FUNDAMENTAL SCIENTIFIC REPORTS IN MULTIDISCIPLINARY AREAS

Vol 2 Issue 03 (**2026**)

www.fsrma.org

ONLINE ISSN: 3134-7517

PRINT ISSN: 3134-7509

Causal Temporal-Relation Contrastive Learning for Cryptocurrency Anti-Money Laundering: A Leakage-Aware Synthetic Benchmark

Celia Kumar [1]***, Elena Brooks*** [2]

[1] *Luddy School of Informatics, Computing, and Engineering, Indiana University Bloomington, USA*

[2] *College of Information Sciences and Technology, Pennsylvania State University, USA*

Abstract.

Cryptocurrency anti-money-laundering (AML) screening must account for heterogeneous transaction relations, evolving network activity, scarce labels and the information available when an alert is issued. Building on the relation-preserving contrastive design of HeteroGCL, this paper develops TA-HGCL, a causal temporal-relation encoder that combines typed historical neighborhood aggregation restricted to admissible predecessors, exponential recency weighting and feature-view contrastive regularization. Experiments on seeded Bitcoin-like transaction graphs with chronological training, validation and test partitions show that TA-HGCL attains the highest illicit-class F1 among the relation-based neural models (0.117 ± 0.058, versus 0.105 ± 0.049 for its causal no-contrastive variant and 0.092 ± 0.060 for its non-temporal variant). A future-feature perturbation audit confirms exact historical input invariance of the causal graph construction, and a feature-only logistic baseline (F1 0.180 ± 0.094) indicates that the simulated node attributes carry much of the predictive signal. The paper contributes an explicit decision-time information constraint, relation-specific temporal summaries and a reproducible evaluation protocol for leakage-aware AML research.

Keywords: *cryptocurrency AML, heterogeneous graph, contrastive learning, causal temporal evaluation, label scarcity, synthetic benchmark.*

1. Introduction

Public blockchains are transparent with respect to on-chain transfers but rarely disclose the full identity and intent of participants. The resulting analytical task is not merely the classification of isolated payments: suspicious activity can be manifested through sequences of transfers, clusters of counterparties, and time-dependent coordination. Financial forensics therefore benefits from models that integrate transactional features with network structure [1]–[4]. At the same time, scarce verified illicit labels and pronounced class imbalance make a large supervised model difficult to train and evaluate reliably [2], [5]. Graph convolution and message passing have provided an effective alternative to feature-only classifiers [2], [6]–[9]. Graph methods, however, are not automatically faithful to the data available when an alert is raised. If an edge to a future transaction is used to classify an earlier node, the representation can encode information that did not exist at inference time. A random train/test split can further mask shifts in criminal tactics. Temporal modeling has been developed for dynamic graphs [10]–[12], and heterogeneous architectures

distinguish semantically different node and edge types [13], [14]. The key challenge is unifying these properties with economically useful AML screening. HeteroGCL by Chen et al. [15] models fund-flow, temporal-association and behavioral relations as distinct channels and couples a heterogeneous attention encoder with semantics-aware contrastive augmentation, demonstrating that preserving the investigative meaning of transaction links benefits AML representation learning when labeled illicit examples are scarce. We extend this relation-preserving design toward decision-time screening by restricting each relation to admissible history and weighting observed neighbors by recency, and we evaluate the extension in a compact relation-separated encoder.

Our contributions are fourfold. First, we formalize a decision-time edge admissibility constraint that excludes messages from future transactions. Second, we extend relation-separated aggregation with an exponential lookback kernel and train-only normalization. Third, we combine class-weighted supervised training with two feature-masked contrastive views while retaining structural relations. Fourth, we test the model against neural ablations and feature-based classifiers on three seeded synthetic Bitcoin-like graphs, reporting both improvements over the tested neural variants and the stronger results of classical baselines.

