

Temporal Motif-aware Graph Test-time Adaptation for OOD Blockchain Anomaly Detection

Runang He¹, Tongya Zheng^{1,2,3}, Huiling Peng¹, Yuanyu Wan^{1,3}, Bingde Hu^{1*},
Jiawei Chen^{1,3}, Canghong Jin², Mingli Song^{1,2,3}, Can Wang^{1,2,3},

¹State Key Laboratory of Blockchain and Data Security, Zhejiang University

²Zhejiang Provincial Engineering Research Center for Real-Time SmartTech in Urban Security
Governance, Hangzhou City University

³Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
herunang@zju.edu.cn, {doujiang_zheng, phlnku}@163.com, {wanyy, tonyhu, sleepyhunt}@zju.edu.cn,
jinch@hzcu.edu.cn, {brooksong, wcan}@zju.edu.cn

Abstract

Ever-evolving transaction patterns have significantly hindered anomaly detection on emerging cryptocurrency blockchains due to the vast number of addresses and diverse anomalous behaviors. Recently, advanced Graph Anomaly Detection (GAD) approaches applied to blockchains have faced two critical challenges: *adversarial pattern evolution by malicious actors* and the *out-of-distribution (OOD) problem caused by varied transaction semantics on blockchains*. To address these challenges, we propose a novel framework termed **TEMPoral Motif-aware Graph Test-Time Adaptation (TEMG-TTA)**. First, we comprehensively capture the 3-node temporal motif distribution of each active address using an efficient computational mechanism, enabling downstream temporal motif-aware graph learning. Second, we design a simple yet effective test-time adaptation strategy to facilitate the sharing of common patterns between training and testing graphs. Extensive experiments on 5 real-world datasets demonstrate that our proposed **TEMG-TTA** outperforms *state-of-the-art* GAD approaches by an average of 54.88%. A further case study on interpretable motif patterns reveals that **TEMG-TTA** explicitly characterizes the complex transaction patterns of anomalous addresses, thereby verifying the effectiveness of our technical designs. Our code is publicly available at <https://github.com/LuoXishuang0712/TEMG-TTA/>.

1 Introduction¹

Blockchain transaction networks consist of large-scale, directed, and temporally ordered interactions, where anomalous behaviors continuously evolve. This dynamic nature makes

*Corresponding Author

¹Due to space limitations, an extended version of this paper is available at arXiv:2605.29526.

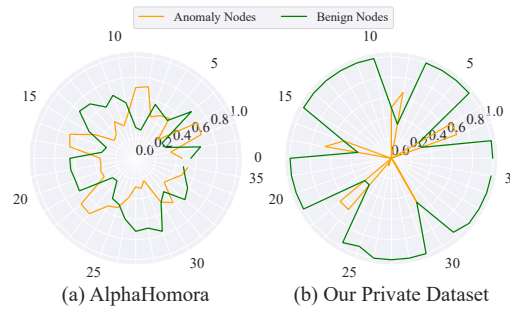


Figure 1: Radar maps of two types of blockchain transactions w.r.t. 36 types of 3-node temporal motifs.

blockchain anomaly detection particularly challenging under distribution shifts and limited supervision. Cryptocurrencies on decentralized blockchains, such as Bitcoin and Ethereum, are widely used and support millions of transactions daily². Although blockchain transparency enables public verification, it also allows illicit activities to hide within legitimate transaction flows, especially in programmable environments enabled by smart contracts [Chen *et al.*, 2020].

From a modeling perspective, blockchain transactions can be represented as directed temporal multi-graphs, motivating graph-based Blockchain Anomaly Detection (BAD) methods for asset flow analysis and anomalous group discovery [Qi *et al.*, 2023; Luo, 2024]. However, anomalous behaviors on blockchains are non-stationary and evolve with emerging cryptocurrency techniques and attack strategies [Yao *et al.*, 2024]. Thus, models trained on historical data often suffer from severe out-of-distribution (OOD) issues in real-world deployment.

Early detection efforts relied on hand-crafted features and traditional machine learning models to detect Ponzi schemes [Chen *et al.*, 2018], trace cryptocurrency scams [Phillips and Wilder, 2020], and identify phishing transactions [Yuan *et al.*, 2020]. With the development of deep graph learning, recent studies have shifted toward end-to-end representation learning for money laundering de-

²<https://etherscan.io/txs>

tection [Weber *et al.*, 2019] and account de-anonymization on Ethereum [Zhou *et al.*, 2022]. More generally, Graph Anomaly Detection (GAD) methods have addressed challenges such as heterophily and label imbalance, as exemplified by DGAGNN [Duan *et al.*, 2024] and SpaceGNN [Dong *et al.*, 2025]. Despite these advances, most existing methods rely on supervised learning and remain sensitive to distribution shifts in blockchain data.

Our closer inspection of transaction behaviors suggests that many complex patterns can be decomposed into fine-grained temporal motifs. As shown in Figure 1, anomalous and benign nodes exhibit distinct temporal motif patterns on both the AlphaHomora dataset and a real-world money-laundering dataset. Moreover, we find that improved detection performance is often accompanied by adaptive changes in adversarial behaviors, further exacerbating distribution shifts. These observations reveal two unresolved challenges: (i) existing GAD methods rarely model temporal motifs explicitly, limiting their ability to capture evolving adversarial patterns; (ii) supervised learning approaches struggle to adapt to structural changes under OOD settings.

In this work, we collaborate with Zhejiang Provincial Public Security Department, which provides reliable on-chain anomaly labels and cases, to detect OOD anomalies on blockchains using public transaction records and provided labels. Specifically, we propose a **TEmporal Motif-aware Graph Test-Time Adaptation (TEMG-TTA)** framework to enable explicit temporal motif perception for OOD blockchain anomaly detection. First, we design an efficient motif matching algorithm that reduces the time complexity from $\mathcal{O}(M^3)$ to $\mathcal{O}(M \cdot k^2)$, where M is the number of transactions and k is the maximum number of edges within a constrained time window. Next, we construct a comprehensive temporal motif representation for each active node by incorporating shared motif prototype embeddings, role representations, and positional encodings. Finally, we introduce a trustable node selection mask and a teacher-student regularization mechanism to facilitate graph TTA on blockchains and mitigate disruptive deviation. Extensive experiments on four public datasets and one private dataset demonstrate that **TEMG-TTA** significantly outperforms *state-of-the-art* GAD approaches by an average of 54.88%.

Our main contributions are summarized as follows:

- We collaborate with a Provincial Public Security Department to detect suspicious Ethereum transactions in real time and report anomalous addresses with specified patterns, such as anomalous exchangers.
- We propose **TEMG-TTA**, a temporal motif-aware graph test-time adaptation framework that improves structural expressiveness and robustness to temporal distribution shifts for blockchain anomaly detection.
- Extensive experiments on multiple real-world blockchain datasets show that our method consistently outperforms state-of-the-art baselines and remains robust under temporal distribution drift. In-depth motif analysis and ablation studies further demonstrate the effectiveness of different components.

