Label Information Enhanced Fraud Detection against Low Homophily in Graphs

Yuchen Wang*
Southeast University
Nanjing, China
yuchen_seu@seu.edu.cn

Jinghui Zhang†
Southeast University
Nanjing, China
jhzhang@seu.edu.cn

Zhengjie Huang
Baidu Inc.
Shenzhen, China
huangzhengjie@baidu.com

Weibin Li
Baidu Inc.
Shenzhen, China
liweibin02@baidu.com

Shikun Feng
Baidu Inc.
Shenzhen, China
fengshikun01@baidu.com

Ziheng Ma
Baidu Inc.
Beijing, China
maziheng01@baidu.com

Yu Sun
Baidu Inc.
Beijing, China
sunyu02@baidu.com

Dianhai Yu
Baidu Inc.
Beijing, China
yudianhai@baidu.com

Fang Dong
Southeast University
Nanjing, China
fdong@seu.edu.cn

Jiahui Jin
Southeast University
Nanjing, China
jjin@seu.edu.cn

Beilun Wang
Southeast University
Nanjing, China
beilun@seu.edu.cn

Junzhou Luo
Southeast University
Nanjing, China
jluo@seu.edu.cn

WWW2023

https://github.com/Orion-wyc/GAGA.

Reported by Xiaoke Li

Introduction

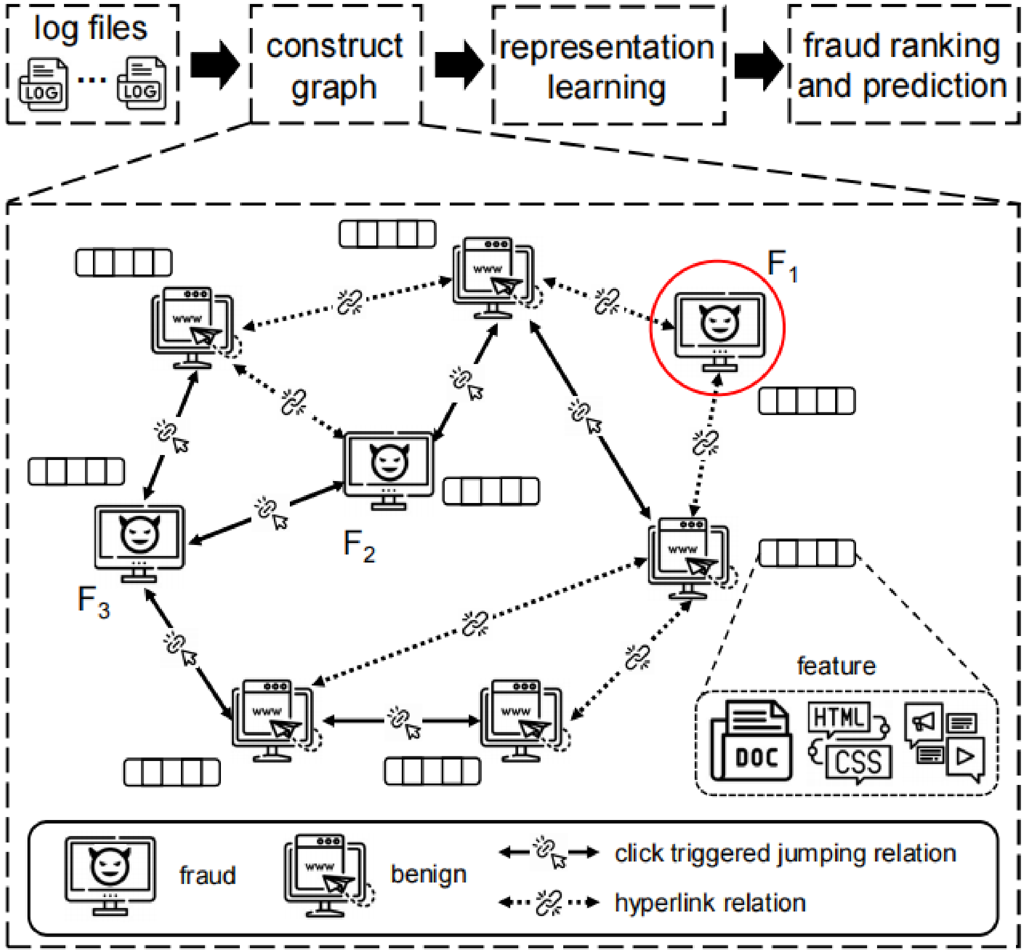

Figure 1: A toy multi-relational fraud graph of a website anti-fraud task. The target node F_1 highlighted with a colored circle is a fraudulent website that disguises itself by connecting to benign nodes (i.e., low homophily patterns).

