LEAP-HGN: A BRAIN-INSPIRED AMYGDALA-PARIETAL DETECTION FRAMEWORK WITH LLM-DRIVEN EVOLUTIONARY RULES FOR ANTI-MONEY LAUNDERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Money laundering activities are increasingly evolving into a time-sensitive, dense micro-circulation pattern, posing severe challenges to existing graph neural network technologies in dealing with the disruption of continuity caused by discrete time slices, the signal drowning due to massive business noise, and the logical lag resulting from the dynamic evolution of strategies. To address these issues, we propose the **LEAP-HGN** (**L**LM-**E**volved **A**mygdala-**P**arietal **H**yper**G**raph **N**etwork), a novel approach that simulates the human brain's threat processing mechanism within a unified brain-like cognitive framework. This model first constructs a spatio-temporal multi-view money laundering hypergraph, explicitly restoring the disrupted temporal causal chains of funds. On this basis, we redesign a biomimetic amygdala-parietal synergy mechanism: the amygdala low pathway uses neural gain modulation technology to transform expert rules into explicit warning factors, forcibly amplifying the response of sparse high-risk nodes before deep feature propagation, thereby preventing signal dilution from the source; the parietal high pathway performs deep spatial reasoning through Markov high-order diffusion and residual attention methods, penetrating the dense noise background from a global perspective and reconstructing hidden long-distance multi-hop links. Additionally, we design a Large Language Model (LLM) self-evolutionary closed loop based on hard sample mining, an LLM-driven loop that parses hard samples into symbolic rules for autonomous re-injection, achieving continuous self-evolution without human intervention. Extensive experiments on six money laundering datasets demonstrate that LEAP-HGN significantly outperforms 12 SOTA baselines. Code and datasets will be made publicly available at `https://anonymous.4open.science/r/LEAP-9AEF` upon acceptance.

1 INTRODUCTION

Syndicated money laundering has emerged as a critical destabilizing factor for global financial systems as it evolves from simple unidirectional transfers into sophisticated micro-cyclic patterns with high temporal sensitivity (Deprez et al., 2025; Han et al., 2026). Rather than utilizing traditional linear flows (Li et al., 2025b), laundering syndicates now leverage short-term and high-frequency hierarchical micro-transactions to construct complex high-order circular transaction linkages. By fragmenting illicit proceeds into granular amounts known as structuring and orchestrating multi-layered transfers across cross-border channels (Oztas et al., 2023; Wu et al., 2024), perpetrators leverage diverse financial institutions (de Perthuis & Pointcheval, 2022) and various account types, including shell companies and public accounts, to effectively obfuscate the origin of funds (Garg & Gupta, 2025). These intricate structural camouflages bury abnormal patterns within massive and heterogeneous transaction networks, posing significant challenges to conventional detection methods (Shi et al., 2022).

Real-world detection scenarios face three serious challenges: **(1) Disruption of Temporal Continuity and Trajectory Fragmentation in Short-Term High-Frequency Transactions.** In high-velocity

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

money laundering scenarios, illicit fund flows often execute multi-layered transfers within extremely compressed timeframes to establish densely connected and continuous transaction trajectories (Li et al., 2020; Shadrooh & Nørvåg, 2024; Chai et al., 2023b). However, conventional monitoring mechanisms predominantly employ discrete time-window partitioning to process data (Seo et al., 2024). Such a fragmented observation paradigm disrupts the temporal continuity and sequential dependencies inherent in money laundering, inevitably segmenting integral fund flows into seemingly unrelated, disjointed transactions (Shadrooh & Nørvåg, 2024; Deprez et al., 2025). A recent study (Han et al., 2026) emphasizes that moving beyond isolated transaction analysis is crucial for reconstructing these severed chains. Consequently, critical short-term structural patterns are obscured within the fragmented data due to trajectory truncation (Han et al., 2026; Shadrooh & Nørvåg, 2024), thereby hindering the system's ability to holistically reconstruct the complete lineage of the fund chain (Ahmed et al., 2025; Li et al., 2023). **(2) Signal Dilution and Submersion of High-Order Micro-Cyclic Links within Dense Noise Backgrounds.** By atomizing large capital into numerous micro-transactions and constructing multi-hop linkages across hierarchical levels, perpetrators effectively sever the direct causal associations between source and destination (Liu et al., 2021). Furthermore, these illicit flows are deeply embedded within massive volumes of legitimate transactions. The resulting overwhelming transaction noise masks the sparse and latent laundering signals (Tian et al., 2025a; Cheng et al., 2023). In such highly noisy environments, critical evidential chains are submerged during deep propagation, rendering the separation of weak laundering indicators from dense noise intractable (Tang et al., 2022). Consequently, this leads to the systematic under-detection of covert syndicates, a challenge that the conventional neighbor aggregation mechanisms of existing Graph Neural Networks (GNNs) fail to effectively address (He et al., 2025). This failure stems from the indiscriminate nature of traditional aggregation, which absorbs massive normal background transactions during deep propagation and causes weak laundering features to be severely over-smoothed and diluted. **(3) Rigidity of Discriminative Logic and Breakdown of Traceable Evidence Chains Due to Dynamic Evolutionary Strategies.** The transaction strategies employed by money laundering syndicates exhibit highly non-stationary distributions characterized by the continuous adaptation of temporal sequences and dense looping motifs (Pareja et al., 2020). However, existing detection models (Sun et al., 2022; Song et al., 2025b) predominantly rely on static discriminative logic derived from historical priors. Consequently, when confronted with the rapid evolution of laundering tactics, these models fail to generalize to abnormal patterns that have shifted beyond their original decision boundaries (Lin et al., 2024). Despite advances in representation learning, the fragmentation of evidence chains remains a critical obstacle (Luo et al., 2024). Furthermore, even when discrete local anomalies are captured, the lagging discriminative mechanisms struggle to reconstruct the evolved and complex topological structures (Chai et al., 2023a). This disconnection results in the fragmentation of the evidence chain, where isolated clues cannot be effectively consolidated into a complete and traceable narrative required for regulatory compliance and real-world deployment (Fu, 2025; Tian et al., 2025b).

To address the above challenges, we propose the **LEAP-HGN** (**L**LM-**E**volved **A**mygdala-**P**arietal **H**yper**G**raph **N**etwork) inspired by the Amygdala-Parietal dual-path risk perception mechanism. Mechanistically, the framework orchestrates two synergistic pathways: (1) Amygdala-Inspired Low Path: It is utilized to rapidly and dynamically amplify the salience of high-risk signals from suspicious nodes, ensuring immediate attention to potential threats despite noisy backgrounds. (2) Parietal-Inspired High Path: Following the initial alert, this mechanism engages in complex spatial reasoning and global cognitive integration to provide a structural context for the detected anomalies.

LEAP-HGN is architected into four synergistic modules: (1) Spatio-Temporal Multi-view Hypergraph construction (STMH) which establishes the structural foundation; (2) Amygdala-driven prior Rule Risk Warning (ARRW) serving as the fast-response mechanism; (3) Parietal-Collaborative Residual high-order learning and Risk assessment (PCRR) functioning as the deep cognitive processor; and (4) Self-evolutionary Rule Closed-Loop via hard sample mining (SRCL) ensuring continuous adaptation. The ARRW module transforms discrete expert rules into explicit risk warning indicators. Through a gain modulation mechanism, the gradient response of high-risk nodes is dynamically amplified prior to feature assimilation by noise, effectively blocking the dilution path of sparse micro-circulation signals during deep propagation. The PCRR module employs residual attention correction and Markov high-order diffusion processes to execute refined spatial reasoning, reconstructing causal structures previously obscured by noise. Within the STMH module, we adopt a synergistic modeling strategy integrating account topology, behavioral pattern dynamics, temporal cascade dependencies, and money laundering homophily. By effectively repairing fragmented temporal linkages, it facilitates

the rapid and holistic reconstruction of the obfuscated fund trajectory. Finally, the SRCL module harnesses the semantic logical reasoning capabilities of Large Language Models (LLMs) to execute a deep diagnosis of structural defects in hard samples that reside beyond legacy decision boundaries. We employ a semantic parser to translate these unstructured diagnostic insights into executable logic, whereby novel corrective symbolic rules are synthesized and retro-injected into the ARRW module. This empowers the system to autonomously adapt to dynamic and variant money laundering behaviors without human intervention.

The main contributions of this study are summarized as follows:

- We propose LEAP-HGN, which incorporates the Amygdala-Parietal dual-path mechanism to address signal dilution and evidence submersion. To our knowledge, this is the first work to apply neuromorphic intelligent perception mechanisms to anti-money laundering detection. By synergizing Amygdala-driven gain modulation for rapid risk amplification with Parietal-guided residual attention correction for global spatial reasoning, we effectively integrate local risk intuition with global structural perception to enhance the discrimination of concealed syndicates.
- Additionally, we design a novel Markov structural diffusion process deployed across node-hyperedge-node paths to capture high-order structural information, successfully bridging time-slicing causal breaks to explicitly reconstruct global long-range fund dependencies.
- We construct a spatio-temporal multi-view hypergraph to overcome trajectory fragmentation in high-velocity transactions and explicitly reconstruct fragmented temporal linkages into high-order hyperedges with clear causal dependencies.
- We establish a self-evolutionary rule closed-loop via hard sample mining to mitigate the rigidity of static controls against dynamic laundering tactics. Leveraging the semantic reasoning of LLM to diagnose structural defects in hard samples, we synthesize corrective symbolic rules for retro-injection into the ARRW module, thereby enabling autonomous neuro-symbolic evolution and continuous adaptation.

2 Related Work

Graph-Based Anti-Money Laundering Detection. While recent Graph Neural Networks (GNNs) have significantly advanced in capturing complex transaction topologies, addressing label imbalance, and modeling high-order dependencies (Egressy et al., 2024; Li et al., 2025a; Song et al., 2025a;c; Duan et al., 2024; Zhang et al., 2025; Zheng et al., 2025; Wei et al., 2025), they face two critical limitations. First, their predominant reliance on discrete time-window partitioning inevitably severs intrinsic temporal dependencies, fragmenting continuous fund trajectories. Second, conventional neighborhood aggregation suffers from severe signal dilution, where sparse micro-cyclic laundering signals are overwhelmed by massive benign background noise during deep propagation.

LLM-Driven Logic Evolution. To address non-stationary laundering strategies, recent studies have explored structural shifts and zero-shot perception using Large Language Models (LLMs) (Gao et al., 2024; He et al., 2025; Niu et al., 2025). Nevertheless, most current models still depend on static discriminative logic derived from historical priors. When confronted with rapid strategic evolution, these fixed decision boundaries fail to generalize, resulting in a cognitive lag where the model's judgment cannot adapt to novel laundering variants. For a more comprehensive and detailed review of related literature, please refer to Appendix A.

3 Methodology

Figure 1 presents the proposed LEAP-HGN framework.

3.1 Spatio-Temporal Multi-view Hypergraph Construction (STMH)

For covert money laundering patterns characterized by time-sensitivity and dense features, we partition the chronologically ordered transaction stream into local windows of duration ΔT. Within each window, we construct $G_{hyper} = \{\mathcal{H}_m\}_{m\in\mathcal{M}}$, where $\mathcal{M} = \{AB, BP, TA, LS\}$ and each view

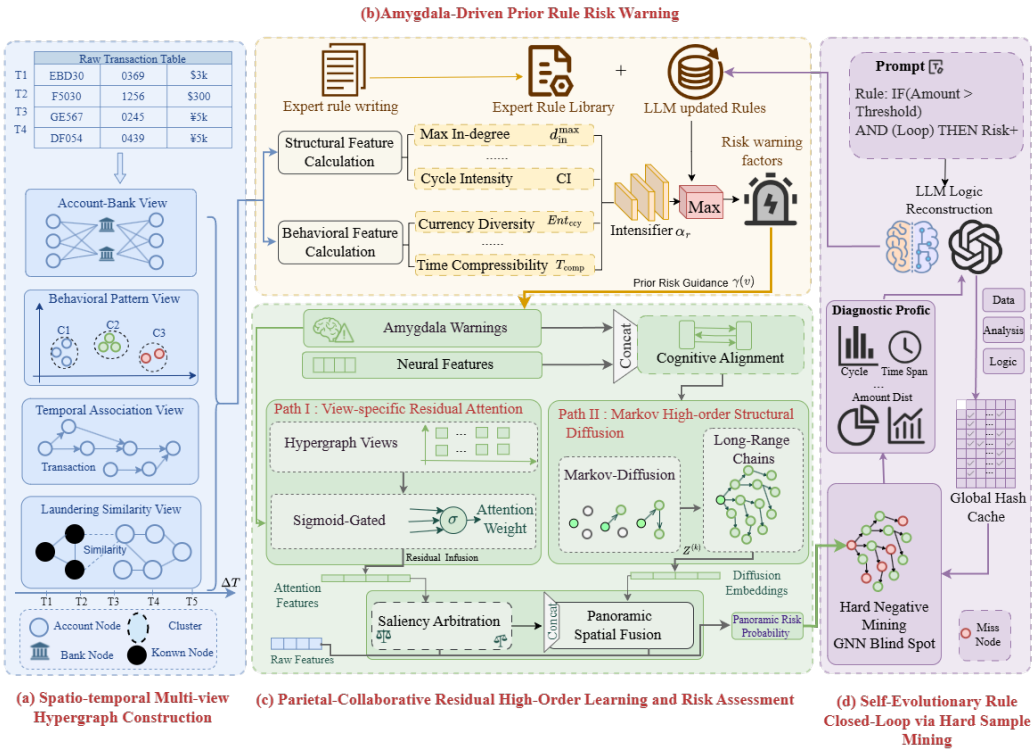


Figure 1: The overall framework of LEAP-HGN.

