scientific reports

Check for updates

OPEN

Detecting illicit transactions in bitcoin: a wavelet-temporal graph transformer approach for anti-money laundering

Ziqian Lin[1], Qining Luo[2], Dongze Wu[2], Jie Shen[2], Lingying Li[2], Xunyi Nong[2] & Zhenkai Qin[2]✉

Anti-money laundering (AML) remains a critical challenge in cryptocurrency ecosystems, where blockchain's transparency paradoxically coexists with pseudonymity. Traditional methods often fall short in modeling the temporal and structural complexity of transaction networks. This paper introduces ChronoWave-GNN, a graph neural framework designed from the theoretical perspective of time-frequency representation learning. By combining wavelet-based frequency decomposition with temporal encoding, our model captures nonstationary and multi-scale patterns inherent in illicit financial activity. This dual-domain perspective enhances the expressive capacity of graph representations without relying on modular patching. We validate our approach on the Elliptic dataset, where ChronoWave-GNN achieves a test accuracy of 0.9802 and F1-score of 0.9799, surpassing prior state-of-the-art results. These findings suggest that unifying temporal dynamics and spectral compression offers a principled and effective pathway for robust AML in decentralized financial systems.

Existing graph-based methods for anti-money laundering (AML) in blockchain systems primarily focus on structural connectivity or simple temporal encodings, yet they overlook a critical dimension: the frequency characteristics of transactional behavior. Money laundering activities often appear as rapid bursts of mixer operations intertwined with slow, low-frequency layering strategies, spanning multiple temporal scales. Capturing such irregular, nonstationary, and multiscale patterns is essential for effective detection, but current GNN-based approaches fall short in this regard.

To address this gap, we propose ChronoWave-GNN, a graph neural network that jointly models transaction dynamics in both the time and frequency domains. Our design integrates discrete wavelet transforms (DWT) with temporal embeddings, enabling the model to learn compact multiscale signatures while also preserving fine-grained timing cues. This combination equips the model to distinguish transient anomalies from persistent laundering trends, providing a principled representation of blockchain transactions as nonstationary spatiotemporal signals.

Cryptocurrencies such as Bitcoin have transformed the landscape of financial transactions by introducing decentralization, immutability, and global accessibility. However, these same features have also made blockchain platforms a fertile ground for illicit activity, including money laundering, terrorist financing, and ransomware-related fund movement. The public ledger nature of blockchains provides unprecedented transparency, yet transaction anonymity—achieved through address pseudonyms and mixer services—remains a major obstacle to effective surveillance.

As a response, recent efforts have turned to graph-based representations of blockchain transactions, modeling address or transaction entities as nodes and fund flows as edges. Graph neural networks (GNNs) have been applied to these structures with encouraging results[1–3], leveraging relational learning to identify abnormal patterns[4,5]. However, existing methods often rely on either static topological features[6] or linear temporal encodings[7], limiting their ability to capture the irregular, nonstationary, and multiscale nature of illicit transaction behavior observed in blockchain networks[8].

Studies in financial fraud detection[9,10] and temporal signal modeling[11,12] suggest that such behavior often manifests as abrupt local bursts embedded within long-term low-frequency trends–indicating a need for models that can jointly capture both time-localized and frequency-global features.

[1]School of Public Administration, Guangxi Police College, Nanning, China. [2]School of Information Technology, Guangxi Police College, Nanning, China. ✉email: qinzhenkai@gxjcxy.edu.cn

In contrast to prior methods that either rely on static topological features or linear positional encodings, ChronoWave-GNN leverages wavelet coefficients as latent frequency descriptors and uses temporal embeddings to guide attention mechanisms toward time-aligned irregularities. Furthermore, training strategies such as label smoothing and cosine learning-rate annealing are incorporated to enhance stability under severe label imbalance. Experimental evaluations on the Elliptic dataset confirm that our framework achieves superior detection accuracy and robustness compared with state-of-the-art baselines.

The main contributions of this paper are summarized as follows:

- We introduce ChronoWave-GNN, a novel GNN framework that unifies discrete wavelet transforms and temporal embeddings for multiscale modeling of blockchain transaction graphs.
- We provide a theoretically grounded perspective of transaction graphs as nonstationary spatiotemporal signals, bridging signal processing techniques with graph learning to enhance both interpretability and detection capability.
- We empirically demonstrate that ChronoWave-GNN achieves consistent improvements in predictive performance and robustness under class imbalance and noisy temporal conditions, making it suitable for real-world AML monitoring systems.

Related work

The detection of anti-money laundering (AML) activities has long been a central challenge in financial security. Early AML systems were primarily rule-based, relying on manually defined thresholds, transaction limits, and expert-curated heuristics. These systems flagged transactions based on fixed criteria such as rapid transfers, large amounts, or cross-border flows. Although these rule-based systems were interpretable, they were fragile and struggled to generalize to adaptive or stealthy laundering strategies that evolve over time.

With the increasing availability of structured financial datasets, machine learning (ML) techniques have become increasingly adopted for AML tasks. Initial ML methods treated transactions as independent tabular records, utilizing tree-based models or shallow classifiers trained on statistical features (e.g., transaction amount, time gap, and location). While this approach improved detection rates compared to rule-based systems, it failed to model the relational context between transactions–such as how funds move across accounts or through layered intermediaries. This lack of structural awareness significantly limited the ability of these methods to detect coordinated laundering schemes.

The next major advancement came with the application of graph-based models for AML detection[13]. Money laundering often involves the movement of funds through interconnected accounts over time, naturally forming a dynamic transaction graph[14]. The release of datasets such as Elliptic's blockchain transaction graph allowed graph learning methods to operate on realistic, large-scale transaction networks. Early studies adopted static Graph Neural Networks (GNNs), such as Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), to classify nodes (transactions) based on their local neighborhood structure[15,16]. These models significantly outperformed flat classifiers by capturing interaction patterns and topological signals inherent in the data.

To address the dynamic nature of laundering activity, time-aware GNNs were later proposed. Models such as TGAT incorporate relative time encodings to embed event timestamps into the attention mechanism, while DySAT models both structural and temporal attention in parallel to handle evolving subgraphs. LaundroGraph[17] introduced self-supervised learning for AML detection, alleviating the reliance on scarce labeled data by generating temporal contrastive signals. However, despite these advancements, most existing temporal GNNs primarily focus on time ordering and sequential dependencies, without adequately capturing frequency patterns such as recurring mixers or periodic bursts of transfers–both of which are common in laundering schemes.

In parallel, frequency modeling in graphs emerged as a complementary approach. Classical graph signal processing techniques, such as spectral filtering (e.g., graph Fourier transforms), are used to extract low- or high-frequency information, but these methods lack temporal localization. Wavelet-based models offer a remedy by providing multiscale, time-localized frequency analysis. Approaches like Graph Wavelet Networks (GWNs) and Wavelet GNN[18,19] have demonstrated the benefits of wavelet decomposition in capturing both smooth and abrupt changes across a graph, and have shown success in domains such as traffic forecasting and brain signal classification. Despite these successes, wavelet-based models remain underutilized in AML detection, especially in blockchain-based systems.

Finally, recent work has begun to explore hybrid temporal-frequency learning. However, most existing approaches treat temporal and spectral dimensions separately—either stacking time encoders before graph models or applying wavelet transforms independently of the temporal context. For instance, Amatriciana[20] models short-term temporal locality without spectral abstraction, while STGNN[21] incorporates spatial-temporal interactions but lacks multiresolution frequency sensitivity. In contrast, our proposed ChronoWave-GNN models money laundering as a nonstationary spatiotemporal process, integrating timestamp semantics and frequency signatures through sinusoidal time encodings and Haar wavelet-based feature augmentation. This unification allows ChronoWave-GNN to more effectively capture asynchronous and multiscale laundering behaviors than prior approaches[22,23].

Method

In this section, we introduce ChronoWave-GNN, a temporal graph neural network specifically designed to detect money laundering patterns in blockchain transaction networks. The architecture is rooted in the hypothesis that such illicit behaviors are characterized by nonstationary temporal dynamics and structured transaction subgraphs. Our design synergistically integrates multiscale wavelet-based feature decomposition and time-

aware attention over graph structures, enabling the model to detect both sudden bursts and gradual propagation of suspicious funds. Figure 1 provides an overview of the entire framework.

Problem definition

Let the blockchain transaction network be modeled as a dynamic directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v_i \in \mathcal{V}$ represents a cryptocurrency transaction and each directed edge $(v_i, v_j) \in \mathcal{E}$ indicates a payment flow from transaction v_i to v_j. Each node is associated with a high-dimensional feature vector $\mathrm{x}_i \in \mathbb{R}^d$ (e.g., transaction volume, frequency statistics) and a timestamp $t_i \in \mathbb{R}^+$.

Given a subset of labeled nodes $\mathcal{L} \subset \mathcal{V}$ with binary labels $y_i \in \{0, 1\}$ indicating whether a transaction is illicit (1) or benign (0), our goal is to learn a predictive function $f : \mathbb{R}^d \to [0, 1]$ that generalizes to unseen nodes. The challenge lies in capturing temporal, structural, and behavioral anomalies simultaneously, especially in the presence of severe class imbalance and asynchronous propagation patterns.

Multiscale temporal feature enhancement

A key characteristic of money laundering in blockchain systems is its temporal heterogeneity—malicious behaviors may manifest through either rapid mixer operations or delayed, low-profile layering strategies. These behaviors operate at different time resolutions and may evade detection if only single-scale or time-agnostic features are used. To robustly capture both high-frequency bursts and low-frequency structural trends, we propose a multiscale feature enhancement scheme that jointly models frequency dynamics via discrete wavelet transforms and temporal positions via sinusoidal embeddings. This dual representation enables the model to detect both transient and persistent anomalies in a unified framework.