Method

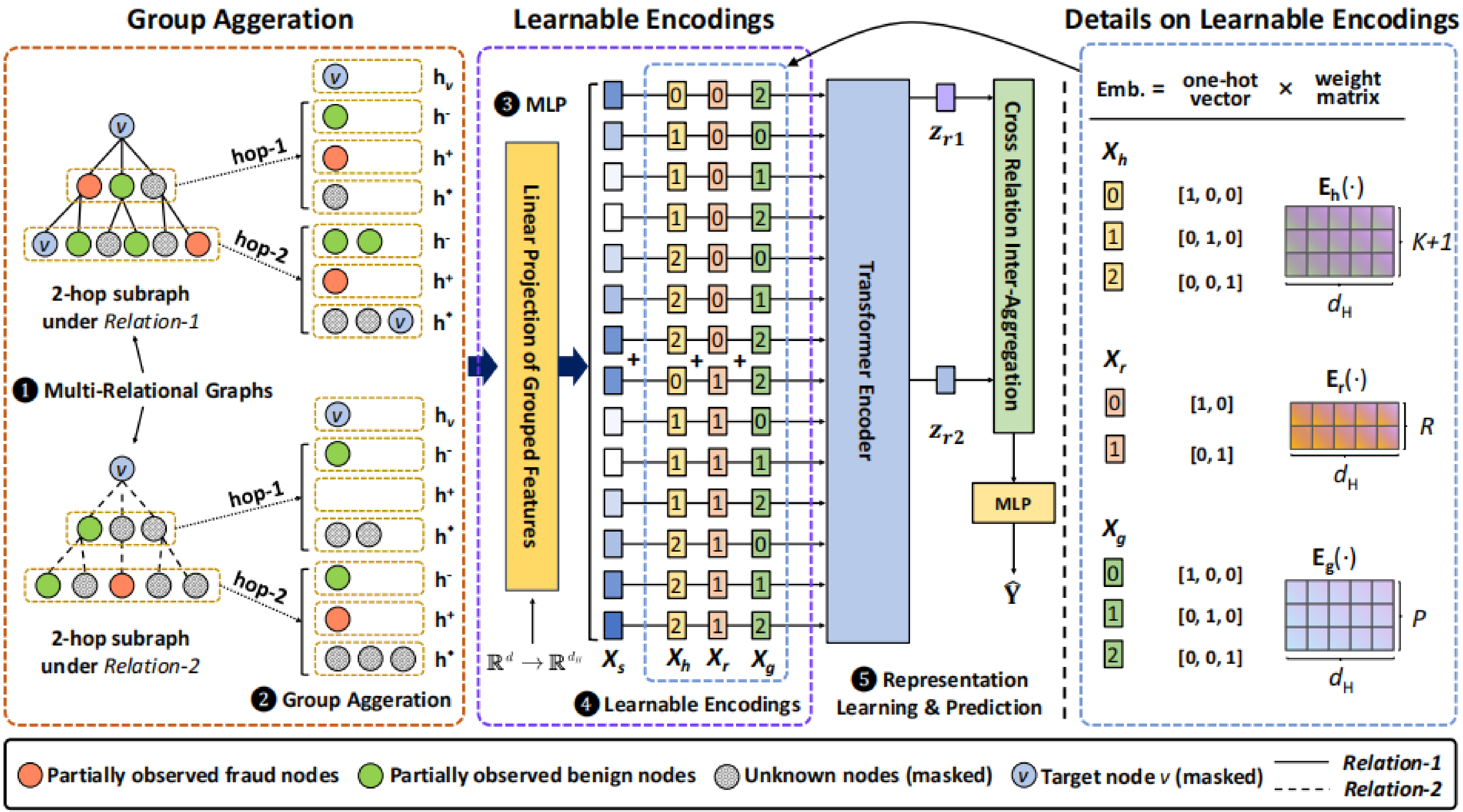

Figure 2: The overall architecture of GAGA. The number of hops K and relations R are set to 2 for simplicity.

