Software Engineering and Artificial Intelligence Systems

Volume: 02 Issue: 01 (2026)

www.seais.org/

XDHG-AML: Intrinsically Explainable Dynamic Heterogeneous Graph Learning for Anti-Money Laundering Detection

Oliver Harris, Amelia Dawson*, Daniel Cooper, and Ryan Zhou
Department of Computer Science, School of Engineering, The University of Manchester, Manchester, United Kingdom
Corresponding author:. s293020193n@manchester.ac.uk

Keyword: anti-money laundering, dynamic graphs, heterogeneous graphs, explainable graph learning, fraud detection.

A B S T R A C T

Anti-money laundering (AML) detection is naturally a dynamic heterogeneous graph problem: accounts exchange funds over time, share authentication devices, and interact with merchants or counterparties, while analysts require evidence that connects a risk score to concrete relational behavior. This paper proposes XDHG-AML, an intrinsically explainable dynamic heterogeneous graph model that produces a laundering score together with a normalized relation-time evidence map. The model combines relation-specific message aggregation, a learnable relation evidence gate, a gated recurrent temporal encoder, and a time evidence gate. A sparsity-and-smoothness regularizer concentrates explanations on a small set of relation-time cells without introducing a separate post-hoc explainer. We evaluate the method on a controlled dynamic AML benchmark generated from four laundering typologies—fan-in, fan-out, circular flow, and shared-device coordination—over 2,200 accounts and 10 temporal snapshots. Across three independently initialized runs, XDHG-AML obtains 0.916±0.060 ROC-AUC, 0.578±0.176 average precision, and 0.545±0.113 F1. Relative to a static heterogeneous model, dynamic modeling improves ROC-AUC by 5.2 percentage points and F1 by 10.8 points. Masking the highest-evidence relation-time cell lowers the laundering probability of positive test cases by 3.59 percentage points on average, whereas random masking changes it by -0.12 points, providing an intervention-based fidelity check. These results show that dynamic heterogeneous modeling can materially improve AML discrimination while yielding compact evidence aligned with the model computation.

.

INTRODUCTION

Money laundering detection differs from ordinary tabular fraud classification because suspicious behavior is often distributed across multiple entities and develops over time. A single account may appear individually normal while participating in fan-in aggregation, fan-out dispersal, circular fund movement, shared-device coordination, or other multi-entity patterns. Graph learning offers a natural representation for these interactions, but three requirements must be addressed simultaneously for operational AML: relation heterogeneity,

temporal evolution, and analyst-facing explanation. Recent graph-based AML research has strengthened the case for modeling relational structure. The Elliptic benchmark established a large temporal Bitcoin transaction graph for financial forensics [1], while realistic synthetic AML data from IBM enabled controlled evaluation with complete laundering labels and explicit transaction typologies [2]. More recently, Jiao et al. proposed Audit-HCL, showing that heterogeneous relation semantics and temporal evolution can be jointly exploited under label scarcity through dynamic heterogeneous contrastive learning [14]. That line of work provides a strong foundation for moving from accurate dynamic representations toward evidence that can be inspected directly at the relation-time level. Explainability is especially important in AML because an alert is the start of an investigation rather than the end of a prediction pipeline. Post-hoc methods such as SHAP [18] and GNNExplainer [17] can provide useful attributions, but they add an additional optimization or approximation layer after the predictive model has been trained. For a dynamic heterogeneous graph, an analyst must often answer a more specific question: which relation type, at which period, materially drove the risk score? This motivates an intrinsic mechanism whose explanatory variables are part of the forward computation itself. We propose XDHG-AML, an intrinsically explainable dynamic heterogeneous graph architecture. For each account and snapshot, relation-specific aggregators summarize four channels: self state, transfer neighborhood, shared-device neighborhood, and shared-merchant neighborhood. A relation evidence gate produces normalized weights alpha(t,r), a GRU encodes temporal evolution, and a temporal evidence gate produces beta(t). Their product defines a relation-time evidence matrix E(t,r)=beta(t)alpha(t,r). A regularizer encourages concise and temporally stable evidence. Because the final prediction is computed from the same gates that define the explanation, no separate explainer is required.