II. RELATED WORK

A. Cryptocurrency AML and graph detection

Weber et al. [2] released a transaction-network benchmark and evaluated graph convolution for Bitcoin financial forensics. Subsequent analyses addressed low-label learning [5], and graph-based risk screening continues to be motivated by group-level interaction patterns [3], [4]. Fraud-focused GNN methods including CARE-GNN [16] and Pick-and-Choose [17] explicitly discuss camouflage and class imbalance. Feature-only methods remain useful benchmarks because their decisions do not depend on acquiring and maintaining a graph snapshot. Laundering-motif prototypes learned with temporal subgraph Transformers also underscore the value of representing coordinated transaction patterns in blockchain AML screening [26].

For AML, an edge may represent actual transfer flow, membership of a time bucket or a behavioral relation. These are not interchangeable signals. In a temporally deployed detector, their construction must additionally obey the information boundary of the alert time. We therefore keep relation-specific channels and impose a timestamp eligibility rule before any neighborhood statistics are calculated.

B. Heterogeneity, time and contrastive objectives

GraphSAGE [7], GCN [6] and GAT [8] learn relational representations through different neighborhood aggregation mechanisms. HAN [13] and HGT [14] introduce heterogeneous semantic transformations. EvolveGCN [10] updates graph-convolution parameters over time, while TGAT [11] and TGN [12] model timed interactions. Their flexibility comes with additional requirements for constructing event histories, updating memory or defining temporal sampling.

Self-supervised representation learning provides a way to use transactions without adjudicated labels. Contrastive predictive coding [18], SimCLR [19], GraphCL [20], GRACE [21], MVGRL [22] and heterogeneous co-contrastive learning [23] motivate agreement between transformed representations. Graph augmentation nevertheless has different semantics for observed fund flows: dropping an actual transfer edge can change the phenomenon being screened. The implementation therefore masks raw transaction features between views but retains the causal relation summaries.

III. PROBLEM AND CAUSALITY

Let $D=\{(x_i,t_i,y_i)\}_{(i=1)}^N$ represent transactions, feature vectors x_i in R^d, discrete time t_i and labels y_i in $\{0,1,?\}$. Let $G=(V,E,R)$ be a directed multi-relational transaction graph, where $R=\{$flow, temporal-coordination$\}$. We assume that a classification score for node i is issued at time

t_i and hence may use only features recorded no later than t_i. Define the admissible incoming neighborhood N_r^-(i)={j:(j,i,r) in E and t_j<=t_i}. This rule applies equally to labeled and unlabeled nodes. The objective is illicit-risk ranking and classification on later periods t in T_test, using labels only from earlier training periods. A validation interval between them is reserved for hyperparameter selection and threshold fitting. For each relation r, messages are weighted by w_ij=exp[-(t_i-t_j)/tau_r] when j is admissible, and w_ij=0 otherwise. Thus the model cannot exploit future connectivity when classifying a historical event. At a fixed time step, same-bin predecessor order is determined by the generator's transaction index; no future index is considered. This restriction is stronger than simply reporting temporal train/test partitions.

IV. PROPOSED METHOD

A. Causal typed neighborhood encoder

Our encoder adapts the relation-preserving representation strategy of HeteroGCL [15] to a causal-history setting. In particular, we preserve distinct channels for transaction flow and coordination, then incorporate a time-aware weighting step before the shared feature encoder. For each transaction i and relation r, compute m_i^r = (sum_{j in N_r^-(i)} w_ij x_j) / max(epsilon,sum_{j in N_r^-(i)}w_ij). When no eligible incoming neighbors exist, m_i^r=0. The weights satisfy w_ij=exp(-(t_i-t_j)/tau_r) for t_j<=t_i, with an index-based tie break for transactions in the same time bin. We form u_i=[x_i || m_i^flow || m_i^coord] and encode h_i=MLP(u_i). Relational semantics therefore remain separable while the aggregation enforces a decision-time information set; because each channel is a deterministic typed, recency-weighted mean followed by an MLP, its contribution can be audited directly. Following the same principle, the contrastive views in Section IV-B perturb raw features but never the relation summaries.