Method

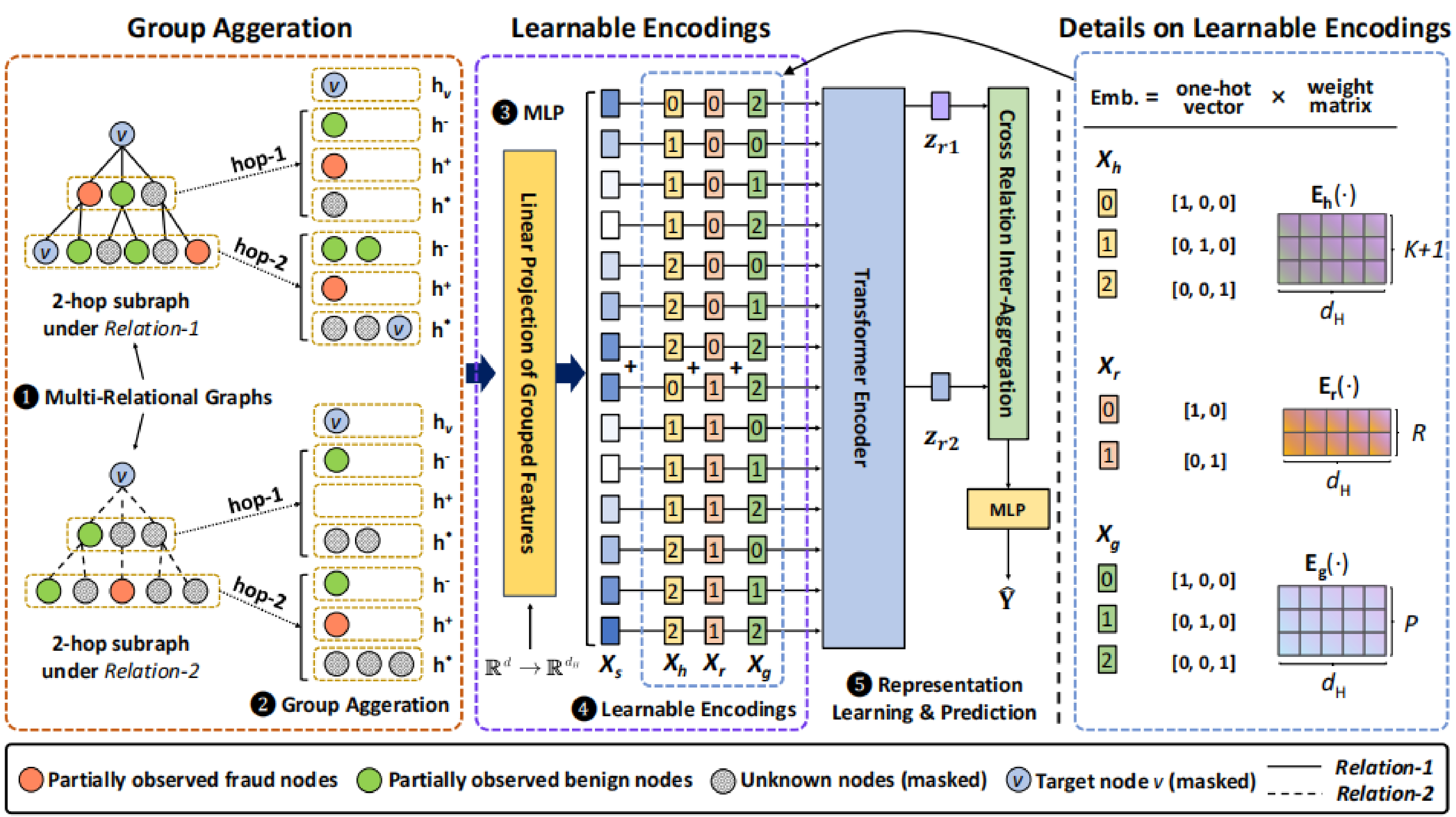

Figure 2: The overall architecture of GAGA. The number of hops K and relations R are set to 2 for simplicity.