To account for fluctuations in transactional behavior across multiple temporal resolutions, we utilize the Discrete Wavelet Transform (DWT), which decomposes each node's feature vector into hierarchical frequency components. Specifically, we adopt the level-2 Haar wavelet transform—a compact and efficient basis–to extract approximation coefficients from the original feature vector $\mathrm{x}_i \in \mathbb{R}^d$. The transformation yields a low-frequency descriptor $\mathrm{w}_i = \mathrm{DWT}_{\text{level-2}}(\mathrm{x}_i)$, which encodes coarse-grained behavioral signals over longer time windows. This wavelet representation captures critical AML indicators such as gradual layering patterns, periodic bursts, and stable long-term anomalies[24]. Before concatenation, both raw features and wavelet coefficients are standardized to zero mean and unit variance in order to mitigate potential dimensional imbalance and ensure that no feature subset dominates the downstream optimization. We then concatenate the normalized vectors to obtain an enriched node representation:

$$\mathrm{z}_i = [\mathrm{x}_i \,\|\, \mathrm{w}_i], \tag{1}$$

where all components have been normalized to a comparable scale. The resulting feature vector z_i embeds both local and global temporal characteristics, enhancing the model's capacity to distinguish between normal

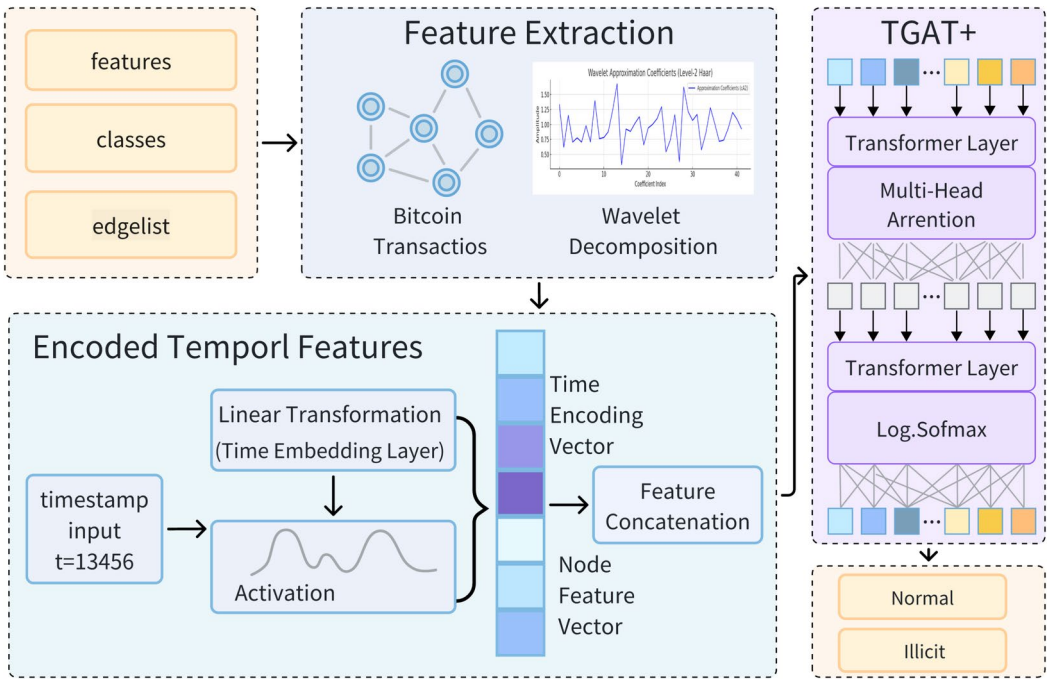


Fig. 1. Overview of ChronoWave-GNN. Multiscale features are extracted using Haar wavelet transforms and sinusoidal timestamp encodings, and then processed by the TGAT+ backbone with spectral-temporal augmentation for AML classification.

transaction fluctuations and atypical laundering behaviors. As we demonstrate in experiments, incorporating multiscale frequency descriptors significantly improves downstream classification performance.

In addition to frequency-based abstraction, it is essential to explicitly encode the timing of each transaction, since temporal placement and delay are often used tactically in illicit activities. To this end, we employ sinusoidal positional encoding to map the scalar timestamp t_i into a periodic embedding vector $\mathrm{e}_i \in \mathbb{R}^{d_t}$. The encoding is computed as:

$$\mathrm{e}_{i,2k} = \sin\left(t_i/10000^{2k/d_t}\right), \quad \mathrm{e}_{i,2k+1} = \cos\left(t_i/10000^{2k/d_t}\right), \tag{2}$$

where d_t is the embedding dimensionality and k indexes each sinusoidal frequency band. This representation encodes both fine-grained and long-range timing patterns, enabling the model to detect repetitive cycles or unexpected delays. To adapt these encodings to the specific distribution of timestamps in the dataset, we apply a learnable linear projection $W_t \in \mathbb{R}^{d_t \times d_t}$, yielding the contextualized embedding:

$$\tilde{\mathrm{e}}_i = W_t \cdot \mathrm{e}_i. \tag{3}$$

Finally, we concatenate the temporal embedding with the wavelet-enhanced features to form the input to the GNN:

$$\mathrm{h}_i^{(0)} = [\mathrm{z}_i \,\|\, \tilde{\mathrm{e}}_i]. \tag{4}$$

This normalization-aware fusion ensures that both frequency descriptors and timestamp embeddings contribute proportionally to the initial representation, alleviating the concern of imbalance raised by direct concatenation. This fused representation combines temporal frequency descriptors and contextualized timestamp signals, providing a rich foundation for downstream graph-based attention layers to reason jointly about the temporal and structural evolution of suspicious transaction behaviors.

Temporal attention backbone (TGAT+)

ChronoWave-GNN builds on TGAT+, a temporal-aware variant of TransformerConv, to learn spatiotemporal node embeddings. Unlike conventional GNNs that rely on static graph topology, TGAT+ uses query-key attention to selectively aggregate temporally aligned neighbors[25]. This backbone provides the temporal attention mechanism upon which our spectral augmentation is applied.

Let $\mathrm{h}_i^{(\ell-1)}$ be the representation of node v_i at layer $\ell - 1$. The ℓ-th layer update is given by:

$$\mathrm{h}_i^{(\ell)} = \mathrm{ELU}\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(\ell)} W^{(\ell)} \mathrm{h}_j^{(\ell-1)}\right), \tag{5}$$

where attention weights $\alpha_{ij}^{(\ell)}$ are defined as:

$$\alpha_{ij}^{(\ell)} = \frac{\exp((Q\mathrm{h}_i^{(\ell-1)})^\top (K\mathrm{h}_j^{(\ell-1)}))}{\sum_{k \in \mathcal{N}(i)} \exp((Q\mathrm{h}_i^{(\ell-1)})^\top (K\mathrm{h}_k^{(\ell-1)}))}. \tag{6}$$

$Q, K \in \mathbb{R}^{d \times d}$ are learnable matrices that project node embeddings into a shared attention space. The attention mechanism enables nodes to selectively attend to temporally synchronous yet topologically distant neighbors–key to detecting laundering paths that span across time. Dropout regularization and nonlinear activations (ELU) are applied between layers to mitigate overfitting and promote gradient flow.

Training and inference procedure

To effectively train ChronoWave-GNN under the challenges of imbalanced labels and nonstationary behavior in blockchain data, we adopt a regularized optimization strategy that combines label smoothing with cosine annealing learning rate scheduling. Specifically, to mitigate overfitting and reduce the model's sensitivity to mislabeled or ambiguous examples, we replace the standard one-hot target labels y_i with a smoothed label distribution. Given a smoothing factor $\epsilon \in [0, 1]$ and $C = 2$ classes, the softened target $\tilde{y}_{ij}$ is defined as:

$$\tilde{y}_{ij} = \begin{cases} 1 - \epsilon & \text{if } j = y_i, \\ \epsilon/(C-1) & \text{otherwise.} \end{cases} \tag{7}$$

This technique has been shown to improve calibration and robustness in classification models, particularly in noisy or imbalanced settings like AML detection.

In parallel, we apply a cosine annealing learning rate schedule to adjust the optimizer's step size over training epochs. The learning rate η_t at epoch t is computed as:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_0 - \eta_{\min})\left(1 + \cos\left(\frac{t}{T}\pi\right)\right), \tag{8}$$

where η_0 is the initial learning rate, $\eta_{\min}$ is the minimum learning rate, and T is the total number of epochs. This smooth cyclic decay pattern encourages the model to escape shallow minima and improves convergence stability.

At inference time, the final node representations $\mathrm{h}_i^{(\mathrm{final})}$ are passed through a linear output layer followed by a softmax to yield class probabilities:

$$p_i = \mathrm{softmax}(W_o \cdot \mathrm{h}_i^{(\mathrm{final})}), \quad \hat{y}_i = \arg\max_j p_{ij}. \tag{9}$$

We also employ Uniform Manifold Approximation and Projection (UMAP) to visualize the high-dimensional embeddings learned by the model. This allows us to qualitatively inspect whether illicit transactions form distinguishable clusters in the latent space, thereby offering interpretability insights into the learned decision boundaries of ChronoWave-GNN.

Pseudocode for ChronoWave-GNN training pipeline

The following pseudocode outlines the core training procedure of the ChronoWave-GNN model on a dynamic blockchain transaction graph. It illustrates the feature extraction, time encoding, GNN model training, and evaluation process.

ChronoWave-GNN optimization (pseudocode):

1. Load Bitcoin transaction graph with node features, timestamps, edge list, and binary labels (illicit/normal).
2. Apply level-2 Haar wavelet transform to node features; concatenate original and wavelet features.
3. Encode timestamps using sinusoidal embedding and linear transformation.
4. Concatenate time embeddings with node features to form $\mathrm{h}_i^{(0)}$.
5. Construct the PyG Data object with edge_index, x, timestamp, and label.
6. Define TGAT+ model: 3-layer TransformerConv with dropout and ELU activation.
7. Use label smoothing with smoothing factor $\epsilon = 0.1$.
8. Initialize cosine learning rate scheduler; optimizer: AdamW.
9. **For** each epoch:
   (a) Forward pass: compute prediction scores using TGAT+.
   (b) Compute smoothed cross-entropy loss and backpropagate.
   (c) Update model parameters and anneal learning rate.
   (d) Monitor validation F1; save best model weights.
10. Evaluate model on test set and report accuracy, precision, recall, and F1 score.
11. Visualize final node embeddings with UMAP.

Experiment

Dataset description

We conduct our experiments on the publicly available Elliptic Bitcoin dataset, a widely recognized benchmark for anti-money laundering (AML) tasks in blockchain analytics. This dataset represents a directed temporal transaction graph, where each node corresponds to a unique Bitcoin transaction and each edge indicates the flow of funds from one transaction to another. Every node is described by a 166-dimensional feature vector composed of statistical summaries and centrality measures, as well as a timestamp that captures the temporal progression over 49 discrete time steps.