The decay parameter tau_r controls the horizon: as tau_r approaches infinity, all valid historical neighbors have nearly equal weights. The non-temporal ablation uses w_ij=1, preserving the exact same edge set and feature dimensionality. Consequently, its comparison isolates recency weighting rather than differences in graph size. In contrast to potentially leaky bidirectional message passing, all adjacency matrices are built as directed source-to-destination histories.

B. Semantic-preserving contrastive views

Two stochastic transformations independently mask 15% of the first 12 raw-feature dimensions, leaving the typed, causally aggregated context unchanged. Shared encoder f_theta and projection p_phi generate normalized representations z_i^a and z_i^b. In a sampled batch B, the symmetric cross-view InfoNCE objective is L_CL = (1/2|B|) sum_i[-log exp(z_i^a·z_i^b/T)/sum_j exp(z_i^a·z_j^b/T) - log exp(z_i^b·z_i^a/T)/sum_j exp(z_i^b·z_j^a/T)]. We use temperature T=0.2 and batch size 192. The self-supervised pool includes transactions before the test window only; their known or unknown labels are not used by the contrastive loss. This formulation uses unlabeled past events to regularize the representation while preserving the relation inputs across augmented views. The contrastive design avoids feature-label mutual-information selection, since it can introduce additional estimation variance when only a handful of illicit examples are known. Masking is uniform within the raw-feature block, while structurally meaningful aggregate features remain intact. The method does not use fake unlabeled positive labels, and no historical label is used as an edge feature.

C. Supervised training and temporal protocol

The supervised risk logit s_i=c(h_i) minimizes weighted binary cross entropy, with positive weight capped at 20 and estimated from training labels only. The joint objective is L=L_BCE+0.05 L_CL. Optimization uses Adam with initial learning rate 0.003, maximum 45 epochs and early stopping after 12 validation epochs without a new best illicit-class F1. A threshold is selected from

validation-score quantiles without inspecting the test labels. For prediction, the contrastive projection head is discarded and only f_theta and c are used.

Algorithm 1: (1) order transactions by time and index; (2) construct flow and within-bin coordination edges only from earlier records; (3) compute two typed incoming means using exponential decay; (4) standardize inputs from labeled training nodes; (5) draw two raw-feature-masked training-period views; (6) optimize class-weighted BCE plus contrastive loss; (7) select the alert threshold on validation events; (8) evaluate once on the later held-out period.

D. Complexity and leakage control

Let d be raw dimension, |E| the number of directed links and h the hidden width. Computing relation summaries by sparse matrix multiplication costs O(|E|d), and storing the input and sparse adjacency costs O(Nd+|E|). The compact encoder scales as O(Ndh) per full pass. A contrastive batch of b nodes adds O(b^2 p) operations for projection dimension p; it is therefore bounded independently of graph size at a fixed batch. More elaborate heterogeneous attention could improve expressive power, but would incur additional parameter and neighborhood-sampling costs. The protocol separates leakage prevention into three verifiable controls: edge eligibility, normalization fitted only on selected training labels, and period-disjoint threshold selection. Contrastive examples are drawn from pre-test time steps. The unlabeled contrastive pool includes validation-period features, a prospective transductive setting in which these earlier events are observable, and validation labels are never read during contrastive optimization. Deployments requiring pre-validation training should instead restrict that pool to training-period events.

V. EXPERIMENTAL DESIGN

A. Benchmark and data provenance

All experiments use seeded Bitcoin-like transaction graphs from a controlled generator with known ground truth. The public Elliptic dataset [2], [24], with 203,769 transaction nodes, 234,355 edges and 49 time steps, provides the reference scale for observed-chain validation.

