# Technical Report

## 1 PROOF OF THEOREM 2

Theorem 2. The overall time complexity of the RADIO model is dominated by $O(deg^k_{\max} \cdot |V|)$, where $deg_{\max}$ denotes the maximum degree of nodes, $k$ is the constant from $k$-ego net.

Proof. The time complexities of the subgraph encoder, prototype discovery, and anomalous subgraph refinement modules in the RADIO model are $O(deg_{\text{avg}} \cdot |V| + k_1(|U| + |Z|)$, $O(deg^k_{\max} \cdot |V| + k_2 \cdot |V| + d_p \cdot p)$, and $O(|\mathcal{G}[S_{a_t}]| \cdot t)$, respectively, where $deg_{\text{avg}}$ denotes the average degree of nodes, $deg_{\max}$ denotes the maximum degree of nodes, $d_p$ denotes the dimension of the feature vector, $p$ is the number of prototypes, $k, k_1, k_2$ are constants, and $t$ denotes the number of states. Since the parameters involved are constants, the overall time complexity is dominated by $O(deg^k_{\max} \cdot |V|)$. Given that $k$ is typically small, the RADIO model's time complexity scales nearly linearly with the size of the nodes $|V|$. □

## 2 PROOF OF LEMMA 1

Lemma 1. Optimizing the loss function in Equation 4 of the paper is equivalent to maximizing $I(f_e(\mathcal{G}[S_1]); f_e(\mathcal{G}[S_2]))$, leading to the maximization of $I(f_e(\mathcal{G}[S_1]); \mathcal{G}[S_2])$, where $I$ represents mutual information.

Proof. Since the objective of the encoder is to maximize the embedding similarity between positive subgraph pairs and minimize the embedding similarity between negative pairs, the loss function for a pair of subgraphs $G[S_1]$ and $G[S_2]$ with embeddings $e(G[S_1])$ and $e(G[S_2])$ is equivalent to:

$$\mathcal{L}_{1,2} = -\log \frac{\exp(-D(G[S_1], G[S_2]))}{\sum_{i,j} \exp(-D(G[S_i], G[S_j]))} . \tag{1}$$

And minimizing Equation 1 above is equivalent to maximizing a lower bound of the mutual information between the learned embeddings of two subgraphs as proven in [5], which is equivalent to maximizing the mutual information between their embeddings $I(f_e(\mathcal{G}[S_1]); f_e(\mathcal{G}[S_2]))$. According to Lemma A.5 in [4], it leads to the maximization of $I(f_e(\mathcal{G}[S_1]); \mathcal{G}[S_2])$. □

## 3 PROOF OF THEOREM 3

Theorem 3. Let $f_{e_1}$ denote our proposed subgraph encoder with financial distribution considered, and let $f_{e_2}$ denote the encoder without financial distribution modeling as in [2, 3]. After sufficient training of $f_{e_1}$ and $f_{e_2}$, $I(f_{e_1}(\mathcal{G}[S_1]); y) > I(f_{e_2}(\mathcal{G}[S_1]); y)$, where $y$ represents the graph label.

Proof. The encoder $f_{e_1}$ can differentiate financial subgraphs with different financial distributions that $f_{e_2}$ cannot. It means that $f_{e_1}$ can capture more information of subgraphs $\mathcal{G}[S_1]$ than $f_{e_2}$:

$$H(\mathcal{G}[S_1]) \geq H(f_{e_1}(\mathcal{G}[S_1])) > H(f_{e_2}(\mathcal{G}[S_1])) , \tag{2}$$

where $H(\cdot)$ represents information entropy. Since $f_{e_1}(\mathcal{G}[S_1])$ and $f_{e_2}(\mathcal{G}[S_1])$ are functions of $\mathcal{G}[S_1]$, we have

$$I(f_{e_1}(\mathcal{G}[S_1]); \mathcal{G}[S_1]) > I(f_{e_2}(\mathcal{G}[S_1]); \mathcal{G}[S_1]) . \tag{3}$$

According to Theorem A.6 in [4] and Lemma 4.1, it holds that $I(f_{e_1}(\mathcal{G}[S_1]); y) > I(f_{e_2}(\mathcal{G}[S_1]); y)$. □

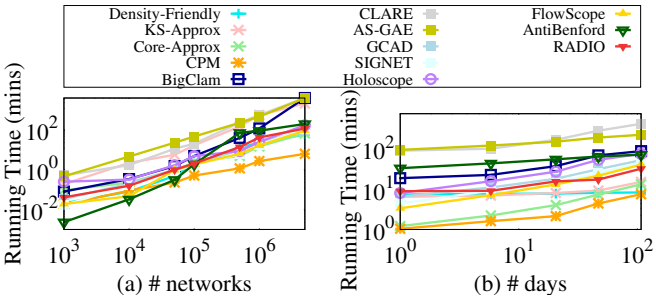


**Figure 1: Scalability on (a) the size of graph nodes; (b) the span of time dimension.**

## 4 SCALABILITY EVALUATION

We use Intel Xeon Platinum 8370C processor 32-core 2.8GHz CPU with NVIDIA GeForce RTX 4090 24G GPU for evaluation. We use five datasets spanning from 1 week to 1 year of 2019 to analyze the running time of our method in Section 4.2 as in [1]. Details are in Table 1.