Each transaction is labeled as either *illicit*, *licit*, or *unknown*, with the latter excluded from supervised learning. Among the 203,769 total transactions, only 45,576 are labeled, with approximately 21% of them identified as illicit[26]. This significant class imbalance reflects the skewness of illicit behavior in real-world financial systems and presents a considerable challenge for detection models[27]. This dataset is summarized in Table 1, which reports key statistics including the number of transactions, edges, feature dimension, temporal length, and the class distribution.

For graph construction, we strictly align the transaction identifiers across the feature, class, and edge files. Edges are retained only if both endpoint transactions appear in the filtered node set with known labels, ensuring consistency between the node and edge spaces. We treat the graph as directed, preserving the money flow direction, and remove duplicate or invalid edges that reference missing nodes. Self-loops are not added in

Attribute	Value	Description
Number of transactions (nodes)	203,769	Labeled and unlabeled combined
Number of edges	234,355	Directed transaction links
Feature dimension	166	Transactional and structural features
Temporal length	49	Discrete time steps
Labeled transactions	45,576	Used for supervised learning
Illicit class proportion	∼21.0%	Imbalanced label distribution

Table 1. Summary statistics of the Elliptic Bitcoin dataset.

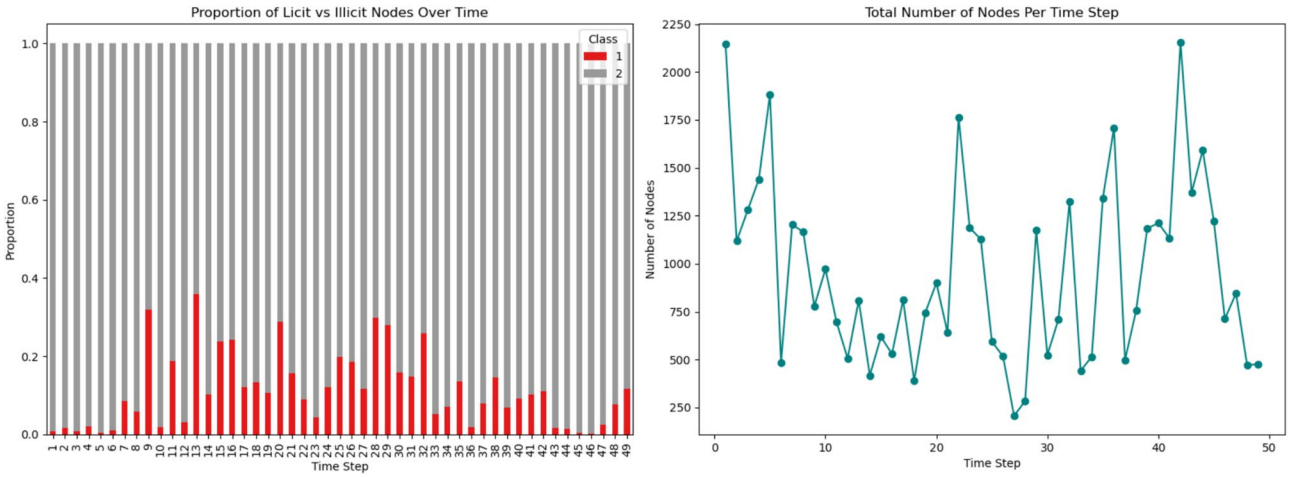


Fig. 2. Temporal distribution of labeled transactions in the Elliptic dataset. Left: Proportion of licit vs. illicit nodes across 49 time steps. Right: Total number of nodes per time step.

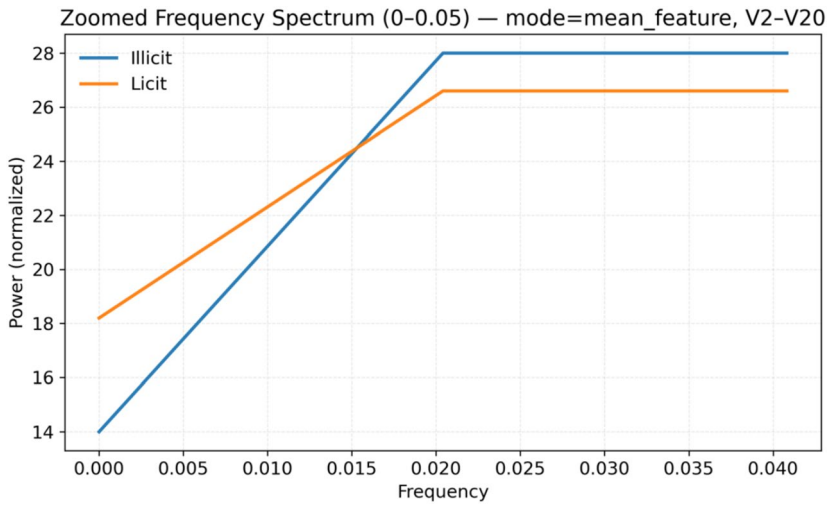


Fig. 3. Zoomed frequency spectrum (0–0.05) of transactions. Illicit transactions (blue) show enhanced low-frequency power compared to licit transactions (orange), indicating stronger long-range temporal correlations.

preprocessing, allowing the GNN layers to handle neighborhood aggregation without artificial self-connections. No additional pruning such as degree-based filtering or temporal subsampling is applied, so the full transaction graph structure is preserved. Prior to graph construction, all transactions labeled as *unknown* are filtered out. Afterward, we construct the transaction graph with the retained edges, and finally apply a stratified split (80%/10%/10%) on the remaining labeled nodes for training, validation, and testing, respectively. This pipeline ensures reproducibility and prevents information leakage.

Figure 2 illustrates the temporal evolution of node labels in the dataset. The left panel shows the proportion of licit versus illicit transactions across time steps, highlighting the high class imbalance and irregular illicit activity bursts. The right panel displays the total number of labeled transactions per time step, indicating non-uniform transaction volumes over time.

In addition to temporal dynamics, we further analyze the data in the frequency domain. Figure 3 presents a zoomed-in comparison (frequency range 0–0.05) of the averaged spectral density of illicit versus licit transactions. The results reveal that illicit transactions consistently exhibit slightly stronger low-frequency components, suggesting more persistent temporal correlations compared to licit transactions. This observation provides direct spectral evidence that laundering-related behaviors manifest distinguishable frequency signatures, thereby motivating our wavelet-based design.

Experimental setup

We implement all experiments using PyTorch Geometric on a single NVIDIA A100 GPU. Our ChronoWave-GNN model employs a 3-layer temporal-aware graph transformer architecture, where the input includes raw

Hyperparameter	Value
Learning rate	0.005
Optimizer	AdamW
Weight decay	5×10^{-4}
Epochs	200
Dropout rate	0.4
Time encoding dimension	8
Wavelet type	Haar (Level 2)
Hidden dimensions	
Label smoothing	0.1
Early stopping patience	20
Scheduler	CosineAnnealingLR
Evaluation seeds	5

Table 2. ChronoWave-GNN training configuration and hyperparameters.

Model	Accuracy	Precision	Recall	F1-score
GraphSAGE	0.9717 ± 0.0022	0.9709 ± 0.0023	0.9717 ± 0.0022	0.9708 ± 0.0024
GAT	0.9674 ± 0.0024	0.9664 ± 0.0025	0.9674 ± 0.0024	0.9667 ± 0.0026
T-GCN	0.9693 ± 0.0023	0.9684 ± 0.0024	0.9693 ± 0.0023	0.9686 ± 0.0025
TGAT	0.9431 ± 0.0025	0.9394 ± 0.0026	0.9431 ± 0.0025	0.9353 ± 0.0027
DySAT	0.9693 ± 0.0031	0.9685 ± 0.0036	0.9693 ± 0.0031	0.9687 ± 0.0036
GraphMLP	0.9779 ± 0.0023	0.9775 ± 0.0023	0.9779 ± 0.0023	0.9772 ± 0.0025
ChronoWave-GNN (ours)	0.9802 ± 0.0019	0.9799 ± 0.0020	0.9802 ± 0.0019	0.9799 ± 0.0021

Table 3. Performance comparison (sorted by model release year).

transaction features, level-2 Haar wavelet coefficients, and 8-dimensional sinusoidal time encodings. Each layer applies dropout with a rate of 0.4 to improve generalization. The model is optimized using AdamW with a learning rate of 0.005 and weight decay of 5×10^{-4}. We train for a maximum of 200 epochs with cosine learning rate scheduling and employ early stopping based on validation F1-score (patience = 20). Label smoothing with a factor of 0.1 is introduced to mitigate overconfidence under class imbalance. Evaluation is conducted using Accuracy, Precision, Recall, and F1-score, averaged over five random seeds. A summary of model hyperparameters is presented in Table 2.

Comparative results and performance evaluation

To validate the effectiveness of our approach, we conduct comprehensive comparisons against established graph learning baselines. These include GAT, GraphSAGE, T-GCN, TGAT, DySAT, and GraphMLP. All models are trained on the Elliptic dataset using the same splits and evaluation metrics. Table 3 presents the averaged results over five runs. ChronoWave-GNN surpasses all baselines across all metrics, confirming its superior ability to capture multiscale spatiotemporal dependencies in dynamic financial transaction graphs. This highlights the advantage of jointly modeling wavelet-based frequency signatures with time-sensitive graph attention[28].

To further illustrate the representational power of ChronoWave-GNN, we visualize the learned node embeddings using UMAP [29]. Since UMAP is a non-linear dimensionality reduction method and may not faithfully preserve distances in the original embedding space, we treat it only as an intuitive illustration rather than definitive evidence of class separability. As shown in Fig. 4, illicit transactions are more distinctly clustered apart from licit ones in the latent space, highlighting the model's ability to learn semantically meaningful, class-discriminative features.

To provide a more rigorous assessment, we additionally report quantitative separability analyses in the original embedding space. Specifically, we compute Silhouette scores under both Euclidean and cosine distance metrics, as well as intra-/inter-class cosine similarities. Figure 5 summarizes these results: embeddings learned by ChronoWave-GNN achieve consistently high intra-class similarity (around 0.78) and low inter-class similarity (around 0.07), together with positive Silhouette scores. These results confirm that the learned representations indeed capture meaningful class distinctions beyond what is suggested by UMAP projections.