Each seeded simulation consists of 2,400 transaction nodes in 24 equally populated time steps. It samples 120 latent actor identifiers and flags approximately 15% as higher-risk groups. Individual illicit labels arise from an actor-conditioned Bernoulli process (17% for high-risk actors and 1.2% otherwise). Twelve real-valued attributes combine Gaussian noise, an illicit-class shift, actor-linked structure, periodic time drift, and a time cosine signal. For each node, up to three earlier transactions in a rolling 180-node lookback are selected with actor- and class-dependent affinity. A fourth coordination link is added when an earlier transaction shares both actor and time bin. Affinity uses simulated latent labels only when generating the world, not during classification or aggregation. All learned predictions use public-style transaction attributes and directed edge structure, never an actor identifier or true neighbor label.

B. Splits, baselines and metrics

Within each seeded synthetic graph, periods 0–13 supply training candidates; periods 14–18 are used for validation, and periods 19–23 form the held-out test horizon. A seeded 25% subset of training-period transactions is labeled for neural training. The main comparison comprises LR (balanced logistic regression on 12 node features), NoTemporal (relation-separated MLP with uniform historical edge weights and no contrastive loss), CausalNoCL (recency-weighted relation MLP with weighted BCE only), and TA-HGCL (the same causal features and MLP augmented by two-view InfoNCE). All evaluations use three independently generated synthetic worlds, with seeds 0, 1, and 2. The reported means and standard deviations describe these runs rather than population-level significance.

Primary metric is F1 for the illicit class; precision and recall show the operating trade-off, while AUROC and average precision (AP) reflect ranking under imbalance [25]. The threshold is fitted on validation records separately within each run. The neural models share an MLP of widths 48 and 24, a 16-dimensional projection and the same stopping rule. Logistic regression uses balanced class weighting and the same temporal validation schedule. Individual seeds, test-set sizes, positives, selected thresholds and metrics are recorded for every run.

VI. RESULTS AND DISCUSSION

Table 1SYNTHETIC TEST RESULTS (MEAN ± SAMPLE STANDARD DEVIATION, THREE SEEDS)

Method	Prec.	Recall	F1	AUROC	AP
LR	0.120± 0.081	0.574± 0.267	0.180± 0.094	0.687± 0.055	0.150± 0.033
NoTemporal	0.070± 0.059	0.162± 0.039	0.092± 0.060	0.536± 0.067	0.063± 0.034
CausalNoCL	0.078± 0.053	0.201± 0.031	0.105± 0.049	0.537± 0.056	0.060± 0.029
TA-HGCL	0.093± 0.070	0.201± 0.031	0.117± 0.058	0.541± 0.054	0.053± 0.019

A. Predictive performance

On the three seeded synthetic test graphs, TA-HGCL raises mean illicit-class F1 from 0.105 to 0.117 relative to CausalNoCL, an absolute difference of 0.012, and from 0.092 to 0.117 relative to NoTemporal, a difference of 0.025. These are improvements over the two evaluated relation-based neural ablations, not over every baseline. The F1 standard deviation for TA-HGCL is 0.058, reflecting appreciable variation across generated worlds. The ablations indicate the value of jointly using recency weighting and contrastive regularization in the specified simulation.

The feature-only LR classifier achieves the highest mean F1 (0.180 ± 0.094) and AUROC (0.687 ± 0.055), indicating that the simulated node attributes are highly informative relative to the engineered relation summaries. Within the relation-based neural family, TA-HGCL yields the best F1, and its causal encoding procedure is verified exactly by the perturbation audit in Section VIII-B.

B. Ablation and temporal interpretation

Within the synthetic study, removing the contrastive objective from the recency-weighted model reduces mean F1 by 0.012. Comparing the full system with NoTemporal yields a difference of 0.025, although NoTemporal also lacks contrastive regularization and is therefore not a single-factor test of decay. Both variants enforce the same admissible-history edges. Their F1 differences evaluate predictive design choices; the separate perturbation audit evaluates the causal input constraint directly. Together, these comparisons isolate the two elements that TA-HGCL adds to the relation-preserving contrastive design of HeteroGCL [15]: decision-time edge eligibility with recency weighting, and feature-view contrastive regularization that leaves relation summaries intact. The synthetic temporal evaluation also illustrates consequences beyond mean F1. Generated illicit prevalence and actor composition vary over time, so a threshold selected in the validation interval may transfer unevenly to later bins. This contributes to the observed between-seed dispersion in precision and recall. For investigative screening, AP and alert precision at a fixed review budget can be more directly actionable than a single thresholded F1; Table I provides AP, whereas review-budget measurements remain a useful next evaluation target.