$$f_{agg}(\{\mathbf{x}_u|\forall u \in \mathcal{N}(v) \cup \{v\}\}) = \frac{1}{\phi(\cdot)} \sum_{u \in \mathcal{N}(v)\cup\{v\}} \mathbf{x}_u, \quad (1)$$

$$\mathbf{H}_g = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_C, \mathbf{h}^*]$$
$$\mathbf{h}_i = f_{agg}(\{\mathbf{x}_u|\forall u \in V_i\}) = \frac{1}{\phi(\cdot)} \sum_{u \in V_i} \mathbf{x}_u, \quad (2)$$
$$\mathbf{H}_g = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3] = [\mathbf{h}^-, \mathbf{h}^+, \mathbf{h}^*],$$

$$\mathbf{H}_r = \big\Vert_{k=1}^{K} \mathbf{H}_g^{(k)},$$
$$\mathbf{H}_g^{(k)} = [\mathbf{h}^-, \mathbf{h}^+, \mathbf{h}^*]^{(k)} \text{ given } \hat{\mathcal{N}}_k(v), \quad (3)$$

$$\mathbf{X}_h = [\underbrace{\mathbf{E}_h(0), \overbrace{\mathbf{E}_h(1), \mathbf{E}_h(1), \mathbf{E}_h(1)}^{\text{1st hop}}, \dots, \overbrace{\mathbf{E}_h(K), \mathbf{E}_h(K), \mathbf{E}_h(K)}^{K-\text{th hop}}}_{\text{1st relation}}, \dots, \underbrace{\mathbf{E}_h(0), \mathbf{E}_h(1), \mathbf{E}_h(1), \mathbf{E}_h(1), \dots, \mathbf{E}_h(K), \mathbf{E}_h(K), \mathbf{E}_h(K)}_{R-\text{th relation}}] \quad (4)$$

$$\mathbf{X}_r = [\underbrace{\mathbf{E}_r(0), \mathbf{E}_r(0), \dots, \mathbf{E}_r(0)}_{\text{1st relation}}, \underbrace{\mathbf{E}_r(1), \mathbf{E}_r(1), \dots, \mathbf{E}_r(1)}_{\text{2nd relation}}, \dots, \underbrace{\mathbf{E}_r(R-1), \mathbf{E}_r(R-1), \dots, \mathbf{E}_r(R-1)}_{R-\text{th relation}}] \quad (5)$$

$$\mathbf{X}_g = [\underbrace{\mathbf{E}_g(*), \overbrace{\mathbf{E}_g(-), \mathbf{E}_g(+), \mathbf{E}_g(*)}^{\text{1st hop}}, \dots, \overbrace{\mathbf{E}_g(-), \mathbf{E}_g(+), \mathbf{E}_g(*)}^{K-\text{th hop}}}_{\text{1st relation}}, \dots, \underbrace{\mathbf{E}_g(*), \mathbf{E}_g(-), \mathbf{E}_g(+), \mathbf{E}_g(*), \dots, \mathbf{E}_g(-), \mathbf{E}_g(+), \mathbf{E}_g(*)}_{R-\text{th relation}}] \quad (6)$$