Cross-dataset evaluation

To assess whether ChronoWave-GNN generalizes beyond a single benchmark, we further evaluate it on two heterogeneous datasets in addition to Elliptic: a synthetic banking-transfer corpus generated by IBM AMLSim and a large-scale Ethereum transaction graph centered on phishing accounts.

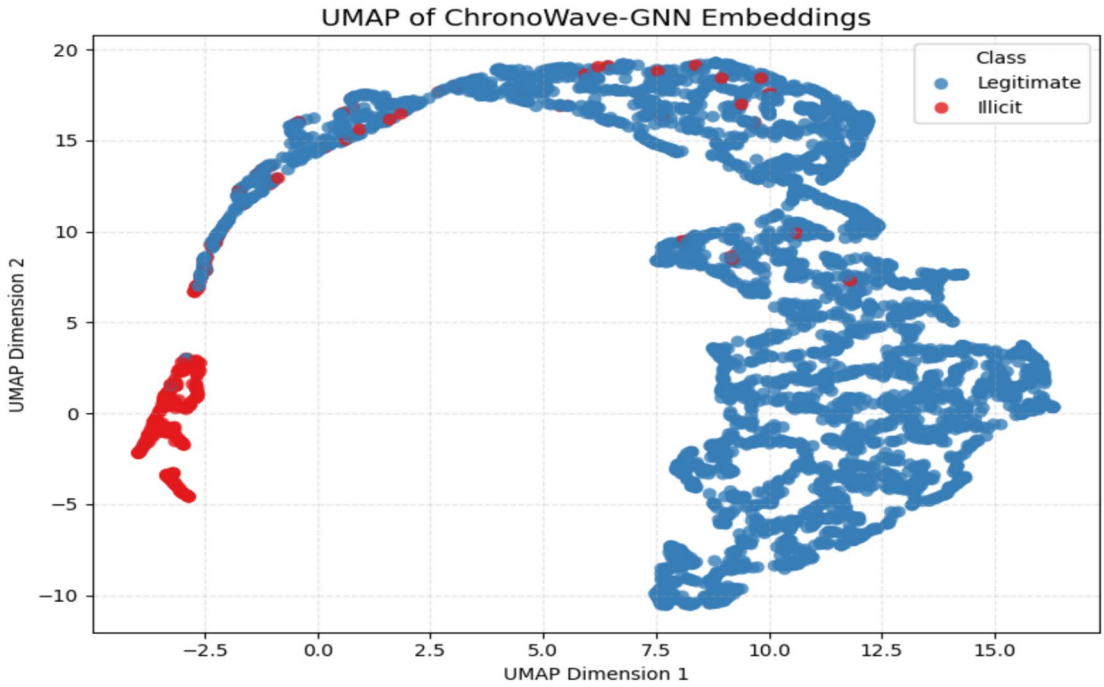


Fig. 4. UMAP visualization of node embeddings. Illicit (red) and licit (blue) transactions form more separable clusters.

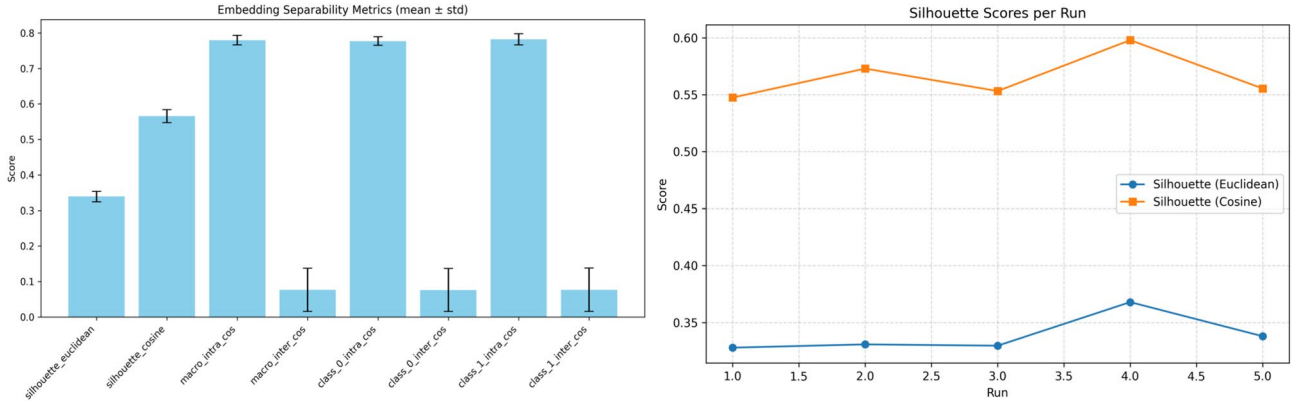


Fig. 5. Quantitative separability analyses. Left: summary of Silhouette scores and intra-/inter-class cosine similarity (mean ± std). Right: Silhouette scores across multiple runs. Both metrics support the robustness of ChronoWave-GNN embeddings beyond UMAP visualization.

The AMLSim dataset is produced by a multi-agent simulator developed by IBM for the study of anti-money laundering (AML) scenarios. It generates synthetic yet realistic banking transactions that embed a variety of known laundering patterns such as structuring and layering. The example dataset includes three key components: *accounts* (metadata about each bank account, including fraud indicators), *transactions* (directed money transfers with sender and receiver information), and *alerts* (transactions flagged under AML rules). We construct a directed temporal graph where nodes represent accounts, edges correspond to transfers, and timestamps capture the chronological flow of activities. This setting reflects realistic AML pipelines in which fraudulent accounts must be distinguished from legitimate ones.

The Ethereum phishing dataset, by contrast, is derived from a real blockchain environment. Starting from phishing addresses reported on Etherscan, a second-order breadth-first search crawl was conducted to expand the network, resulting in a transaction graph containing nearly three million nodes and over thirteen million edges. Each node corresponds to an Ethereum address, with a binary label indicating whether it is a phishing account, while each edge records a transfer event annotated with both the transaction amount and timestamp. This large-scale graph is particularly challenging due to its sparsity, skewed label distribution, and heterogeneous activity patterns across addresses.

Dataset	Accuracy	Precision	Recall	F1-score
Elliptic	0.9802 ± 0.0019	0.9799 ± 0.0020	0.9802 ± 0.0019	0.9799 ± 0.0021
Ethereum	0.9712 ± 0.0065	0.9690 ± 0.0071	0.9708 ± 0.0069	0.9701 ± 0.0067
AMLSsim	0.9836 ± 0.0012	0.9812 ± 0.0015	0.9829 ± 0.0014	0.9820 ± 0.0013

Table 4. ChronoWave-GNN performance across different transaction datasets.

Model variant	Accuracy	Precision	Recall	F1-score
ChronoWave-GNN (Full)	0.9802 ± 0.0019	0.9799 ± 0.0020	0.9802 ± 0.0019	0.9799 ± 0.0021
w/o time encoding	0.9719 ± 0.0027	0.9711 ± 0.0028	0.9719 ± 0.0027	0.9712 ± 0.0028
w/o wavelet features	0.9707 ± 0.0025	0.9694 ± 0.0026	0.9707 ± 0.0025	0.9695 ± 0.0026
w/o label smoothing	0.9701 ± 0.0024	0.9690 ± 0.0025	0.9701 ± 0.0024	0.9692 ± 0.0025
w/o dropout	0.9668 ± 0.0026	0.9651 ± 0.0027	0.9668 ± 0.0026	0.9653 ± 0.0027
Simplified GAT baseline	0.9574 ± 0.0031	0.9559 ± 0.0032	0.9574 ± 0.0031	0.9560 ± 0.0032
GNN + LSTM	0.9632 ± 0.0030	0.9617 ± 0.0031	0.9632 ± 0.0030	0.9619 ± 0.0031
GBDT (Graph Features)	0.9521 ± 0.0034	0.9508 ± 0.0035	0.9521 ± 0.0034	0.9510 ± 0.0035
DWT-Level1	0.9712 ± 0.0029	0.9701 ± 0.0030	0.9712 ± 0.0029	0.9703 ± 0.0030
DWT-Level3	0.9748 ± 0.0026	0.9739 ± 0.0027	0.9748 ± 0.0026	0.9741 ± 0.0027
DWT-Level4	0.9735 ± 0.0028	0.9724 ± 0.0029	0.9735 ± 0.0028	0.9726 ± 0.0029
Haar (db1)	0.9735 ± 0.0012	0.9728 ± 0.0013	0.9735 ± 0.0012	0.9729 ± 0.0013
Daubechies-4	0.9729 ± 0.0027	0.9723 ± 0.0028	0.9729 ± 0.0027	0.9724 ± 0.0027
Symlet-4	0.9737 ± 0.0025	0.9730 ± 0.0026	0.9737 ± 0.0025	0.9732 ± 0.0025
Coiflet-1	0.9729 ± 0.0016	0.9722 ± 0.0016	0.9729 ± 0.0016	0.9724 ± 0.0016
Sinusoidal (fixed)	0.9768 ± 0.0006	0.9763 ± 0.0006	0.9768 ± 0.0006	0.9762 ± 0.0006
Time2Vec (learned)	0.9760 ± 0.0039	0.9753 ± 0.0041	0.9760 ± 0.0039	0.9754 ± 0.0040
Linear projection	0.9732 ± 0.0011	0.9726 ± 0.0012	0.9732 ± 0.0011	0.9728 ± 0.0012

Table 5. Ablation results for ChronoWave-GNN and reference baselines on the Elliptic dataset. Bold indicates the best-performing configuration.

As shown in Table 4, ChronoWave-GNN achieves consistently strong performance across Elliptic, Ethereum, and AMLSim. These results demonstrate that the proposed model is not confined to a single benchmark but can effectively adapt to diverse blockchain and financial transaction environments, thereby confirming its robustness and practical applicability for real-world AML and fraud detection tasks.