2 Related Works

Graph Anomaly Detection. Graph neural networks (GNNs) have shown strong effectiveness in graph anomaly detection (GAD) [Qiao *et al.*, 2025b]. Recent studies address heterophily and imbalance in GAD, such as H₂GCN [Zhu *et al.*, 2020], PMP [Zhuo *et al.*, 2024], and ConsisGAD [Chen *et al.*, 2024]. Graph foundation models (GFMs), including UNPrompt [Niu *et al.*, 2024], ARC [Liu *et al.*, 2024], and AnomalyGFM [Qiao *et al.*, 2025a], further aim to improve generalization to complex abnormal patterns. However, these methods generally assume stationary distributions and struggle with continuously evolving blockchain transaction patterns.

Dynamic and Spatiotemporal Graph Modeling. Dynamic and spatiotemporal graph modeling has been widely studied for evolving behavioral patterns. Representative studies model temporal aggregation and propagation for dynamic graph representation [Zheng *et al.*, 2023], incorporate spatiotemporal graph structures for human mobility simulation [Wang *et al.*, 2024b], and use Transformer-based transfer learning for cross-city trajectory generation [Wang *et al.*, 2024a]. These methods highlight the importance of temporal evolution and transferable behavioral patterns, but mainly target mobility or general dynamic representation learning rather than label-scarce anomaly detection on rapidly evolving cryptocurrency transaction graphs.

Motif in Graph. Traditional GNNs aggregate pairwise connections and are limited by the expressive power of the 1-WL test [Lee *et al.*, 2019]. Graph motifs, as recurring higher-order subgraphs, have been used to enhance structural expressiveness, including motif-based attention [Lee *et al.*, 2019], motif-augmented attributed networks [Huang *et al.*, 2021], and motif-based GNNs [Monti *et al.*, 2018]. Nevertheless, most motif-based methods ignore temporal and directional information or require explicit graph augmentation, limiting their scalability on large dynamic transaction graphs.

Graph Test-time Adaptation. Test-time adaptation (TTA) adapts pre-trained models to distribution shifts during inference without labeled data. Existing graph TTA methods rely on min-max optimization for test-domain adaptation [Chen *et al.*, 2022], graph structure editing [Jin *et al.*, 2022], regularized prototype supervision [Zhao *et al.*, 2025], edge-importance-based graph augmentation [Zhang *et al.*, 2024], or homophily-based pseudo-label denoising [Zheng *et al.*, 2025]. However, they mainly handle global distribution shifts and may overlook fine-grained transactional patterns crucial for cryptocurrency fraud detection.

Summary of Differences. Overall, existing studies provide valuable foundations but remain insufficient for blockchain fraud detection. GAD and graph foundation models mainly rely on static representations, motif-based GNNs often suffer from temporal or scalability limitations, and graph TTA methods usually overlook fine-grained higher-order transaction structures. Moreover, real-time intelligent big data processing emphasizes scalable analysis over continuously generated large-scale streams [Zheng *et al.*, 2019], which is important for practical blockchain monitoring. Different

Algorithm 1 Motif Matching

Input: Transaction set $\mathcal{R}$

Parameter: Maximum time window t_w , edge limit k , aggregation range Δt

Output: Motif count matrix $\mathbf{C}$

```

1: Sort transactions in  $\mathcal{R}$  by timestamp
2: Initialize motif counts  $\mathbf{C}$ 
3: if time aggregation is enabled then
4:   Aggregate same-direction transactions within  $[t_i - \Delta t, t_i]$  into  $\tilde{\mathcal{R}}$ 
5: else
6:    $\tilde{\mathcal{R}} = \mathcal{R}$ 
7: end if
8: for each transaction  $r_i \in \mathcal{R}$  do
9:   Sample at most  $k$  historical transactions from  $\tilde{\mathcal{R}}$  within  $[t_i - t_w, t_i]$  as  $\mathcal{S}_i$ 
10:  for each pair  $(r_j, r_m)$  in  $\mathcal{S}_i$  with  $t_j < t_m < t_i$  do
11:    if  $\{r_j, r_m, r_i\}$  forms a 3-node temporal motif then
12:      Update the corresponding motif-role count in  $\mathbf{C}$ 
13:    end if
14:  end for
15: end for
16: return  $\mathbf{C}$ 

```

from these works, our approach jointly integrates motif-aware modeling with label-free test-time adaptation, enabling efficient adaptation to evolving higher-order transaction patterns in dynamic blockchain systems.

3 Method

3.1 Blockchain Transaction Graph

Given a cryptocurrency transaction set $\mathcal{R} = \{r_1, \dots, r_M\}$, each transaction is denoted as $r_i = (s_i, e_i, t_i, c_i)$, where s_i , e_i , t_i , and c_i denote the sender, receiver, timestamp, and amount, respectively. We formulate $\mathcal{R}$ as a directed temporal multi-graph $\mathcal{G} = (V, E, \mathbf{X})$, where $V = \{s_i\} \cup \{e_i\}$ is the address set with $|V| = N$, $E = \{r_i\}_{i=1}^M$ is the transaction set, and $\mathbf{X} \in \mathbb{R}^{N \times d}$ denotes node features. Our goal is to learn a GNN-based classifier $\hat{y}_i = \text{GNN}_\theta(v_i | \mathcal{G})$ to distinguish benign and anomalous addresses.

3.2 Temporal Motif-aware Graph Representation

Temporal Motif Computation. Existing GAD methods mainly rely on multi-hop graph aggregation and node-level classifiers, making it difficult to capture fine-grained intra- and inter-anomaly transaction patterns. We therefore introduce temporal motifs to characterize higher-order transaction behaviors and enhance node representations.

Motifs are recurring subgraph patterns. In this work, we focus on 3-node and 3-edge temporal motifs, which capture interactions beyond pairwise connections while remaining computationally tractable [Paranjape *et al.*, 2017], as shown in Figure 3. Two-node patterns can be captured by ordinary GNN aggregation, whereas four-node motifs are much more expensive and harder to represent compactly. Algorithm 1 shows our motif matching procedure. With edge limitation

and time aggregation, the complexity is reduced from $\mathcal{O}(M^3)$ to $\mathcal{O}(M \cdot k^2)$. The resulting motif-role counts are stored in $\mathbf{C} \in \mathbb{R}^{N \times (3 \times 36)}$.