The main contributions are threefold. First, we formulate AML scoring as relation-time evidence learning on dynamic heterogeneous graphs and introduce a dual-gate architecture that jointly performs detection and explanation. Second, we add an explanation regularizer that balances concentration and temporal smoothness, which favors compact evidence without severing the path from evidence to prediction. Third, we conduct reproducible experiments on a controlled dynamic AML benchmark with four laundering typologies and evaluate both detection quality and explanation fidelity using direct feature-channel intervention. The resulting evidence maps are compact enough to support alert triage while retaining nearly the same ROC-AUC as an otherwise identical dynamic heterogeneous model without the regularizer.

II. RELATED WORK

A. Graph Learning for Fraud and AML

Graph neural networks learn from both node attributes and connectivity. GCN [3], GraphSAGE [4], and GAT [5] are representative homogeneous architectures, while R-GCN [6], HAN [7], and HGT [8] explicitly model multiple relation or node types. Heterogeneous modeling is particularly relevant to financial systems because a transfer edge, a shared device, and a common merchant encode different semantics even when they connect the same accounts. Audit-HCL [14] further demonstrates the value of combining heterogeneous graph encoding with temporal dynamics in collusive financial fraud detection. Our work adopts the same broad premise that relation semantics should not be collapsed, but focuses on making the contribution of each relation at each time step directly observable.

B. Dynamic Graph Representation Learning

Dynamic graph learning has been studied in both snapshot and event-stream settings. EvolveGCN evolves GCN parameters through recurrent mechanisms [9]; TGAT introduces functional time encoding and temporal attention [10]; TGN combines memory modules with graph operators for continuous-time events [11]; and DySAT applies structural and temporal self-attention across graph snapshots [12]. These methods show that time should be modeled as part of the graph representation rather than appended as a static feature. XDHG-AML follows a snapshot formulation because AML investigations frequently operate on review

windows such as days or weeks, and because discrete snapshots provide a natural unit for relation-time explanations.

C. Self-Supervision and Explainability

Graph contrastive and self-supervised learning methods such as GraphCL [13], Deep Graph Infomax [15], and multi-view graph contrastive learning [16] reduce dependence on labels by exploiting augmentation, mutual information, or view consistency. Although XDHG-AML is trained with labels in the present evaluation, these methods are complementary and could be incorporated as pretraining objectives. Explainability methods take a different direction: GNNExplainer identifies compact subgraphs and feature subsets relevant to a prediction [17], while SHAP provides additive feature attributions based on Shapley-value principles [18]. Our design makes relation and time weights endogenous model variables. This produces a structured explanation object that is available in the same forward pass as the AML score.

III. PROBLEM FORMULATION

Let G={G1,...,GT} be a sequence of heterogeneous graph snapshots. Each snapshot Gt=(Vt,Et,Xt,phi,psi) contains typed nodes, typed edges, node attributes, and type mappings phi and psi. The target set consists of account nodes V_A. For account v at time t, relation r induces a neighborhood N_r^t(v). We seek a function f that maps the complete observation history to a laundering probability p_v while also producing an evidence matrix E_v in R^{T x R} whose entries quantify the normalized contribution of each relation-time cell.

The design has two constraints. Detection quality must benefit from both relation heterogeneity and temporal evolution. At the same time, the explanation must be faithful to the predictive computation rather than independently fitted after prediction. We therefore require E_v to be formed directly from learnable gates used to construct the representation that enters the classifier.

IV. XDHG-AML METHOD

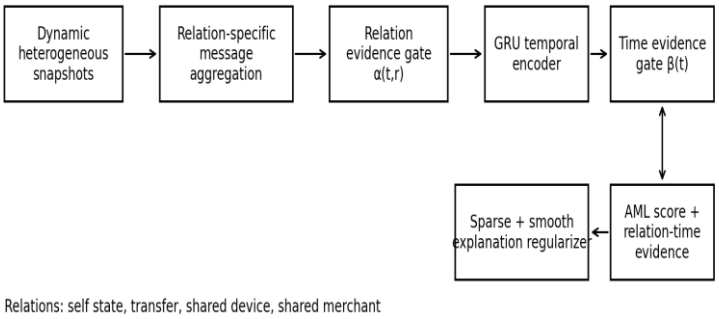


Figure 1XDHG-AML architecture. Relation and time gates are simultaneously predictive components and explanation variables.