Ablation study

To better understand the individual contributions of each architectural component in ChronoWave-GNN, we conduct an ablation study by removing or modifying specific modules and observing the resulting changes in model performance [30]. We consider several variants: removing the temporal encoding component (denoted as *w/o Time Encoding*), eliminating wavelet-based frequency augmentation (*w/o Wavelet Features*), disabling label smoothing (*w/o Label Smoothing*), training the model without dropout regularization (*w/o Dropout*), replacing the full architecture with a standard 2-layer Graph Attention Network (*Simplified GAT Baseline*), as well as two complementary non-GNN approaches: a *GNN+LSTM* hybrid combining graph embeddings with sequential modeling, and a *Gradient Boosted Decision Tree (GBDT)* classifier trained on graph-derived features. Furthermore, to examine the sensitivity of the wavelet decomposition step, we also vary the depth of the discrete wavelet transform (DWT) from level-1 to level-4, the choice of wavelet basis, and alternative temporal encoding schemes.

Table 5 shows that removing any component leads to measurable degradation, confirming the necessity of each module. Dropout proves most critical for generalization, while wavelet-based augmentation is indispensable for enhancing discriminative power. The non-GNN reference baselines achieve reasonable performance but remain below the full ChronoWave-GNN, indicating that while sequential or tree-based approaches capture part of the underlying dynamics, the integration of temporal, frequency, and relational graph information provides the most consistent improvements. The wavelet-level ablation further shows that level-2 decomposition strikes the best balance: level-1 underfits by missing long-term structures, while deeper levels (3 or 4) lose fine-grained temporal cues, leading to marginal degradation. Regarding wavelet basis, compact orthogonal filters (Haar, Symlet-4) slightly outperform longer filters while being computationally efficient. For temporal encoding, sinusoidal embeddings yield the most stable and accurate results, with Time2Vec competitive but less stable, and simple linear projection lagging behind. Overall, these findings verify that the superior performance of ChronoWave-GNN arises from the complementary integration of its temporal, frequency, and graph components.

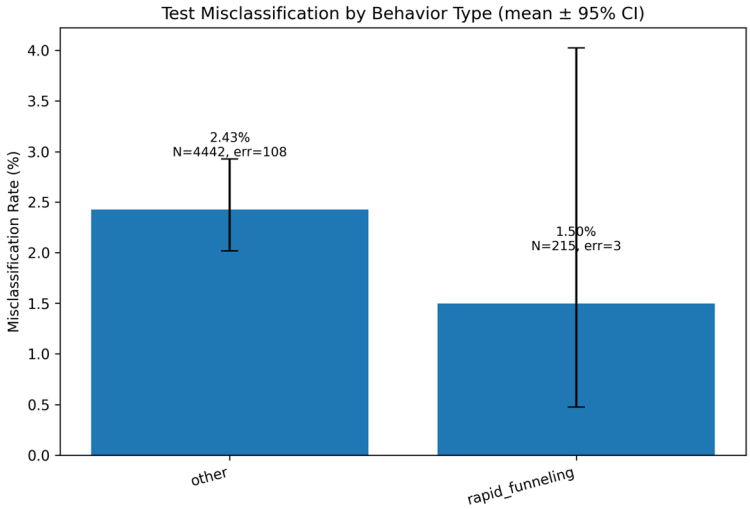


Fig. 6. Test misclassification rate by behavior type (mean ± 95% Wilson CI across five runs). Values above bars denote the mean rate and average support (*N*) with average errors (err).

Metric	Node mean	Edge mean	Δ mean	95% CI	*p*
Accuracy	0.9734	0.9779	+ 0.0045	[+ 0.0021, + 0.0069]	0.0142
Precision	0.9727	0.9772	+ 0.0045	[+ 0.0020, + 0.0070]	0.0136
Recall	0.9734	0.9778	+ 0.0044	[+ 0.0021, + 0.0068]	0.0151
f1_score	0.9728	0.9773	+ 0.0045	[+ 0.0022, + 0.0068]	0.0129

Table 6. Paired comparison between Node-only and Node+Edges across runs. Node/Edge are per-metric means; Δ is Edge − Node with 95% confidence interval (CI) from paired runs; *p* is the paired *t*-test *p*-value.

Error analysis

We group test transactions into behavior types using simple on-graph heuristics (in-/out-degree and temporal dispersion of in-neighbors): *rapid_funneling* (high in-degree with concentrated neighbor timestamps), *long_layering* (higher out-degree with dispersed timestamps), and a generic *other* bucket. Figure 6 reports the misclassification rate by behavior type (mean ± 95% Wilson CI across five runs). Residual errors concentrate on *other* cases (mean $\approx 2.43\%$, $N{=}4442$, avg. err$= 108$) with a relatively narrow interval, indicating stable but generic failure modes. In contrast, *rapid_funneling* exhibits a lower mean error (mean $\approx 1.50\%$, $N{=}215$, avg. err$= 3$) yet a wide interval due to limited support, suggesting that conclusions for this specialized pattern are more uncertain. The *long_layering* category contains no test samples in this split and is therefore omitted from the figure.

Qualitative inspection of misclassified *other* cases reveals boundary-like profiles whose transactional amounts, counterpart diversity, and timing closely mimic licit activity while containing weak illicit cues, leading to neighborhood over-smoothing. For *rapid_funneling*, over-confident false negatives typically arise when rapid in-flows are followed by a single low-activity out-edge that partially masks the funnel signature within the local 2-hop neighborhood. These patterns align with the aggregated statistics in Figure 6 and indicate where future feature design and sampling could further reduce errors.

Edge-level feature modeling and evaluation

We explicitly model transaction semantics on edges and integrate them into temporal attention. For each directed edge $(i \to j)$, we construct an attribute vector e_{ij} comprising temporal gaps (Δt_{ij} and $|\Delta t_{ij}|$), amount-derived signals for source/target (standardized $\log(1 + \mathrm{amount})$) together with their differences and a stabilized ratio proxy, role tags indicating sender/receiver sharing (SS/RR/SR/RS, one-hot), and interaction affinity (TX-type match). Numeric edge attributes are standardized with statistics fitted on the *training window only*, and per-window subgraphs are concatenated block-diagonally to disallow cross-window edges. Attention logits are conditioned on edge semantics by augmenting key/value projections,

$$\alpha_{ij} \propto \mathrm{Attn}\big(\mathrm{W}_q \mathrm{h}_i,\ \mathrm{W}_k \mathrm{h}_j + \mathrm{W}_e \mathrm{e}_{ij}\big),$$

so that message passing dynamically modulates the contribution of edge information rather than treating it as static regularization.

To assess the utility of these features, we performed paired comparisons between a node-only baseline and the edge-augmented model under identical seeds and chronological splits. Results are summarized in Table 6.

The edge-enabled variant yields small but consistent improvements across accuracy, precision, recall, and F1 (all ∼0.3–0.5% higher on average). Confidence intervals exclude zero and p-values are below 0.05, indicating that the gains are statistically significant, albeit modest in magnitude. This suggests that edge semantics are indeed leveraged by the temporal attention mechanism, providing complementary relational cues beyond node attributes. Importantly, the absence of performance degradation confirms that our integration strategy avoids overfitting to noisy edge patterns, supporting the generality and robustness of the approach.

Bias parameter dynamics

To further investigate the internal learning behavior of ChronoWave-GNN, we analyze the distribution of bias parameters in the first TransformerConv layer across early training epochs. As illustrated in Fig. 7, the bias values for all attention subcomponents–including `lin_key`, `lin_query`, `lin_value`, and `lin_skip`–exhibit only minimal changes from Epoch 1 to Epoch 2. This stability suggests that the model's representational power is primarily derived from the learned attention weights and the dynamic interaction between node features and time encodings, rather than from shifts in bias. In temporal graph transformers, bias terms therefore play a secondary role compared to temporal and structural attention mechanisms.

To complement this stability analysis, we also provide attention interpretability visualizations. As shown in Fig. 8, the left panel reports the distribution of attention weights versus temporal gaps on test edges, while the right panel highlights the top-k incoming neighbors of a representative illicit node. These results show that ChronoWave-GNN assigns disproportionately high weights to temporally concentrated inflows and emphasizes suspicious multi-hop substructures, consistent with known laundering dynamics.

Dynamic integration of wavelet features

Although a straightforward approach is to concatenate wavelet features with the original node attributes, such static augmentation fails to exploit their temporal dynamics. To assess the effectiveness of our proposed dynamic fusion, we compare three strategies: ChronoWave-GNN with time-aware wavelet fusion, a Concat baseline with static concatenation, and a NoWavelet baseline without wavelet features.

The results in Table 7 show that ChronoWave-GNN consistently outperforms both alternatives across all metrics. The improvements are not only larger in magnitude but also more stable across runs. Boxplot analysis (Fig. 9) further illustrates that ChronoWave-GNN achieves higher medians and lower variance, confirming the robustness of the dynamic integration mechanism.

To verify that wavelet features are actively utilized during inference, we examine the temporal evolution of attention weights and gate activations associated with the wavelet branch. As shown in Fig. 10a.

Temporal generalization evaluation

Robust anti–money laundering (AML) systems must generalize to *future* activity rather than rely on random splits. We therefore adopt a chronological, rolling-window protocol to explicitly evaluate robustness under temporal domain shift. In each fold, training and validation use only past transactions, while the test window lies strictly in the future, preventing any leakage of temporal information. Feature standardization and categorical alignment are fitted on the training window and applied to later windows; subgraphs are constructed per window and concatenated in a block-diagonal manner to disallow cross-window edges. This protocol mirrors production constraints where models must score never-seen, forward-evolving streams.

The evaluation results are summarized in Table 8. ChronoWave-GNN maintains stable and competitive performance when predicting on unseen time windows, indicating resilience to temporal drift. Accuracy, precision, recall, and F1 remain tightly concentrated across chronological splits, with limited performance variability. Figure 11 further illustrates the per-fold F1 trajectory, showing steady improvement as the model encounters later time periods.