Temporal Motif Representation. Given 36 temporal motif types, we learn a shared motif prototype matrix $\mathbf{M} = [\mathbf{m}_1, \dots, \mathbf{m}_{36}]$. For each motif type m_k and each node role q , we construct a role-aware motif embedding using the motif prototype, a learnable role embedding $\mathbf{r}_{k,q}$, and a positional encoding $\mathbf{p}_{k,q}$. For a motif $m_k = (V', E')$, where $E' = \{e_j\}_{j=1}^3$ and $e_j = (e_j^{src}, e_j^{dst}, t_j)$, the positional encoding of node v_q is defined as

$$\mathbf{p}_{k,q} = \sum_{e_j \in E'} \mathbb{1}(v_q \in e_j) \cdot (\mathcal{T}(t_j) + \mathcal{E}(v_q, e_j)), \quad (1)$$

where $\mathcal{T}(t_j) = \sin(\omega t_j + \theta)$ encodes temporal order, and $\mathcal{E}(v_q, e_j) = \text{MLP}([\mathbb{1}(v_q = e_j^{src}), \mathbb{1}(v_q = e_j^{dst})])$ encodes edge-level source/destination roles. The motif-role embedding is then obtained by

$$\mathbf{h}_{k,q}^m = \mathbf{m}_k + \mathbf{r}_{k,q} + \mathbf{p}_{k,q}. \quad (2)$$

We collect all motif-role embeddings into $\mathbf{H}_m \in \mathbb{R}^{(3 \times 36) \times d_h}$ and compute node-level motif features as

$$\mathbf{X}_{motif} = \mathbf{C} \mathbf{H}_m. \quad (3)$$

Temporal Motif-enhanced Node Representation. We fuse original node features with motif features by

$$\mathbf{X} = \text{MLP}(W_f \mathbf{X}_{orig} + \mathbf{X}_{motif} + b), \quad (4)$$

where W_f and b are learnable parameters, and $\mathbf{X}$ denotes the fused node features.

3.3 Graph Pre-training

We divide learning into graph pre-training and test-time adaptation. In pre-training, the motif-enhanced graph is fed into a GNN backbone, which can be GCN [Kipf, 2016], GraphSAGE [Hamilton *et al.*, 2017], SpaceGNN [Dong *et al.*, 2025], or DGAGNN [Duan *et al.*, 2024]. The general GNN pipeline is written as

$$\hat{y}_i = \text{MLP} \left(\text{Update}^L \left(\{\mathbf{z}_{v_j}^{L-1} \mid v_j \in \{v_i\} \cup \mathcal{N}(v_i)\} \right) \right), \quad (5)$$

where Update^L is the L -th layer update function, $\mathbf{z}_{v_i}^L$ is the representation of v_i , and $\mathbf{z}_{v_i}^0 = \mathbf{x}_i \in \mathbf{X}$. The model is optimized with binary cross-entropy:

$$\mathcal{L}_{tr} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]. \quad (6)$$

We optimize θ by back-propagation and apply early stopping based on validation AUC-PRC with tolerance $\tau_{tolerance}$.

3.4 Graph Test-time Adaptation for Blockchain

In test-time adaptation (TTA), we adapt the pre-trained model to an unlabeled test graph $\mathcal{G}_{test}$ whose distribution differs from the training graph. We initialize two identical GNNs from the pre-trained model: a student GNN_θ^S for adaptation and a teacher GNN_θ^T for stable supervision.

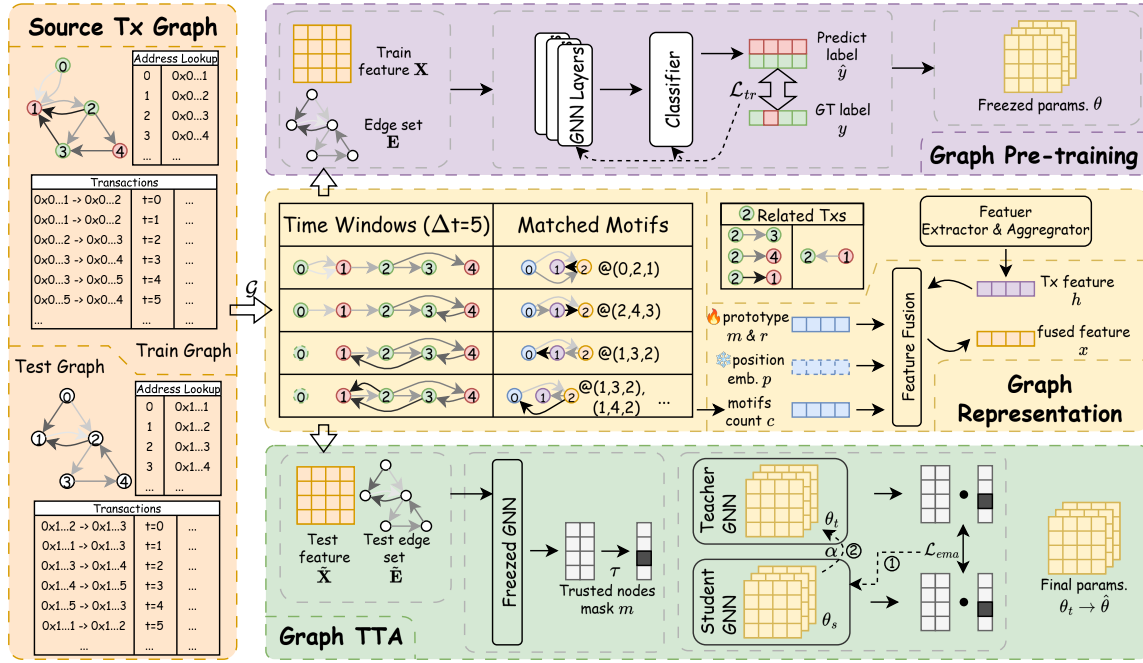


Figure 2: Overall framework of our proposed TEMG-TTA.

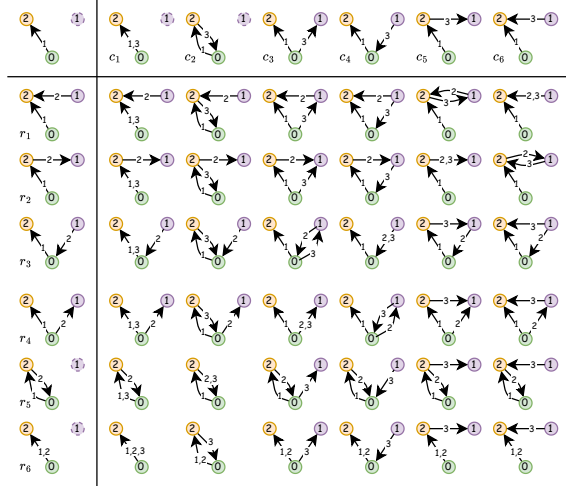


Figure 3: All possible directed motifs with 3 nodes and 3 edges.