**Table 1: Statistics of experimented datasets.**

| Datasets | # Nodes | # Edges | # Transactions |
|---|---|---|---|
| ETH-1 Week | 779,237 | 1,016,481 | 1,438,641 |
| ETH-1 Month | 2,199,347 | 3,331,594 | 6,128,061 |
| ETH-3 Month | 2,877,055 | 8,504,999 | 17,529,925 |
| ETH-6 Month | 5,018,047 | 21,572,152 | 43,885,527 |
| ETH-1 Year | 6,820,719 | 38,917,136 | 85,055,054 |

### 4.1 Scalability w.r.t. the size of graph nodes

We evaluate the scalability of RADIO with respect to the size of graph nodes in Figure 1 (a). We find that the running time of RADIO is approximately linear w.r.t. the size of financial networks in the log-scale axes, which means that the time complexity of RADIO is polynomial.


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

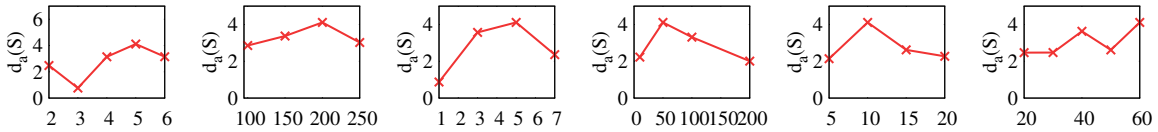


**Figure 2: Hyperparameter effects of RADIO on ETH-Jan-18 dataset.**

## 4.2 Scalability w.r.t. the span of time dimension

We also evaluate RADIO's scalability w.r.t. the time dimension. A longer time span results in a larger size of transaction edges $|E|$ when we fix the size of nodes. Figure 1 (b) shows the running time trend as the time span increases from 7 to 365 days. The trend is approximately linear with respect to the time span. The graph node size is fixed at 500,000 to ensure the observations relate only to the time dimension change.

# 5 SENSITIVITY ANALYSIS OF HYPERPARAMETERS

We omit other datasets due to similar trends to ETH-Jan-18. Similarly, we show $d_a(S)$ results due to similar trends in other metrics.

**Depth $l$ of Graph Neural Networks.** Figure 2 (a) shows that increasing $l$ leads to worse result when $l = 3$, after that achieves the best at 5, then works worse afterward for the over-fitting problems.

**Maximum Size $m$ of Nodes in the Boundary.** Figure 2 (b) shows that increasing $m$ beyond a certain point, i.e., 200, leads to more noise, resulting in degraded results.

**Layer of the $k$-ego Net.** Figure 2 (c) shows that enlarging the layer of the $k$-ego net beyond a certain point, i.e., 5, brings large-size subgraphs with lower abnormal degrees, leading to poor effect.

**Size of the Top-$p$ Candidates.** Figure 2 (d) shows that increasing the size of candidates beyond a certain point, i.e., 50, results in more sub-optimal subgraphs, generating worse performance.

**Number of Sample Pairs.** Figure 2 (e) shows that increasing the number of sample pairs beyond a certain point, specifically 10, results in degraded performance due to potentially increased noise.

**Number of Training subgraphs.** Figure 2 (f) shows that enlarging the number of training subgraphs initially improves performance but eventually leads to increased variability.

# REFERENCES

[1] Tianyi Chen and Charalampos Tsourakakis. 2022. Antibenford subgraphs: Unsupervised anomaly detection in financial networks. In *SIGKDD*. 2762–2770.
[2] Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, Jure Leskovec, et al. 2020. Neural subgraph matching. *arXiv preprint arXiv:2007.03092* (2020).
[3] Xixi Wu, Yun Xiong, Yao Zhang, Yizhu Jiao, Caihua Shan, Yiheng Sun, Yangyong Zhu, and Philip S Yu. 2022. Clare: A semi-supervised community detection algorithm. In *SIGKDD*. 2059–2069.
[4] Yucheng Wu, Leye Wang, Xiao Han, and Han-Jia Ye. 2024. Graph Contrastive Learning with Cohesive Subgraph Awareness. In *Proceedings of the ACM on Web Conference 2024*. 629–640.
[5] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.