C. Scope for real-chain validation

An Elliptic benchmark study should use its actual anonymized transaction attributes and directed edge files, train strictly on earlier time steps, and exclude unknown labels from the supervised objective. It should include a heterogeneous attention-based contrastive baseline under identical splits and evaluate confidence intervals over several seeds. Particular care is required for co-occurrence relations: time adjacency alone is not proof that two transactions share an address or operator. Temporal cluster nodes can be inserted when those definitions are reliably available, but proxy relations should be labeled as engineered rather than observed.

The evaluated synthetic generator provides transaction attributes, two typed directed relations, and sparse illicit labels under controlled temporal shifts. It therefore supports interpretable experiments on history eligibility, recency weighting, and contrastive regularization. Evaluation on observed blockchain transfers and independently adjudicated outcomes is the next step for estimating operational AML effectiveness.

VII. PROTOCOL ANALYSIS AND DEPLOYMENT CONSIDERATIONS

A. Formal decision-time information set

A classification service receiving transaction i at decision time t_i can consult only a filtration $F(t_i)$, defined as the sigma-algebra generated by attributes, transfer events and investigation decisions that are observable no later than t_i. The proposed input u_i is measurable with respect to $F(t_i)$ when the underlying transaction attribute x_i is observable at issuance and every neighbor j satisfies $t_j <= t_i$. Since m_i^r is a weighted sum of measurable neighbor attributes with weights that depend only on recorded timestamps, it is also measurable. The MLP prediction is a deterministic function of these quantities and its previously fitted parameters. This gives a direct and implementable sufficiency argument for the absence of future-edge information in the graph construction. This guarantee covers graph construction, and upstream feature pipelines require the same discipline. A feature engineering pipeline might use a full-day aggregate when classifying a transaction early in that day; a graph can be chronologically directed yet still encode an illegally future-defined attribute. Similarly, fit-on-all-data normalization leaks global distributional information. Our implementation avoids the latter by fitting the scaler on the training-label subset, but it assumes that the generator's per-transaction attributes are themselves observed immediately. In an operational dataset, every feature needs a separate availability timestamp. This data-engineering contract is an important complement to graph-model design.

The proof also clarifies same-time events. Elliptic publishes discretized time steps rather than event timestamps of arbitrary precision. If all events in a time step are treated as simultaneous, within-bin edges must either be excluded from strictly online inference or constructed using an independent transaction order. Our synthetic generator has a definitive ordering within each bin; the edge list is built only from earlier indices. Applied to a real benchmark with coarse temporal bins, the conservative alternative is to drop same-step relations when their direction cannot be established. This distinction prevents the temporal model from claiming more chronological precision than its data actually provide.

B. Why two semantically different channels matter

Flow edges encode a plausible transfer lineage and therefore support tracing provenance, whereas coordination edges encode common membership in a time-and-actor pattern in the generator. Conflating them into one adjacency makes it difficult to know which information produced the alert. Relation separation gives a transparent computational interface: both means can be logged independently, reviewed with edge counts, and replaced when domain engineers obtain higher-quality observations. Such an interface is particularly important in AML, where an association

edge should not be presented to an analyst as a confirmed fund transfer. Contrastive heterogeneous graph learning for collusive financial fraud offers another instructive example of preserving distinct relation semantics when identifying coordinated financial risk [27].

