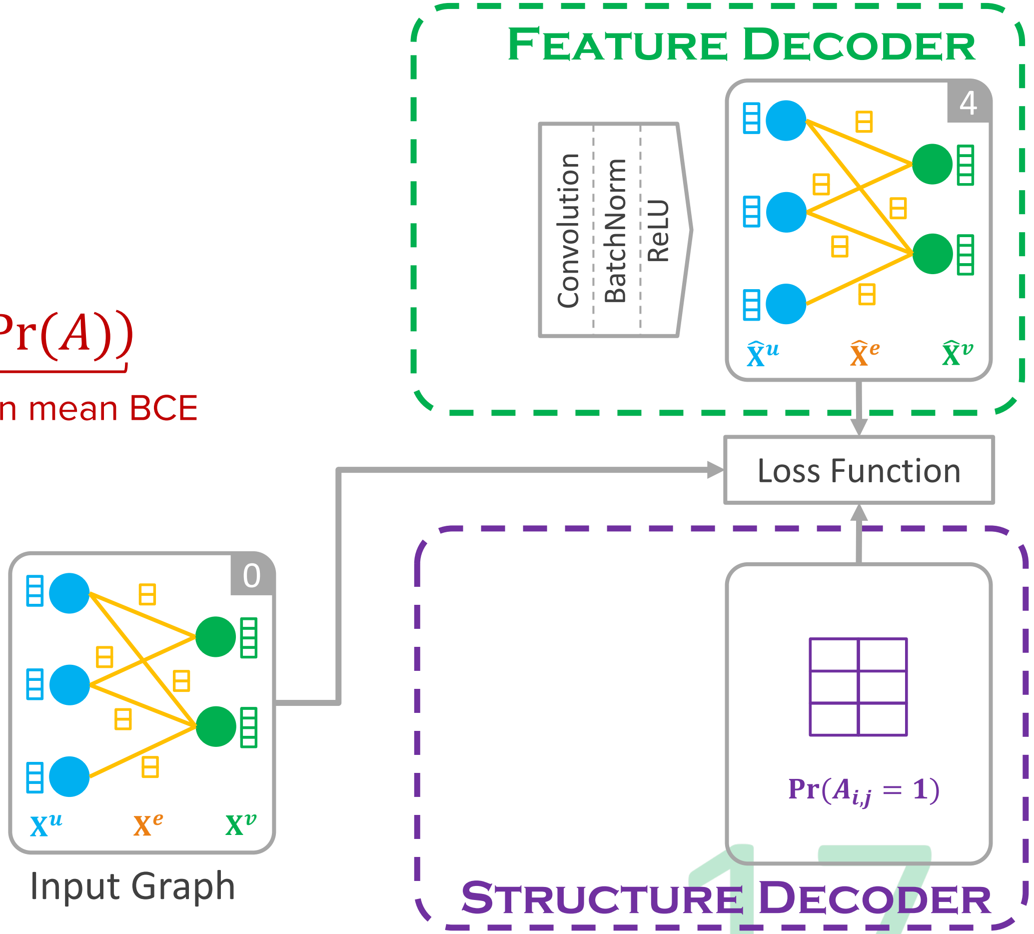
Loss Function

Our optimization objective:

→ Reconstruction Loss

- feature reconstruction error (MSE) of the feature decoder
- edge prediction error (BCE) of the structure decoder

$$\min \underbrace{MSE(X_u, X'_u)}_{\text{MSE of nodes in } U} + \underbrace{MSE(X_v, X'_v)}_{\text{MSE of nodes in } V} + \underbrace{MSE(X_e, X'_e)}_{\text{MSE of the edges}} + \underbrace{\eta BCE(A, \Pr(A))}_{\text{edge prediction mean BCE}}$$