Method

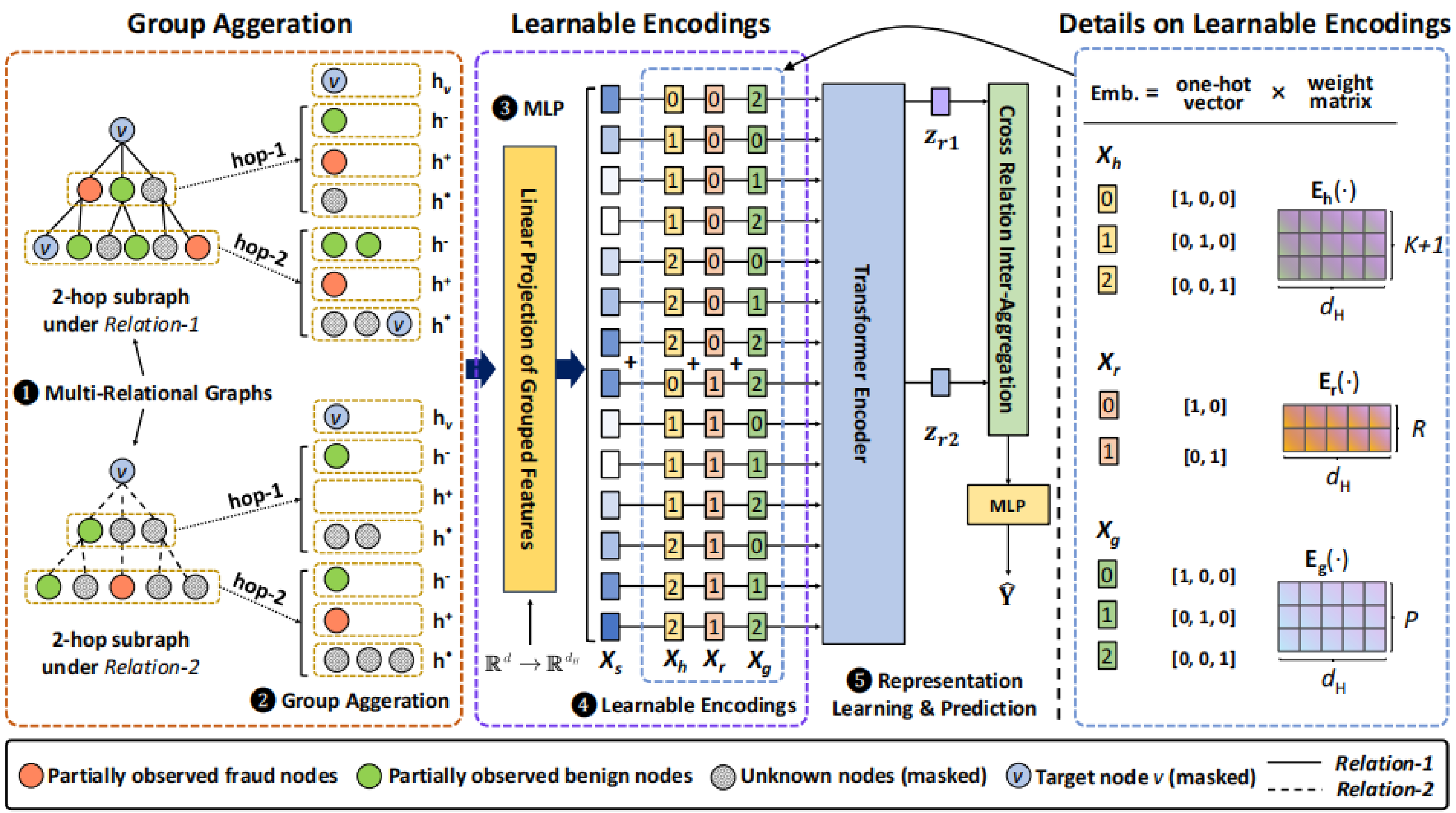

Figure 2: The overall architecture of GAGA. The number of hops K and relations R are set to 2 for simplicity.

$$\mathbf{X}_{in} = \mathbf{X}_s + \mathbf{X}_h + \mathbf{X}_r + \mathbf{X}_g. \tag{7}$$

$$\begin{aligned}
\mathbf{x}_i^0 &= \mathbf{x}_i, \\
\mathbf{x}_i^{l+1} &= \text{Concat}\left(\text{head}_1, \ldots, \text{head}_M\right) O^l, \\
\text{head}_m &= \text{Attention}\left(\mathbf{Q}^{m,l}\mathbf{x}_i^l,\ \mathbf{K}^{m,l}\mathbf{x}_j^l,\ \mathbf{V}^{m,l}\mathbf{x}_j^l\right) \\
&= \sum_{j \in S} w_{ij}\left(\mathbf{V}^{m,l}\mathbf{x}_j^l\right), \\
w_{ij} &= \text{softmax}_j\left(\frac{\mathbf{Q}^{m,l}\mathbf{x}_i^l \cdot \mathbf{K}^{m,l}\mathbf{x}_j^l}{\sqrt{d_H}}\right),
\end{aligned} \tag{8}$$

$$\begin{aligned}
\mathcal{L} &= -\sum_{v \in \hat{\mathcal{V}}} \left[y_v \log p_v + (1 - y_v)\log(1 - p_v)\right] + \lambda \|\theta\|_2^2, \\
p_v &= \text{sigmoid}(\text{MLP}(\mathbf{z}_v))
\end{aligned} \tag{9}$$

Experiments

Table 1: Statistics of Datasets. IR represents the class imbalance ratio. $\varphi_r = \frac{|\{(u,v)|A_r[u,v]=1 \wedge y_u=y_v\}|}{\sum A_r[u,v]}$ is the homophily ratio which calculates the proportion of the immediate neighbors that share the same class label.

Dataset	#Nodes (IR)	Relations	#Relations	φ_r	#Feat
Amazon	11,944 (13.5)	U-P-U	175,608	0.1673	25
		U-S-U	3,566,479	0.0558	
		U-V-U	1,036,737	0.0532	
YelpChi	45,954 (5.9)	R-U-R	49,315	0.9089	32
		R-S-R	3,402,743	0.1857	
		R-T-R	573,616	0.1764	
BF10M	13,251,571 (12.4)	U-J-U	5,815,738	0.4092	161
		U-L-U	65,530,647	0.1371	

Experiments

Table 2: Performance Comparison on public spam review datasets.