To avoid over-confident pseudo supervision [Wu *et al.*, 2020], we select reliable but non-trivial nodes using a confidence mask:

$$B_i = \mathbb{1}(\tau_{low} < p_i \leq \tau_{high}), \quad (7)$$

where $p_i = \max \text{softmax}(\hat{y}_i^T)$ is the teacher confidence, and $B = \{i \mid B_i = 1\}$. At each adaptation step, we compute

$$(\hat{Y}^T, H^T) = \text{GNN}_{\theta}^T(\mathcal{G}_{test}), \quad (\hat{Y}^S, H^S) = \text{GNN}_{\theta}^S(\tilde{\mathcal{G}}_{test}), \quad (8)$$

where $\tilde{\mathcal{G}}_{test}$ is obtained by random edge dropping to produce a semantic-preserving perturbation. We regularize the student

by matching teacher and student embeddings:

$$\mathcal{L}_{sim} = \frac{1}{|B|} \sum_{i \in B} (1 - \cos(\mathbf{h}_i^T, \mathbf{h}_i^S)). \quad (9)$$

To improve robustness to unstable blockchain patterns, we further adopt an InfoNCE loss [Oord *et al.*, 2018]:

$$\mathcal{L}_{Info} = -\frac{1}{|B|} \sum_{i \in B} \log \frac{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)}{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+) + \sum_{\mathbf{h}_k^- \in \mathcal{N}_i^-} \text{sim}(\mathbf{h}_i, \mathbf{h}_k^-)}, \quad (10)$$

where $\text{sim}(\mathbf{h}, \mathbf{h}') = \exp(\cos(\mathbf{h}, \mathbf{h}')/\tau)$. For each node v_i , positives are sampled from nodes with similar motif features, i.e., $\cos(\mathbf{x}_{m_i}, \mathbf{x}_{m_j}) > \gamma$, while the remaining sampled nodes are treated as negatives $\mathcal{N}_i^-$. The final TTA objective is

$$\mathcal{L}_{TTA} = \mathcal{L}_{sim} + \beta \mathcal{L}_{Info}, \quad (11)$$

where β controls the contrastive regularization strength. We update the student parameters θ^S by back-propagation and update the teacher by exponential moving average:

$$\theta^T = \alpha \theta^T + (1 - \alpha) \theta^S. \quad (12)$$

With $\alpha \in [0.9, 0.999]$, the teacher preserves source-domain knowledge while gradually tracking the adapted student, reducing catastrophic forgetting during TTA. During each training progress, α is fixed.

4 Experiments

4.1 Experiment Setting

Datasets. We conduct experiments on five real-world blockchain anomaly detection datasets, as shown in Table 1. AlphaHomora (Alpha), CryptopiaHacker (Crypto), PlusTokenPonzi (Plus), and UpbitHack (Upbit) are from [Lin *et al.*, 2024], while our private dataset Trace is collected through collaboration with our real-world partner.

Dataset	#Nodes	#Edges	%Anomaly	Description
Alpha	115,488	897,308	8.07	Exploitation
Crypto	222,761	835,302	4.38	Hacker Attack
Plus	38,327	93,493	80.34	Ponzi Scheme
Upbit	577,994	1,213,049	3.24	Hacker Attack
Trace	1,698,331	2,969,691	0.16	Money Laundering

Table 1: Statistics of blockchain anomaly detection datasets.

Baselines. We evaluate **TEMG-TTA** with several GNN backbones, including Graph Convolutional Network (GCN) [Kipf, 2016], GraphSAGE (SAGE) [Hamilton *et al.*, 2017], SpaceGNN (SGNN) [Dong *et al.*, 2025], and DGAGNN (DGA) [Duan *et al.*, 2024]. We use GAD-Bench [Tang *et al.*, 2023] implementations for the first three models and the official implementation for DGAGNN. We also attempted temporal GNN baselines such as DyGFormer [Yu *et al.*, 2023] and SALoM [Liu *et al.*, 2026], but their training cost is prohibitive on blockchain transaction graphs, with several runs exceeding 24 hours on public datasets and becoming harder on Trace. Following prior GAD studies, we therefore use representative GNN-based anomaly detection backbones for the main comparison and focus on the consistent gains brought by **TEMG-TTA**.

Evaluation protocol. To prevent data leakage, we record the activation time of each anomalous node and split the training, validation, and test sets chronologically with a ratio of 6:2:2. Due to severe class imbalance, we do not use accuracy. Instead, we evaluate anomaly detection performance using Area Under the Precision-Recall Curve (AUC-PRC), Recall@ k (Rec@ k), and F1-score. AUC-PRC measures the overall ranking quality under imbalance, Rec@ k evaluates the coverage of high-risk nodes in practical top- k inspection, and F1-score reflects the balance between precision and recall under a decision threshold of 0.5. We set k as the number of anomalous nodes in the test set.

Implementation details. We train each model for 200 epochs with early stopping based on validation AUC-PRC and run 10 epochs for TTA. Unless otherwise specified, motif extraction uses $k = 100$ and $\Delta t = 3600$. Since TTA requires hidden embeddings on the test graph, we modify all backbones to output the hidden embeddings before the classifier. For DGAGNN, we disable its source graph oriented super mask during TTA. All experiments are conducted on 2 Intel Xeon Platinum 8260L CPUs, 256 GiB RAM, and an NVIDIA GeForce RTX 4090 GPU with 24 GiB VRAM.

4.2 Overall Comparison

Table 2 compares the performance of “Original” models without TTA and “Ours” with **TEMG-TTA**. Here, “-” denotes out-of-memory (OOM) errors, “tr.” and “te.” denote “train on” and “test on”, respectively. Overall, **TEMG-TTA** preserves the anomaly detection ability learned from the source dataset while adapting the model to shifted test distributions.

Following Table 1, we use the anomaly ratio of each test dataset as the random AUC-PRC baseline. When the original model performs poorly, such as transferring from CryptopiaHacker to PlusTokenPonzi, the source and target datasets ex-

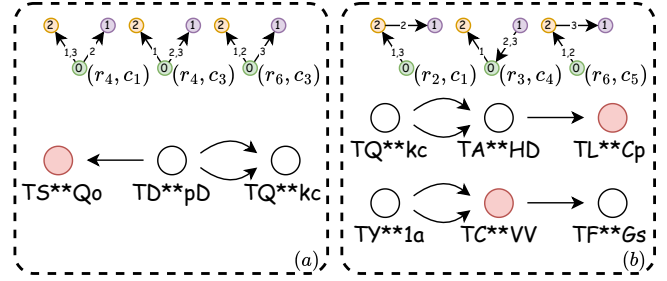


Figure 4: Common temporal motif patterns in real-world blockchain datasets.

hibit clear distribution shifts, and **TEMG-TTA** consistently improves performance through test-time adaptation. When the original model already performs well, such as transferring from AlphaHomora to CryptopiaHacker, **TEMG-TTA** still maintains or further improves the performance, indicating that the adaptation process does not severely damage source-domain knowledge.

