A TECHNICAL REPORT

A.1 Efficiency Evaluation

We use Intel Intel(R) Xeon(R) Platinum 8358P processor CPU @2.60GHz with NVIDIA A40 48GB for evaluation. The implementation is done in Python 3.8 with PyTorch.

We conduct experiments on Digg, Yelp, and Amazon datasets to compare the time required for training and testing phases of all models. As shown in Figure 4, our model demonstrates consistent runtimes across different datasets for both training and testing. In contrast, the THGNN model exhibits significant variation in training time across datasets due to its early stopping strategy. Since MIDAS is a traditional non-learning based approach, only a testing phase exists and it is very short. Despite incorporating dynamic structural evolution learning in our proposed ReadGraph model, which is not included in other models, our testing time remains comparable as other methods.

A.2 Convergence of Relation Subgraph Representation Learning

The message passing function for each relation subgraph is defined as:

$$\mathbf{x}'_{\phi(e)}(v_i^t) = \sigma\left(\sum_{j\in\mathcal{N}_{\phi(e)}(v_i^t)} g_{ij}^{\phi(e)} \cdot \mathbf{x}_{\phi(e)}(v_j^t)\right), \tag{7}$$

where

$$g_{ij}^{\phi(e)} = \text{softmax}\left(w_{\phi(e)}^\top \cdot [\mathbf{x}(v_i^t)||\mathbf{x}(v_j^t)]\right). \tag{8}$$

Theorem 1. If the message passing function $g_{ij}^{\phi(e)}$ and the activation function σ are Lipschitz continuous and the graph is connected, then the relation subgraph representation learning will converge to a stable representation.

Proof. 1). **Lipschitz Continuity of $g_{ij}^{\phi(e)}$**:

Let $\mathbf{x}^1$ and $\mathbf{x}^2$ be two different sets of node embeddings. Since the softmax function is Lipschitz continuous, we can conclude that $g_{ij}^{\phi(e)}$ is Lipschitz continuous. The Lipschitz continuity of the function $g_{ij}^{\phi(e)}$ implies there exists a constant L such that:

$$\left|g_{ij}^{\phi(e)}(\mathbf{x}^1) - g_{ij}^{\phi(e)}(\mathbf{x}^2)\right| \leq L\|\mathbf{x}^1 - \mathbf{x}^2\|, \tag{9}$$

where $\|\cdot\|$ denotes the norm.

2). **Fixed-Point Convergence**:

Consider the message passing process as an iterative update:

$$\mathbf{x}_{\phi(e)}^{(k+1)}(v_i^t) = \sigma\left(\sum_{j\in\mathcal{N}_{\phi(e)}(v_i^t)} g_{ij}^{\phi(e)} \cdot \mathbf{x}_{\phi(e)}^{(k)}(v_j^t)\right). \tag{10}$$

We can analyze the convergence using a fixed-point theorem. Specifically, we are interested in the fixed point $\mathbf{x}_{\phi(e)}^*$ such that:

$$\mathbf{x}_{\phi(e)}^*(v_i^t) = \sigma\left(\sum_{j\in\mathcal{N}_{\phi(e)}(v_i^t)} g_{ij}^{\phi(e)} \cdot \mathbf{x}_{\phi(e)}^*(v_j^t)\right). \tag{11}$$

Since σ is a contraction mapping and $g_{ij}^{\phi(e)}$ is Lipschitz continuous, the Banach fixed-point theorem guarantees the existence and uniqueness of the fixed point. Therefore, the iterative process converges to this fixed point.

Thus, under the conditions that $g_{ij}^{\phi(e)}$ and σ are Lipschitz continuous and the graph is connected, the message passing function for relation subgraphs converges to a stable representation.