Methods	YelpChi			Amazon		
	AUC	AP	F1-macro	AUC	AP	F1-macro
$\text{GCN}_{(\text{ICLR'17})}$	0.5924±0.0030	0.2176±0.0119	0.5072±0.0271	0.8405±0.0075	0.4660±0.0131	0.6985±0.0046
$\text{GAT}_{(\text{ICLR'17})}$	0.6796±0.0070	0.2807±0.0048	0.5773±0.0080	0.8096±0.0113	0.3082±0.0067	0.6681±0.0076
$\text{HAN}_{(\text{WWW'19})}$	0.7420±0.0009	0.2722±0.0036	0.5472±0.0097	0.8421±0.0062	0.4631±0.0185	0.7016±0.0126
$\text{GraphSAGE}_{(\text{NeurIPS'17})}$	0.7409±0.0000	0.3258±0.0000	0.6001±0.0002	0.9172±0.0001	0.8268±0.0002	0.9029±0.0004
$\text{Cluster-GCN}_{(\text{KDD'19})}$	0.7623±0.0069	0.3691±0.0179	0.6204±0.0557	0.9211±0.0256	0.8075±0.0566	0.8853±0.0272
$\text{GraphSAINT}_{(\text{ICLR'20})}$	0.7412±0.0143	0.3641±0.0304	0.5974±0.0728	0.8946±0.0176	0.7956±0.0091	0.8888±0.0244
$\text{CARE-GNN}_{(\text{CIKM'20})}$	0.7854±0.0111	0.3972±0.0208	0.6064±0.0186	0.8823±0.0305	0.7609±0.0904	0.8592±0.0574
$\text{FRAUDRE}_{(\text{ICDM'21})}$	0.7588±0.0078	0.3870±0.0186	0.6421±0.0135	0.9308±0.0180	0.8433±0.0089	0.9037±0.0031
$\text{PC-GNN}_{(\text{WWW'21})}$	0.8154±0.0031	0.4797±0.0064	0.6523±0.0197	0.9489±0.0067	0.8435±0.0166	0.8897±0.0144
$\text{RioGNN}_{(\text{TOIS'21})}$	0.8144±0.0050	0.4722±0.0079	0.6422±0.0233	0.9558±0.0019	0.8700±0.0044	0.8848±0.0125
$\text{H}^2\text{-FDetector}_{(\text{WWW'22})}$	0.8892±0.0020	0.5543±0.0135	0.7345±0.0086	0.9605±0.0008	0.8494±0.0023	0.8010±0.0058
$\text{SIGN}_{(\text{ICML-GRL'20})}$	0.8569±0.0051	0.5801±0.0191	0.7308±0.0053	0.9404±0.0033	0.8483±0.0031	0.9046±0.0012
$\text{GA+RNN}_{(\text{Ablation})}$	0.9073±0.0237	0.6727±0.0672	0.7713±0.0313	0.9563±0.0075	0.8688±0.0086	0.9081±0.0055
$\text{GA+LSTM}_{(\text{Ablation})}$	0.9278±0.0025	0.7358±0.0119	0.7994±0.0041	0.9539±0.0089	0.8655±0.0117	0.9079±0.0066
$\text{GT}_{(\text{Ablation})}$	0.9084±0.0065	0.6979±0.0143	0.7839±0.0091	0.9514±0.0078	0.8581±0.0085	**0.9137±0.0035**
$\text{GAGA}_{(\text{Ours})}$	**0.9439±0.0016**	**0.8014±0.0063**	**0.8323±0.0041**	**0.9629±0.0052**	**0.8815±0.0095**	0.9133±0.0040

Experiments

Table 3: Performance Comparison on BF10M.

Methods	BF10M		
	AUC	AP	F1-macro
MLP	0.9066±0.0015	0.5135±0.0063	0.6840±0.1073
GCN	0.9194±0.0034	0.5282±0.0128	0.7478±0.0037
GAT	0.9143±0.0336	0.5487±0.0855	0.7469±0.0340
GraphSAGE	0.9597±0.0020	0.6965±0.0107	0.8103±0.0062
Cluster-GCN	0.9480±0.0039	0.6383±0.0221	0.7898±0.0084
GraphSAINT	0.9659±0.0016	0.7156±0.0092	0.8227±0.0061
SIGN	0.9652±0.0112	0.7262±0.0592	0.8228±0.0225
H^2-FDetector		OOM	
$\text{GAGA}_{(\text{Ours})}$	**0.9923±0.0004**	**0.9249±0.0036**	**0.9097±0.0032**

Table 4: Comparison of label utilization methods.