The OOM cases mainly occur when combining the large Trace dataset with SpaceGNN. During TTA, maintaining both teacher and student models further increases memory consumption, making SpaceGNN infeasible on Trace under our hardware setup.

4.3 Case Study on Temporal Motif Patterns

To verify whether temporal motifs can capture meaningful anomalous behaviors, we analyze common motif patterns in our private dataset. As shown in Figure 4, we group motifs by their basic structures and present representative cases verified by our partner.

Group (a) shows a distribution pattern from node-0. In our cases, the most common receiver in this pattern is a private exchange service frequently used in fraud cases, while the other receiver is the suspicious node labeled as fraud. Although such distribution patterns are common on blockchains, they become strong fraud signals within a short time window because malicious actors often avoid using the same account for a long period.

Group (b) shows aggregation patterns, where assets are collected into one account before payment. Due to temporal-order constraints, only motif (r_6, c_3) clearly indicates this behavior. In the first case, the middle node aggregates assets from an exchange service and sends them to the fraud-related target node. In the second case, the aggregation node itself is suspicious, corresponding to a cash-out pattern according to our partner. This indicates that fraud-related addresses may repeatedly receive assets from upstream addresses and forward them downstream, forming distinctive temporal aggregation motifs.

4.4 Ablation Study

Temporal Motif. To verify the effectiveness of the proposed motif representation, we conduct ablation experiments on four public datasets with different GNN backbones. As shown in Table 3, motif representation consistently improves anomaly detection performance, especially on AlphaHomora

GNN	Metric	TTA	Tr. Te.	Alpha				Crypto				Plus			
				Crypto	Plus	Upbit	Trace	Alpha	Plus	Upbit	Trace	Alpha	Crypto	Upbit	Trace
GCN	AUC-PRC	Origin		0.2292	0.6241	0.1752	0.0047	0.5552	0.6333	0.1101	0.0022	0.1490	0.0562	0.0974	0.0012
		Ours		0.3636 ±0.0008 **	0.7647 ±0.0128 *	0.4457 ±0.0016 **	0.0399 ±0.0002 **	0.6439 ±0.0006 **	0.8159 ±0.0169 *	0.3037 ±0.0007 **	0.0060 ±0.0000 **	0.1673 ±0.0005 **	0.0613 ±0.0001 **	0.1013 ±0.0001 **	0.0012 ±0.0000 **
	Rec@ <i>k</i>	Origin		0.2838	0.7562	0.1961	0.0000	0.4978	0.7576	0.2323	0.0000	0.3246	0.2274	0.3049	0.0007
		Ours		0.3466 ±0.0011 **	0.7680 ±0.0000 **	0.4705 ±0.0005 **	0.1044 ±0.0002 **	0.5833 ±0.0001 **	0.7624 ±0.0001 **	0.4319 ±0.0002 **	0.0000 ±0.0000 **	0.0000 ±0.0001	0.0000 ±0.0000	0.0000 ±0.0000	0.0018 ±0.0000 **
	F1	Origin		0.3062	0.8910	0.4183	0.0106	0.5354	0.8910	0.2929	0.0044	0.2516	0.1154	0.1844	0.0032
		Ours		0.3787 ±0.0000	0.8910 ±0.0000	0.5004 ±0.0004 **	0.1461 ±0.0004 **	0.5964 ±0.0004 **	0.8911 ±0.0000 **	0.4422 ±0.0009 **	0.0567 ±0.0002 **	0.2855 ±0.0008 **	0.1260 ±0.0000 **	0.1922 ±0.0001 **	0.0034 ±0.0000 **
SAGE	AUC-PRC	Origin		0.1946	0.8278	0.0664	0.0021	0.3541	0.6145	0.1047	0.0024	0.1093	0.0456	0.0503	0.0017
		Ours		0.2080 ±0.0007 *	0.9985 ±0.0000 **	0.2429 ±0.0011 **	0.0023 ±0.0000 **	0.4880 ±0.0016 **	0.9907 ±0.0000 **	0.0832 ±0.0007	0.0015 ±0.0000	0.1195 ±0.0000 **	0.0557 ±0.0000 **	0.0777 ±0.0000 **	0.0017 ±0.0000
	Rec@ <i>k</i>	Origin		0.2269	0.9396	0.0000	0.0018	0.4567	0.7690	0.0616	0.0089	0.0000	0.0000	0.0000	0.0000
		Ours		0.2847 ±0.0057 *	0.9926 ±0.0002 **	0.3843 ±0.0007 **	0.0111 ±0.0000 **	0.5400 ±0.0029 **	0.9855 ±0.0000 **	0.0145 ±0.0034	0.0000 ±0.0000	0.0001 ±0.0000 **	0.0000 ±0.0000	0.0000 ±0.0000	0.0000 ±0.0000
	F1	Origin		0.4004	0.9924	0.3588	0.0050	0.4734	0.8970	0.1654	0.0032	0.2119	0.1119	0.1354	0.0036
		Ours		0.4138 ±0.0033 *	0.9940 ±0.0002 *	0.3975 ±0.0009 **	0.0147 ±0.0000 **	0.5518 ±0.0017 **	0.9925 ±0.0000 **	0.1808 ±0.0006 **	0.0037 ±0.0000 **	0.2139 ±0.0000 **	0.1117 ±0.0000	0.1460 ±0.0000 **	0.0039 ±0.0000 **
DGA	AUC-PRC	Origin		0.2181	0.7195	0.1121	0.0020	0.3709	0.9869	0.1331	0.0017	0.0652	0.0705	0.0343	0.0053
		Ours		0.4152 ±0.0010 **	0.9931 ±0.0002 **	0.4510 ±0.0046 **	0.0174 ±0.0001 **	0.5100 ±0.0079 *	0.9922 ±0.0000 **	0.2153 ±0.0004 **	0.0184 ±0.0009 *	0.0984 ±0.0035 *	0.1017 ±0.0021 *	0.2802 ±0.0137 *	0.0091 ±0.0001 **
	Rec@ <i>k</i>	Origin		0.2883	0.7564	0.2334	0.0037	0.3840	0.9383	0.2890	0.0000	0.0677	0.0845	0.0384	0.0126
		Ours		0.4125 ±0.0029 **	0.9819 ±0.0017 **	0.4313 ±0.0022 **	0.0334 ±0.0002 **	0.5263 ±0.0071 *	0.9852 ±0.0000 **	0.3872 ±0.0051 *	0.0435 ±0.0011 **	0.0813 ±0.0097	0.1635 ±0.0017 **	0.2792 ±0.0104 **	0.0479 ±0.0003 **
	F1	Origin		0.3399	0.8910	0.3962	0.0212	0.3592	0.9897	0.2826	0.0032	0.1530	0.0945	0.0632	0.0056
		Ours		0.4413 ±0.0005 **	0.9825 ±0.0013 **	0.4684 ±0.0023 **	0.0568 ±0.0007 **	0.5334 ±0.0072 **	0.9925 ±0.0000 **	0.4141 ±0.0053 **	0.0700 ±0.0029 **	0.2056 ±0.0026 *	0.1888 ±0.0009 **	0.3244 ±0.0165 *	0.0558 ±0.0002 **
SGNN	AUC-PRC	Origin		0.5077	0.9963	0.3772	0.0256	0.6418	0.9895	0.7074	0.0033	0.2108	0.1065	0.2055	0.0054
		Ours		0.5904 ±0.0001 **	0.9956 ±0.0000	0.7443 ±0.0003 **	-	0.6094 ±0.0005	0.9974 ±0.0000 **	0.6972 ±0.0007	-	0.2221 ±0.0000 **	0.1324 ±0.0000 **	0.3398 ±0.0002 **	-
	Rec@ <i>k</i>	Origin		0.4915	0.9915	0.4070	0.0588	0.6374	0.9795	0.6693	0.0052	0.0000	0.0000	0.0000	0.0044
		Ours		0.5486 ±0.0007 **	0.9937 ±0.0001 **	0.6698 ±0.0000 **	-	0.5832 ±0.0006	0.9875 ±0.0000 **	0.6470 ±0.0007	-	0.0000 ±0.0000	0.0000 ±0.0000	0.2857 ±0.0554 *	-
	F1	Origin		0.5747	0.9917	0.6165	0.0162	0.5535	0.9440	0.5199	0.0087	0.3842	0.2529	0.4727	0.0899
		Ours		0.5763 ±0.0004 *	0.9942 ±0.0001 **	0.6933 ±0.0002 **	-	0.5852 ±0.0004 **	0.9921 ±0.0000 **	0.6561 ±0.0005 **	-	0.3674 ±0.0000	0.5009 ±0.0001	0.5009 ±0.0001 **	-