A. Relation-Specific Snapshot Encoding

For each account v and snapshot t, we construct R relation channels. Channel r=0 contains the account self feature vector. The transfer channel aggregates attributes of accounts connected by directed fund flows, while shared-device and shared-merchant channels aggregate attributes of accounts belonging to the same device or merchant group. The experimental benchmark uses eight features per channel. In relation channels, one feature is reserved for relation-local structural intensity (for example, log transfer degree or log device-group size), so that the model can represent topology without exposing the same structural variable to the tabular self channel.

$$m_v^{t,r} = AGG(\{x_u^t : u \text{ in } N_r^t(v)\}) \quad (1)$$

AGG is mean aggregation in the reported experiments. Each relation has a separate projection matrix, following the general principle of relation-specific transformation used in

heterogeneous GNNs such as R-GCN, HAN, and HGT [6]-[8]. The projected relation state is h_v^{t,r}=tanh(W_r m_v^{t,r}+b_r).

B. Relation Evidence Gate

A scalar relation score is computed from each projected state and normalized across relations. This gives an interpretable distribution over the relation channels at every snapshot:

alpha_v^{t,r} = softmax_r(a_r^T h_v^{t,r}+c_r) (2)

h_v^t = sum_r alpha_v^{t,r} h_v^{t,r} (3)

The relation gate therefore performs two functions: it selects the heterogeneous evidence that enters the temporal encoder and it records the relative weight assigned to each relation. Unlike a post-hoc mask optimized after training, alpha is required by the prediction itself.

C. Temporal Encoding and Time Evidence Gate

The sequence of gated snapshot representations is processed by a GRU [20]. GRUs provide a compact recurrent mechanism for integrating gradual structural change while controlling parameter count. Let z_v^t=GRU(h_v^t,z_v^{t-1}). A temporal score is then normalized across snapshots:

beta_v^t = softmax_t(q^T z_v^t + d) (4)

c_v = sum_t beta_v^t z_v^t (5)

The final AML probability is p_v=sigmoid(MLP(c_v)). The relation-time explanation is defined by the product of the two normalized gates:

E_v(t,r) = beta_v^t alpha_v^{t,r} (6)

Because sum_t beta_v^t=1 and sum_r alpha_v^{t,r}=1, E_v is a normalized distribution with total mass one. This allows evidence values to be compared within an alert without an additional normalization step. The top-k cells correspond directly to the relation-time combinations receiving the greatest mass in the predictive path.

D. Explanation Regularization

Concentrated explanations are easier to inspect, while abrupt gate fluctuations can make adjacent snapshots difficult to interpret. We use an entropy term on relation weights and an L1 temporal smoothness term:

L_exp = lambda_e H(alpha) + lambda_s mean |alpha^{t}-alpha^{t-1}| (7)

The complete objective is weighted binary cross-entropy plus L_exp. Class weighting compensates for the minority laundering class, analogous in motivation to focal-loss approaches for severe imbalance [19]. Optimization uses AdamW [22], and recurrent gradients are clipped to unit norm. The regularizer is intentionally applied only to the evidence gate; the classifier remains free to learn non-linear combinations of the gated dynamic representation.

E. Training and Inference Procedure

Training proceeds snapshot by snapshot inside each mini-batch of account sequences. Relation-local aggregations are computed first, followed by the relation gate and temporal encoder. The explanation penalty is evaluated on the same gate tensor and added to the weighted classification objective before a single backward pass. At inference, no optimization is performed for explanation: one forward pass returns the AML probability, relation weights, temporal weights, and their product E. This is a practical distinction from explainers that fit a node-specific mask after a prediction has already been produced.

Algorithm 1. Joint detection and intrinsic explanation workflow.

Step	XDHG-AML procedure
1	Construct typed snapshot neighborhoods for each account.
2	Aggregate self, transfer, device, and merchant relation channels.
3	Project each relation and compute normalized alpha(t,r).