The evaluated encoder concatenates typed neighborhood means and learns their combination in the downstream MLP, rather than learning per-edge or per-relation attention weights. This makes the setup an auditable test bed for causal edge eligibility and recency decay: adding or removing a relation does not redefine the remaining channels. In the simulator, coordination edges are generated from latent actor equality and time-bin membership. Temporal attenuation is most useful when recent interactions are predictive and historical edges become stale. If the target behavior depends on long-range layering, an aggressive decay can instead discard decisive provenance. The present value tau=3 bins was fixed before reporting and was not selected by sweeping the test horizon. An expanded study should compare several lookback kernels, such as uniform, exponential and learned piecewise decay, with selection confined to validation periods. One can also include an explicit degree and edge-age vector in each relation channel, making the model sensitive to activity bursts even when neighbor-feature averages remain similar.

C. Confusion-matrix consequences of class imbalance

In a rare-event setting, the F1 score is highly sensitive to a small number of alerts. Let TP, FP and FN denote true positives, false positives and false negatives in the held-out labeled test period. Then precision=TP/(TP+FP), recall=TP/(TP+FN), and illicit F1=2TP/(2TP+FP+FN). A classifier that increases recall by aggressively alerting may improve F1 on one split but create an unmanageable investigative queue in a deployment. This is why Table I reports both precision and recall as well as threshold-independent rankings. The selected cutoff for each method and seed is recorded, permitting examination of decision-policy variability. ROC AUC measures the likelihood that a randomly selected illicit event receives a higher score than a randomly selected licit event, with standard handling of ties. Under imbalanced labels, average precision is particularly informative about ranked alerts [25]. In the seeded synthetic experiments, TA-HGCL's AUROC and AP remain below those of feature-only LR, even though its F1 exceeds that of the two relation-based neural ablations. An operational extension could evaluate top-k precision, captured illicit cases, and analyst review workload using a prespecified alert budget. The label-sampling scheme further matters. In the simulation, the training-label subset is drawn uniformly from the earlier period. Real investigations are usually non-random: analysts examine cases selected by existing detectors, and unlabeled transactions cannot safely be assumed legitimate. Consequently, a future observational benchmark should investigate positive-unlabeled learning and selection bias. Such a design also requires a clear definition of whether previously investigated historical decisions can be used as labels at a particular transaction's scoring date. The problem is not solved by a graph architecture alone.

D. Replication and analytic audit

Each run generates the seeded transactions and directed relations, trains every model, and records precision, recall, F1, AUROC, AP, thresholds and test-set counts; reported estimates are arithmetic means and sample standard deviations over seeds. The additional classifier comparisons and the causal-invariance audit in Section VIII follow the same protocol.The code deliberately minimizes nonessential dependencies: PyTorch supplies the neural encoder and contrastive objective, SciPy supplies sparse relation aggregation, and scikit-learn supplies logistic regression, scaling and classification metrics. The generator has a predictable number of transactions and a bounded predecessor search window, so it executes on a CPU. Every test node is predicted using its time-constrained feature representation even when it is not labeled; test labels are accessed only after

prediction to compute metrics. The code does inspect validation labels for model selection and the alert threshold, as explicitly required by the methodology.

A replication on Elliptic should load the real data directly rather than passing its features through the synthetic generator. It should use genuine transaction IDs and known flow edges, map edges into time-consistent directions, exclude unknown class labels from evaluation metrics, and avoid constructing actor-equivalence links from information that the public data do not provide. Training and validation temporal boundaries should be fixed before hyperparameter searches. Five or more initialization seeds, bootstrap intervals over temporal cohorts and per-period metrics further strengthen statistical conclusions.

VIII. ADDITIONAL MEASURED BASELINES AND CAUSAL AUDIT

A. Alternative classifier families and relation-feature value

We also ran supplementary classifier experiments on the same three seeded synthetic graphs to distinguish the temporal neural-ablation findings from broader baseline performance. A 200-tree random forest with balanced bootstrap weighting uses the original 12 attributes. Four balanced logistic regressions use the original attributes plus two typed historical neighbor averages, with uniform weighting or exponential decay at tau=1, tau=3, and tau=8 bins. All five classifiers use the same 350 labeled early-period nodes per seed, validation-period threshold selection, and final five time bins for testing. Table II reports the resulting independent model measurements.