□

A.3 Stability of the Dynamic Structural Evolution

The GRU model in sequential evolution learning is defined by the following recurrence relations:

$$\begin{aligned}
\mathbf{h}_t &= \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \bar{\mathbf{h}}_t \\
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1}) \\
\bar{\mathbf{h}}_t &= \tanh(\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t \circ \mathbf{h}_{t-1})) \\
\mathbf{c}_t &= \sigma(\mathbf{W}_r \mathbf{m}_{r^t} + \mathbf{U}_r \mathbf{h}_{t-1}),
\end{aligned} \tag{12}$$

where h_t is the hidden state at time t, z_t is the update gate, $\bar{h}_t$ is the candidate activation, c_t is the reset gate, and σ and tanh are activation functions.

Theorem 2. The GRU-based sequential learning algorithm will converge to a stable state if the following conditions are satisfied: the weight matrices $\mathbf{W}_z$, $\mathbf{U}_z$, $\mathbf{W}_h$, $\mathbf{U}_h$, $\mathbf{W}_r$, and $\mathbf{U}_r$ are bounded, the activation functions σ and tanh are Lipschitz continuous, and the input sequence $\mathbf{m}_{r^t}$ is bounded.

Proof. 1). **Bounded Weight Matrices and Activation Functions**:

Assume that all weight matrices $\mathbf{W}_z$, $\mathbf{U}_z$, $\mathbf{W}_h$, $\mathbf{U}_h$, $\mathbf{W}_r$, and $\mathbf{U}_r$ are bounded by a constant M, and the activation functions σ and tanh are Lipschitz continuous with constants L_σ and $L_{\tanh}$, respectively. Then:

$$|\sigma(\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1})| \leq L_\sigma \|\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1}\|, \tag{13}$$

$$\left|\tanh(\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t \circ \mathbf{h}_{t-1}))\right| \leq L_{\tanh} \|\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t \circ \mathbf{h}_{t-1})\|. \tag{14}$$

2). **Fixed-Point Stability**:

We analyze the fixed point h_t^* such that:

$$\mathbf{h}_t^* = \mathbf{z}_t^* \circ \mathbf{h}_{t-1}^* + (1 - \mathbf{z}_t^*) \circ \bar{\mathbf{h}}_t^*, \tag{15}$$

where:

$$\mathbf{z}_t^* = \sigma(\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1}^*), \tag{16}$$

$$\bar{\mathbf{h}}_t^* = \tanh(\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t^* \circ \mathbf{h}_{t-1}^*)), \tag{17}$$

$$\mathbf{c}_t^* = \sigma(\mathbf{W}_r \mathbf{m}_{r^t} + \mathbf{U}_r \mathbf{h}_{t-1}^*). \tag{18}$$

Given the boundedness of the weight matrices and the bounded input sequence, the update equation is guaranteed to remain within a bounded region. By the Banach fixed-point theorem, the iterative process will converge to a unique fixed point if the mappings are contraction mappings.

3). **Contraction Mapping:**

We need to show that the GRU update rules form a contraction mapping. For a given input sequence $\mathbf{m}_{r^t}$, let $\|\cdot\|$ denote the norm. The GRU update function is:

$$\Phi(\mathbf{h}_{t-1}) = \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \bar{\mathbf{h}}_t. \tag{19}$$

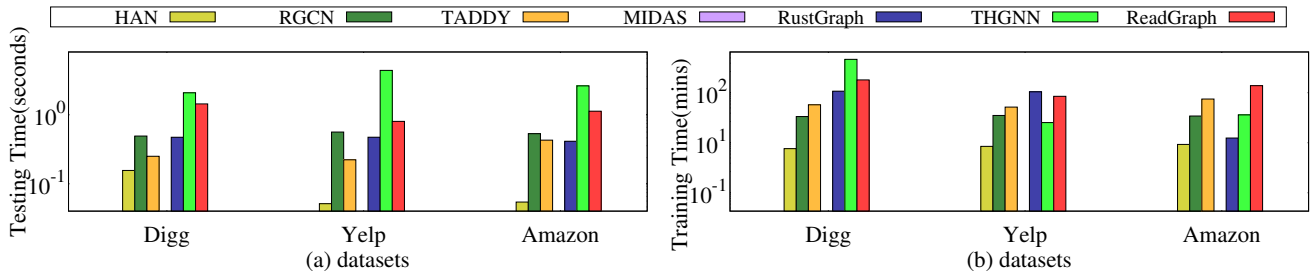

Figure 4: Efficiency of all methods on three datasets.