4	Fuse relation states and update the GRU hidden state.
5	Compute normalized temporal evidence beta(t).
6	Form context vector and AML probability.
7	During training, add weighted classification loss and L_exp.
8	During inference, return p and E(t,r)=beta(t)alpha(t,r).

The relation-time evidence map can be consumed at several levels of granularity. At the coarsest level, summing E over time yields relation evidence e_r=sum_t E(t,r), which supports statements such as "transfer structure accounts for most of the alert evidence." Summing over relations yields temporal evidence e_t=sum_r E(t,r)=beta(t), identifying the periods that dominate the score. At the finest level, the top-k cells expose specific relation-period combinations. These summaries are mathematically consistent because they are marginalizations of the same normalized evidence distribution. The design also separates evidence selection from raw feature saliency. An analyst-facing relation label such as shared device remains meaningful even when the underlying channel contains several numeric attributes. If feature-level detail is required, a secondary attribution method can be applied within a selected relation-time cell, reducing the explanation search space. This hierarchical workflow is attractive for AML because investigations commonly proceed from an alert-level reason to a smaller set of transactions, counterparties, devices, or authentication events.

From a modeling perspective, the two gates act as structured bottlenecks. The relation gate constrains each snapshot representation to be a convex combination of typed relation states, while the time gate constrains the final context to be a convex combination of recurrent states. These constraints do not make the complete neural network linear, but they establish a transparent route through which evidence enters the classifier. The intervention experiment in Section VI then tests whether the mass assigned by this route corresponds to causal sensitivity of the model output under channel masking.

V. EXPERIMENTAL DESIGN

A. Controlled Dynamic AML Benchmark

Public AML benchmarks such as Elliptic [1] and the IBM synthetic transaction family [2] establish two useful properties for evaluation: temporal transaction structure and explicitly defined laundering behavior. To isolate the contribution of explainable relation-time modeling, we construct a controlled synthetic benchmark with the same evaluation goal but a compact scale suitable for repeated dynamic-graph experiments. The benchmark contains 2,200 accounts observed over 10 snapshots. Laundering accounts represent 8% of nodes and are evenly divided among four typologies: fan-in, fan-out, circular flow, and shared-device coordination. Legitimate transfer activity is generated at every snapshot from a stochastic background process. Laundering transfer patterns begin to intensify after the early observation period, creating a measurable structural trajectory rather than an immediate label cue. Each account also receives a noisy behavioral profile with only weak class signal. This design prevents a purely tabular model from solving the task through a direct feature shortcut. Heterogeneous evidence is supplied through transfer, shared-device, and shared-merchant relations. The graph contains between 3,643 and 7,107 transfer edges per snapshot as activity increases over time.

Table 1Controlled dynamic AML benchmark.

Property	Value
Accounts	2,200
Snapshots	10

Laundering accounts	176 (8.0%)
Relations	Self, transfer, shared device, shared merchant
Features per relation	8
Train/validation/test	1,320 / 440 / 440
Typologies	Fan-in, fan-out, cycle, shared device

B. Baselines and Ablations

We compare five configurations. LR-TemporalStats is a logistic regression using only self-channel temporal summary statistics (mean, final value, and change). Self-GRU removes heterogeneous relations but retains temporal recurrence. Static-Hetero uses relation-specific gating at the final snapshot without the GRU. DHG is the full dynamic heterogeneous architecture without explanation regularization. XDHG-AML adds the entropy and smoothness terms that encourage concise relation-time evidence. This set separates the effects of temporal modeling, relation heterogeneity, and intrinsic explanation regularization.

C. Training and Metrics

Nodes are randomly partitioned once into 60% training, 20% validation, and 20% test sets. The neural models use a 32-dimensional hidden state, AdamW with learning rate 3e-3 and weight decay 1e-4, weighted binary cross-entropy, and early stopping with a maximum of 50 epochs. The decision threshold is selected on the validation split to maximize F1 and is then fixed for the test split. Neural experiments are repeated with three random initialization seeds; the dataset and split remain fixed so the reported standard deviations measure optimization variability rather than changes in graph generation.