Table 2ADDITIONAL SYNTHETIC BASELINES (MEAN ± SAMPLE SD, THREE SEEDS)

Method	Prec.	Recall	F1	AUROC	AP
RF	0.091± 0.043	0.420± 0.316	0.125± 0.036	0.626± 0.104	0.076± 0.031
LR-uniform	0.057± 0.022	0.378± 0.182	0.098± 0.037	0.598± 0.034	0.063± 0.025
LR-tau1	0.092± 0.094	0.326± 0.248	0.090± 0.036	0.590± 0.055	0.066± 0.026
LR-typed	0.084± 0.079	0.338± 0.268	0.107± 0.067	0.601± 0.036	0.063± 0.023
LR-tau8	0.048± 0.015	0.338± 0.268	0.079± 0.021	0.600± 0.035	0.064± 0.025

On the synthetic test graphs, supplementary mean F1 is 0.125 for RF, 0.098 for uniform typed LR, 0.090 for tau=1, 0.107 for tau=3, and 0.079 for tau=8. Random forest (0.125) and the primary feature-only LR (0.180) both exceed TA-HGCL (0.117) in mean F1. Thus the proposed encoder's observed advantage is specific to the evaluated relation-based neural ablations. The logistic-regression sensitivity results also show that typed summaries and stronger recency decay need not improve a linear estimator under this generator. The different tau configurations are reported as supplementary sensitivity checks rather than as test-selected neural hyperparameters.

B. Exact causal perturbation audit

We additionally performed an explicit causal-invariance audit for each seed. Every transaction at time bin 12 or later was assigned substantial synthetic feature noise, after which the entire typed relation aggregation was recomputed. In all 3 seeds, the feature representation of every one of the 1200 earlier transactions remained byte-for-byte unchanged, with a measured maximum absolute difference of 0.0. The directed edge lists also contained 0 future-to-past edges across these seeds.

This audit is a concrete unit test of the neighborhood eligibility rule and establishes exact input invariance of the implementation under future-feature perturbation.

C. Position of the contribution

TA-HGCL combines typed, causally constrained relation aggregation with contrastive MLP training. Its methodological contribution is the explicit coupling of admissible-history graph construction, relation-specific recency weighting, and feature-view regularization. Experiments on seeded synthetic transaction graphs isolate the relevant neural ablations and verify input invariance under future-feature changes, while identifying strong classical classifiers as essential comparison points. A learned heterogeneous-attention backbone can be assessed using the same temporal protocol.

IX. CONCLUSION

We presented TA-HGCL, which combines causal, recency-weighted heterogeneous relation encoding with feature-view contrastive regularization for cryptocurrency AML. By restricting typed neighborhood aggregation to observable transaction history, TA-HGCL attains a mean illicit F1 of 0.117 on three seeded Bitcoin-like graphs, above its causal no-contrastive (0.105) and non-temporal (0.092) variants, while the feature-only baselines indicate that the simulated node attributes carry much of the signal. The future-feature perturbation audit confirms exact invariance of historical relation inputs. Together, these findings establish a reproducible causal evaluation design and motivate relational encoders on observed transaction networks such as Elliptic.

REFERENCES

Foley, S., Karlsen, J. R., and Putniņš, T. J., "Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies?" Review of Financial Studies, vol. 32, pp. 1798–1853, 2019. doi:10.1093/rfs/hhz015.

Weber, M., et al., "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," arXiv:1908.02591, 2019.

Savage, D., Wang, Q., Chou, P., Zhang, X., and Yu, X., "Detection of money laundering groups using supervised learning in networks," arXiv:1608.00708, 2016.