$\mathcal{H}_m = (V, \mathcal{E}_m)$ shares the account-node set V but has its own hyperedge family $\mathcal{E}_m$. The four views preserve short-term transaction coherence while encoding complementary behavioral and structural relations.

Account-Bank path (AB). This view identifies hidden cross-bank connections and spatial aggregation features. The behavioral intensity score s_i of account v_i within a specific time window is calculated as follows:

$$s_i = \mu_1 f_i + \mu_2 Zscore_{v_i} + \mu_3 Burst_{v_i}, \tag{1}$$

where f_i, $Zscore_{v_i}$, and $Burst_{v_i}$ are the normalized transaction frequency, standardized amount score, and temporal burstiness of account v_i, respectively; $\mu_1, \mu_2, \mu_3 \geq 0$ are component weights satisfying $\sum_{q=1}^{3} \mu_q = 1$. For each bank context b, the node set $e_b^{AB} = \{v_i \in V \mid b(v_i) = b,\ s_i \geq \tau_{AB}\}$ forms an AB hyperedge and $\mathcal{E}_{AB}$ collects all such non-singleton sets; $b(v_i)$ is the bank of v_i and τ_{AB} is the intensity threshold (Appendix B.1).

Behavioral Pattern (BP). This view isolates collaborative behaviors such as structured splitting in high-dimensional space. Density clustering maps the d_x-dimensional feature vector $x_k \in \mathbb{R}^{d_x}$ of account v_k to a cluster C_j; each cluster induces the node-set hyperedge $e_j^{BP} = \{v_k \in V \mid v_k \in C_j\}$, and $\mathcal{E}_{BP}$ collects all non-singleton e_j^{BP}. Thus, x_k determines membership whereas accounts, rather than feature vectors, are the hypergraph vertices (Appendix B.2).

Temporal Association (TA). This view captures time-sensitive and amount-correlated multi-hop chains. Let $t_i = (A, B, a_i, T_i)$ and $t_j = (B, C, a_j, T_j)$ be consecutive transfers sharing intermediary account B, with amounts a_i, a_j and timestamps T_i, T_j. The indicator $I_{cascade}(t_i, t_j) = 1$ when $0 < T_j - T_i \leq \Delta t$ and their relative amount difference does not exceed τ_{TA}, where Δt and τ_{TA} are the time and amount tolerances. Each qualifying pair induces $e_{ij}^{TA} = \{A, B, C\}$ (Appendix B.3).

Laundering Similarity (LS). This view groups candidates similar to laundering templates derived exclusively from the chronological training period. Here, $q(t) \in \mathbb{R}^{d_q}$ is the normalized d_q-dimensional representation of transaction t, $\mathcal{U}^{(s)}$ is the candidate set in split s, $\mathcal{T}_{train}^{+}$ contains positive training

templates, and s_{kj} is the cosine similarity between $t_k \in \mathcal{U}^{(s)}$ and $t_j^+ \in \mathcal{T}_{train}^+$. With τ_{sim} fixed on training/validation data, each template induces $e_j^{LS} = \{v(t_k) \in V \mid s_{kj} \geq \tau_{sim}\}$; no test label or future positive is queried (Appendix B.4).

3.2 Amygdala-Driven Prior Rule Risk Warning (ARRW)

This module simulates the rapid response mechanism of the amygdala. It does not directly perform rule discrimination but converts expert experience into explicit risk warning factors. These serve as gain control signals for the neural network, dynamically amplifying the risk features of dense micro-cycles during deep propagation.

Structural Risk: For the directed transaction subgraph induced by hyperedge e, we extract the maximum in-degree d_{in}^{max}, maximum out-degree d_{out}^{max}, cycle intensity CI, and degree-based Gini imbalance $Gini$ (Appendix C.1). These features describe aggregation, dispersion, fund reflux, and flow asymmetry. Their structural risk score is

$$S_{struct}(e) = \Phi_{logic}(d_{in}^{max}, d_{out}^{max}, CI, Gini; \Theta_{struct}), \tag{2}$$

where Φ_{logic} is a bounded nonlinear symbolic mapping and Θ_{struct} contains its thresholds, slopes, and operator parameters. The parameters of every active rule remain fixed within each neural training cycle, and newly validated corrective rules are appended only at asynchronous SRCL boundaries.

Behavioral Risk: This captures short-term high-frequency and collaborative behavior using normalized amount-template similarity S_{amt}^e, currency diversity Ent_{ccy}^e, format inconsistency I_{format}^e, and time compression T_{comp}^e (Appendix C.2):

$$S_{beh}(e) = w_b^\top [S_{amt}^e, Ent_{ccy}^e, I_{format}^e, T_{comp}^e]. \tag{3}$$

where, $w_b \in \mathbb{R}_{\geq 0}^4$ is the behavioral weight vector with $\mathbf{1}^\top w_b = 1$; the four indicators are scaled to comparable ranges before fusion.

Risk Fusion: For each active rule $r \in \mathcal{R}^{(c)}$, structural and behavioral evidence is fused into $M_r(e)$, and the strongest rule match defines the effective hyperedge risk $M^*(e)$:

$$M_r(e) = \alpha_r S_{struct,r}(e) + (1 - \alpha_r) S_{beh,r}(e), \qquad M^*(e) = \max_{r \in \mathcal{R}^{(c)}} M_r(e), \tag{4}$$

$$\gamma_A(v) = \max_{e \ni v} M^*(e), \tag{5}$$

where, $\alpha_r \in [0, 1]$ balances both risks; the rule-wise maximum is followed by a maximum over all hyperedges incident to v (Appendix F).

3.3 Parietal-Collaborative Residual High-Order Learning and Risk Assessment (PCRR)

As the high-order spatial cognition center, the Parietal module receives the Amygdala warning signal before deep feature processing. Its input is $h^{(0)}(v) = \text{Concat}(X_{raw}(v), X_{rule}(v))$, where X_{raw} contains observed account/transaction attributes and X_{rule} encodes rule-derived features including $\gamma_A(v)$ (Appendix D.1).

Circuit 1: View-specific Residual Attention. This circuit addresses signal dilution through risk-aware gain modulation:

$$H_{guided}^{(l)}(v) = h_{attn}(v) \odot (1 + \lambda_{alert} \gamma_A(v)), \tag{6}$$

$$h^{(l)}(v) = \sigma(PReLU(LN(W_{res} h^{(l-1)}(v) + W_{trans} H_{guided}^{(l)}(v)))), \tag{7}$$

where, l indexes GNN layers; $h_{attn}(v)$ is the attention-aggregated context; $\gamma_A(v)$ is channel-broadcast; and $\lambda_{alert} \geq 0$ controls gain. W_{res} and W_{trans} align the residual and guided paths, while LN, $PReLU$, and σ denote LayerNorm, parametric ReLU, and sigmoid gating (Appendix D.2).

Circuit 2: Markov High-Order Structural Diffusion. For $N = |V|$ accounts, let $H \in \{0, 1\}^{N \times E}$ concatenate the four view-incidence matrices, where $E = \sum_m |\mathcal{E}_m|$. With $\mathbf{1} \in \mathbb{R}^N$ denoting the

all-ones vector, we form $A = HH^\top$, $D_A = \mathrm{diag}(A\mathbf{1})$, and the row-stochastic transition $P = D_A^{-1}A$ for Random Walk with Restart over node–hyperedge–node paths:

$$Z^{(0)} = h^{(0)}, \tag{8}$$

$$Z^{(k)} = (1-\beta)PZ^{(k-1)} + \beta Z^{(0)}, \tag{9}$$

where $Z^{(k)} \in \mathbb{R}^{N\times d_h}$ is the step-k representation and $\beta \in (0,1)$ the restart probability; zero-degree handling, multi-view assembly, and probability conservation are detailed in Appendix D.3.

Parietal Panoramic Decision and Fusion. For each node and view, a view-specific projection produces $s_m(v)$ before normalization:

$$s_m(v) = W_{attn}^{(m)}[h_{global}^{(m)} \parallel \gamma_A(v)], \quad \omega_m(v) = \frac{\exp(s_m(v))}{\sum_{j=1}^{M}\exp(s_j(v))}, \tag{10}$$

where $s_m(v)$ and $\omega_m(v)$ are the saliency logit and normalized view weight; $h_{global}^{(m)}$ is the pooled context, $W_{attn}^{(m)}$ its projection, and $M = 4$. View-specific projections retain node-risk dependence (Appendix D.4):

$$h_{final} = g \odot h_{parietal} + (1-g) \odot \frac{1}{M}\sum_{m=1}^{M} h_{feedback}^{(m)}. \tag{11}$$

where, $g(v) \in (0,1)^{d_h}$ is a learned element-wise gate, $h_{parietal}(v)$ is the saliency-weighted multi-view consensus, and $h_{feedback}^{(m)}(v)$ is the refined feature of view m. Finally, $H_{combined}(v) = h_{final}(v) \parallel Z^{(k)}(v) \parallel X_{raw}(v)$ is classified by $\hat{y}_v = \sigma(w^\top H_{combined}(v) + b)$, where w and b are the classifier parameters and $\hat{y}_v$ is the predicted laundering probability. The detailed assembly and Algorithm 1 are in Appendices D.4 and E.

3.4 Self-Evolutionary Rule Closed-Loop via Hard Sample Mining (SRCL)

To address the rigidity and lag of discrimination logic caused by the dynamic evolution of money laundering methods, this module constructs a Diagnosis-Reconstruction-Feedback self-evolutionary closed-loop. The system automatically identifies decision boundary defects (hard samples) and utilizes an LLM to translate these error features into symbolic rules. By injecting this new knowledge back into the Amygdala, it achieves a synergistic evolution of the neural and symbolic systems (details in Appendix F).

Hard Sample Mining & Error Profile Construction. At the end of each evolutionary cycle, full reasoning is performed on the training set to identify hard negative samples that possess laundering features but are incorrectly suppressed by the neural network. Given a miss-report threshold τ_{miss}, the hard sample set V_{hard} is defined as follows:

$$V_{hard} = \{v \in V \mid y_v = 1 \wedge \hat{y}_v < \tau_{miss}\}. \tag{12}$$

where for V_{hard}, the system backtracks all hyperedges in which they participate, calculates key topological indicators $f(e)$, and constructs an error diagnostic vector z_{profile}:

$$z_{\text{profile}} = \text{Agg}\{f(e) \mid e \in \mathcal{N}(V_{hard})\}, \tag{13}$$

where $y_v \in \{0,1\}$ is the ground-truth label, $\hat{y}_v \in [0,1]$ is the predicted probability, and τ_{miss} is the miss-report threshold. $\mathcal{N}(V_{hard}) = \{e \mid e \cap V_{hard} \neq \emptyset\}$ is the incident-hyperedge set, $f(e)$ is its structural-behavioral feature vector, and Agg is component-wise aggregation (mean in our implementation).

LLM Logic Reconstruction & Evolutionary Closed-Loop. Guided by Table 9 in Appendix J, the LLM converts z_{profile} into a smooth, bounded symbolic rule. The accepted symbolic parameters remain fixed during the subsequent neural training cycle. Newly validated corrective rules are appended to the active rule set only at evolutionary-cycle boundaries; a same-name candidate updates its named entry without removing unrelated rules:

$$S_{new}(e) = \Phi_{logic}(f_e; \Theta_{new}) = \sigma(k \cdot (f_e - \tau_{new})), \tag{14}$$

where f_e is the selected hyperedge feature, τ_{new} its synthesized threshold, $k > 0$ the sigmoid slope, and $\Theta_{new} = \{k, \tau_{new}\}$ the accepted symbolic parameters. The bounded score $S_{new}(e) \in (0,1)$ modulates the Amygdala warning signal; AST-evaluated Θ_{new} remains outside the automatic-differentiation graph, so alternating symbolic and neural updates form the closed loop.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate LEAP-HGN on six diverse datasets: synthetic banking simulations (**HI-Series** Small/Medium/Large, **PaySim**) for testing topological robustness, and real-world distributions (**XploreML**, **IEEE-CIS**) covering bank transfers and e-commerce fraud. Detailed dataset statistics are in Appendix G.

Baselines. To demonstrate the effectiveness of our framework, we compare it against 12 competitive SOTA methods, categorized into three groups: **Specialized anti-money laundering**, including FlowScope Li et al. (2020), SGNN Song et al. (2025b), MonLAD Sun et al. (2022), and GARG-AML Deprez et al. (2025); **Graph-based Fraud Detection**, including HUGE Pan et al. (2025), RGTAN Xiang et al. (2025), HOGRL Zou & Cheng (2024), TROPICAL Haghighi et al. (2024), and SpaceGNN Dong et al. (2025); and **recent LLM-based models**, including ARC Liu et al. (2024), EdgePrompt Fu et al. (2025), and Fraud-R1 Yang et al. (2025). (Appendix I)

Evaluation Metrics and Implementation Details. We evaluate all methods using AUC, F1, and illicit-class Recall under a strict chronological Out-of-Time (OOT) protocol. The framework is optimized via Adam with Focal Loss ($\gamma_F = 2.0, \alpha = 0.85$) on an NVIDIA RTX 5090 GPU (32GB VRAM). Metric definitions, hyperparameters, LLM settings, and dataset-specific OOT boundaries are detailed in Appendix H.