Methods	AUC	AP	F1-fraud	F1-benign
LPA	0.6248	0.2567	0.2903	0.7021
GCN	0.5894	0.2032	0.2685	0.7085
GCN (w/ C&S)	0.6188↑	0.2588↑	0.2864↑	0.7301↑
GAT	0.6707	0.2737	0.3375	0.7872
GAT (w/ C&S)	0.6762↑	0.2875↑	0.3414↑	0.8083↑
GT	0.9059	0.6802	0.6322	0.9308
GT (w/ C&S)	0.9241↑	0.7352↑	0.6596↑	0.9390↑
GT (w/ BoT)	0.9281↑	0.7478↑	0.6672↑	0.9352↑
GAGA (Ours)	**0.9423**	**0.7944**	**0.7135**	**0.9525**

Experiments

Table 5: Ablation study to demonstrate the effectiveness of the Learnable Encodings.

Ablation (n_hops=2)		AUC	
		YelpChi	Amazon
w/ GA	X_g, X_r, X_h	**0.9439±0.0016**	**0.9629±0.0052**
	X_r, X_h	0.9274±0.0028	0.9584±0.0082
	X_g	0.9355±0.0032	0.9584±0.0099
	X_r	0.9217±0.0036	0.9581±0.0106
	X_h	0.9144±0.0039	0.9540±0.0099
	–	0.9020±0.0066	0.9524±0.0070
w/o GA	X_r, X_h	0.9219±0.0047	0.9578±0.0095
	X_r	0.9197±0.0048	0.9574±0.0075
	X_h	0.9172±0.0046	0.9520±0.0081
	–	0.9084±0.0065	0.9528±0.0077

Table 6: Sensitivity analysis to verify the performance under different percentage (1~40%) of training data and percentage (1~40%) of partially observed labels. (YelpChi)

Train (%)	Label (%)	AUC	AP
40	40	**0.9439±0.0016**	**0.8014±0.0063**
40	30	0.9416±0.0020	0.7904±0.0067
40	20	0.9328±0.0034	0.7619±0.0112
40	10	0.9232±0.0024	0.7335±0.0060
30	30	0.9320±0.0030	0.7641±0.0076
20	20	0.9081±0.0189	0.6894±0.0456
10	10	0.8782±0.0107	0.5924±0.0207
5	5	0.8130±0.0064	0.4703±0.0315
1	1	0.7000±0.0710	0.3104±0.0744

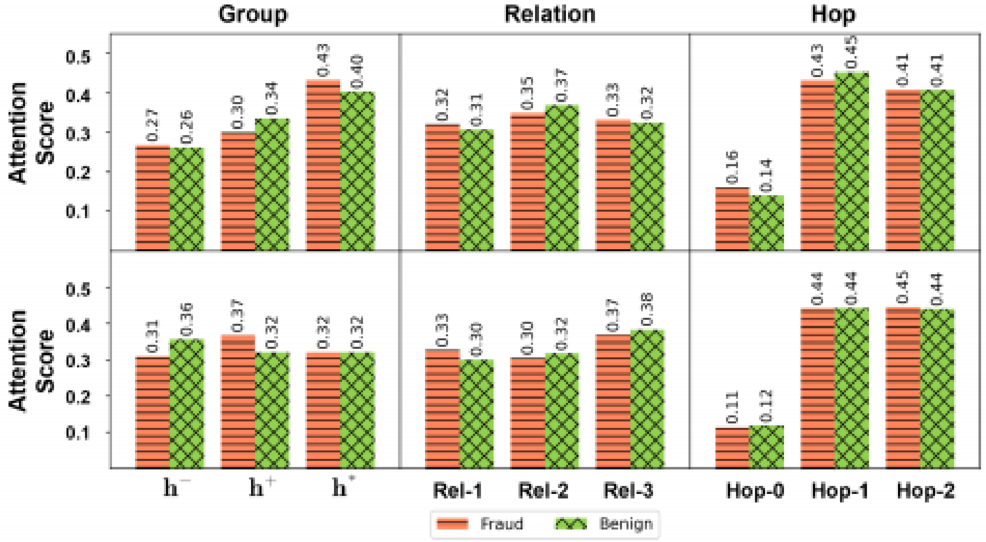

Figure 3: Visualization of attention scores on Amazon (upper) and YelpChi (lower) based on statistics. (Rel-1, Rel-2, Rel-3) denotes (U-P-U, U-S-U, U-V-U) for Amazon and (R-S-R, R-T-R, R-U-R) for YelpChi, respectively.