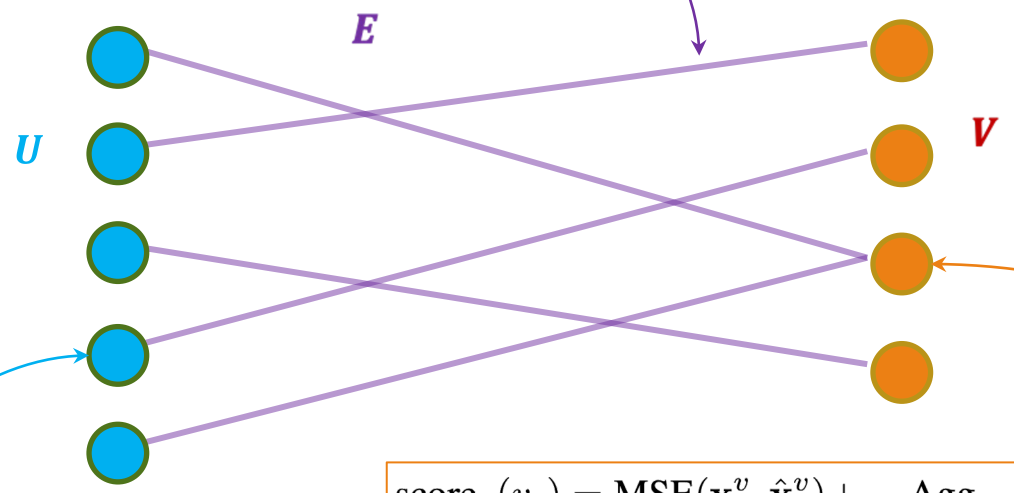


Anomaly Score Construction

Reconstruction-based anomaly score

Normal behaviors : common ➜ can be easily reconstructed
Anomalous behaviors : rare ➜ cannot be reconstructed easily

$$\mathrm{score}_e(e_{i,j}) = \mathrm{MSE}(\mathbf{x}^e_{i,j}, \hat{\mathbf{x}}^e_{i,j}) + \eta \mathrm{BCE}(e_{i,j})$$



Edge-Level Anomaly Score

edge reconstruction error

Node-Level Anomaly Score

node features reconstruction error

+ aggregate over edge scores

$$\mathrm{score}_v(v_j) = \mathrm{MSE}(\mathbf{x}^v_j, \hat{\mathbf{x}}^v_j) + \operatorname*{Agg}_{e_{i,j} \in \mathcal{M}(v_j)} \mathrm{score}_e(e_{i,j})$$

Aggregate operator:
max or **mean**

$$\mathrm{score}_u(u_i) = \mathrm{MSE}(\mathbf{x}^u_i, \hat{\mathbf{x}}^u_i) + \operatorname*{Agg}_{e_{i,j} \in \mathcal{M}(u_i)} \mathrm{score}_e(e_{i,j})$$

Experiments

Model evaluation

Datasets

Wikipedia
contributor–edit–wikipage graph

Reddit
user–post–subreddit graph

Amazon | Finefoods category
Amazon | Movies category
consumer–review–product graph

Datasets with various characteristics

Regular size datasets

Large size dataset

Anomaly ratio (edge): 0.2% - 3.7%

Injection: - topological structure anomaly
- attributes anomaly

DATASET PROPERTIES

Dataset	#node $\mathcal{U}$	#node $\mathcal{V}$	#edge	$\mathcal{U}$ deg.	$\mathcal{V}$ deg.	#ft. $\mathcal{U}$	#ft. $\mathcal{V}$	#ft. $\mathcal{E}$	+ratio $\mathcal{U}$	+ratio $\mathcal{V}$	+ratio $\mathcal{E}$
FINEFOODS-SMALL	9,705	4,879	18,523	1.9	3.8	11	11	384	0.013	0.026	0.037
MOVIES-SMALL	9,622	6,366	28,147	2.9	4.4	11	11	384	0.013	0.019	0.020
WIKIPEDIA	8,227	1,000	18,257	2.2	18.3	174	174	345	0.023	0.071	0.025
REDDIT	10,000	984	78,516	7.9	79.8	174	174	345	0.038	0.102	0.012
FINEFOODS-LARGE	256,059	74,258	560,804	2.2	7.6	11	11	384	0.004	0.014	0.013
MOVIES-LARGE	889,176	253,059	7,834,236	8.8	31.0	11	11	384	0.001	0.005	0.002

Baselines & Evaluation Metric

BASELINES

Graph Structure Only

FRAUDAR (Hooi et.al; 2016)

*dense block anomaly detection on **bipartite** graphs*

Non-attributed graphs

Features Only

Isolation Forest (Liu et.al; 2008)

Classical, non graph model

GNN Models

DOMINANT (Ding et.al; 2019)

AnomalyDAE (Fan, et.al; 2020)

AdOne (Bandyopadhya, et.al; 2020)

CONAD (Xu et.al; 2022)

Convert bipartite graphs into homogeneous graph

Edge anomaly score = average of the connected node scores

EVALUATION METRIC

Area under the Precision Recall Curve (AUC-PR)

Suitable for a very imbalance dataset
like in the anomaly detection task.