4.2 OVERALL EXPERIMENTAL RESULTS

Table 1 compares LEAP-HGN against 12 baselines across six datasets. LEAP-HGN achieves the highest AUC, F1, and Recall on four of the six datasets. On IEEE-CIS, MonLAD has slightly higher F1 and Recall, whereas LEAP-HGN has the highest AUC; on HI-Small, HOGRL has the highest AUC, whereas LEAP-HGN leads in F1 and Recall.

Compared to specialized AML models. On HI-Small, LEAP-HGN attains 92.89% F1, exceeding MonLAD (79.00%) by 13.89 percentage points. On HI-Medium, its AUC reaches 97.95%, compared with 92.39% for SGNN. These results indicate that LEAP-HGN more effectively preserves discriminative evidence in temporally evolving transaction graphs.

Compared to advanced graph/hypergraph models. On PaySim, LEAP-HGN obtains 96.71% F1, compared with 89.10% for TROPICAL. On XploreML, its AUC reaches 88.45%, exceeding HOGRL (67.57%) by 20.88 percentage points. These gains support the effectiveness of risk-signal amplification and multi-view high-order aggregation in noisy transaction graphs.

Compared to recent LLM-based paradigms. LEAP-HGN obtains 83.72% F1 on IEEE-CIS, compared with 75.10% for ARC. On PaySim, it improves F1 over Fraud-R1 from 90.10% to 96.71%, an absolute gain of 6.61 percentage points. The results show that the Self-evolutionary Rule Closed-Loop remains effective under chronological distribution shifts.

4.3 ABLATION STUDY

To quantify the contribution of each module, we compare the full LEAP-HGN against four variants on the XploreML and HI-Small datasets: (a) **w/o LLM Feedback** (removes iterative optimization), (b) **w/o Amygdala** (excludes low-pathway gain modulation), (c) **w/o Parietal** (removes global diffusion integration), and (d) **w/o Multi-View** (replaces the hypergraph with a homogeneous view).

As shown in Figure 2, the full model achieves the highest F1 on both XploreML and HI-Small. Removing LLM feedback causes the largest degradation on XploreML, supporting the role of iterative rule evolution in adapting the detector to hard samples. Removing the multi-view construction produces the largest F1 decline on HI-Small, supporting the role of the spatio-temporal hypergraph in preserving fragmented transaction evidence, while the remaining variants confirm the complementary contributions of the Amygdala and Parietal pathways.

Table 1: Detection performance (%) on six datasets. Rec. denotes illicit-class recall; bold values indicate the best result for each metric.

Types	Models	PaySim (PS)			XploreML			IEEE-CIS		
		AUC	F1	Rec.	AUC	F1	Rec.	AUC	F1	Rec.
Specialized AML	SGNN	92.74	86.08	93.33	86.35	80.90	77.26	70.69	54.76	69.30
	MonLAD	92.50	87.00	91.00	81.12	73.00	76.00	90.41	**84.00**	**88.00**
	GARG-AML	77.70	1.16	42.65	76.29	53.52	54.36	71.90	38.83	33.61
	FlowScope	89.16	70.24	72.83	84.18	64.44	69.96	85.07	63.89	70.03
Graph-based Fraud Detection	SpaceGNN	95.93	89.97	89.20	70.77	62.75	62.20	77.61	69.34	68.94
	RGTAN	94.12	75.32	79.81	79.81	46.15	30.00	78.56	51.88	56.55
	HOGRL	96.56	84.74	80.60	67.57	53.83	59.31	72.35	66.35	66.03
	TROPICAL	96.13	89.10	91.77	87.57	79.40	63.79	77.15	42.60	80.20
	HUGE	96.63	18.40	35.52	66.24	32.77	40.37	86.22	44.57	41.06
LLM-based	EdgePrompt	79.16	71.49	71.44	86.00	74.17	75.64	80.98	68.57	68.38
	Fraud-R1	96.99	90.10	89.23	87.15	79.92	78.45	87.56	79.74	78.54
	ARC	94.09	85.86	85.29	87.75	76.39	74.65	84.43	75.10	71.67
Proposed	**LEAP-HGN**	**99.55**	**96.71**	**97.30**	**88.45**	**81.36**	**79.09**	**90.94**	83.72	82.87
Types	Models	HI-Small			HI-Medium			HI-Large		
		AUC	F1	Rec.	AUC	F1	Rec.	AUC	F1	Rec.
Specialized AML	SGNN	77.70	71.35	93.98	92.39	89.09	94.78	93.21	87.39	89.47
	MonLAD	80.49	79.00	65.00	81.56	71.00	62.00	85.38	87.00	77.00
	GARG-AML	77.79	6.24	55.15	76.00	54.68	45.37	74.10	45.08	49.53
	FlowScope	84.97	64.33	74.11	84.88	81.98	75.07	85.41	64.40	73.36
Graph-based Fraud Detection	SpaceGNN	83.67	78.75	79.06	89.19	81.97	81.98	87.50	73.91	74.00
	RGTAN	81.61	70.16	80.69	88.85	74.63	69.35	86.88	68.91	30.34
	HOGRL	**94.57**	87.06	87.79	94.83	83.06	81.63	84.32	77.55	81.24
	TROPICAL	82.41	61.69	71.28	80.00	73.85	94.45	74.65	61.30	53.82
	HUGE	89.29	27.49	24.34	83.62	22.23	20.92	83.84	22.91	21.34
LLM-based	EdgePrompt	82.32	69.31	69.26	74.21	68.98	68.72	74.06	66.43	67.00
	Fraud-R1	91.04	87.63	87.25	93.85	88.62	91.94	93.64	88.73	92.00
	ARC	88.78	81.72	88.70	94.80	67.78	92.67	94.08	68.50	91.98
Proposed	**LEAP-HGN**	91.16	**92.89**	**99.36**	**97.95**	**94.74**	**96.53**	**94.98**	**90.03**	**92.02**

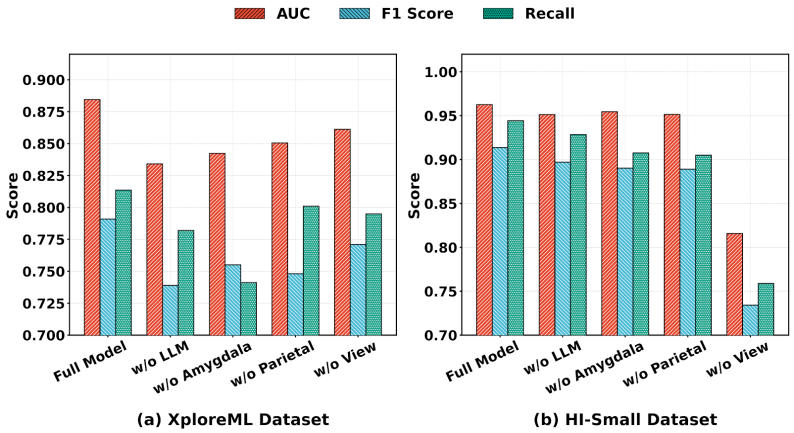


Figure 2: Comparative ablation results on XploreML and HI-Small.

4.4 Parameter Sensitivity Analysis

Numerical sensitivity analysis on key hyperparameters identifies the optimal configuration for LEAP-HGN as illustrated in Figure 3.

Concerning the learning rate η and regularization coefficient λ_{reg}, the model achieves its best observed F1 at $\eta = 0.001$ and $\lambda_{reg} = 0.1$. Smaller learning rates hinder convergence, while regularization values above 0.5 suppress feature expression and reduce performance through underfitting.

Regarding the hidden dimension d and focal loss parameter γ_F, representational capacity improves as d increases to 64. Expansion to 128 yields negligible gains while increasing computational overhead, marking $d = 64$ as the saturation point for feature extraction. For class imbalance, focal loss is most effective at γ_F between 1.0 and 2.0. Excessive γ_F values such as 5.0 make the model overly sensitive to outliers and label noise, thereby reducing F1. Additionally, the alert mechanism weight λ_{alert} demonstrates high stability across varied settings.

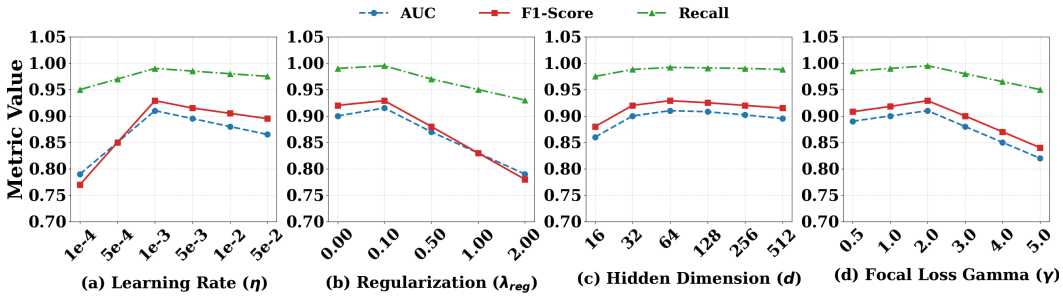


Figure 3: Parameter sensitivity analysis.

4.5 Iterative Optimization Analysis for Neuro-Symbolic Closed-Loop

To validate the dynamic adaptability of LEAP-HGN, we track its performance across iterative optimization cycles. The results show that LLM-generated prior rules enable the Amygdala pathway to amplify sparse risk signals, while the Parietal pathway integrates local warnings with global structural dependencies. As optimization proceeds, model performance improves steadily and approaches a favorable Pareto frontier, reaching its best overall state around Cycle 7. The slight performance retraction afterwards suggests that the proposed closed-loop can effectively balance continuous adaptation and structural generalization. Detailed evolutionary dynamics and holographic capability analysis are provided in Appendix M.

4.6 Dynamic Evolution Analysis of Decision Boundaries

To further elucidate how LLMs bridge the semantic gap, we conduct a comparative convergence analysis of risk decision boundaries across different LLM backbones. Stronger semantic reasoning generally accelerates the establishment of robust decision boundaries and improves the identification of implicit and camouflaged laundering patterns. Moreover, the F1 and Recall trajectories reveal distinct adaptation behaviors across LLMs, with some converging rapidly in early stages while others respond more strongly in later cycles. These observations show that the framework progressively reconciles semantic risk signals with structural evidence during iterative optimization. Further trajectory analyses and discussion are provided in Appendix N.

5 Conclusion

In this paper, we propose LEAP-HGN, a novel brain-inspired Neuro-Symbolic framework for Anti-Money Laundering integrating a Spatio-Temporal Multi-view Hypergraph with an Amygdala-Parietal dual-pathway. The Amygdala low pathway employs neural gain modulation to pre-amplify sparse risk signals, while the Parietal high pathway utilizes Markov high-order diffusion to reconstruct global long-range laundering chains. Furthermore, a Self-Evolutionary Rule Closed-Loop employs an LLM to translate boundary defects into smooth symbolic rules that act as periodically refreshed neural gain priors. Extensive experiments on six financial datasets demonstrate that LEAP-HGN significantly outperforms 12 SOTA baselines and exhibits superior adaptability to dynamic laundering patterns. LEAP-HGN currently relies on LLMs to parse regulatory knowledge, and its performance may still be affected by the capability of the underlying models. Future work will further investigate the robustness, generalizability, and practical applicability of the proposed framework in more complex and realistic scenarios.

AI USE STATEMENT

In this work, generative AI tools were used in two distinct roles. First, multiple large language models were employed as semantic reasoning backbones within the proposed Self-Evolutionary Rule Closed-Loop (SRCL). At each evolutionary stage, an LLM receives an aggregated error profile constructed from hard false-negative samples and generates candidate symbolic rules under predefined structural and output constraints. The generated rules are not directly adopted as unconstrained model outputs; they are subjected to schema checking, sandbox-based validation, and rule screening before being incorporated into the subsequent neural optimization cycle. The detailed self-evolutionary procedure is described in Appendix F, while the prompt template, output schema, and rule-generation constraints are provided in Appendix J. We additionally evaluate different LLM backbones to examine the robustness of the rule-evolution mechanism, with the corresponding analysis reported in Appendix N.

Second, generative AI tools, including Gemini 3 Pro and DeepSeek-4-Pro, were employed for grammar checking, language polishing, and improving presentation clarity. These tools were not used for methodology design, code implementation, data processing, experiment execution, or the analysis and interpretation of experimental results. All AI-assisted text and literature summaries were reviewed and verified by the authors. We take full responsibility for the final methodology, experimental results, claims, and manuscript.

ETHICS STATEMENT

This work studies anti-money laundering as a defensive financial-security application. Our experiments are conducted on existing synthetic or anonymized financial-transaction datasets, and we do not collect new personally identifiable financial information or conduct experiments involving human participants. Detailed descriptions and characteristics of the datasets used in this study are provided in Appendix G.

As with other automated financial-risk detection systems, LEAP-HGN may produce false-positive or false-negative predictions and may be affected by biases, incomplete observations, or temporal distribution shifts in the underlying transaction data. Consequently, the proposed framework is intended to support risk analysis rather than serve as the sole basis for account restrictions, punitive actions, or other consequential financial decisions. Practical deployment should incorporate appropriate human review, domain expertise, institutional policies, and applicable regulatory requirements. The proposed methods are intended for defensive detection and research purposes and should not be used to facilitate illicit financial activities. Further discussion of data privacy, potential risks, and responsible use is provided in Appendix L.5.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed descriptions of the model, training procedure, datasets, evaluation protocol, and LLM integration throughout the main text and appendices. The construction of the spatio-temporal multi-view hypergraph and the definitions of the risk indicators are detailed in Appendices B and C. The Parietal-collaborative high-order learning mechanism and the complete algorithmic procedure are described in Appendices D and E, respectively. The implementation of the Self-Evolutionary Rule Closed-Loop, including hard-sample profiling, symbolic rule generation, sandbox validation, and alternating symbolic–neural updating, is provided in Appendix F.