Experiments

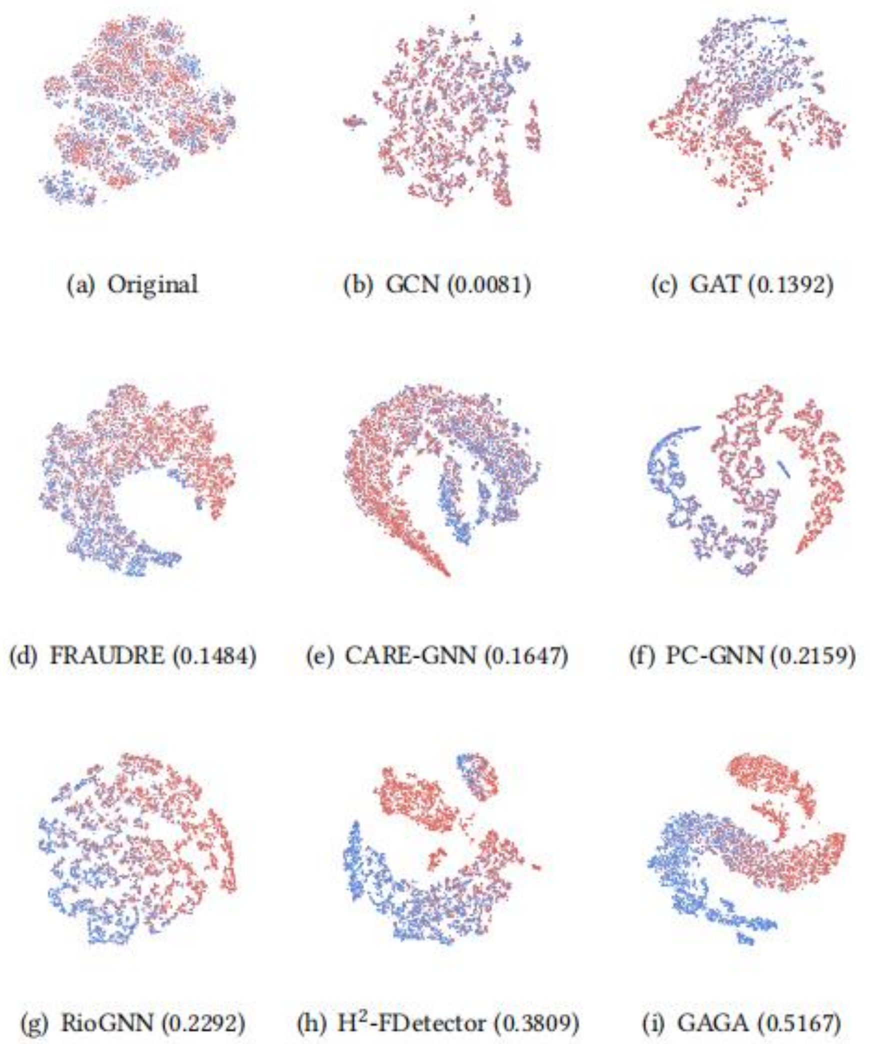

Figure 4: The t-SNE visualization of latent representations and the Adjusted Rand Index of each cluster. (Dataset: YelpChi, Fraud: red, Benign: blue)

Table 7: Comparison of training throughput (sample/s).

Methods	YelpChi	Amazon	BF10M
GAGA	21739.13	17241.38	12820.51
FRAUDRE	244.92	403.06	-
CARE-GNN	13888.89	3125.00	-
PC-GNN	9900.99	1937.98	-
RioGNN	17241.38	3436.43	-
H^2-FDetetor	11185.72	336.60	OOM

1% and 5% training nodes). GAGA still shows a competitive result. Besides, we explore the effect of the percentage of the ground truth labels in training and prediction phase in Tab. 6 (the upper half). It is observed that the accumulated ground-truth labels can boost the performance of fraud detection.

Thanks