Overall Results

AUC-PR for node and edge anomaly detections

Edge-Level Anomaly: GraphBEAN significantly outperforms all models, often by a large margin

Node-Level Anomaly: GraphBEAN also outperforms the baselines, often by a considerably large margin

Large Size Datasets: GraphBEAN are **scalable** to the datasets, whereas most of the GNN baselines are not

Table 1. Results (average and std. dev) for standard size datasets

Model \ Dataset	FINEFOODS-SMALL			MOVIES-SMALL			WIKIPEDIA			REDDIT		
	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{E}$	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{E}$	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{E}$	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{E}$
FRAUDAR	0.256 (0.07)	0.392 (0.07)	0.279 (0.13)	0.229 (0.12)	0.188 (0.11)	0.260 (0.16)	0.102 (0.03)	0.085 (0.09)	0.043 (0.04)	0.059 (0.02)	0.101 (0.01)	0.011 (0.007)
IsoForest	0.090 (0.02)	0.166 (0.04)	0.794 (0.12)	0.127 (0.05)	0.181 (0.08)	0.827 (0.10)	0.226 (0.06)	0.499 (0.12)	0.278 (0.10)	0.361 (0.08)	**0.608** (0.07)	0.172 (0.039)
DOMINANT	0.735 (0.10)	0.721 (0.10)	0.686 (0.12)	0.631 (0.09)	0.708 (0.08)	0.389 (0.16)	0.164 (0.03)	0.179 (0.04)	0.049 (0.02)	0.121 (0.02)	0.186 (0.01)	0.016 (0.003)
AnomalyDAE	0.770 (0.09)	0.773 (0.09)	0.683 (0.12)	0.679 (0.12)	0.753 (0.10)	0.556 (0.10)	0.174 (0.03)	0.193 (0.04)	0.051 (0.02)	0.128 (0.02)	0.192 (0.02)	0.015 (0.003)
CONAD	0.740 (0.10)	0.721 (0.10)	0.691 (0.12)	0.684 (0.09)	0.695 (0.08)	0.564 (0.09)	0.165 (0.03)	0.182 (0.04)	0.052 (0.05)	0.116 (0.02)	0.180 (0.18)	0.016 (0.003)
AdOne	0.239 (0.05)	0.162 (0.03)	0.048 (0.01)	0.164 (0.03)	0.129 (0.03)	0.021 (0.01)	0.205 (0.04)	0.128 (0.03)	0.025 (0.01)	0.138 (0.02)	0.133 (0.01)	0.008 (0.001)
GraphBEAN (ours)	**0.855** (0.08)	**0.875** (0.07)	**0.876** (0.09)	**0.911** (0.04)	**0.911** (0.04)	**0.888** (0.08)	**0.441** (0.09)	**0.571** (0.03)	**0.415** (0.11)	**0.427** (0.06)	**0.631** (0.04)	**0.296** (0.038)