Dataset statistics, feature specifications, and dataset-specific characteristics are reported in Appendix G. The experimental hyperparameters and optimization settings are provided in Appendix H.2, the LLM configuration and training-efficiency settings in Appendix H.3, and the dataset-specific chronological Out-of-Time (OOT) partitions in Appendix H.4. The complete evidence-driven prompt template, input variables, output schema, and symbolic rule-generation constraints are provided in Appendix J.

For additional reproducibility information, including model configuration, evaluation protocols, LLM integration, and responsible-use considerations, we provide a consolidated description in Appendix L. An anonymized implementation is available as described in Appendix L.1. Additional analyses of the iterative neuro-symbolic optimization process and the behavior of different LLM backbones are reported in Appendices M and N.

REFERENCES

Nourhan Ahmed, Vijaya Krishna Yalavarthi, and Lars Schmidt-Thieme. Motif-aware graph neural networks for networked time series imputation. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 11409–11417. AAAI Press, 2025.

Erik R. Altman, Jovan Blanusa, Luc von Niederhäusern, Beni Egressy, Andreea Anghel, and Kubilay Atasu. Realistic synthetic financial transactions for anti-money laundering models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Ziwei Chai, Yang Yang, Jiawang Dan, Sheng Tian, Changhua Meng, Weiqiang Wang, and Yifei Sun. Towards learning to discover money laundering sub-network in massive transaction network. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 14153–14160. AAAI Press, 2023a. doi: 10.1609/AAAI.V37I12.26656.

Ziwei Chai, Yang Yang, Jiawang Dan, Sheng Tian, Changhua Meng, Weiqiang Wang, and Yifei Sun. Towards learning to discover money laundering sub-network in massive transaction network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pp. 14153–14160. AAAI Press, 2023b.

Dawei Cheng, Yujia Ye, Sheng Xiang, Zhenwei Ma, Ying Zhang, and Changjun Jiang. Anti-money laundering by group-aware deep graph learning. *IEEE Trans. Knowl. Data Eng.*, 35(12):12444–12457, 2023.

Paola de Perthuis and David Pointcheval. Two-client inner-product functional encryption with an application to money-laundering detection. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi (eds.), *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pp. 725–737. ACM, 2022.

Bruno Deprez, Bart Baesens, Tim Verdonck, and Wouter Verbeke. GARG-AML against smurfing: A scalable and interpretable graph-based framework for anti-money laundering. *CoRR*, abs/2506.04292, 2025.

Xiangyu Dong, Xingyi Zhang, Lei Chen, Mingxuan Yuan, and Sibo Wang. Spacegnn: Multi-space graph neural network for node anomaly detection with extremely limited labels. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Mingjiang Duan, Tongya Zheng, Yang Gao, Gang Wang, Zunlei Feng, and Xinyu Wang. DGA-GNN: Dynamic grouping aggregation GNN for fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11820–11828, 2024. doi: 10.1609/aaai.v38i10.29067.

Béni Egressy, Luc von Niederhäusern, Jovan Blanuša, Erik R. Altman, Roger Wattenhofer, and Kubilay Atasu. Provably powerful graph neural networks for directed multigraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11838–11846, 2024. doi: 10.1609/aaai.v38i10.29069.

Qishuang Fu. Towards explainable and effective anti-money laundering for cryptocurrency. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier (eds.), *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS 2025, Taipei, Taiwan, October 13-17, 2025*, pp. 4875–4877. ACM, 2025. doi: 10.1145/3719027.3765574.

Xingbo Fu, Yinhan He, and Jundong Li. Edge prompt tuning for graph neural networks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Yuan Gao, Jinghan Li, Xiang Wang, Xiangnan He, Huamin Feng, and Yongdong Zhang. Revisiting attack-caused structural distribution shift in graph anomaly detection. *IEEE Trans. Knowl. Data Eng.*, 36(9):4849–4861, 2024. doi: 10.1109/TKDE.2024.3380709.

Pragati Garg and Vanshika Gupta. Governance of shell corporations and their role in enabling white-collar crimes. *International Journal of Law*, 11(9):32–38, 2025.

Venus Haghighi, Behnaz Soltani, Nasrin Shabani, Jia Wu, Yang Zhang, Lina Yao, Quan Z. Sheng, and Jian Yang. TROPICAL: transformer-based hypergraph learning for camouflaged fraudster detection. In Elena Baralis, Kun Zhang, Ernesto Damiani, Mérouane Debbah, Panos Kalnis, and Xindong Wu (eds.), *IEEE International Conference on Data Mining, ICDM 2024, Abu Dhabi, United Arab Emirates, December 9-12, 2024*, pp. 121–130. IEEE, 2024. doi: 10.1109/ICDM59182.2024.00019.

Dongmei Han, Min Min, Yuchen Wang, Guoming Xu, and Xiaofeng Zhou. Beyond single transactions: D-EMAML - dual-edge motif neural networks for enhanced anti-money laundering detection. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 14792–14801. AAAI Press, 2026.

Xin He, Yili Wang, Wenqi Fan, Xu Shen, Xin Juan, Rui Miao, and Xin Wang. Mamba-based graph convolutional networks: Tackling over-smoothing with selective state space. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 5345–5353. ijcai.org, 2025.

Pengbo Li, Hang Yu, and Xiangfeng Luo. Context-aware graph neural network for graph-based fraud detection with extremely limited labels. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 12112–12120. AAAI Press, 2025a. doi: 10.1609/AAAI.V39I11.33319.

Xiangfeng Li, Shenghua Liu, Zifeng Li, Xiaotian Han, Chuan Shi, Bryan Hooi, He Huang, and Xueqi Cheng. Flowscope: Spotting money laundering based on graphs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4731–4738. AAAI Press, 2020.

Xujia Li, Yuan Li, Xueying Mo, Hebing Xiao, Yanyan Shen, and Lei Chen. Diga: Guided diffusion model for graph recovery in anti-money laundering. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (eds.), *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 4404–4413. ACM, 2023.

Zhong Li, Xueting Yang, and Changjun Jiang. Multi-view graph-based hierarchical representation learning for money laundering group detection. *IEEE Trans. Inf. Forensics Secur.*, 20:2035–2050, 2025b.

Dan Lin, Jiajing Wu, Yunmei Yu, Qishuang Fu, Zibin Zheng, and Changlin Yang. Denseflow: Spotting cryptocurrency money laundering in ethereum transaction graphs. In Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (eds.), *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pp. 4429–4438. ACM, 2024. doi: 10.1145/3589334.3645692.

Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (eds.), *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 3168–3177. ACM / IW3C2, 2021.

Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan. ARC: A generalist graph anomaly detector with in-context learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in*

Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024.

Xuejiao Luo, Xiaohui Han, Wenbo Zuo, Xiaoming Wu, and Wenyin Liu. Mlad^2: A semi-supervised money laundering detection framework based on decoupling training. *IEEE Trans. Inf. Forensics Secur.*, 19:4518–4533, 2024. doi: 10.1109/TIFS.2024.3380262.

Chaoxi Niu, Hezhe Qiao, Changlu Chen, Ling Chen, and Guansong Pang. Zero-shot generalist graph anomaly detection with unified neighborhood prompts. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 3226–3234. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/359.

Berkan Oztas, Deniz Cetinkaya, Festus Fatai Adedoyin, Marcin Budka, Huseyin Dogan, and Gökhan Aksu. Enhancing anti-money laundering: Development of a synthetic transaction monitoring dataset. In *IEEE International Conference on e-Business Engineering, ICEBE 2023, Sydney, Australia, November 4-6, 2023*, pp. 47–54. IEEE, 2023.

Junjun Pan, Yixin Liu, Xin Zheng, Yizhen Zheng, Alan Wee-Chung Liew, Fuyi Li, and Shirui Pan. A label-free heterophily-guided approach for unsupervised graph fraud detection. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 12443–12451. AAAI Press, 2025. doi: 10.1609/AAAI.V39I12.33356.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5363–5370. AAAI Press, 2020.

Sangwoo Seo, Sungwon Kim, Jihyeong Jung, Yoonho Lee, and Chanyoung Park. Self-explainable temporal graph networks based on graph information bottleneck. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 2572–2583. ACM, 2024.

Shiva Shadrooh and Kjetil Nørvåg. Smotef: Smurf money laundering detection using temporal order and flow analysis. *Appl. Intell.*, 54(15-16):7461–7478, 2024.

Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (eds.), *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 1486–1494. ACM, 2022.

Chuancheng Song, Xixun Lin, Hanyang Shen, Yanmin Shang, and Yanan Cao. Uniform: Towards unified framework for anomaly detection on graphs. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 12559–12567. AAAI Press, 2025a. doi: 10.1609/AAAI.V39I12.33369.

Jie Song, Sijia Zhang, Pengyi Zhang, Junghoon Park, Yu Gu, and Ge Yu. Illicit social accounts? anti-money laundering for transactional blockchains. *IEEE Trans. Inf. Forensics Secur.*, 20:391–404, 2025b. doi: 10.1109/TIFS.2024.3518068.

Yudan Song, Yuecen Wei, Yuhang Lu, Qingyun Sun, Minglai Shao, Li-E Wang, Chunming Hu, Xianxian Li, and Xingcheng Fu. Mitigating message imbalance in fraud detection with dual-view graph representation learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 3281–3289. ijcai.org, 2025c. doi: 10.24963/IJCAI.2025/365.

Xiaobing Sun, Wenjie Feng, Shenghua Liu, Yuyang Xie, Siddharth Bhatia, Bryan Hooi, Wenhan Wang, and Xueqi Cheng. Monlad: Money laundering agents detection in transaction streams. In K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (eds.), *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pp. 976–986. ACM, 2022. doi: 10.1145/3488560.3498418.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21076–21089. PMLR, 2022.

Zhihua Tian, Yuan Ding, Xiang Yu, Enchao Gong, Jian Liu, and Kui Ren. Towards collaborative anti-money laundering among financial institutions. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 4722–4733. ACM, 2025a.

Zhihua Tian, Yuan Ding, Xiang Yu, Enchao Gong, Jian Liu, and Kui Ren. Towards collaborative anti-money laundering among financial institutions. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 4722–4733. ACM, 2025b. doi: 10.1145/3696410.3714576.

Chunyu Wei, Wenji Hu, Xingjia Hao, Yunhai Wang, Yueguo Chen, Bing Bai, and Fei Wang. Graph evidential learning for anomaly detection. In Luiza Antonie, Jian Pei, Xiaohui Yu, Flavio Chierichetti, Hady W. Lauw, Yizhou Sun, and Srinivasan Parthasarathy (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pp. 3122–3133. ACM, 2025. doi: 10.1145/3711896.3736989.

Jiajing Wu, Dan Lin, Qishuang Fu, Shuo Yang, Ting Chen, Zibin Zheng, and Bowen Song. Toward understanding asset flows in crypto money laundering through the lenses of ethereum heists. *IEEE Trans. Inf. Forensics Secur.*, 19:1994–2009, 2024.

Sheng Xiang, Guibin Zhang, Dawei Cheng, and Ying Zhang. Enhancing attribute-driven fraud detection with risk-aware graph representation. *IEEE Trans. Knowl. Data Eng.*, 37(5):2501–2512, 2025. doi: 10.1109/TKDE.2025.3543887.

Shu Yang, Shenzhe Zhu, Zeyu Wu, Keyu Wang, Junchi Yao, Junchao Wu, Lijie Hu, Mengdi Li, Derek F. Wong, and Di Wang. Fraud-r1 : A multi-round benchmark for assessing the robustness of LLM against augmented fraud and phishing inducements. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 4374–4420. Association for Computational Linguistics, 2025.

Xiong Zhang, Hong Peng, Zhenli He, Cheng Xie, Xin Jin, and Hua Jiang. GCTAM: global and contextual truncated affinity combined maximization model for unsupervised graph anomaly detection. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 3642–3650. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/405.

Jianbo Zheng, Chao Yang, Tairui Zhang, Longbing Cao, Bin Jiang, Xuhui Fan, Xiao-Ming Wu, and Xianxun Zhu. Dynamic spectral graph anomaly detection. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 13410–13418. AAAI Press, 2025. doi: 10.1609/AAAI.V39I12.33464.

Yao Zou and Dawei Cheng. Effective high-order graph representation learning for credit card fraud detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 7581–7589. ijcai.org, 2024.

