Technical Appendix

A. 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 6, 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 ExpGraph model, which is not included in other models, our testing time remains comparable as other methods.

B. PROOFS

B.1 Convergence of Relation Subgraph Representation Learning

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

$$\mathbf{x}'_{\phi(e)}(v_i^t) = \sigma\left(\sum_{j\in\mathcal{N}_{\phi(e)}(v_i^t)} g_{ij}^{\phi(e)} \cdot \mathbf{x}_{\phi(e)}(v_j^t)\right), \quad (15)$$

where

$$g_{ij}^{\phi(e)} = \text{softmax}\left(w_{\phi(e)}^{\top} \cdot [\mathbf{x}(v_i^t)||\mathbf{x}(v_j^t)]\right). \quad (16)$$

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:

$$\left|g_{ij}^{\phi(e)}(\mathbf{x}^1) - g_{ij}^{\phi(e)}(\mathbf{x}^2)\right| \leq L\|\mathbf{x}^1 - \mathbf{x}^2\|, \quad (17)$$

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

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

Consider the message passing process as an iterative update:

$$\mathbf{x}_{\phi(e)}^{(k+1)}(v_i^t) = \sigma\left(\sum_{j\in\mathcal{N}_{\phi(e)}(v_i^t)} g_{ij}^{\phi(e)} \cdot \mathbf{x}_{\phi(e)}^{(k)}(v_j^t)\right). \quad (18)$$

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

$$\mathbf{x}_{\phi(e)}^*(v_i^t) = \sigma\left(\sum_{j\in\mathcal{N}_{\phi(e)}(v_i^t)} g_{ij}^{\phi(e)} \cdot \mathbf{x}_{\phi(e)}^*(v_j^t)\right). \quad (19)$$

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.

□

B.2 Stability of the Dynamic Structural Evolution

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

$$\begin{aligned}
\mathbf{h}_t &= \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \bar{\mathbf{h}}_t \\
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1}) \\
\bar{\mathbf{h}}_t &= \tanh(\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t \circ \mathbf{h}_{t-1})) \\
\mathbf{c}_t &= \sigma(\mathbf{W}_r \mathbf{m}_{r^t} + \mathbf{U}_r \mathbf{h}_{t-1}),
\end{aligned} \quad (20)$$

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:

$$|\sigma(\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1})| \leq L_\sigma \|\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1}\|, \quad (21)$$

$$|\tanh(\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t \circ \mathbf{h}_{t-1}))| \leq L_{\text{tanh}} \|\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t \circ \mathbf{h}_{t-1})\|. \quad (22)$$

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

We analyze the fixed point h_t^* such that:

$$\mathbf{h}_t^* = \mathbf{z}_t^* \circ \mathbf{h}_{t-1}^* + (1 - \mathbf{z}_t^*) \circ \bar{\mathbf{h}}_t^*, \quad (23)$$

where:

$$\mathbf{z}_t^* = \sigma(\mathbf{W}_z \mathbf{m}_{r^t} + \mathbf{U}_z \mathbf{h}_{t-1}^*), \quad (24)$$

$$\bar{\mathbf{h}}_t^* = \tanh(\mathbf{W}_h \mathbf{m}_{r^t} + \mathbf{U}_h(\mathbf{c}_t^* \circ \mathbf{h}_{t-1}^*)), \quad (25)$$

$$\mathbf{c}_t^* = \sigma(\mathbf{W}_r \mathbf{m}_{r^t} + \mathbf{U}_r \mathbf{h}_{t-1}^*). \quad (26)$$

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:

$$\Phi(\mathbf{h}_{t-1}) = \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \bar{\mathbf{h}}_t. \quad (27)$$

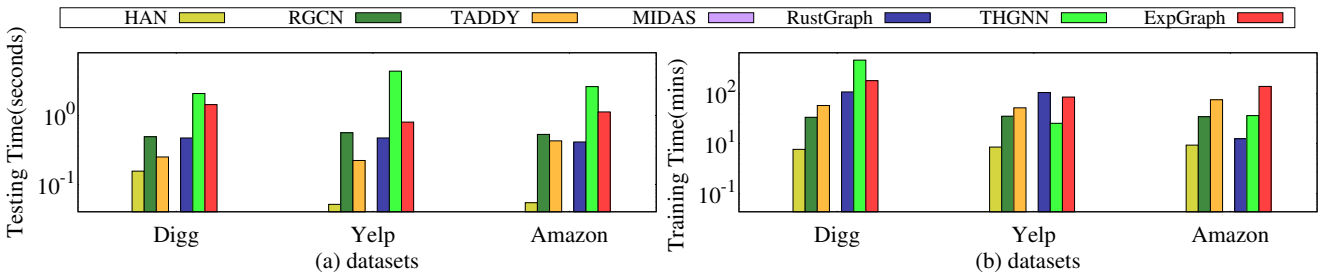

Figure 6: Efficiency of all methods on three datasets.