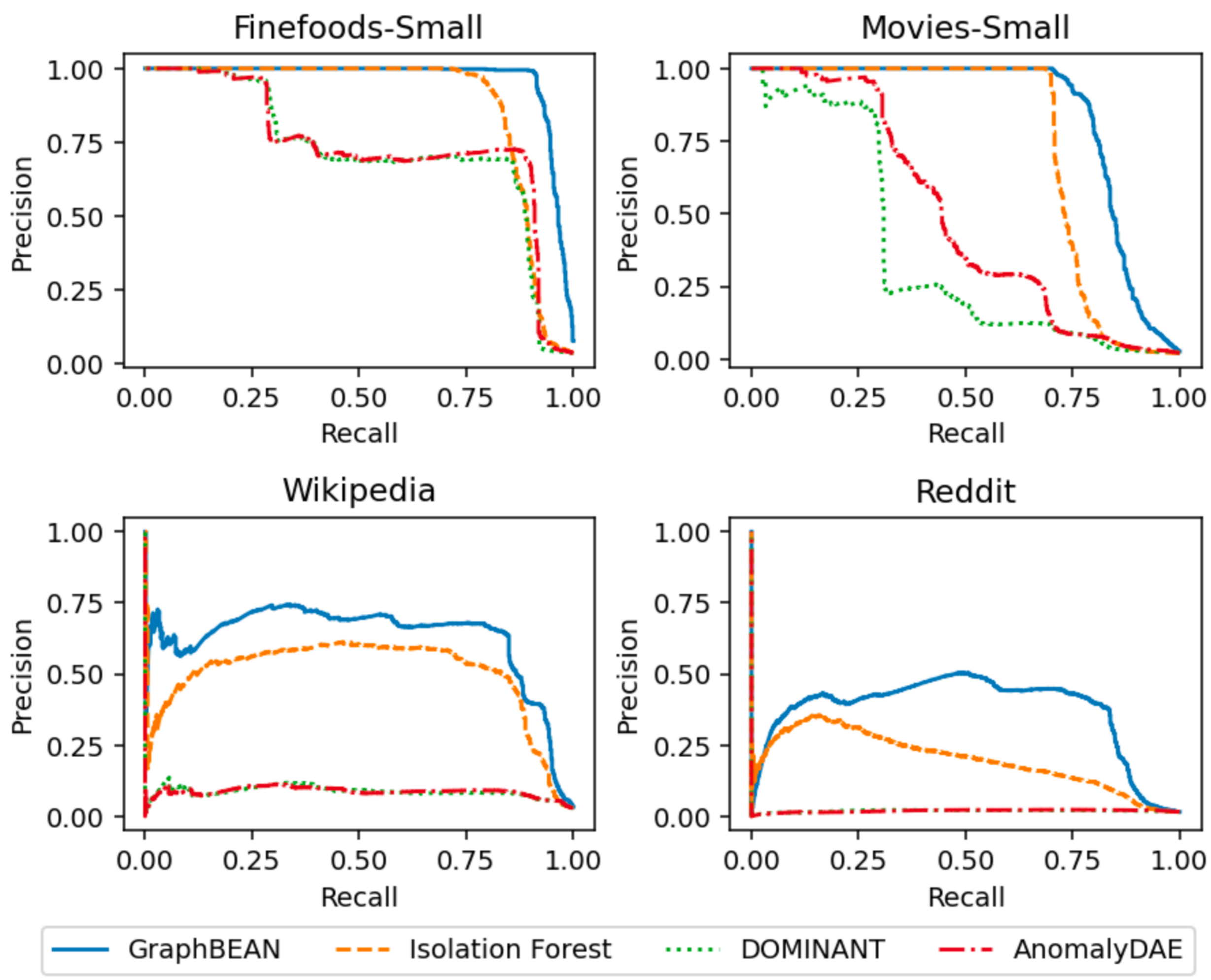
Table 2. Results for large size datasets

Model \ Dataset	FINEFOODS-LARGE			MOVIES-LARGE		
	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{E}$	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{E}$
FRAUDAR	0.093 (0.02)	0.195 (0.03)	0.077 (0.02)	0.004	0.007	0.001
IsoForest	0.023 (0.00)	0.098 (0.02)	0.805 (0.05)	0.008	0.025	0.722
GraphBEAN	**0.701** (0.07)	**0.813** (0.05)	**0.875** (0.03)	**0.413**	**0.547**	**0.779**

Precision-Recall Curve

Precision Recall Trade-off
at any given point in PR Curve

at almost all thresholding points
GraphBEAN outperforms all the baselines,
sometimes by a **significant** margin.



Conclusions

Conclusions and Remarks

Conclusions

Interaction Graphs

All available information is required

to build a high-performing anomaly detection model

Our proposed model

GraphBEAN is effective in detecting

entity level and interaction level anomaly
on bipartite interaction graphs

Grab

Open-Source Implementation

https://github.com/grab/GraphBEAN/

PyTorch PyG

Thank You