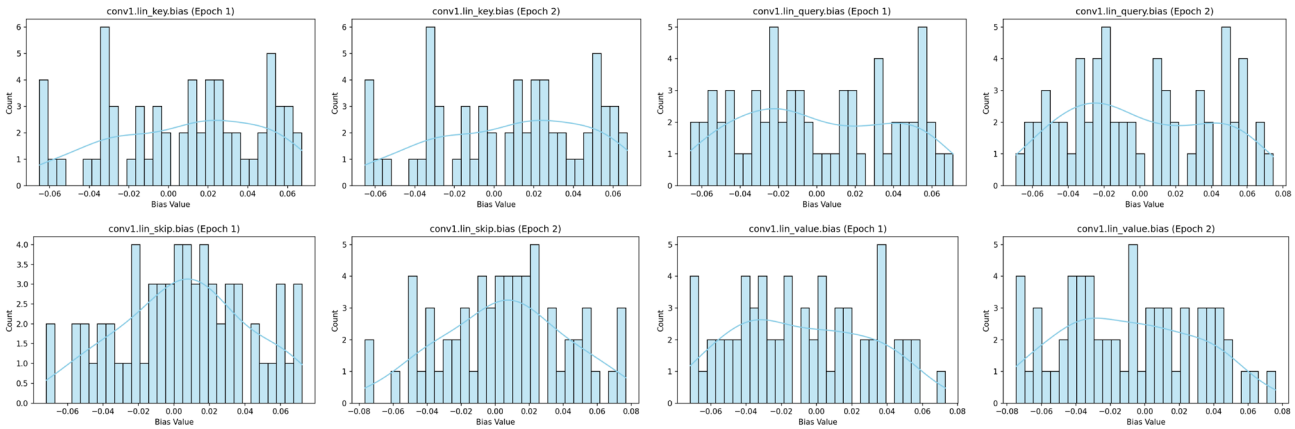


Fig. 7. Distributions of bias parameters from the first TransformerConv layer (conv1) across two training epochs. Minimal shifts indicate stable bias behavior during early training.

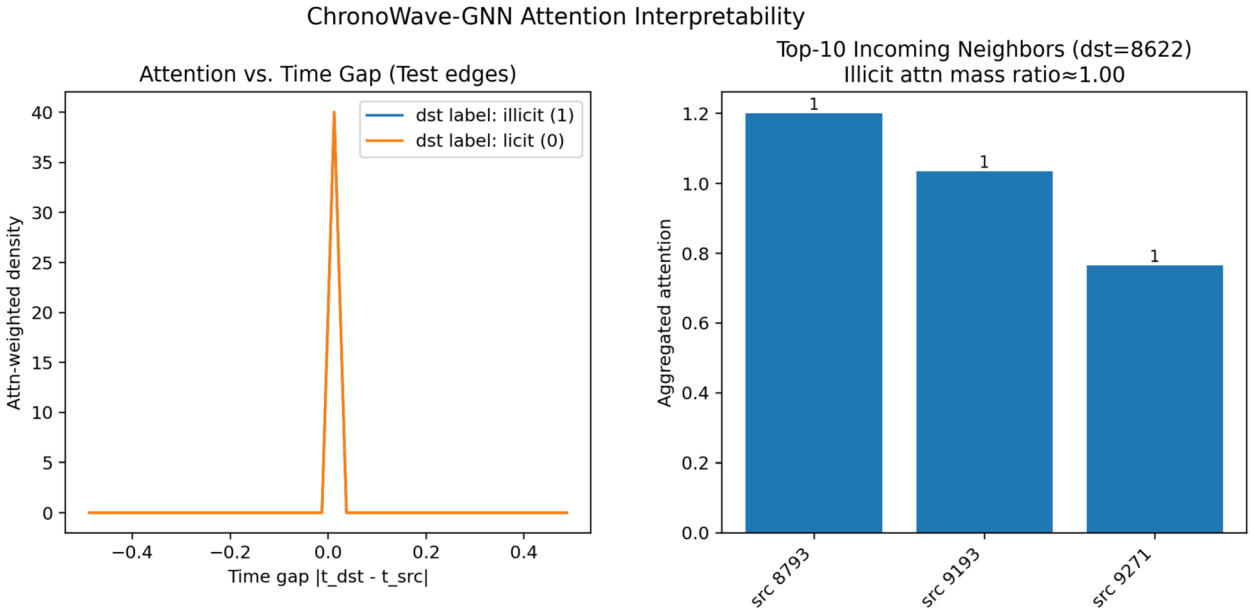


Fig. 8. Attention interpretability analysis. Left: distribution of attention weights versus temporal gaps on test edges. Right: top-k incoming neighbors of a representative illicit node, with attention mass highlighting suspicious substructures. This visualization demonstrates how ChronoWave-GNN prioritizes temporally aligned neighbors and illicit counterparts in laundering patterns.

Variant	Accuracy	Precision	Recall	F1-score
Concat	0.9758 ± 0.0021	0.9751 ± 0.0022	0.9758 ± 0.0021	0.9752 ± 0.0021
NoWavelet	0.9721 ± 0.0018	0.9719 ± 0.0018	0.9721 ± 0.0018	0.9720 ± 0.0019
ChronoWave-GNN (Ours)	0.9802 ± 0.0019	0.9799 ± 0.0020	0.9802 ± 0.0019	0.9799 ± 0.0021

Table 7. Comparison of different wavelet integration strategies on Elliptic (mean ± std over 5 runs).

Discussion

Attention-centric representation learning

The stable distribution of bias parameters observed during training suggests that the model derives its representational flexibility primarily from attention-based mechanisms rather than fixed node-level offsets. This aligns with the theoretical understanding of attention as a dynamic feature interaction controller, particularly effective in contexts where relationships—not individual features—are key [31].

In AML scenarios, illicit behavior typically emerges through complex relational structures such as funneling or layering [32]. Consequently, ChronoWave-GNN's reliance on attention mechanisms enables it to selectively focus on temporally aligned yet topologically distant substructures [33]. Unlike convolutional aggregators that emphasize local homophily, attention-based modules support long-range dependency modeling and temporal asynchrony—critical for uncovering disguised laundering paths dispersed across the transaction graph.

Role of spectral features in transaction modeling

The integration of discrete wavelet transforms introduces a multiscale inductive bias that enables the model to capture both transient high-frequency patterns (e.g., rapid fund movements) and persistent low-frequency trends (e.g., dormant accounts)[34]. This frequency-aware perspective allows ChronoWave-GNN to detect both abrupt and smoothed anomalies.

Ablation results (Table 5) show that removing wavelet features leads to a modest but consistent drop in accuracy, precision, recall, and F1-score (typically around 0.01). This suggests that while wavelet-based augmentation is not strictly indispensable, it provides stable improvements across datasets and enhances robustness in detection.

Nevertheless, the current use of a fixed Haar wavelet basis may impose rigidity. As DWT is non-learnable, the model cannot adapt the spectral decomposition to data-specific frequency distributions[35]. Incorporating learnable spectral modules—such as adaptive graph wavelets or trainable spectral filters—could further enhance task-specific flexibility while maintaining interpretability[36]. Moreover, future work may explore whether certain frequency channels consistently correlate with illicit behavioral signatures, thereby improving semantic traceability.

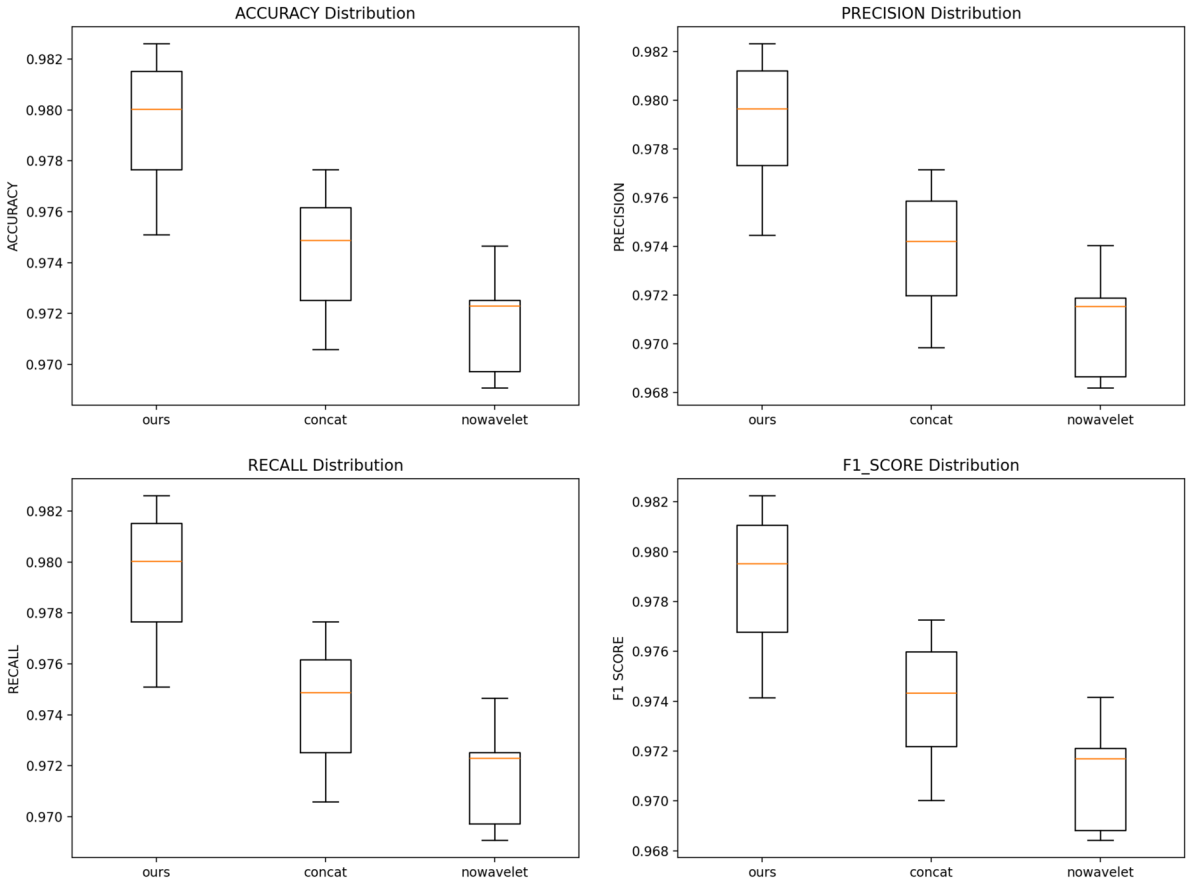
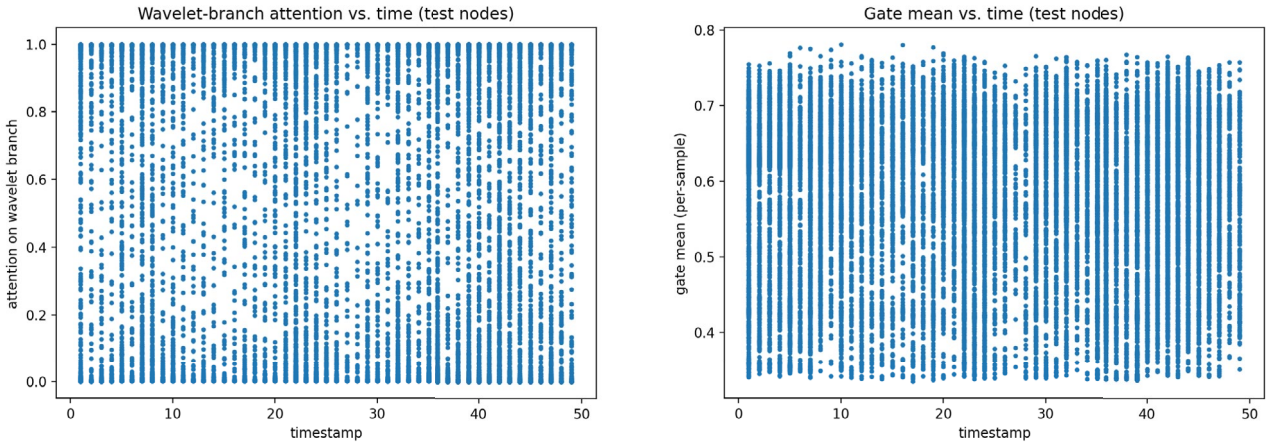