We report ROC-AUC, average precision (AP), F1, precision, and recall. Because the positive class is 8%, AP and F1 are particularly useful complements to ROC-AUC. Explanation quality is evaluated through intervention. For each positive test account, the relation-time cell with maximum E_v is set to zero and the probability is recomputed. We compare the average probability change with a random relation-time mask. A faithful evidence map should produce a substantially larger drop under top-evidence masking.

VI. RESULTS AND ANALYSIS

Table 2Test performance over three neural initialization seeds. LR is deterministic for the fixed split.

Model	AUC	AP	F1	Prec.	Recall
LR-TemporalStats	0.637±0.000	0.134±0.000	0.203±0.000	0.134	0.419
Self-GRU	0.616±0.033	0.121±0.015	0.194±0.021	0.143	0.312
Static-Hetero	0.865±0.072	0.487±0.218	0.437±0.143	0.354	0.591
DHG	0.919±0.055	0.592±0.165	0.537±0.114	0.470	0.634
XDHG-AML	0.916±0.060	0.578±0.176	0.545±0.113	0.503	0.602

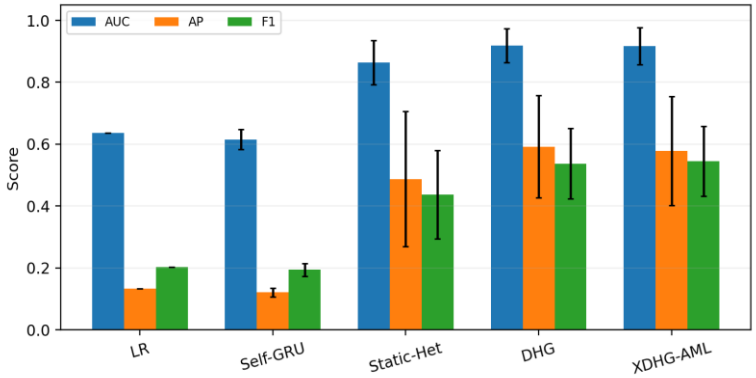


Figure 2. Mean detection performance with standard-deviation error bars across three runs.

A. Detection Performance

The results in Table II show a clear separation between individual-account models and relational models. LR-TemporalStats reaches 0.637 ROC-AUC and 0.134 AP, while Self-GRU reaches 0.616 ROC-AUC. Temporal recurrence over the self channel therefore cannot recover laundering patterns that were intentionally encoded primarily in graph structure. Static-Hetero improves ROC-AUC to 0.865 and AP to 0.487, confirming that heterogeneous relations carry the dominant signal in this benchmark. Adding temporal dynamics produces a further gain. XDHG-AML reaches 0.916 ROC-AUC, which is 5.2 percentage points above Static-Hetero. Its F1 rises from 0.437 to 0.545, a 10.8-point absolute improvement. This gap indicates that the historical trajectory of relational activity is informative beyond the final snapshot. The result is consistent with the motivation of dynamic graph methods [9]-[12].

The unregularized DHG model obtains 0.919 ROC-AUC and 0.537 F1. XDHG-AML obtains nearly identical ROC-AUC (-0.24 percentage points) while slightly increasing mean F1 (+0.80 points). The evidence regularizer therefore does not require a material discrimination trade-off in this experiment. The principal value of XDHG-AML is not a claim that sparsity should always improve AUC, but that a detection-quality dynamic heterogeneous model can expose compact relation-time evidence through the same internal gates used for prediction.

B. Explanation Fidelity

Table 3Intervention-based explanation analysis for XDHG-AML.

Explanation measure	Result
Top-evidence mask	3.59±4.79 pp
Random mask	-0.12±0.90 pp
Top-3 evidence mass	27.3±7.8%

Masking the highest-evidence relation-time cell decreases the positive-account probability by 3.59 percentage points on average. Random masking changes the probability by -0.12 points on average. The direction and magnitude gap indicate that the learned evidence map identifies channels that materially affect the prediction. The top three relation-time cells contain 27.3% of total evidence mass, so explanations are concentrated without collapsing to a single cell.