Table 2: Overall comparison results. Models are pre-trained on AlphaHomora, CryptopiaHacker, and PlusTokenPonzi. ** indicates $p < 0.001$ and * indicates $p < 0.1$ in the paired t-test compared with the baseline.

Dataset	Metric	GCN	SAGE	DGA	SGNN
Alpha	AUC-PRC	0.7173	0.7368	0.6756	0.7713
	w/o motif	0.6034	0.7214	0.6323	0.7753
	Rec@ <i>k</i>	0.6676	0.6849	0.6224	0.7159
	w/o motif	0.5653	0.6758	0.5839	0.7179
Crypto	AUC-PRC	0.5713	0.4611	0.4768	0.7051
	w/o motif	0.4869	0.5223	0.4404	0.7061
	Rec@ <i>k</i>	0.5655	0.5158	0.4526	0.6588
	w/o motif	0.4745	0.5164	0.4474	0.6614
Plus	AUC-PRC	0.9960	0.9919	0.9931	0.9987
	w/o motif	0.9968	0.9944	0.9955	0.9993
	Rec@ <i>k</i>	0.9908	0.9736	0.9798	0.9966
	w/o motif	0.9869	0.9902	0.9881	0.9968
Upbit	AUC-PRC	0.4346	0.4114	0.3553	0.7329
	w/o motif	0.4023	0.5221	0.3480	0.7177
	Rec@ <i>k</i>	0.4788	0.4482	0.4100	0.7173
	w/o motif	0.4783	0.5357	0.4528	0.7063

Table 3: Ablation study on GNNs with motif representation.

and UpbitHack. GCN obtains the most significant improvement, suggesting that motif features can compensate for its limited structural expressiveness. The minor improvement on PlusTokenPonzi may be due to its high anomaly ratio and nearly saturated prediction performance, leaving limited room for further gains.

4.5 Parameter Analysis

Edge limit k and time aggregation range Δt . We analyze the impact of k and Δt on motif representation, with results shown in Table 4. In most settings, a proper edge limit and time aggregation range improve anomaly detection performance. The edge limit focuses motif extraction on temporally relevant transactions, while time aggregation reduces duplicate transaction patterns. We further evaluate the runtime efficiency of motif matching on real-world transaction graphs, as reported in Table 5. The results show that k and Δt substantially reduce computational cost, demonstrating the practical scalability of our motif extraction method on large-scale blockchain datasets with over 1 million nodes and 2 million edges.

In deployment, **TEMG-TTA** runs on localized k -hop sub-graphs around seed accounts instead of the full blockchain graph. Motif matching costs about 0.12 ms per edge, and forward/backward passes take 40.25/169.69 ms, leading to second-level latency under Ethereum-scale throughput and supporting real-time on-chain fraud detection.

Teacher model momentum α . The teacher momentum α controls how much information the teacher model receives from the student model. We evaluate different values of α using GCN. As shown in Table 6, a larger α usually weakens TTA performance, such as in the transfer from AlphaHomora to PlusTokenPonzi, because it slows down knowledge adaptation from the student model.

Model			GCN				SAGE				SGNN			
dataset	k	Δt	None	1800	3600	7200	None	1800	3600	7200	None	1800	3600	7200
Alpha	None		-	0.6077	0.7172	0.7117	-	0.5876	0.7310	0.7213	-	0.7512	0.7677	0.7699
	50		0.4357	0.6406	<u>0.7195</u>	0.7144	0.3928	0.6606	0.7254	0.7209	0.7347	0.7215	0.7742	0.7720
	100		0.3952	0.6556	0.7205	0.7145	0.2397	0.6503	<u>0.7327</u>	0.7268	0.7373	0.7494	<u>0.7727</u>	0.7695
	200		0.3549	0.6448	0.7111	0.7160	0.2801	0.6006	0.7338	0.7282	0.7312	0.7520	0.7675	0.7673
Crypto	None		-	0.4214	0.5774	0.5855	-	0.3050	0.5268	0.4944	-	0.7013	0.6945	0.6844
	50		0.3136	0.5264	0.5276	0.5610	0.1302	0.4030	0.4903	0.5666	0.6734	0.6860	0.6840	0.6812
	100		0.3118	0.3962	0.5749	0.5594	0.1303	0.2982	0.4830	0.4931	0.6824	0.6853	0.6986	<u>0.7033</u>
	200		0.2625	0.3664	0.5473	0.5663	0.1012	0.2064	0.5205	<u>0.5504</u>	0.6725	0.7105	0.6969	0.6856
Plus	None		-	0.9972	0.9970	0.9974	-	0.9909	0.9952	0.9925	-	0.9983	0.9988	0.9985
	50		0.9957	<u>0.9975</u>	0.9964	0.9962	0.9930	0.9932	0.9914	0.9930	0.9987	0.9981	0.9986	0.9987
	100		0.9957	0.9966	0.9964	0.9957	0.9938	0.9926	0.9920	0.9934	<u>0.9987</u>	0.9984	<u>0.9987</u>	0.9985
	200		0.9953	0.9977	0.9973	0.9969	0.9932	0.9934	0.9944	<u>0.9949</u>	<u>0.9987</u>	0.9983	<u>0.9987</u>	0.9985
Upbit	None		-	0.2074	0.3719	0.3946	-	0.3636	0.3288	0.3317	-	0.7418	0.7406	0.7349
	50		0.1676	0.3889	0.4376	0.4294	0.1016	0.4120	0.4408	0.4783	0.7347	0.7379	0.7294	0.7322
	100		0.1553	0.3690	0.4045	0.4132	0.0407	0.4156	0.4390	0.4398	0.7317	0.7337	0.7345	0.7375
	200		0.1297	0.3170	0.3972	0.4128	0.0308	0.3069	0.3248	0.3483	0.7324	0.7364	0.7343	<u>0.7410</u>