Table of Contents for Appendices

A. Comprehensive Review of Related Work

B. Detailed Formulation of Hypergraph Views

C. Detailed Formulation of Risk Indicators

D. Detailed Formulation of Parietal-Collaborative High-Order Learning

E. Algorithmic Framework

F. Detailed Implementation of the Self-Evolutionary Rule Closed-Loop

G. Dataset Specifications

H. Detailed Experimental Configurations

I. Detailed Baseline Description

J. Prompt Engineering and Neuro-Symbolic Synthesis

K. Complexity and Efficiency Analysis

L. Reproducibility, Resources, and Responsible Use

M. Iterative Optimization Analysis for Neuro-Symbolic Closed-Loop

N. Dynamic Evolution Analysis of Decision Boundaries

A Comprehensive Review of Related Work

Graph-Based Anti-Money Laundering Detection. Recent anti-money laundering studies Egressy et al. (2024); Li et al. (2025a) using GNNs have shifted from processing simple homogeneous graphs to adapting to complex financial transaction scenarios. To address multi-channel fund transfers, Egressy et al. (2024) established a theoretical framework for directed multigraphs, optimizing message passing schemes to enhance detection performance. Targeting the diversity and concealment of anomalies in real-world scenarios, Song et al. (2025a) proposed the UniFORM framework, which achieves joint anomaly detection across node, edge, and subgraph granularities. To address the signal imbalance caused by camouflage, Song et al. (2025c) proposed a dual-view graph representation learning method, while Li et al. (2025a) designed context-aware strategies to handle label scarcity. Moreover, modern money laundering activities often exhibit highly structured dense micro-cyclic patterns. Addressing this, Duan et al. (2024) proposed DGA-GNN to suppress background noise via dynamic grouping aggregation. To accurately capture high-order associations, Zhang et al. (2025) utilized truncated affinity maximization to peel off high-order dependencies. From a spectral perspective, Zheng et al. (2025) introduced DSGAD with dynamic wavelet transforms to capture anomalous energy distributions, while Wei et al. (2025) shifted the paradigm to uncertainty modeling.

However, most existing approaches predominantly employ discrete time-window partitioning to process continuous transaction streams. This discontinuous observational paradigm inevitably severs the intrinsic temporal dependencies and sequential coherence between transactions, causing originally integral fund chains to be fragmented into isolated segments. Additionally, detecting dense micro-cycles hidden within massive benign transactions remains challenging. Conventional neighborhood aggregation approaches often suffer from signal dilution, where sparse laundering signals are overwhelmed by dominant background noise during deep propagation. Consequently, these methods struggle to accurately disentangle faint high-order laundering links from dense interference, leading to the systematic under-detection of covert syndicates.

LLM-Driven Logic Evolution. To address non-stationary strategies, Gao et al. (2024) investigated attack-induced Structural Distribution Shift (SDS), revealing the lag of static models. He et al. (2025) incorporated State Space Models to capture long-range dependencies, and Niu et al. (2025) explored LLM-based zero-shot detection (UNPrompt) to enable adaptive perception. Nevertheless, money laundering strategies exhibit highly dynamic evolutionary characteristics. Most current models still rely on static discriminative logic derived from historical priors. When confronted with rapid strategic evolution and parameter shifts, these fixed decision boundaries fail to generalize, resulting in a cognitive gap where the model's decision logic becomes rigid and lags behind novel laundering variants.

B Detailed Formulation of Hypergraph Views

Let the chronological transaction stream be partitioned into windows $\{\mathcal{T}^{(s)}\}_{s=1}^{S}$, with $\mathcal{T}^{(s)} = \{t \mid T_s \le T(t) < T_s + \Delta T\}$. For each split and window, $V^{(s)}$ contains the account nodes observed up to the permitted chronological boundary, and every $\mathcal{E}_m^{(s)}$ is a family of subsets of $V^{(s)}$. Superscript (s) is omitted below when the window is unambiguous. This notation distinguishes a hyperedge family $\mathcal{E}_m$ from an individual node-set hyperedge $e \in \mathcal{E}_m$ and does not introduce links from future windows.

B.1 Detailed Formulation of the AB View

To capture the intensity of abnormal behavior for an account v_i within a discrete time window, we define the standardized amount score ($Zscore_{v_i}$) and the temporal burstiness score ($Burst_{v_i}$) using the following formulations:

(a) **Standardized Amount Score** ($Zscore_{v_i}$): This metric measures the cumulative deviation of transaction amounts from the historical baseline of the account:

$$Zscore(v_i) = \text{Norm}\left(\sum_{t \in T_{v_i}} \frac{x_t - \mu}{\sigma}\right), \tag{15}$$

where x_t is the amount of a single transaction t within the window, μ is the mean of historical transaction amounts, and σ represents the standard deviation (volatility) of those amounts.

(b) **Temporal Burstiness Score** ($Burst_{v_i}$): This score characterizes the high-frequency nature of transactions by aggregating the proximity of consecutive trades:

$$Burst(v_i) = \text{Norm}\left(\sum(\Delta T_{\max} - \Delta T)\right), \tag{16}$$

where ΔT is the time interval between two adjacent transactions, and $\Delta T_{\max}$ is the maximum allowable time interval defined for the current time window. A smaller ΔT results in a higher burstiness contribution. The function $\text{Norm}(\cdot)$ denotes the normalization process (e.g., Min-Max scaling) used to ensure all components are within a comparable range.

(c) **Normalization, Threshold, and Hyperedge Family**: The nonnegative coefficients satisfy $\mu_q \geq 0$ and $\sum_{q=1}^{3} \mu_q = 1$, so s_i remains on the common normalized scale of its components; the construction threshold is $\tau_{AB} = 0.6$. Let $\mathcal{B}$ be the set of bank contexts and $b(v_i)$ the bank associated with account v_i. Each bank context induces one node-set hyperedge

$$e_b^{AB} = \{v_i \in V \mid b(v_i) = b,\ s_i \geq \tau_{AB}\}, \qquad \mathcal{E}_{AB} = \{e_b^{AB} \mid b \in \mathcal{B},\ |e_b^{AB}| \geq 2\}. \tag{17}$$

Thus, $\mathcal{E}_{AB}$ is a family of account-node sets rather than a single thresholded vertex set.

B.2 Detailed Formulation of the BP View

To identify collaborative patterns in high-dimensional feature spaces, the BP view utilizes density-based clustering to aggregate similar transaction points. The mathematical formulation and implementation details are as follows:

(a) **Cluster Formulation**: The clustering mapping $x_k \to C_j$ aggregates transaction points that are density-reachable within the feature space. A cluster C_j is defined as:

$$C_j = \{v_k \mid \text{dist}(x_k, Core_j) \leq \tau_{BP}\}, \tag{18}$$

where $x_k \in \mathbb{R}^{d_x}$ is the normalized feature vector associated with account node v_k, $Core(C_j)$ denotes the set of core points within cluster C_j, and $\text{dist}(\cdot)$ is the Euclidean distance between account feature vectors.

(b) **Hyperedge Construction Rule**: Each identified cluster induces a distinct node-set hyperedge:

$$e_j^{BP} = \{v_k \in V \mid v_k \in C_j\}, \qquad \mathcal{E}_{BP} = \{e_j^{BP} \mid C_j \in \mathcal{C},\ |e_j^{BP}| \geq 2\}. \tag{19}$$

The feature vectors x_k determine cluster membership but are not themselves hyperedge vertices; the vertices are account nodes v_k.

(c) **Experimental Hyperparameters**: In our implementation, the neighborhood radius threshold τ_{BP} is set to 0.8. This value is optimized to balance the sensitivity of detecting structuring behaviors against the risk of including random background noise.

B.3 Detailed Formulation of the TA View

To accurately identify funds-flow chains that are both rapid and consistent in volume, we define the temporal and amount constraints for consecutive transactions $t_i = (A, B, \cdot, T_i)$ and $t_j = (B, C, \cdot, T_j)$ as follows:

(a) **Cascade Indicator Function**: The correlation between two transactions is determined by the indicator function $I_{cascade}$:

$$I_{cascade}(t_i, t_j) = \begin{cases} 1, & \text{if } 0 < T_j - T_i \leq \Delta t \text{ and } D_{amt}(i,j) \leq \tau_{TA} \\ 0, & \text{otherwise} \end{cases}, \tag{20}$$

where Δt is the maximum allowable time interval between the transfer-out (t_i) and transfer-in (t_j), and τ_{TA} is the threshold for relative amount difference.

(b) **Relative Amount Difference (D_{amt})**: To capture the amount conservation characteristic of money laundering chains (where funds are moved with minimal retention), we calculate D_{amt} as:

$$D_{amt}(i,j) = \frac{|amt(t_i) - amt(t_j)|}{\max(amt(t_i), amt(t_j))}, \tag{21}$$

where $amt(t_i)$ and $amt(t_j)$ represent the transaction amounts of the transfer-out (amt_P) and transfer-in (amt_R) stages, respectively.

(c) **Hyperedge Construction**: For every valid consecutive pair $t_i = (A, B, \cdot, T_i)$ and $t_j = (B, C, \cdot, T_j)$, define

$$e_{ij}^{TA} = \{A, B, C\}, \qquad \mathcal{E}_{TA} = \{e_{ij}^{TA} \mid I_{cascade}(t_i, t_j) = 1\}. \tag{22}$$

Thus, each qualifying cascade contributes one high-order node set, with B acting as the layering or mixing node.

B.4 Detailed Formulation of the LS View

The LS view uses laundering templates fixed from the chronological training partition. Let $\mathcal{T}_{train}^{+}$ denote positive training transactions, $\mathcal{U}^{(s)}$ denote the unlabeled candidate transactions available in split s, and $q(t) \in \mathbb{R}^{d_q}$ denote the normalized transaction representation used by the implementation. For a candidate t_k and a training template t_j^{+}, the cosine similarity is

$$s_{kj} = \frac{q(t_k)^{\top} q(t_j^{+})}{\|q(t_k)\|_2 \, \|q(t_j^{+})\|_2}. \tag{23}$$

Each training template induces a candidate node-set hyperedge

$$e_j^{LS} = \{v(t_k) \in V \mid t_k \in \mathcal{U}^{(s)},\ s_{kj} \geq \tau_{sim}\}, \qquad \mathcal{E}_{LS}^{(s)} = \{e_j^{LS} \mid t_j^{+} \in \mathcal{T}_{train}^{+},\ |e_j^{LS}| \geq 2\}. \tag{24}$$

The similarity threshold τ_{sim} is selected using training/validation data and then fixed for OOT evaluation. At test time, membership depends only on $q(t_k)$, the frozen training templates, and τ_{sim}; test labels and future positive transactions are never queried. This construction therefore avoids look-ahead label leakage while retaining similarity-based coordination evidence.

C Detailed Formulation of Risk Indicators

C.1 Structural Risk Indicators

To characterize the abnormal association structures within hyperedge local subgraphs G_e, we define the following four topological indicators:

(a) **Maximum In-degree (d_{in}^{max})**: Measures the prominence of fund aggregation. It identifies potential fund pools that collect scattered transactions:

$$d_{in}^{max} = \max_{v \in e}(\text{Indegree}(v)), \tag{25}$$

where $v \in e$ represents the account nodes within hyperedge e. In AML tasks, a node with a significantly higher in-degree than its peers often serves as a centralized collection point for dispersed funds.

(b) **Maximum Out-degree (d_{out}^{max})**: Measures the dominance of fund distribution. It identifies critical hubs used to disperse illicit capital:

$$d_{out}^{max} = \max_{v \in e}(\text{Outdegree}(v)), \tag{26}$$

Accounts with high out-degrees typically act as distribution centers, splitting aggregated funds into multiple downstream accounts to obfuscate the money trail.

(c) **Cycle Intensity** (CI): Simulates the fear response to recursive patterns. A high density of cycles indicates fund layering which is a classic tactic to hide the true source and destination of money:

$$CI = \frac{1}{|V_{G_e}|} \sum_i \mathbb{I}(\text{Cycle}_i), \tag{27}$$

where $|V_{G_e}|$ is the total number of nodes in the local subgraph, and $\mathbb{I}(\text{Cycle}_i)$ is an indicator function that equals 1 if a closed loop (e.g., $A \to B \to C \to A$) exists.

(d) **Gini Imbalance** ($Gini$): Measures the non-natural tilt or asymmetry of transaction flows. In amygdala-inspired logic, high asymmetry signals a disruption of environmental stability:

$$Gini = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |d_{(i)} - d_{(j)}|}{2n \sum_{i=1}^{n} d_{(i)}}, \tag{28}$$

where $d_{(i)}$ represents the sorted sequence of node degrees within the hyperedge ($d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(n)}$). Here, d can represent either in-degree or out-degree depending on the direction of the risk assessment.

After these indicators are placed on comparable normalized scales, $\Phi_{logic}(\cdot; \Theta_{struct})$ maps them to $S_{struct}(e) \in [0, 1]$. The parameter set Θ_{struct} contains rule thresholds, slopes, and permitted operators. It is not optimized by neural backpropagation: the parameters of each active rule remain fixed during a neural cycle, and the active set is augmented only after SRCL sandbox validation.

C.2 Behavioral Risk Indicators

Behavioral risk analysis identifies potential money laundering by quantifying the degree of deviation in transaction attributes. The four specific metrics are defined as follows:

(a) **Amount-Template Similarity** (S_{amt}): This metric uses the complement of Total Variation Distance to measure similarity between the current hyperedge amount distribution and a predefined risk baseline:

$$S_{amt}(e) = 1 - \frac{1}{2} \sum_{k=1}^{K_a} |P_e(k) - P_0(k)|, \tag{29}$$

where $P_e(k)$ and $P_0(k)$ are the probabilities of the k-th amount bin and K_a is the number of amount bins. Hence $S_{amt}(e) \in [0, 1]$, with larger values indicating closer agreement with the risk template. This is a similarity score, not an amount difference; the relative amount difference used by the TA cascade remains the separate quantity $D_{amt}(i, j)$.

(b) **Currency Diversity** (Ent_{ccy}): We use normalized information entropy to measure multi-currency dispersion. Let K_e be the number of distinct currencies observed in hyperedge e:

$$Ent_{ccy}(e) = \begin{cases} 0, & K_e \leq 1, \\ -\dfrac{1}{\ln K_e} \displaystyle\sum_{i=1}^{K_e} p_i \ln p_i, & K_e > 1, \end{cases} \tag{30}$$

where p_i is the observed proportion of currency i. The explicit $K_e \leq 1$ branch avoids the undefined division by $\ln 1$.