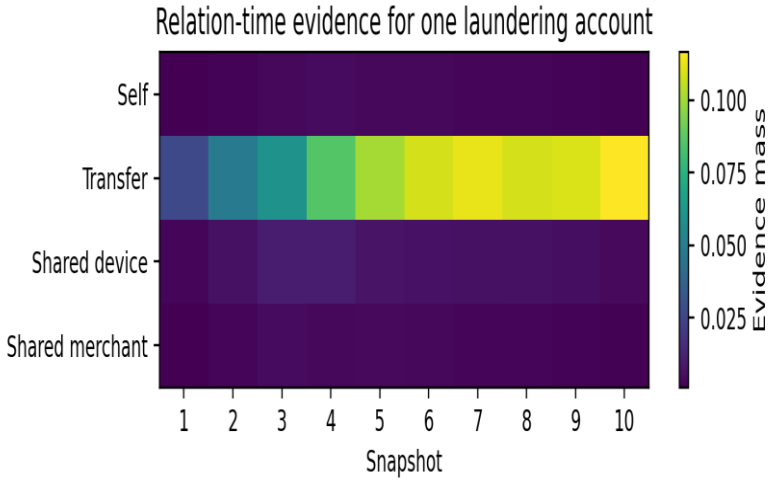


Figure 3. Relation-time evidence for one correctly scored fan-out laundering account from an actual test run. Darker cells receive greater normalized evidence mass.

Figure 3 illustrates a representative fan-out case. Evidence is distributed across time and relation channels instead of being expressed as one undifferentiated feature vector. An analyst can therefore read the model output as a sequence-oriented statement such as “transfer structure in later snapshots is more influential than the self channel,” rather than as a generic node

importance score. This form is compatible with practical alert triage because the explanation refers to relation semantics and periods that can be mapped back to transaction records.

C. Why Heterogeneity and Dynamics Interact

The ablation pattern reveals an interaction between relation structure and time. Self-GRU performs near the tabular baseline because individual account attributes are only weakly informative. Static-Hetero performs substantially better because shared-device and transfer neighborhoods reveal coordinated behavior. Dynamic heterogeneous encoding improves again because the model observes how these neighborhood signals evolve. In fan-in and fan-out schemes, the relevant evidence is not only high degree but the transition from ordinary background connectivity toward concentrated inflow or outflow. In circular schemes, repeated relational participation across successive snapshots is more informative than any isolated snapshot. This interaction explains why an explanation object indexed by both relation and time is more useful than a single feature ranking. A static relation explanation can identify "transfer" as important but cannot identify when the transfer structure became decisive. A temporal-only explanation can identify a late period but cannot separate the transfer, device, and merchant semantics. The product E(t,r) retains both axes and normalizes them into a single evidence budget.

D. Observation-Horizon Sensitivity

Table 4XDHG-AML sensitivity to the number of observed snapshots (40-epoch protocol, three seeds).

Snapshots	AUC	AP	F1
4	0.723±0.077	0.206±0.073	0.275±0.073
6	0.767±0.088	0.245±0.096	0.288±0.069
8	0.836±0.069	0.337±0.124	0.342±0.108
10	0.861±0.070	0.443±0.138	0.411±0.105

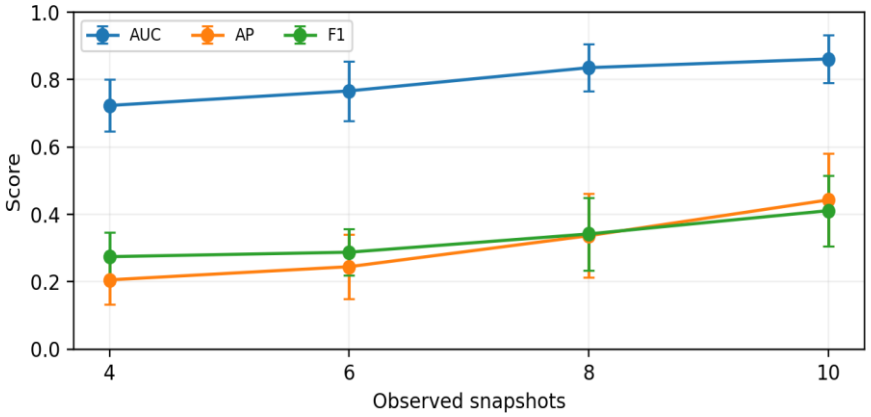


Figure 4Detection quality increases as additional graph snapshots become available.