Algorithm 1: The ReadGraph Model

Input : The target edge $e = (v_{s_1}, v_{s_2})$, the graph $G(\mathcal{V}, \mathcal{E})$
Output: The anomaly score $\hat{y}_e$

```
/* Edge-Centric Heterogeneous Subgraph Sampling */
```
$\mathcal{G}_e \leftarrow \emptyset$
for *each sampling strategy in {meta-path, k-hop, node-importance}* **do**
  Sample nodes to form subgraph $\mathcal{G}_s$
  $\mathcal{G}_e \leftarrow \mathcal{G}_e \cup \mathcal{G}_s$
Initialize node embeddings in $\mathcal{G}_e$ using Equation 2
```
/* Relation Subgraph Representation Learning */
```
for *each relation type* $\phi(e)$ **do**
  Propagate messages within relation subgraph $\mathcal{G}_{\phi(e)}$
  Compute node embeddings $\mathbf{x}'_{\phi(e)}(v_i^t)$ using Equation 3
Fuse relation-level messages using Equation 4
Aggregate node embeddings to compute $\mathbf{m}_{r^t}$
```
/* Relation-Aware Dynamic Structural Evolution */
```
for *each snapshot* t **do**
  Update hidden state using GRU equations in Equation 5
Form sequential outputs $\mathbf{H}'$
```
/* Prototype Alignment */
```
Initialize prototypes $\mathbf{P}^0$
for *each prototype layer* l **do**
  Update prototypes $\mathbf{P}^l$
Update prototype representations $\mathbf{H}_p$
```
/* Optimization */
```
Compute loss $\mathcal{L}$ using Equation 6
Minimize $\mathcal{L}$ using optimization techniques until converge
```
/* Anomaly Prediction */
```
Compute $\hat{y}_e$
return $\hat{y}_e$

The Lipschitz continuity of the activation functions ensures that Φ is a contraction mapping, meaning that:

$$\|\Phi(\mathbf{h}_{t-1}) - \Phi(\mathbf{h}'_{t-1})\| \leq \alpha \|\mathbf{h}_{t-1} - \mathbf{h}'_{t-1}\|, \tag{20}$$

where $\alpha < 1$. Therefore, the GRU-based sequential learning algorithm converges to a stable representation.

□

A.4 The Overall Complexity

Theorem 3. The overall complexity of ReadGraph is dominated by $O(|\mathcal{E}|(|\mathcal{V}_h| + |\mathcal{E}_h|))$, where $|\mathcal{V}_h|$ and $|\mathcal{E}_h|$ denote the number of nodes and edges in the sampled heterogeneous subgraph for each target edge e.

Proof. The time complexity of edge-centric heterogeneous subgraph sampling module is $O\big(|\mathcal{E}|(|\mathcal{V}_h| + |\mathcal{E}_h| + N_m \cdot |\mathcal{V}_h| \cdot L_m + deg^k_{max})\big)$, where N_m denotes the number of meta-paths, L_m represents the maximum length of meta-paths, deg_{max} is the maximal degree of nodes in the heterogeneous subgraph. The time complexity of relation subgraph representation learning module is $O\big(|\mathcal{E}|(|\mathcal{R}| \cdot |\mathcal{V}_r| \cdot deg_{avg})\big)$, where $|\mathcal{R}|$ denotes the number of relations, $|\mathcal{V}_r|$ represents the number of nodes in the relation subgraph, deg_{avg} denotes the average number of neighbors in the relation subgraph. The time complexity of relation-aware dynamic structural evolution module is $O\big(|\mathcal{E}|(|\mathcal{T}|\cdot|\mathcal{R}|\cdot d_u)\big)$, where d_u denotes the dimension of hidden units in the GRU. The time complexity of the prototype alignment module is $O\big(|\mathcal{E}|(N_p \cdot d_p^2)\big)$, where N_p denotes the number of prototypes, d_p is the dimension of the prototype representations. Therefore, the overall complexity of ReadGraph is dominated by $O(|\mathcal{E}|(|\mathcal{V}_h| + |\mathcal{E}_h|))$.