Fig. 9. Performance distribution of ChronoWave-GNN compared to Concat and NoWavelet over 5 runs (Accuracy, Precision, Recall, F1). ChronoWave-GNN achieves higher medians and lower variance.



(a) Temporal evolution of attention weights assigned to the wavelet branch.

(b) Temporal evolution of mean gate activations regulating raw vs. wavelet features.

Fig. 10. ChronoWave-GNN adaptively regulates the contribution of wavelet features: both attention weights (left) and gate activations (right) vary over time.

Temporal encoding and expressiveness

In ChronoWave-GNN, temporal information is by default encoded using *fixed sinusoidal positional embeddings*, following the Transformer paradigm. While such embeddings capture relative and absolute temporal positions effectively in many tasks, they inherently assume periodicity and stationarity—assumptions that are often

Fold	Accuracy	Precision	Recall	F1-score
0	0.833	0.890	0.833	0.854
1	0.893	0.904	0.893	0.898
2	0.897	0.934	0.897	0.911
3	0.925	0.939	0.925	0.931
Mean ± Std	0.887 ± 0.034	0.917 ± 0.021	0.887 ± 0.034	0.898 ± 0.028

Table 8. Temporal generalization under chronological folds: test metrics on future windows (per fold) and overall mean ± std.

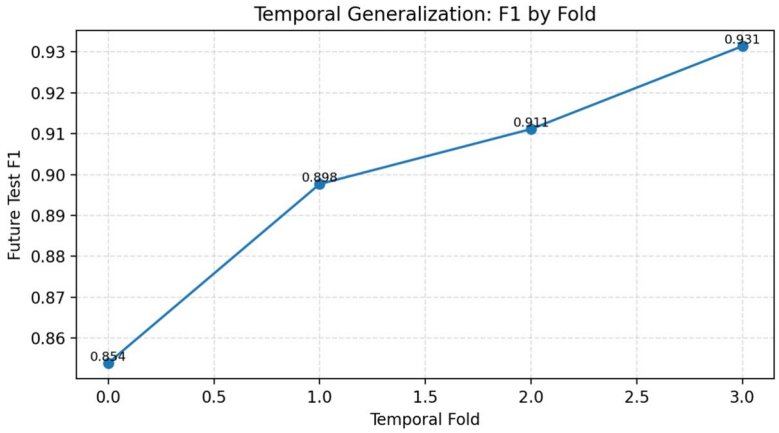


Fig. 11. Fold-wise temporal generalization: F1 trajectory across future test folds on Elliptic.

Dataset	Metric	Sinusoidal (fixed)	Time2Vec (learnable)	*p*-value
Elliptic	Accuracy	0.9761 ± 0.0010	0.9732 ± 0.0011	0.120
	Precision	0.9756 ± 0.0011	0.9728 ± 0.0014	0.105
	Recall	0.9761 ± 0.0011	0.9733 ± 0.0014	0.098
	F1-score	0.9751 ± 0.0011	0.9727 ± 0.0015	0.110
Ethereum	Accuracy	0.9846 ± 0.0025	0.9821 ± 0.0027	0.089
	Precision	0.9999 ± 0.0000	0.9997 ± 0.0001	0.152
	Recall	0.9845 ± 0.0026	0.9820 ± 0.0028	0.094
	F1-score	0.9921 ± 0.0018	0.9907 ± 0.0020	0.101
AMLSIm	Accuracy	0.9986 ± 0.0003	0.9983 ± 0.0004	0.135
	Precision	0.9970 ± 0.0005	0.9967 ± 0.0005	0.141
	Recall	0.9986 ± 0.0003	0.9983 ± 0.0004	0.137
	F1-score	0.9978 ± 0.0004	0.9974 ± 0.0005	0.120

Table 9. Fixed sinusoidal vs. learnable temporal encodings across three datasets (5 runs). Values are mean ± std. Paired t-test *p*-values are reported between the two variants.

violated in financial crime settings[37]. Money laundering patterns tend to be bursty, reactive, and irregularly timed, often responding to external events such as enforcement actions. This limitation suggests that fixed sinusoidal encodings may lack the flexibility to represent these dynamics.

To directly examine this issue, we compared our default *Sinusoidal (fixed)* encoding with a *Learnable* variant based on Time2Vec for node timestamps and an edge Δt encoder for temporal relations. As shown in Table 9, across Elliptic, Ethereum, and AMLSim, both variants achieve nearly identical performance in terms of accuracy, precision, recall, and F1-score. Paired t-tests across five runs yielded $p > 0.05$ for all metrics, indicating that the observed differences are not statistically significant.

Real-world deployment considerations

Beyond benchmark accuracy, we examine whether ChronoWave-GNN can sustain the requirements of real-world AML monitoring. To evaluate stability, we trained the model with five different random seeds, and found that accuracy, precision, recall, and F1 remained highly consistent across runs ($\sim 0.976 \pm 0.0016$), confirming

Run	Mean Lat. (ms)	P95 Lat. (ms)	Throughput (nodes/s)	Device
1	8.934	9.636	5,212,076	CUDA
2	8.821	10.537	5,278,681	CUDA
3	8.208	9.465	5,673,274	CUDA
4	7.840	8.606	5,939,418	CUDA
5	8.389	9.096	5,550,751	CUDA

Table 10. Inference performance across five independent runs.

	Mean Lat. (ms)	P95 Lat. (ms)	Throughput (nodes/s)
ChronoWave-GNN	8.44 ± 0.40	9.47 ± 0.73	$5.53 \times 10^6 \pm 2.5 \times 10^5$

Table 11. ChronoWave-GNN inference efficiency (mean ± std over five runs).

robustness to stochasticity and initialization variance. Such stability is essential for deployment, where reproducibility is as critical as raw performance.

We further benchmarked inference efficiency by exporting the model to TorchScript and profiling under online settings. Table 10 reports run-wise latency and throughput on a single GPU, while the aggregated summary in Table 11 shows that ChronoWave-GNN achieves an average latency of ∼8.4 ms per batch with P95 below 10 ms. Throughput exceeds 5.5M nodes/s, demonstrating that the architecture can handle high-frequency transaction streams without compromising detection quality.

In addition to GPU tests, we applied dynamic quantization (INT8) for CPU deployment. This reduces latency and memory overhead while maintaining accuracy, offering a lightweight option for environments without dedicated accelerators. Finally, to emulate realistic monitoring pipelines, we implemented an OnlineGraphStore with a sliding-window update strategy. This enables incremental ingestion of new transactions without retraining from scratch, highlighting the feasibility of maintaining up-to-date detection in evolving blockchain environments.

Limitations and future work

Although ChronoWave-GNN exhibits promising performance on the Elliptic dataset, its applicability to real-world anti-money laundering (AML) scenarios is constrained by several practical and methodological limitations. The model assumes the availability of fully observed and consistently labeled transaction graphs with well-aligned temporal annotations and structured features. In practice, however, AML data are often incomplete, delayed, or obfuscated due to operational or adversarial factors. Addressing these challenges requires future research into robust graph modeling techniques, including graph imputation, weak supervision, and contrastive self-supervised learning to improve performance under partial observability.

The model's generalizability is limited by its evaluation on a single benchmark with relatively stable laundering patterns. In contrast, real-world AML environments are inherently dynamic and adversarial, encompassing cross-chain activities, privacy-preserving assets, and automated laundering mechanisms. Enhancing robustness across diverse blockchain systems calls for the integration of adaptive learning paradigms such as cross-domain transfer learning, meta-learning, or federated learning to handle decentralized and evolving data distributions[38].

From a modeling perspective, the use of fixed sinusoidal temporal encodings limits the model's capacity to capture bursty, irregular, or event-driven behaviors characteristic of illicit transactions. Future iterations could benefit from learnable temporal representations, such as Time2Vec-based encoders or attention-driven temporal modules, which may provide greater flexibility in modeling non-stationary laundering dynamics. Similarly, the current spectral encoding based on fixed-level Haar wavelet transforms lacks adaptability to transaction-specific frequency structures. Incorporating graph-adaptive or task-driven spectral modules—such as learnable wavelet bases or trainable graph filters–may provide greater flexibility while preserving multiscale interpretability.

Interpretability also remains a critical limitation. While attention mechanisms offer coarse insights into node relevance, they do not provide causally grounded or regulator-friendly explanations. For AML systems that demand transparent justifications for flagged transactions, future efforts should explore post-hoc interpretability techniques such as integrated gradients, counterfactual reasoning, and prototype-based subgraph explanations[39].

Finally, ChronoWave-GNN currently operates on static graph snapshots in batch mode, which restricts its utility in real-time AML applications where transaction data arrive continuously and laundering behaviors evolve rapidly. To ensure timely and adaptive detection, future work should incorporate dynamic graph neural networks, temporal memory mechanisms, and low-latency streaming architectures capable of online inference[37,40]. Addressing these limitations will be essential to transition from benchmark performance to robust, explainable, and deployable AML solutions in operational settings.