(c) **Transaction Format Inconsistency** (I_{format}): Frequent switching between transaction formats is treated as a behavioral warning. We therefore use the complement of the dominant-format share:

$$I_{format}(e) = 1 - \frac{1}{|e|} \sum_{v \in e} \mathbb{I}(\text{format}(v) = \text{mode}(e)), \tag{31}$$

where $\text{format}(v)$ is the transaction format associated with node v and $\text{mode}(e)$ is the most frequent format in the hyperedge. The score is 0 under a uniform format and increases as format switching becomes more diverse, matching its positive contribution to S_{beh}.

(d) **Time Compression** (T_{comp}): This quantifies the density of transaction timing, simulating the bursty firing of bio-electrical signals. It serves as a physical marker for rapid fund transfers:

$$T_{comp}(e) = \frac{1}{\Delta T} \sum_{v \in e} f_v, \tag{32}$$

where ΔT is the observation window and f_v denotes the transaction frequency of node v.

Collecting the four normalized quantities as $r_{beh}(e) = [S_{amt}(e), Ent_{ccy}(e), I_{format}(e), T_{comp}(e)]^\top$, the main-text behavioral score is $S_{beh}(e) = w_b^\top r_{beh}(e)$. We use $w_b \in \mathbb{R}^4_{>0}$ with $\mathbf{1}^\top w_b = 1$, which makes the score an interpretable convex combination rather than an unbounded sum.

D Detailed Formulation of Parietal-Collaborative High-Order Learning

As the high-order spatial cognition hub of the brain, the Parietal Lobe is responsible for receiving warning signals from the Amygdala and initiating high-pathway deep processing. This process achieves deep integration from micro-local correction to macro-panoramic decision-making within a unified Neuro-Symbolic Cognitive Space. Throughout this section, let $N = |V|$ be the number of account nodes, d_h the hidden dimension, and $v \in V$ an individual node. Notation formatted as $(\cdot)(v)$ is node-specific, and matrices collecting all node representations have shape $N \times d_h$ unless otherwise stated.

D.1 Construction of Neuro-Symbolic Cognitive Space

To enable effective transmission of low-pathway signals in the deep network, different types of information are first mapped to unified features:

(a) **Structural Sensory Features (X_{raw}):** Includes basic statistics such as transaction frequency and cycle intensity.

(b) **Prior Warning Features (X_{rule}):** Derived from the Amygdala warning factor $\gamma_A(v)$ and risk one-hot encoding.

The final node feature is $h^{(0)}(v) = \text{Concat}(X_{raw}(v), X_{rule}(v))$, providing complete input for the deep reasoning of the Parietal Lobe.

D.2 Circuit 1: View-specific Residual Attention

This circuit simulates the fine processing of local details by the Parietal Lobe to solve signal dilution. We design a risk-aware residual strategy using $\gamma_A(v)$ to amplify high-risk feature amplitudes:

$$H^{(l)}_{guided}(v) = h_{attn}(v) \odot (1 + \lambda_{alert} \cdot \gamma_A(v)), \tag{33}$$

$$h^{(l)}(v) = \sigma(PReLU(LN(W_{res} h^{(l-1)}(v) + W_{trans} H^{(l)}_{guided}(v)))), \tag{34}$$

where $h_{attn}(v) \in \mathbb{R}^{d_h}$ is the context feature of node v after GNN sigmoid-gated aggregation; $\odot$ is the Hadamard product; the scalar $\gamma_A(v)$ is broadcast to all d_h channels; and $\lambda_{alert} \geq 0$ controls gain intensity. Moreover, $h^{(l-1)}(v)$ is the preceding-layer state, W_{res} and W_{trans} are learnable projections with outputs in $\mathbb{R}^{d_h}$, LN is LayerNorm, $PReLU$ is the parametric rectifier, and the outer σ is sigmoid gating.

D.3 Circuit 2: Markov High-order Structural Diffusion

Let $\mathcal{M} = \{AB, BP, TA, LS\}$. For each view m, let $H_m \in \{0,1\}^{N \times |\mathcal{E}_m|}$ be its incidence matrix, with $(H_m)_{ve} = 1$ iff node v belongs to hyperedge e. The implementation concatenates all view-specific incidences,

$$H = [H_{AB} \parallel H_{BP} \parallel H_{TA} \parallel H_{LS}], \tag{35}$$

so the projected node association matrix is

$$A = HH^\top = \sum_{m\in\mathcal{M}} H_m H_m^\top. \tag{36}$$

Thus $H \in \{0,1\}^{N\times E}$ with $E = \sum_m |\mathcal{E}_m|$, and concatenation aggregates the four node–hyperedge–node relations without enumerating cross-view subgraphs. Define the projected node degree and transition matrices as

$$D_A = \mathrm{diag}(A\mathbf{1}), \qquad P = D_A^{-1}A. \tag{37}$$

Isolated zero-degree nodes are assigned self-transitions $P_{vv} = 1$. For every non-isolated node,

$$P\mathbf{1} = D_A^{-1}A\mathbf{1} = D_A^{-1}D_A\mathbf{1} = \mathbf{1}, \tag{38}$$

which proves that P is row-stochastic. Notice that D_A is the degree matrix of the projected association A, not merely the incidence count $\mathrm{diag}(H\mathbf{1})$; this distinction is what ensures probability conservation when hyperedges contain multiple nodes. The Random Walk with Restart is initialized and updated as

$$Z^{(0)} = h^{(0)}, \tag{39}$$

$$Z^{(k)} = (1-\beta)PZ^{(k-1)} + \beta Z^{(0)}, \tag{40}$$

where $Z^{(k)} \in \mathbb{R}^{N\times d_h}$, $k \in \{1,\dots,K\}$ is the diffusion step, and $\beta \in (0,1)$ is the probability of restarting from $Z^{(0)}$. The use of $Z^{(0)}$ in the restart term makes the walk return explicitly to the initialized neuro-symbolic representation.

D.4 Detailed Step-by-Step Parietal Panoramic Assembly

Following the deep processing in Circuit 1, the Parietal module executes a three-step assembly process to synthesize multi-view evidence into a final panoramic decision.

Prerequisite: View Projection. To ensure all view-specific hidden states are comparable in the Neuro-Symbolic Cognitive Space, the output $\mathbf{h}^{(l)}(v)$ from Circuit 1 for each view m is first projected through a fully connected layer:

$$h_{proj}^{(m)}(v) = \mathrm{ELU}(W_{proj}^{(m)} \cdot \mathbf{h}^{(l)}(v)), \tag{41}$$

where $W_{proj}^{(m)}$ is the learnable projection matrix for view m, and $\mathrm{ELU}(\cdot)$ is the activation function.

Step 1: Saliency Weight Arbitration. Faced with evidence conflicts from heterogeneous views, the Parietal Lobe assigns each view a separate projection vector $W_{attn}^{(m)} = [(w_h^{(m)})^\top, a_m]$. For node v, the score and normalized weight are

$$s_m(v) = (w_h^{(m)})^\top h_{global}^{(m)} + a_m\gamma_A(v), \qquad \omega_m(v) = \frac{\exp(s_m(v))}{\sum_{j=1}^{M}\exp(s_j(v))}, \tag{42}$$

where $h_{global}^{(m)} = \frac{1}{|V|}\sum_{u\in V} h_{proj}^{(m)}(u) \in \mathbb{R}^{d_h}$ is the global context of view m, $w_h^{(m)} \in \mathbb{R}^{d_h}$, $a_m \in \mathbb{R}$, and $M = |\mathcal{M}| = 4$. Because a_m is view-specific, the warning contribution is not an additive constant shared by every softmax logit and therefore does not cancel. The resulting node-specific weights satisfy $\omega_m(v) > 0$ and $\sum_m \omega_m(v) = 1$.

Step 2: Generation of Parietal and Feedback Features. Two distinct feature sets are generated to resolve signal conflicts:

(a) **Parietal Feature (Weighted Consensus):** A panoramic summary for node v based on saliency arbitration: $h_{parietal}(v) = \sum_{m=1}^{M}\omega_m(v)\odot h_{proj}^{(m)}(v)$.

(b) **Deep Feedback Feature:** Uses fused macro-information to reinforce local view details:

$$h_{feedback}^{(m)}(v) = h_{proj}^{(m)}(v) \odot (1 + \lambda_{alert}\cdot \mathrm{Mean}_{feat}(h_{parietal}(v))), \tag{43}$$

where $\mathrm{Mean}_{feat}(\cdot)$ denotes the feature-wise mean of the fused parietal representation.

Step 3: Prefrontal Gating and Spatial Atlas Fusion. A Prefrontal Gating variable $g(v)$ is introduced to adaptively regulate the trade-off for node v:

$$g(v) = \sigma(W_{gate} \cdot h_{parietal}(v)). \tag{44}$$

The final assembled focal feature $h_{final}(v)$ is obtained as:

$$h_{final}(v) = g(v) \odot h_{parietal}(v) + (1 - g(v)) \odot \frac{1}{M} \sum_{m=1}^{M} h_{feedback}^{(m)}(v), \tag{45}$$

where $W_{gate} \in \mathbb{R}^{d_h \times d_h}$ is the gating matrix and $g(v) \in (0,1)^{d_h}$ controls the feature-wise trade-off between consensus and feedback. Finally, the focal features h_{final}, the panoramic spatial atlas $Z^{(k)}$, and the raw attributes X_{raw} are concatenated to obtain the final classification for node v:

$$H_{combined}(v) = h_{final}(v) \parallel Z^{(k)}(v) \parallel X_{raw}(v), \tag{46}$$

$$\hat{y}_v = \sigma(w^\top H_{combined}(v) + b), \tag{47}$$

where w and b are the learnable classifier weight and bias, and $\hat{y}_v \in [0,1]$ is the predicted laundering probability of node v.

E Algorithmic Framework

Algorithm 1 formalizes the complete LEAP-HGN learning procedure, separating the outer rule-evolution cycles from the inner neural epochs and multi-layer graph propagation.

Algorithm 1 LEAP-HGN: Brain-inspired Dual-Pathway Learning

Input: Transaction Graph $\mathcal{G} = \{V, E\}$; Prior Rule Set $\mathcal{R}$; Hyperparameters $\{\alpha_r, \lambda_{alert}, \beta, C, E_c, L\}$.
Output: Risk Probabilities $\hat{Y}$; Optimized Parameters Θ^*.

Initialization: Construct multi-view hypergraphs $\{\mathcal{H}_m\}_{m\in\{AB,BP,TA,LS\}}$ and initialize rule set $\mathcal{R}^{(1)}$
for evolutionary cycle $c = 1, \ldots, C$ **do**
  for neural epoch $q = 1, \ldots, E_c$ **do**
    ▷ Phase I: Amygdala Low-Pathway Risk Warning
    for each hyperedge e in the four views **do**
      for each active rule $r \in \mathcal{R}^{(c)}$ **do**
        Calculate $S_{struct,r}(e)$ and $S_{beh,r}(e)$;
        $M_r(e) \leftarrow \alpha_r S_{struct,r}(e) + (1 - \alpha_r) S_{beh,r}(e)$;
      end for
      $M^*(e) \leftarrow \max_{r\in\mathcal{R}^{(c)}} M_r(e)$;
    end for
    $\gamma_A(v) \leftarrow \max_{e\ni v} M^*(e)$;
    ▷ Phase II: Parietal High-Pathway Deep Reasoning
    for GNN layer $l = 1, \ldots, L$ **do**
      for each view $m \in \{AB, BP, TA, LS\}$ **do**
        $\mathbf{H}^{(m,l)}_{attn} \leftarrow \text{HyperAttn}(\mathbf{H}^{(m,l-1)}, \mathcal{H}_m)$;
        $\mathbf{H}^{(m,l)}_{guided} \leftarrow \mathbf{H}^{(m,l)}_{attn} \odot (1 + \lambda_{alert}\gamma_A)$;
        $\mathbf{H}^{(m,l)} \leftarrow \text{LayerNorm}(\text{Residual}(\mathbf{H}^{(m,l)}_{guided}))$;
      end for
    end for
    ▷ Phase III: Global Diffusion and Prefrontal Assembly
    $\mathbf{Z}^{(k)} \leftarrow \text{RWR}(\mathbf{Z}^{(k-1)}, \mathbf{H}, \beta)$;
    Compute $\mathbf{h}_{final}$ via Prefrontal Gating and set $\mathbf{H}_{combined} \leftarrow \mathbf{h}_{final} \parallel \mathbf{Z}^{(k)} \parallel \mathbf{X}_{raw}$;
    Compute $\hat{Y}$ using Equation 47;
    Update neural parameters Θ by minimizing $\mathcal{L}_{Focal}(\hat{Y}, Y_{true})$ via Adam;
  end for
  ▷ Phase IV: Self-Evolutionary Rule Closed-Loop
  Mine $V^{(c)}_{hard}$ from training predictions and aggregate $z^{(c)}_{\text{profile}}$;
  Generate candidate rule $\mathcal{R}^{(c)}_{new}$ from $z^{(c)}_{\text{profile}}$ using the LLM;
  Parse and evaluate $\mathcal{R}^{(c)}_{new}$ in the AST sandbox on the historical cache;
  if $\mathcal{R}^{(c)}_{new}$ satisfies the configured acceptance criteria **then**
    $\mathcal{R}^{(c+1)} \leftarrow \mathcal{R}^{(c)} \cup_{\text{id}} \{\mathcal{R}^{(c)}_{new}\}$;
    ▷ $\cup_{\text{id}}$: append a new name or update the same-name entry
  else
    $\mathcal{R}^{(c+1)} \leftarrow \mathcal{R}^{(c)}$;
  end if
end for
return $\hat{Y}$ and Θ^*.