Table 4: Parameter analysis results of motif matching in AUC-PRC.

Dataset	w/ k (s)	w/ Δt (s)	w/ $k, \Delta t$ (s)
Alpha	1,307.01	174.06	176.04
Crypto	2,134.01	129.03	139.08
Plus	20.08	21.02	19.02
Upbit	462.07	63.10	61.08
Trace	732.01	3,005.07	349.05

Table 5: Running time of motif matching.

α	Alpha			Crypto		
	Crypto	Plus	Upbit	Crypto	Plus	Upbit
0.9	0.3850	0.8501	0.2564	0.6253	0.9713	0.4108
0.99	0.3889	0.8283	0.2672	0.6241	0.9712	0.4300
0.999	0.3892	0.8242	0.2678	0.6240	0.9713	0.4312

Table 6: AUC-PRC results of the momentum coefficient α .

Trustable node selection thresholds τ_{low} and τ_{high} . The thresholds τ_{low} and τ_{high} define the confidence range for selecting trustable nodes, as summarized in Table 7. Increasing τ_{high} or decreasing τ_{low} generally expands the selected node set and helps the model capture more useful adaptation signals. However, this benefit only holds within a proper range: an excessively high τ_{high} may introduce over-confident predictions, while an overly low τ_{low} may include noisy low-confidence nodes, both of which degrade TTA performance.

5 Conclusion

In this work, we propose **TEMG-TTA**, a temporal motif-aware graph test-time adaptation framework for blockchain anomaly detection. **TEMG-TTA** integrates 3-node temporal motif representations with original node features to capture fine-grained higher-order transaction patterns. It further introduces a simple yet effective test-time adaptation strategy to mitigate distribution shifts caused by evolving malicious behaviors and adversarial transaction patterns.

τ_{low}	τ_{high}	Alpha			Crypto		
		Crypto	Plus	Upbit	Crypto	Plus	Upbit
0.5	0.8	0.3309	0.9471	0.3030	0.6359	0.8290	0.2983
	0.9	0.3295	0.9509	0.3061	0.6367	0.8308	0.2978
	0.95	0.3281	0.9469	0.3123	0.6366	0.8291	0.2981
0.7	0.8	0.3270	0.9616	0.3066	0.6360	0.8288	0.2972
	0.9	0.3288	0.9438	0.3034	0.6357	0.8245	0.2977
	0.95	0.3257	0.9222	0.3058	0.6349	0.8305	0.2979
0.9	0.95	0.3255	0.9499	0.2969	0.6363	0.8285	0.2997

Table 7: AUC-PRC results of the trustable node threshold τ .

Extensive experiments on real-world blockchain datasets demonstrate that **TEMG-TTA** consistently outperforms classical and *state-of-the-art* graph anomaly detection models. The case study on temporal motif patterns provides interpretable evidence that our method can characterize meaningful fraud-related transaction structures. Ablation studies further verify the effectiveness of both temporal motif representation and test-time adaptation, showing how different components contribute to the overall performance.

Beyond experimental evaluation, **TEMG-TTA** has also been validated in collaboration with a public security department. In 2025, it was applied to real-world blockchain transaction data and supported the detection of suspicious transactions involving more than 1 million USDT. This practical validation demonstrates the potential of **TEMG-TTA** for real-time on-chain risk monitoring and provides evidence of its real-world impact in blockchain fraud detection.

For future work, it is promising to incorporate large language models (LLMs) into blockchain anomaly detection, given their strong zero-shot inference and natural-language understanding capabilities. LLMs could help integrate heterogeneous information sources, such as smart-contract code, transaction metadata, and off-chain reports, thereby improving detection performance while providing more interpretable explanations for detected anomalies.

Acknowledgments

This work is supported by the Zhejiang Province “Jian-BingLingYan+X” Research and Development Plan (2025C02020).