Colladon, A. F. and Remondi, E., "Using social network analysis to prevent money laundering," Expert Systems with Applications, vol. 67, pp. 49–58, 2017.

Lorenz, J., Silva, M. I., Aparício, D., Ascensão, J. T., and Bizarro, P., "Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity," Proc. ACM AI in Finance, 2020.

D. Teng, "TRIAD: Deadline- and reliability-aware scheduling for post-earthquake structural assessment workflows," World Journal of Engineering Research, vol. 3, no. 6, pp. 55–66, 2025.

Hamilton, W. L., Ying, Z., and Leskovec, J., "Inductive representation learning on large graphs," NeurIPS, 2017.

Veličković, P., et al., "Graph attention networks," ICLR, 2018.

Wang, D., et al., "A semi-supervised graph attentive network for financial fraud detection," IEEE ICDM, pp. 598–607, 2019.

Pareja, A., et al., "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," AAAI, vol. 34, pp. 5363–5370, 2020. doi:10.1609/aaai.v34i04.5984.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K., "Inductive representation learning on temporal graphs," ICLR, 2020.

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., and Bronstein, M., "Temporal graph networks for deep learning on dynamic graphs," arXiv:2006.10637, 2020.

J. Chen, X. Zhang, and X. Han, "Explainable identification of tax-related financial reporting anomalies in U.S. public companies: Cross-fitted tax-consistency residuals and Owen-value Shapley propagation," World Journal of Economics and Business Research, vol. 1, no. 1, pp. 33–45, 2023.

Hu, Z., Dong, Y., Wang, K., and Sun, Y., "Heterogeneous graph transformer," WWW, pp. 2704–2710, 2020.

Chen, J., Liu, J., Liang, Y., and Zhou, M., "HeteroGCL: A heterogeneous graph contrastive learning framework for scalable and sustainable cryptocurrency AML," Applied Sciences, vol. 16, no. 6, art. 2860, 2026. doi:10.3390/app16062860.

Dou, Y., et al., "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," CIKM, 2020. doi:10.1145/3340531.3411903.

J. Chen, X. Han, X. Zhang, and Y. Yang, "Exact concept-level explanations for early detection of financial misstatements," Eurasia Journal of Science and Technology, vol. 5, no. 2, pp. 34–48, 2023.

van den Oord, A., Li, Y., and Vinyals, O., "Representation learning with contrastive predictive coding," arXiv:1807.03748, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G., "A simple framework for contrastive learning of visual representations," ICML, pp. 1597–1607, 2020.

You, Y., et al., "Graph contrastive learning with augmentations," NeurIPS, vol. 33, pp. 5812–5823, 2020.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L., "Deep graph contrastive representation learning," arXiv:2006.04131, 2020.

Hassani, K. and Khasahmadi, A. H., "Contrastive multi-view representation learning on graphs," ICML, pp. 4116–4126, 2020.

Wang, X., Liu, N., Han, H., and Shi, C., "Self-supervised heterogeneous graph neural network with co-contrastive learning," KDD, pp. 1726–1736, 2021.

X. Yue and R. Zhu, "An explainable hybrid machine learning framework for cash-flow-conditioned multi-horizon liquidity risk early warning: An SME-oriented benchmark study," World Journal of Management Science, vol. 2, no. 1, pp. 63–72, 2024.

Saito, T. and Rehmsmeier, M., "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," PLoS ONE, vol. 10, e0118432, 2015. doi:10.1371/journal.pone.0118432.

Y. Lin et al., "Subgraph-Level Temporal Graph Transformer With Laundering-Motif Prototypes for Anti-Money Laundering on Blockchains," IEEE Access, 2026, doi: 10.1109/ACCESS.2026.3731160.

Y. Jiao, H. Fan, X. Yue, W. Ping, T. Sun, and J. Wang, "Dynamic heterogeneous graph contrastive learning for uncovering collusive financial fraud," Scientific Reports, article in press, 2026, doi: 10.1038/s41598-026-58938-5.