Summary of Computational Logic:

(a) **Rapid Perception:** The Amygdala pathway generates a warning engram γ_A by matching transaction flows against structural and behavioral risk templates.

(b) **Cognitive Gain:** Warning signals γ_A serve as a top-down attentional bias that amplifies high-risk features before multi-layer Parietal propagation.

(c) **Periodic Evolution:** After each evolutionary cycle, hard negatives are aggregated into an error profile, and an accepted sandbox-validated rule is activated in the subsequent cycle.

F Detailed Implementation of the Self-Evolutionary Rule Closed-Loop

To address the operational transparency of the neuro-symbolic architecture, this section details the precise engineering logic, mathematical formulations, and execution workflow of the Self-Evolutionary Rule Closed-Loop. The process is decoupled into four phases: Hard Negative Profiling, Dynamic Prompting, Sandbox Execution, and Alternating Symbolic-Neural Updating.

F.1 Subgraph Backtracking and Hard Negative Profiling

At the end of each GNN training cycle, the system identifies blind spots by isolating Hard Negatives or False Negatives. A node v is classified as a hard negative if its ground truth label indicates money laundering ($y_v = 1$), yet the GNN predicted risk probability ($\hat{y}_v$) falls below the strict miss-report threshold τ_{miss} (empirically set to 0.4 in our implementation):

$$V_{hard} = \{v \in \mathcal{V} \mid y_v = 1 \wedge \hat{y}_v < \tau_{miss}\}, \tag{48}$$

where $\mathcal{V}$ is the global set of account nodes.

To understand why the GNN fails, the system performs a topological backtrack. It retrieves the set of hyperedges $\mathcal{E}_{hard} = \{e \in \mathcal{E} \mid e \cap V_{hard} \neq \emptyset\}$ associated with these missed accounts. Utilizing a globally cached feature engine, the system extracts real-time subgraph indicators $\mathbf{f}(e)$ for each hyperedge, encompassing structural metrics (e.g., cycle strength cyc, maximum in-degree $\mathrm{indeg_max}$) and behavioral metrics (e.g., transaction scale $\mathrm{total_money}$).

The error profile vector z_{profile} is then constructed by aggregating these features:

$$z_{\mathrm{profile}} = [\mu(cyc), \mu(\mathrm{time_span}), \mu(\mathrm{total_money}), \dots], \tag{49}$$

where $\mu(\cdot)$ denotes the mean statistical value of a specific feature across all hyperedges in $\mathcal{E}_{hard}$. This vector explicitly quantifies the latent patterns that the current neural network fails to capture.

F.2 Data-Driven Dynamic Prompting and Rule Instantiation

The LLM does not generate natural language advice; rather, it functions as a semantic parser that translates z_{profile} into strictly formatted, smooth, and bounded mathematical expressions. For the main-text form $S_{new}(e) = \sigma(k(f_e - \tau_{new}))$, f_e is a selected normalized feature of hyperedge e, τ_{new} is its synthesized decision threshold, $k > 0$ controls transition sharpness, and $\Theta_{new} = \{k, \tau_{new}\}$ denotes these rule parameters. To prevent invalid or unstable logic, we implement a Feature Adaptive Masking Prompt strategy.

The prompting strategy dynamically adjusts its constraints based on z_{profile}. For instance, if $\mu(cyc) > 0.01$, the LLM is forcibly instructed to incorporate the cyc feature using a specific activation function such as sigmoid$((cyc - \min) * 10)$. Conversely, if the cycle feature is negligible ($\mu(cyc) \leq 0.01$), the system masks this feature and redirects the LLM attention to higher-order non-linear combinations of $\mathrm{edge_size}$ and $\mathrm{total_money}$.

The instantiated rule $\mathcal{R}_{new}$ strictly adheres to a predefined JSON schema and is formally defined as a 5-tuple:

$$\mathcal{R}_{new} = \langle \mathcal{F}, \alpha_r, \mathbf{W}_{beh}, \Phi_{logic}(\cdot; \Theta_{new}), c_{prior} \rangle, \tag{50}$$

where $\mathcal{F}$ represents the predefined topological family space including FAN IN, FAN OUT, and CYCLE; α_r is a rule-level balance coefficient selected within $[0.5, 0.7]$ during rule synthesis and held fixed throughout the subsequent neural training cycle; $\mathbf{W}_{beh}$ signifies the behavioral weight distribution composed of money, currency, time, and format weights summing to one; Φ_{logic} is the core symbolic logic string such as a sigmoid function of maximum in-degree and threshold; and c_{prior} is a rule-specific coefficient used during risk-grade calibration. In the six-field JSON schema, `Parameter_List` specifies Θ_{new} within Φ_{logic} and therefore does not constitute an additional top-level component of the mathematical 5-tuple.

F.3 Neuro-Symbolic Sandbox Execution and Fast Verification

To apply the LLM generated string logic to the computational graph, the system employs a secure execution sandbox (Sandbox Evaluator). The sandbox preloads fundamental mathematical operators $\mathcal{O}_{safe} = \{\sigma, \max, \min, \exp, |\cdot|\}$ and maps the cached raw features $\mathbf{f}(e)$ as local variables.

For any hyperedge e, the structural risk score of rule r, $S_{struct,r}(e)$, is dynamically evaluated by parsing the Abstract Syntax Tree (AST) of $\Phi_{logic,r}$:

$$S_{struct,r}(e) = \text{Eval}(\Phi_{logic,r}, \text{env} = \{\mathcal{O}_{safe} \cup \mathbf{f}(e) \cup \Theta_r\}), \tag{51}$$

Simultaneously, $S_{beh,r}(e)$ is computed from the normalized behavioral indicators using the rule-specific weight vector $\mathbf{W}_{beh,r}$. The continuous matching score is

$$M_r(e) = \alpha_r S_{struct,r}(e) + (1 - \alpha_r) S_{beh,r}(e). \tag{52}$$

The rule-specific coefficient $c_{prior,r}$ is passed to the subsequent risk-grade calibration routine, which maps the continuous match to its risk grade and associated alert weight. It therefore does not directly rescale $S_{struct,r}(e)$, $S_{beh,r}(e)$, or the continuous matching score $M_r(e)$.

Rule Screening Criterion. Before a candidate rule is activated in the Amygdala rule base, it is evaluated on the historical rule-screening cache. A candidate is retained only if its rule-level Recall exceeds that of the current rule configuration and its rule-level F1 remains above the configured minimum screening threshold:

$$\text{Recall}_{\text{new}} > \text{Recall}_{\text{current}}, \qquad F1_{\text{rule}} > 0.1. \tag{53}$$

Both quantities are computed from the candidate rule's cache-level matches rather than from the final node classifier. This lightweight heuristic screening mechanism rejects degenerate candidates; it should not be interpreted as a guarantee of monotonic improvement in the final node-level classification metrics.

Multi-Rule Aggregation and Prior Calibration. Validated corrective rules accumulate across evolutionary cycles. Hence, every active rule is evaluated independently on each hyperedge, and the effective hyperedge risk is the strongest continuous match:

$$M^*(e) = \max_{r \in \mathcal{R}^{(c)}} M_r(e), \qquad M_r(e) = \alpha_r S_{\text{struct},r}(e) + (1 - \alpha_r) S_{\text{beh},r}(e). \tag{54}$$

The corresponding account-level warning factor is

$$\gamma_A(v) = \max_{e \ni v} M^*(e). \tag{55}$$

Operationally, the first maximum selects the best-matching active rule for each hyperedge, whereas the second propagates the strongest incident-hyperedge evidence to account v. After the best-matching rule has been identified, its $c_{prior,r}$ is used by the risk-grade calibration routine; the coefficient does not alter either maximum or the continuous scores entering them. The complete flow is therefore rule accumulation, followed by rule-wise maximization and then hyperedge-wise maximization.

F.4 Alternating Symbolic Modulation and Neural Updating

At the boundary of evolutionary cycle c, a validated corrective rule is appended to the active symbolic rule set:

$$\mathcal{R}^{(c+1)} = \mathcal{R}^{(c)} \cup_{\text{id}} \{\mathcal{R}^{(c)}_{new}\}. \tag{56}$$

Here, $\cup_{\text{id}}$ denotes identifier-aware union. Rules with previously unseen identifiers are appended; if a candidate reuses an existing identifier, the implementation updates that named entry without removing any unrelated active rule. Thus, the rule base accumulates validated knowledge while avoiding duplicate entries with the same identifier.

The AST sandbox returns ordinary scalar rule values, so the parameters of every active rule are held fixed during the subsequent neural training cycle. Formally,

$$\frac{\partial \mathcal{L}_{Focal}}{\partial \Theta_r} = 0, \qquad \forall r \in \mathcal{R}^{(c+1)}, \quad \text{within a neural optimization cycle.} \tag{57}$$

The aggregated scalar $\gamma_A(v)$ is injected into the Parietal module as a fixed neuromodulatory prior during that cycle:

$$\mathbf{H}^{(l)}_{guided}(v) = \mathbf{h}_{attn}(v) \odot (1 + \lambda_{alert} \cdot \gamma_A(v)), \tag{58}$$

where λ_{alert} controls the gain intensity. Gradients update the neural parameters conditioned on this fixed prior. After the cycle, a new hard-sample profile is constructed and the LLM may synthesize an additional corrective rule. Once validated, that rule is appended to the active symbolic rule set and becomes available in the subsequent evolutionary cycle. The closed loop is therefore an alternating symbolic-neural update, not end-to-end gradient optimization of the AST rule parameters.

G Dataset Specifications

The used datasets span diverse financial domains including synthetic banking, e-commerce, and mobile money, ensuring a robust evaluation of model cross-domain generalization and the ability to handle varying feature densities. Table 2 summarizes the overall statistics of these six datasets.

Table 2: Statistics of six money laundering datasets.

Datasets	#Nodes	#Edges	#Fraud	Fraud Ratio
HI-Small	515,080	5,078,345	3,209	0.06%
HI-Medium	2,135,071	32,546,846	20,387	0.06%
HI-Large	2,135,071	180,387,249	200,854	0.11%
XploreML	9,999	992,425	1,279	0.13%
PaySim	6,353,307	6,362,620	8,213	0.13%
IEEE-CIS	217,935	2,653,878	20,663	3.50%

Task Definition and Transaction-to-Node Label Mapping. LEAP-HGN formulates detection as an account-level node classification task. For datasets originally annotated at the transaction level, an account v is labeled positive within chronological split s if it participates as either the sender or the receiver in at least one positively labeled transaction from that split:

$$y_v^{(s)} = \max_{t \in \mathcal{T}^{(s)}:\, v \in \{\text{sender}(t), \text{receiver}(t)\}} y_t. \tag{59}$$

Here, s denotes the chronological training, validation, or test split, and $\mathcal{T}^{(s)}$ is the set of transactions assigned to that split. Otherwise, $y_v^{(s)} = 0$. For each chronological split, node labels are constructed exclusively from transactions within the corresponding temporal window, ensuring that no future transaction labels are used to define node labels in earlier splits. Accordingly, the ground-truth vector is defined over account nodes rather than transactions, and $\hat{y}_v \in [0, 1]$ denotes the predicted risk probability of account node v throughout the paper.

G.1 IBM-HI Series (Synthetic Banking)

The IBM-HI series (comprising Small, Medium, and Large scales) serves as the primary benchmark for evaluating our model scalability Altman et al. (2023). Generated via the **IBM AMLSim** simulator, these datasets are specifically engineered to reflect the structuring and layering phases of modern money laundering. By providing explicit attributes such as `Currency` and `Format`, the HI series allows our Amygdala pathway to perform precise threat template matching against complex cross-border transaction flows. The detailed schema of the IBM-HI Series is outlined in Table 3.

Table 3: Detailed specification of IBM-HI series schema

Attribute	Description
Data Source	IBM AMLSim (Synthetic Banking Transaction)
Data Type	Multi-Bank Cross-Border Transaction Graph
Topological Scale	From 515K to 2.13M nodes; up to 180M edges
Key Features	`Timestamp`, `From Bank`, `Account0`, `To Bank`, `Account1`, `Amount`, `Currency`, `Format`
Ground Truth	`Is Laundering` (Binary Label)

G.2 XPLOREML (E-COMMERCE INTERACTION)

The XploreML(GDSC) dataset is a specialized benchmark focusing on user-device and account-sharing behaviors within e-commerce ecosystems. Unlike traditional banking data, this dataset emphasizes high-order associations between IP addresses, device fingerprints, and transaction identities. We utilize this benchmark to verify the **LLM-driven rule evolution** module, testing whether evolved rules can effectively generalize to non-banking anomalies such as brushing or fake-order laundering. The key attributes of this dataset are detailed in Table 4.

Table 4: Detailed specification of XploreML (GDSC) schema

Attribute	Description
Data Source	Kaggle / IEEE Global Data Science Challenge
Data Type	E-commerce transaction and user interaction graph
Interaction Logic	User-Device-IP multi-order associations
Key Features	`Timestamp`, `DeviceID`, `IP_Address`, `TransactionAmt`, `Card_Type`, `Addr1`, `P_emaildomain`
Ground Truth	`isFraud` (Binary Label)

G.3 PAYSIM (MOBILE FINANCIAL SERVICES)

PaySim provides a high-fidelity simulation of mobile money transactions based on real-world logs from an African financial service. It is characterized by its high temporal granularity, making it an ideal candidate for validating the **Temporal Association (TA)** view of LEAP-HGN. The dataset specifically captures the rapid fund depletion and pass-through patterns common in mobile wallet layering, where funds are moved through multiple accounts within extremely short intervals. Table 5 presents the detailed schema specification.