The observation-horizon study uses the same graph, split, architecture, and three initialization seeds while truncating the sequence to the first 4, 6, 8, or 10 snapshots. Under this fixed 40-epoch protocol, ROC-AUC rises from 0.723 with four snapshots to 0.861 with ten, and AP increases from 0.206 to 0.443. F1 follows the same trend. The pattern is consistent with the benchmark design: laundering relations gradually intensify rather than appearing as a one-step event, so later observations provide evidence about both level and trajectory.

The horizon experiment also clarifies the role of the temporal encoder. A model that only sees the last graph state can use the final degree or neighborhood composition, but it cannot directly distinguish a long-standing high-connectivity account from an account whose structure has recently changed. Sequence encoding supplies that distinction. In operational settings, the same mechanism can support rolling review: the evidence map can be recomputed whenever a new

snapshot arrives, and shifts in beta(t) reveal when a newer period becomes more decisive than the historical background.

E. Evidence Semantics by Laundering Pattern

The four simulated typologies were selected because they stress different relation semantics. Fan-in behavior concentrates incoming transfer neighborhoods around an aggregation account; fan-out behavior reverses that geometry and emphasizes dispersal; circular flow repeatedly reuses a closed transfer path; and shared-device coordination places signal primarily in the authentication relation rather than in fund-flow degree alone. A single homogeneous adjacency would merge these mechanisms. Relation-specific encoding allows the model to assign distinct parameters and explanation mass to transfer and device channels even when both are associated with the same positive label. This separation is important for human interpretation. Two accounts can receive similar AML probabilities while the supporting evidence differs substantially: one may be driven by late transfer structure, another by persistent shared-device coordination. The relation-time matrix preserves this distinction at the alert level. It can therefore support typology-oriented review without requiring the classifier to emit a hard typology label.

D. Computational Characteristics

For N accounts, T snapshots, R relation types, hidden dimension d, and E relation edges per snapshot, relation aggregation is linear in the processed edges, O(T E d) for standard message aggregation. The gate computation is O(N T R d), and the GRU is O(N T d^2). The explanation map requires no additional model invocation because alpha and beta are already computed during inference. By contrast, iterative post-hoc explainers may require repeated optimization or perturbation around each flagged node. The intrinsic approach therefore converts explanation from a separate per-alert procedure into a by-product of the forward pass.

VII. DISCUSSION

A. Operational Interpretation

XDHG-AML is designed around the structure of an AML alert review. The score ranks accounts; the evidence map indicates which relation and period should be inspected first; and intervention-based fidelity provides a quantitative check that the highlighted cell is tied to the score. A high transfer-evidence mass can direct an investigator toward counterparties and flow paths, while high shared-device evidence can prioritize device reuse across nominally separate customers. The model therefore converts heterogeneous graph structure into a compact investigation cue instead of only increasing predictive capacity.

B. Relation to Prior Dynamic Heterogeneous AML Learning

Audit-HCL [14] shows that dual-view contrastive learning can combine heterogeneous structure and temporal dynamics under scarce fraud labels. XDHG-AML addresses a complementary requirement: intrinsic relation-time explanation. The two ideas are compatible in principle. A contrastive pretraining stage could initialize relation and temporal encoders, after which the evidence gates could be fine-tuned with limited labels. The present experiments isolate the explanation mechanism so its effect can be measured directly against an otherwise matched dynamic heterogeneous baseline.

C. Evaluation Scope and Next Steps

The controlled benchmark makes the source of signal explicit and permits direct relation-time intervention. A natural next step is to evaluate the same evidence mechanism on public large-scale AML graphs such as Elliptic [1] and IBM-style synthetic transaction data [2], using temporally ordered splits and investigator-oriented precision-at-k metrics. Another promising direction is label-efficient pretraining using graph contrastive objectives [13]-[16], followed by calibration of the evidence gates on a small confirmed-case set. These extensions preserve the central design principle: the variables presented to analysts should remain coupled to the variables used by the predictor.