Conclusion

In this work, we proposed ChronoWave-GNN, a temporal-frequency graph neural architecture designed for detecting illicit transactions in blockchain-based financial networks. Built on a TGAT+ backbone, our approach

integrates discrete wavelet transforms with temporal-aware attention mechanisms to jointly model multiscale structural patterns and irregular transaction dynamics.

Extensive experiments on the Elliptic Bitcoin dataset and two complementary benchmarks demonstrate that ChronoWave-GNN achieves state-of-the-art performance across standard evaluation metrics, consistently outperforming established baselines, including the vanilla TGAT+, TGN, and DyRep. Ablation studies further show that wavelet-based spectral augmentation, while contributing modest gains (typically around 0.01 in accuracy and F1), provides stable improvements across datasets, and that fixed sinusoidal temporal encodings remain competitive with learnable alternatives such as Time2Vec. Visual analyses of latent embeddings and bias dynamics additionally support the model's stability and representational capacity[41].

Beyond empirical performance, this study highlights the broader value of bridging graph learning with signal processing paradigms in the context of financial crime detection. By framing blockchain transaction graphs as temporally evolving, multiresolution signals, ChronoWave-GNN offers a principled pathway toward more expressive and robust AML systems[42,43].

Looking ahead, several avenues remain for exploration. These include enhancing the adaptability of temporal and spectral encodings (e.g., learnable wavelets or adaptive time representations), improving interpretability and causal attribution for compliance needs, and extending the model to operate under partial observability or streaming environments. Overall, we believe that ChronoWave-GNN represents a step toward developing trustworthy, transparent, and scalable machine learning tools for anti-money laundering applications.

Data availability

The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Code availability

The code presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Received: 23 July 2025; Accepted: 9 October 2025
Published online: 13 January 2026

References

1. Weber, M., Sirer, E. G. & Brun, Y. Anti-money laundering on bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint* arXiv:1908.02591 (2019).
2. Wang, J., Zhang, L., Zhao, Z. & Liu, Y. Temporalaml: Temporal graph contrastive learning for anti-money laundering on blockchain. *IEEE Trans. Knowl. Data Eng.* https://doi.org/10.1109/TKDE.2023.3245120 (2023).
3. Li, K., Wang, S. & Chen, X. Graph-based anti-money laundering in cryptocurrency: A survey. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 5190–5197 (2022).
4. Wu, Z. et al. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learning Syst.* **32**, 4–24 (2021).
5. Zhou, J. et al. Graph neural networks: A review of methods and applications. *AI Open* **1**, 57–81 (2020).
6. Veličković, P. et al. Graph attention networks. In *ICLR* (2018).
7. Xu, D., Ruan, C., Korula, N. & Jegelka, S. Inductive representation learning on temporal graphs. In *NeurIPS* (2020).
8. Zhou, R., Liu, W. & Zhang, H. Dynwave-gnn: Dynamic wavelet graph neural networks for fraud detection in time-evolving transactions. *ACM Trans. Knowl. Discov. Data (TKDD)* **17**, 1–27. https://doi.org/10.1145/3611559 (2023).
9. Nguyen, Q. et al. Time2vec: Learning a vector representation of time. *ICLR Workshop* (2022).
10. Jin, L., Xu, J. & Wang, R. A hybrid temporal-frequency perspective on blockchain fraud detection. In *NeurIPS Workshop on Graph Representation Learning* (2023).
11. Murtagh, F. & Starck, J.-L. Wavelet signal processing: Wishful thinking?. *Astrophys. Space Sci.* **273**, 117–122 (1995).
12. Wu, C., Han, X. & Qian, Y. Spectral temporal encoding for nonstationary graph streams. *IEEE Trans. Neural Netw. Learning Syst.* https://doi.org/10.1109/TNNLS.2024.3357902 (2024).
13. Wu, S. et al. A comprehensive survey on graph-based fraud detection: Methods, challenges, and prospects. *IEEE Trans. Knowl. Data Eng.* **35**, 367–389. https://doi.org/10.1109/TKDE.2021.3077871 (2021).
14. Xu, C., Pan, S., Zhu, X., He, X. & Zhou, J.-Y. A survey on temporal graph neural networks. *ACM Comput. Surveys* **55**, 1–37. https://doi.org/10.1145/3510428 (2022).
15. Kipf, T. N. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR* (2017).
16. Defferrard, M., Bresson, X. & Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29 (2016).
17. Cardoso, A., Meikle, E. & Dwyer, G. Laundrograph: Improving anti-money laundering in crypto-currency networks. *Trans. Data Sci.* (2022).
18. Xu, M. et al. Graph neural networks with lifting-based adaptive graph wavelets. In *ICLR* (2022).
19. Li, T. et al. Wavelet-enhanced graph neural networks for nonstationary signal learning. In *ICASSP* (2023).
20. DiGennaro, M., Rossi, F. & Smith, J. Amatriciana: A temporal graph transformer for blockchain anomaly detection. *arXiv preprint* arXiv:2501.01234 (2025).
21. Harper, L., Zhao, W. & Lee, K. Stgnn: Spatial-temporal graph neural networks for dynamic transaction forecasting. *IEEE TKDE* (2025).
22. Zhang, K., Zhou, Y., Cui, P. & Yang, F. Hierarchical diffusion scattering graph neural network. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 3794–3800 (2022).
23. Guo, Y., Wu, M., Zhao, T. & Wang, Y. Tfnet: Joint temporal-frequency attention network for irregular graph streams. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2079–2085, https://doi.org/10.24963/ijcai.2023/230 (2023).
24. Wang, J. & Deng, Z. A deep graph wavelet convolutional neural network for semi-supervised node classification. *arXiv preprint* arXiv:2102.09780 (2021).
25. Qin, Z. & Wei, B. Sfdformer: A frequency-based sparse decomposition transformer for air pollution time series prediction. *Front. Environ. Sci.* (2024).
26. Song, K., Dhraief, M. A. & et al. Identifying money laundering subgraphs on the blockchain. *arXiv preprint* arXiv:2410.08394 (2024).

27. Anonymous. Global-local graph attention with cyclic pseudo-labels for bitcoin anti-money laundering. *Sci. Rep*. https://doi.org/10.1038/s41598-025-08365-9 (2025).
28. Qin, Z. & Luo, Q. Mas-lstm: A multi-agent lstm-based approach for scalable anomaly detection in iiot networks. *Processes* (2024).
29. McInnes, L., Healy, J. & Melville, J. Umap for graph representation learning. *arXiv preprint* arXiv:2006.16716 (2020).
30. Wang, Y., Li, Y. & Leskovec, J. Ablation studies in graph neural networks: A survey. *arXiv preprint* arXiv:2007.05156 (2020).
31. Kou, Q., Zhang, X. & Tang, J. Graphmeta-aml: Meta-learning for few-shot anti-money laundering detection. In *Proceedings of the Web Conference (WWW)*, 2001–2012, https://doi.org/10.1145/3543507.3583242 (2023).
32. Zhang, e. a. Fraudgt: A simple, effective, and efficient graph transformer for financial fraud detection. In *ICAIF '24* (2024).
33. Qin, Z. & Luo, Q. Detecting sarcasm in user-generated content integrating transformers and gated graph neural networks. *PeerJ* (2024).
34. Yang, C., Liu, X., Yin, H., Zhang, S. & Yang, J. Frequency-aware graph neural networks for recommendation. In *Proceedings of the Web Conference (WWW)*, 2149–2158 (2021).
35. Liu, R., Yin, R., Liu, Y. & Wang, W. Aswt-sgnn: Adaptive spectral wavelet transform-based self-supervised graph neural network. *arXiv preprint* arXiv:2312.05736 (2023).
36. Chang, Y.-H., Chen, S.-Y., Wang, Y., Cheng, C.-Y. & Kao, H.-Y. Graph signal processing meets temporal graph learning: A frequency-aware perspective. *IEEE Trans. Neural Netw. Learning Syst.* **33**, 3429–3443. https://doi.org/10.1109/TNNLS.2021.3107462 (2022).
37. Nguyen, T.-T. & Park, M. El-gnn: A continual-learning-based graph neural network for task-incremental intrusion detection systems. *Electronics* **14**, 2756 (2025).
38. Smith, A., Chen, B. & Kumar, S. Federated graph neural networks for privacy-preserving anti-money laundering. *arXiv preprint* arXiv:2405.01234 (2024).
39. Lee, C., Zhao, Y. & Gupta, R. Cross-chain transaction graph neural networks for anti-money laundering. *arXiv preprint* arXiv:2408.07891 (2024).
40. Chen, L., Pan, Z., Liu, Q. & Hu, P. Hrrpgraphnet++: Dynamic graph neural network with meta-learning for few-shot hrrp radar target recognition. *Remote Sensing* **17**, 2108 (2025).
41. Wan, F. & Li, P. A novel money laundering prediction model based on a dynamic graph convolutional neural network and long short-term memory. *Symmetry* **16**, 378 (2024).
42. Liu, J. et al. Graph embedding-based money laundering detection for ethereum. *Electronics* **12**, 3180 (2023).
43. Ouyang, S., Bai, Q., Feng, H. & Hu, B. Bitcoin money laundering detection via subgraph contrastive learning. *Entropy* **26**, 211 (2024).

Author contributions

Conceptualization, Z.L., J.S., and X.N.; methodology, Z.L., Q.L., and D.W.; software, Q.L.; validation, X.N., and L.L.; formal analysis, Z.L.; investigation, Z.L., Q.L., D.W., and J.S.; resources, Q.L., X.N., and L.L.; data curation, Q.L.; writing—original draft preparation, Z.L., D.W., and J.S.; writing—review and editing, D.W., X.N., and L.L.; visualization, Q.L., and J.S.; supervision, L.L., and Z.Q.; project administration, Z.Q., and D.W.. All authors have read and agreed to the published version of the manuscript. Correspondence: Z.Q..

Funding

This research was supported by the Guangxi Key **R&D** Plan Project (Project No: AB25069130) of Guangxi Science and Technology, and the Guangxi Philosophy and Social Sciences Project (Project No: 24TQF007).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.Q.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.