□

A.5 Pseudocode

We show the pseudocode of our proposed ReadGraph in Algorithm 1. It begins by sampling edge-centric heterogeneous subgraphs using three strategies (meta-path, k-hop, and node-importance) and initializing node embeddings. Next, it learns relation subgraph representations through message passing and aggregation. The model then captures dynamic structural evolution using Gated Recurrent Units (GRUs) across snapshots. It aligns relation subgraph representations with prototypes. And the model optimizes the combined loss function to refine the anomaly predictions. The anomaly score for the target edge is computed based on relation-aware structural evolution modeling.

A.6 Ablation Study

We conduct an ablation study to assess the impact of key components in ReadGraph. The first variant ignores the meta-path enhanced modeling, denoted as ReadGraph$_{\text{nm}}$. The second variant excludes all losses except the cross-entropy loss, denoted as ReadGraph$_{\text{nl}}$. The last one ignores the dynamic structural evolution,

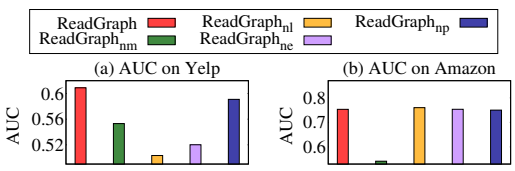

Figure 5: The ablation experiments of the ReadGraph model.

denoted as ReadGraph$_{ne}$. The fourth variant excludes the prototype alignment, denoted as ReadGraph$_{np}$. From Figure 5, we observe:

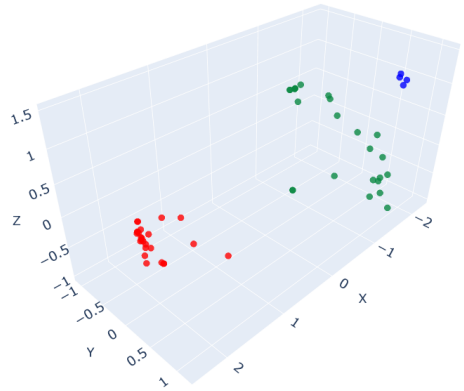

Figure 6: Case Study on Digg dataset, where the red points represent abnormal edges, green points denote normal edges, and blue points represent prototypes.

(1) ReadGraph is significantly more effective than ReadGraph$_{nm}$ on both datasets. It shows that the modeling of the meta-paths is necessary in the model.

(2) ReadGraph shows superior performance compared to ReadGraph$_{nl}$ on the Yelp dataset and performs comparably to ReadGraph$_{nl}$ on the Amazon dataset. This indicates that including additional losses enhances anomaly detection.

(3) ReadGraph has better performance than ReadGraph$_{ne}$ and has a significant effect on the Yelp dataset. It shows that the dynamic structural evolution plays an important role.

(4) ReadGraph has better results than ReadGraph$_{np}$, which shows that the prototype alignment part also plays an essential role in the model.

A.7 Case Study

We illustrate the results of our proposed ReadGraph through a case study. Using the PCA method, we map the high-dimensional embeddings of edges as well as prototypes into three dimensions, as shown in Fig 6. In the visualization, blue points denote the embeddings of prototypes, red points represent the embeddings of abnormal edges, and green points correspond to the embeddings of normal edges. The visualization reveals that the embeddings of normal edges are generally closer to the prototypes, while the embeddings of abnormal edges are more distant. This suggests that abnormal edge detection can be effectively guided by the prototypes.