References

- [Chen *et al.*, 2018] Weili Chen, Zibin Zheng, Jiahui Cui, Edith Ngai, Peilin Zheng, and Yuren Zhou. Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In *Proceedings of the 2018 world wide web conference*, pages 1409–1418, 2018.
- [Chen *et al.*, 2020] Weili Chen, Tuo Zhang, Zhiguang Chen, Zibin Zheng, and Yutong Lu. Traveling the token world: A graph analysis of ethereum erc20 token ecosystem. In *Proceedings of The Web Conference 2020*, pages 1411–1421, 2020.
- [Chen *et al.*, 2022] Guanzi Chen, Jiying Zhang, Xi Xiao, and Yang Li. Graphtta: Test time adaptation on graph neural networks. *arXiv preprint arXiv:2208.09126*, 2022.
- [Chen *et al.*, 2024] Nan Chen, Zemin Liu, Bryan Hooi, Bingsheng He, Rizal Fathony, Jun Hu, and Jia Chen. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *The twelfth international conference on learning representations*, 2024.
- [Dong *et al.*, 2025] Xiangyu Dong, Xingyi Zhang, Lei Chen, Mingxuan Yuan, and Sibow Wang. Spacegcn: Multi-space graph neural network for node anomaly detection with extremely limited labels. *arXiv preprint arXiv:2502.03201*, 2025.
- [Duan *et al.*, 2024] Mingjiang Duan, Tongya Zheng, Yang Gao, Gang Wang, Zunlei Feng, and Xinyu Wang. Dga-gnn: Dynamic grouping aggregation gnn for fraud detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 11820–11828, 2024.
- [Hamilton *et al.*, 2017] Will Hamilton, Zitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- [Huang *et al.*, 2021] Ling Huang, Ye Zhu, Yuefang Gao, Tuo Liu, Chao Chang, Caixing Liu, Yong Tang, and Changdong Wang. Hybrid-order anomaly detection on attributed networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12249–12263, 2021.
- [Jin *et al.*, 2022] Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering graph representation learning with test-time graph transformation. *arXiv preprint arXiv:2210.03561*, 2022.
- [Kipf, 2016] TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [Lee *et al.*, 2019] John Boaz Lee, Ryan A Rossi, Xiangnan Kong, Sungchul Kim, Eunye Koh, and Anup Rao. Graph convolutional networks with motif-based attention. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 499–508, 2019.
- [Lin *et al.*, 2024] Dan Lin, Jiajing Wu, Yunmei Yu, Qishuang Fu, Zibin Zheng, and Changlin Yang. Denseflow: Spotting cryptocurrency money laundering in ethereum transaction graphs. In *Proceedings of the ACM Web Conference 2024*, pages 4429–4438, 2024.
- [Liu *et al.*, 2024] Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan. Arc: A generalist graph anomaly detector with in-context learning. *Advances in Neural Information Processing Systems*, 37:50772–50804, 2024.
- [Liu *et al.*, 2026] Hanwen Liu, Longjiao Zhang, Rui Wang, Tongya Zheng, Sai Wu, Chang Yao, and Mingli Song. Salom: Structure aware temporal graph networks with long-short memory updater. *Advances in Neural Information Processing Systems*, 38:22843–22871, 2026.
- [Luo, 2024] Bingqiao Luo. When crypto economics meet graph analytics and learning. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1186–1189, 2024.
- [Monti *et al.*, 2018] Federico Monti, Karl Otness, and Michael M Bronstein. Motifnet: a motif-based graph convolutional network for directed graphs. In *2018 IEEE data science workshop (DSW)*, pages 225–228. IEEE, 2018.
- [Niu *et al.*, 2024] Chaoxi Niu, Hezhe Qiao, Changlu Chen, Ling Chen, and Guansong Pang. Zero-shot generalist graph anomaly detection with unified neighborhood prompts. *arXiv preprint arXiv:2410.14886*, 2024.
- [Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [Paranjape *et al.*, 2017] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 601–610, 2017.
- [Phillips and Wilder, 2020] Ross Phillips and Heidi Wilder. Tracing cryptocurrency scams: Clustering replicated advance-fee and phishing websites. In *2020 IEEE international conference on blockchain and cryptocurrency (ICBC)*, pages 1–8. IEEE, 2020.
- [Qi *et al.*, 2023] Yuxin Qi, Jun Wu, Hansong Xu, and Mohsen Guizani. Blockchain data mining with graph learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):729–748, 2023.
- [Qiao *et al.*, 2025a] Hezhe Qiao, Chaoxi Niu, Ling Chen, and Guansong Pang. Anomalygfm: Graph foundation model for zero/few-shot anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2326–2337, 2025.
- [Qiao *et al.*, 2025b] Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. Deep graph anomaly detection: A survey and new perspectives.

- IEEE Transactions on Knowledge and Data Engineering*, 2025.
- [Tang *et al.*, 2023] Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Advances in Neural Information Processing Systems*, 36:29628–29653, 2023.
- [Wang *et al.*, 2024a] Yu Wang, Tongya Zheng, Yuxuan Liang, Shunyu Liu, and Mingli Song. Cola: Cross-city mobility transformer for human trajectory simulation. In *Proceedings of the ACM on Web Conference 2024*, pages 3509–3520, 2024.
- [Wang *et al.*, 2024b] Yu Wang, Tongya Zheng, Shunyu Liu, Zunlei Feng, Kaixuan Chen, Yunzhi Hao, and Mingli Song. Spatiotemporal-augmented graph neural networks for human mobility simulation. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7074–7086, 2024.
- [Weber *et al.*, 2019] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.
- [Wu *et al.*, 2020] Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. Conditional negative sampling for contrastive learning of visual representations. *arXiv preprint arXiv:2010.02037*, 2020.
- [Yao *et al.*, 2024] Mingxuan Yao, Runze Zhang, Haichuan Xu, Shih-Huan Chou, Varun Chowdhary Paturi, Amit Kumar Sikder, and Brendan Saltaformaggio. Pulling off the mask: Forensic analysis of the deceptive creator wallets behind smart contract fraud. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 2236–2254. IEEE, 2024.
- [Yu *et al.*, 2023] Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.
- [Yuan *et al.*, 2020] Zihao Yuan, Qi Yuan, and Jiajing Wu. Phishing detection on ethereum via learning representation of transaction subgraphs. In *International conference on blockchain and trustworthy systems*, pages 178–191. Springer, 2020.
- [Zhang *et al.*, 2024] Jiaxin Zhang, Yiqi Wang, Xihong Yang, and En Zhu. A fully test-time training framework for semi-supervised node classification on out-of-distribution graphs. *ACM Transactions on Knowledge Discovery from Data*, 18(7):1–19, 2024.
- [Zhao *et al.*, 2025] Yusheng Zhao, Qixin Zhang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. Test-time adaptation on graphs via adaptive subgraph-based selection and regularized prototypes. In *Forty-second International Conference on Machine Learning*, 2025.
- [Zheng *et al.*, 2019] Tongya Zheng, Gang Chen, Xinyu Wang, Chun Chen, Xingen Wang, and Sihui Luo. Real-time intelligent big data processing: technology, platform, and applications. *Science China Information Sciences*, 62(8):82101, 2019.
- [Zheng *et al.*, 2023] Tongya Zheng, Xinchao Wang, Zunlei Feng, Jie Song, Yunzhi Hao, Mingli Song, Xingen Wang, Xinyu Wang, and Chun Chen. Temporal aggregation and propagation graph neural networks for dynamic representation. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10151–10165, 2023.
- [Zheng *et al.*, 2025] Xin Zheng, Wei Huang, Chuan Zhou, Ming Li, and Shirui Pan. Test-time graph neural dataset search with generative projection. In *Forty-second International Conference on Machine Learning*, 2025.
- [Zhou *et al.*, 2022] Jiajun Zhou, Chenkai Hu, Jianlei Chi, Jiajing Wu, Meng Shen, and Qi Xuan. Behavior-aware account de-anonymization on ethereum interaction graph. *IEEE Transactions on Information Forensics and Security*, 17:3433–3448, 2022.
- [Zhu *et al.*, 2020] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.
- [Zhuo *et al.*, 2024] Wei Zhuo, Zemin Liu, Bryan Hooi, Bingsheng He, Guang Tan, Rizal Fathony, and Jia Chen. Partitioning message passing for graph fraud detection. *arXiv preprint arXiv:2412.00020*, 2024.