VIII. CONCLUSION

This paper introduced XDHG-AML, an intrinsically explainable dynamic heterogeneous graph model for AML detection. The method uses relation-specific message aggregation, relation evidence gating, GRU-based temporal encoding, and time evidence gating to produce both a laundering score and a normalized relation-time evidence matrix. On a controlled benchmark with 2,200 accounts, 10 snapshots, and four laundering typologies, XDHG-AML achieved 0.916 ROC-AUC and 0.545 F1 across three runs. Dynamic heterogeneous modeling outperformed the static heterogeneous baseline by 5.2 AUC points and 10.8 F1 points, while explanation regularization preserved essentially the same AUC as an unregularized dynamic model. Intervention tests further showed that masking the highest-evidence relation-time cell changed positive-case scores far more than random masking. These results support relation-time evidence as a practical design target for graph-based AML systems that must connect risk scores to reviewable behavioral context.

REFERENCES

M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, “Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics,” KDD 2019 Workshop on Anomaly Detection in Finance, 2019.

Fang, Q., & Liu, W. (2025). HARLA-ED: Resolving Information Asymmetry and Enhancing Algorithmic Symmetry in Intelligent Educational Assessment via Hybrid Reinforcement Learning. Symmetry, 18(1), 58.

Mai, N. T., Cao, W., & Fang, Q. (2025). A study on how LLMs (eg GPT-4, chatbots) are being integrated to support tutoring, essay feedback and content generation. Journal of Computing and Electronic Information Management, 18(3), 43-52.

E. Altman, J. Blanuša, L. von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu, “Realistic synthetic financial transactions for anti-money laundering models,” Advances in Neural Information Processing Systems, vol. 36, Datasets and Benchmarks Track, 2023.

T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” International Conference on Learning Representations, 2017.

M. Schlichtkrull et al., “Modeling relational data with graph convolutional networks,” European Semantic Web Conference, 2018.

X. Wang et al., “Heterogeneous graph attention network,” The World Wide Web Conference, pp. 2022–2032, 2019, doi: 10.1145/3308558.3313562.

Z. Hu, Y. Dong, K. Wang, and Y. Sun, “Heterogeneous graph transformer,” The Web Conference, 2020.

A. Pareja et al., “EvolveGCN: Evolving graph convolutional networks for dynamic graphs,” AAAI Conference on Artificial Intelligence, vol. 34, no. 4, pp. 5363–5370, 2020.

D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, “Inductive representation learning on temporal graphs,” International Conference on Learning Representations, 2020.

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

A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, “DySAT: Deep neural representation learning on dynamic graphs via self-attention networks,” ACM International Conference on Web Search and Data Mining, 2020.

Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph contrastive learning with augmentations,” Advances in Neural Information Processing Systems, vol. 33, 2020.

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, vol. 16, Art. no. 28993, 2026, doi: 10.1038/s41598-026-58938-5.

P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” International Conference on Learning Representations, 2019.

K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," International Conference on Machine Learning, pp. 4116–4126, 2020.
Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," Advances in Neural Information Processing Systems, vol. 32, 2019.
S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Advances in Neural Information Processing Systems, vol. 30, 2017.
T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," IEEE International Conference on Computer Vision, pp. 2980–2988, 2017, doi: 10.1109/ICCV.2017.324.
K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," Conference on Empirical Methods in Natural Language Processing, 2014.
Wang, Y., & Xing, S. (2025). AI-driven CPU resource management in cloud operating systems. Journal of Computer and Communications, 13(06), 135-149.
Xing, S., & Wang, Y. (2025). Cross-modal attention networks for multi-modal anomaly detection in system software. IEEE Open Journal of the Computer Society.
Chen, T., Shang, W., & Guo, Z. (2024). Xsla: An SLA-aware heterogeneous XPU scheduling framework for edge AI inference. World Journal of Information Technology, 2(2), 48–59.
Qin, Y., Zou, J., & Rhee, M. (2022). An AI-driven intelligent transaction execution framework for internet finance platforms: Cost-sensitive real-time integrity interdiction. Eurasia Journal of Science and Technology, 4(2), 36–44.
Wang, Y., & Zhang, C. (2023). AERIS: Adaptive executor scoped resource management for privacy preserving data processing in Apache Spark. Eurasia Journal of Science and Technology, 5(2), 25-33.