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

$$\|\Phi(\mathbf{h}_{t-1}) - \Phi(\mathbf{h}'_{t-1})\| \leq \alpha \|\mathbf{h}_{t-1} - \mathbf{h}'_{t-1}\|, \tag{28}$$

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

□

B.3 The Overall Complexity

Theorem 3. The overall complexity of ExpGraph is dominated by $\mathcal{O}\big(|\mathcal{E}|(|\mathcal{V}_h| + |\mathcal{E}_h|)\big)$, where $|\mathcal{V}_h|$ and $|\mathcal{E}_h|$ denotes 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 $\mathcal{O}\big(|\mathcal{E}|(|\mathcal{V}_h|+|\mathcal{E}_h|+N_m\cdot|\mathcal{V}_h|\cdot L_m + deg^k_{max})\big)$, 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 $\mathcal{O}\big(|\mathcal{E}|(|\mathcal{R}| \cdot |\mathcal{V}_r| \cdot deg_{avg})\big)$, 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 $\mathcal{O}\big(|\mathcal{E}|(|\mathcal{T}| \cdot |\mathcal{R}| \cdot d_u)\big)$, where d_u denotes the number of hidden units in the GRU. The time complexity of the explainable prototype alignment module is $\mathcal{O}\big(|\mathcal{E}|(N_p \cdot d_p^2)\big)$, where N_p denotes the number of prototypes, d_p is the dimension of the prototype representations. The time complexity of meta-path enhanced semantic learning module is $\mathcal{O}\big(|\mathcal{E}|(N_m \cdot |\mathcal{V}_h| \cdot d_n)\big)$, where d_n denotes the dimension of node embedding in node-level propagation. Therefore, the overall complexity of ExpGraph is dominated by $\mathcal{O}\big(|\mathcal{E}|(|\mathcal{V}_h| + |\mathcal{E}_h|)\big)$.

□

C. Pseudocode

We show the pseudocode of our proposed ExpGraph 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

Algorithm 1: The ExpGraph 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 $\mathcal{G}_s$
  $\mathcal{G}_e \leftarrow \mathcal{G}_e \cup \mathcal{G}_s$
end
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
end
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
end
Form sequential outputs $\mathbf{H}'$
```
/* Explainable Prototype Alignment           */
```
Initialize prototypes $\mathbf{P}^0$
for *each prototype layer* l **do**
  Update prototypes using Equation 6
end
Update prototype representations $\mathbf{H}_p$ using Equation 7
```
/* Meta-Path Enhanced Semantic Learning
   */
```
for *each meta-path* α **do**
  Update node presentation $\mathbf{m}_i^{\alpha t}$ using Equation 9
end
Aggregate node presentations to form $\mathbf{m}_s^t$ using Equation 10
Fuse snapshot-level representations to compute $\mathbf{m}_e$ using Equation 11
```
/* Optimization                              */
```
Compute loss $\mathcal{L}$ using Equation 14
Minimize $\mathcal{L}$ using optimization techniques until converge
```
/* Anomaly Prediction                        */
```
Compute $\hat{y}_e$ using Equation 13
return $\hat{y}_e$

dynamic structural evolution using Gated Recurrent Units (GRUs) across snapshots. It aligns relation subgraph representations with prototypes and enhances semantic learning through meta-paths. 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 and meta-path based semantic learning.

D. Reproducibility Checklist

This paper:

• Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes, in Section C. Pseudocode.)

• Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes)

• Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper (yes, in README of our code's GitHub repository)

Does this paper make theoretical contributions? (yes)

If yes, please complete the list below.

• All assumptions and restrictions are stated clearly and formally. (yes)

• All novel claims are stated formally (e.g., in theorem statements). (yes)

• Proofs of all novel claims are included. (yes)

• Proof sketches or intuitions are given for complex and/or novel results. (yes)

• Appropriate citations to theoretical tools used are given. (yes)

• All theoretical claims are demonstrated empirically to hold. (yes)

• All experimental code used to eliminate or disprove claims is included. (yes)

Does this paper rely on one or more datasets? (yes)

If yes, please complete the list below.

• A motivation is given for why the experiments are conducted on the selected datasets (yes)

• All novel datasets introduced in this paper are included in a data appendix. (yes)

• All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)

• All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (yes)

• All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (yes)

• All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (NA)

Does this paper include computational experiments? (yes)

If yes, please complete the list below.

• Any code required for pre-processing data is included in the appendix. (yes).

• All source code required for conducting and analyzing the experiments is included in a code appendix. (yes)

• All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)

• All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes)

• If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (NA)

• This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (yes)

• This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (yes)

• This paper states the number of algorithm runs used to compute each reported result. (yes)

• Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (no)

• The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (no)

• This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (partial)

• This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (yes)