Table 5: Detailed specification of PaySim schema

Attribute	Description
Data Source	Mobile Money Simulator (PaySim)
Data Type	Temporal mobile financial service (MFS) transaction flow
Temporal Logic	Sequential time steps (steps) with transaction ordering
Key Features	`step`, `type` (CASH_OUT, TRANSFER, etc.), `amount`, `nameOrig`, `oldbalanceOrg`, `nameDest`
Ground Truth	`isFraud` (Binary Label)

G.4 IEEE-CIS (HETEROGENEOUS PAYMENT DETECTION)

The IEEE-CIS dataset, provided by Vesta Corporation, represents a massive-scale, high-dimensional heterogeneous credit card transaction network. This dataset poses a significant challenge for GNN-based models due to its extreme feature sparsity and anonymized high-dimensional feature space

(V1–V339). We employ this benchmark to test the robustness of our **Behavioral Pattern (BP)** view and the ability of the Parietal module to extract high-order signals from background noise. The detailed specification is provided in Table 6.

Table 6: Detailed specification of IEEE-CIS schema

Attribute	Description
Data Source	Vesta Corporation Payment Dataset (Kaggle)
Data Type	Heterogeneous credit card transaction network
Feature Density	394 high-dimensional features with significant anonymity
Key Features	TransactionDT, TransactionAmt, ProductCD, card1–card6, dist1, V1–V339
Ground Truth	isFraud (Binary Label)

H Detailed Experimental Configurations

To ensure the reproducibility of the **LEAP-HGN** framework, we provide the exhaustive configuration of our experimental environment, hyperparameter settings, and the rigorous Out-of-Time (OOT) evaluation protocol.

H.1 Unified Evaluation Metrics

All methods in Table 1 are reported using the same three metrics. Let $y_i \in \{0, 1\}$ denote the ground-truth label, where class 1 is illicit, and let $\hat{y}_i$ and s_i denote the thresholded prediction and continuous risk score. For the illicit class, we compute

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad F1 = \frac{2PR}{P + R}. \tag{60}$$

If a denominator is zero, the corresponding score is set to zero. The reported Recall is the illicit-class recall, $R = TP/(TP + FN)$. AUC denotes the area under the receiver operating characteristic curve computed from the continuous scores s_i. All values are expressed as percentages in the main table.

H.2 Hyperparameter Specifications

The optimal hyperparameters for LEAP-HGN were determined via systematic grid search and are standardized across all six datasets. Table 7 summarizes the actual configuration used in the final execution loop.

Table 7: Detailed hyperparameter settings for LEAP-HGN reproducibility

Category	Hyperparameter	Value
Architecture	Input Feature Dimension	128
	Hidden Representation Dimension (d)	64
	Number of GNN Layers	3
Optimization	Initial Learning Rate (η)	1×10^{-3}
	Evolutionary Decay Rate	5×10^{-4}
	SMOTE Sampling Strategy	0.8
	Optimizer	Adam
Loss Function	Focal Loss Focusing Parameter (γ_F)	2.0
	Focal Loss Balancing Factor (α)	0.85
	Regularization Coefficient (λ_{reg})	0.1
Evolutionary	Max Evolutionary Cycles	10
	Epochs per Cycle	30
	Early Stopping Patience	4 Cycles

H.3 LLM Configuration and Training Efficiency Analysis

In the Self-evolutionary Rule Closed-Loop, we utilize **qwen-plus** as the core semantic reasoning engine. The integration follows an asynchronous periodic-evolution strategy to maintain high computational efficiency.

(a) **Model Engine:** qwen-plus, accessed via the DashScope API with a temperature setting of 0.2 and top_p of 0.7 to ensure rule stability and reasoning quality.

(b) **Reasoning Protocol:** At the conclusion of each evolutionary cycle, the system mines hard negative samples (false negatives with predicted probability < 0.4) and extracts their structural engrams for symbolic rule generation.

(c) **Efficiency Analysis:** Because LLM inference is decoupled from the GPU-intensive GNN gradient descent, the training speed is preserved. Each evolutionary step incurs an average latency of 12–18 seconds. For a 10-cycle run, the total LLM overhead accounts for approximately 3.8% to 4.5% of the total wall-clock time.

H.4 Temporal Partitioning and OOT Protocols

To evaluate LEAP-HGN resilience against evolving money laundering strategies, we implement a strict chronological split (OOT) instead of random shuffling. This ensures the test set reflects a future environment with potential concept drift. Table 8 presents the specific chronological partitioning boundaries.

Table 8: Chronological partitioning boundaries for Out-of-Time (OOT) Evaluation

Dataset	Time Feature	Train (70%)	Val (10%)	Test/OOT (20%)
HI-Series	`Timestamp`	Initial 70% window	Middle 10% window	Final 20% window
PaySim	`Step`	Steps 1–500	Steps 501–600	Steps 601–743
IEEE-CIS	`TransactionDT`	Months 1–4	Month 5	Month 6
XploreML	`Timestamp`	Initial 80% interactions	–	Final 20% OOT records

Implementation Rationale: By using the final window as a future test set, the model cannot rely on static node identities. It must leverage the Amygdala-inspired structural engrams and the neuro-symbolic collaborative spatial features to recognize generalized laundering topologies that persist across tactical shifts. All label construction and data-dependent preprocessing are restricted to information available within the corresponding chronological split, preventing any look-ahead from validation or test periods.

I Detailed Baseline Description

To strictly evaluate the performance of **LEAP-HGN**, we compare it against 12 state-of-the-art baselines. The detailed functional descriptions and underlying methodologies for each group are as follows:

I.1 Specialized AML

(a) **FlowScope** Li et al. (2020): It models money laundering detection as a max-flow problem on graphs, utilizing a greedy algorithm to mine dense bipartite sub-structures that characterize the aggregation and distribution of illicit funds.

(b) **SGNN** Song et al. (2025b): Designed for transactional blockchains, this method employs a signed graph neural network to capture the polarity of relationships (trust vs. distrust) to identify illicit social accounts.

(c) **MonLAD** Sun et al. (2022): Targeting streaming transaction data, MonLAD employs a dynamic memory architecture to capture both long-term and short-term temporal dependencies and neighborhood changes for real-time detection.

(d) **GARG-AML** Deprez et al. (2025): Specifically targeting smurfing patterns, this framework combines graph-based features with risk guarding rules to identify dense, small-amount transaction chains used to evade regulatory thresholds.

I.2 Graph-based Fraud Detection

(a) **TROPICAL** Haghighi et al. (2024): Tailored for camouflaged fraudster detection, this method utilizes a dual-channel hypergraph Transformer to aggregate intra- and inter-hyperedge correlations, capturing high-order implicit associations.

(b) **HOGRL** Zou & Cheng (2024): This approach introduces a high-order graph representation learning framework using motifs to capture complex structural patterns, overcoming the limitations of simple pairwise message passing.

(c) **HUGE** Pan et al. (2025): A label-free framework that addresses the heterophily problem in fraud detection by introducing a heterophily-guided metric to separate benign and anomalous patterns in unsupervised settings.

(d) **SpaceGNN** Dong et al. (2025): Focusing on scenarios with extremely limited labels, it proposes a multi-space graph neural network that projects nodes into distinct feature spaces to capture diverse anomaly patterns.

(e) **RGTAN** Xiang et al. (2025): This approach introduces a risk-aware graph tandem network that incorporates a relation-dependent routing function to dynamically weigh the risk contribution of neighbors.

I.3 LLM-based Methods

(a) **ARC** Liu et al. (2024): Designed as a generalist detector, it leverages in-context learning to reformat graph data into text, enabling LLMs to identify structural anomalies via analogical reasoning.

(b) **EdgePrompt** Fu et al. (2025): Addressing the gap between pre-training and downstream tasks, this method applies prompt tuning specifically to graph neural networks, optimizing edge-level prompts for specific anomaly detection tasks.

(c) **Fraud-R1** Yang et al. (2025): A benchmark and framework designed to assess the robustness of LLMs against augmented fraud and phishing inducements, providing insights into their semantic reasoning capabilities in fraud contexts.

J Prompt Engineering and Neuro-Symbolic Synthesis

The Self-Evolutionary Rule Closed-Loop (SRCL) uses the LLM as a constrained semantic reasoning component rather than as an unconstrained decision maker. Its role is to translate the aggregated structural and behavioral characteristics of hard false-negative samples into candidate symbolic rules that can be subsequently validated and incorporated into the Amygdala warning mechanism. The detailed construction of the hard-sample error profile and the rule validation procedure are provided in Appendix F.

To reduce stochastic and structurally unsupported rule generation, the prompt interface imposes explicit constraints on the admissible functional form, feature selection, structural–behavioral balance, and output structure. The same constrained interface is applied across the LLM backbones considered in our comparative analysis, while the corresponding backbone-dependent evolutionary behavior is discussed in Appendix N. The complete prompt specification used for rule synthesis is summarized in Table 9.

Table 9: The evidence-driven prompt template for rule self-evolution. The System Prompt constrains the generation of smooth and bounded symbolic rules, while the User Prompt adaptively injects the error profile (z_{profile}) derived from false-negative hard-sample mining.

System Prompt: Meta-Cognitive Role Formalization

Instructional Identity: You are a Neuro-Symbolic AML Rule Architect tasked with translating high-dimensional statistical anomalies into human-interpretable, smooth, and bounded symbolic logic.

[CORE OPERATIONAL CONSTRAINTS]

(a) **Smooth Risk Mapping:** All thresholding logic MUST utilize the `sigmoid((x - τ) * k)` operator. The sigmoid formulation replaces brittle binary thresholding with a continuous and bounded risk response in $(0, 1)$, allowing the generated rule to serve as a stable neuromodulatory prior during the subsequent neural optimization cycle. The generated symbolic parameters remain fixed during that cycle and are not updated by backpropagation.

(b) **Heuristic Sensitivity:** When addressing high-risk false negatives, adjust the threshold parameters (τ) for informative structural descriptors such as `cyc` or `indeg_max` according to the observed error profile, so that previously under-detected structural patterns can receive stronger risk responses without introducing unconstrained rule changes.

(c) **Cognitive Weighting:** Maintain the structural bias α within the predefined range $[0.5, 0.7]$ to balance structural and behavioral evidence while preserving a topology-oriented inductive bias.

[SCHEMA ENFORCEMENT]

Synthesize exactly 6 configuration parameters strictly adhering to the `rule_schema.json` protocol:

1. `Family`
2. `Alpha`
3. `Prior`
4. `Weight_Components`
5. `Logic_Expression`
6. `Parameter_List`

User Prompt: Adaptive Contextual Injection

[DIAGNOSTIC CONTEXT]

False-Negative Hard-Sample Mining Mode: The current GNN state exhibits a detection gap in which laundering-related structures remain under-estimated amid high-frequency background transactions.

[ERROR PROFILE (z_{profile})]

The following aggregated statistics summarize the structural and behavioral characteristics of the currently missed laundering subgraphs:

(a) **Mean Cycle Strength** (μ_{cyc}): `{cyc_mean}`.

(b) **Mean Temporal Span** (μ_{time}): `{time_span_h_mean}`.

(c) **Mean Financial Volume** (μ_{money}): `{total_money_mean}`.

[ADAPTIVE LOGIC CONSTRAINTS]

(a) **Condition ($z_{\text{cyc}} > 0.01$):** The current error profile contains a non-negligible cyclic component. You MUST incorporate `cyc` through a sigmoid-based expression so that cyclic topology contributes explicitly to the structural risk score.

(b) **Condition ($z_{\text{cyc}} \approx 0$):** The current error profile contains little cyclic evidence. Mask `cyc` and instead construct the rule from informative descriptors such as `edge_size` and `total_money`.

Infer a candidate set of symbolic parameters that addresses the observed error profile while preserving structural generalization and satisfying the predefined rule constraints.

J.1 Prompt Design Rationale and Theoretical Foundation

The prompt design is guided by three principles that restrict the LLM to a controlled symbolic search space and support stable alternating symbolic–neural evolution.

(a) **Smooth Semantic Grounding.** Hard thresholding can cause abrupt changes in the rule-derived risk prior. We therefore constrain threshold-based rule generation to a smooth, bounded mapping. For a selected hyperedge descriptor f_e, the basic response takes the form

$$S_{\text{new}}(e) = \sigma(k\,(f_e - \tau_{\text{new}}))\,, \tag{61}$$

where τ_{new} is the synthesized threshold and $k > 0$ controls transition sharpness. equation 61 maps the rule response to $(0, 1)$, allowing the generated symbolic score to act as a bounded prior rather than as a brittle binary decision.

The accepted symbolic parameters remain fixed during the subsequent neural optimization cycle. Their detailed validation and integration into the Amygdala warning mechanism are described in Appendix F.

(b) **Adaptive Evidence Selection.** The prompt does not require the same structural descriptors for every evolutionary cycle. Instead, feature selection is conditioned on the aggregated error profile of the currently missed laundering structures. For example, sufficiently strong cyclic evidence activates a cycle-sensitive component, whereas weak cyclic evidence redirects rule synthesis toward alternative supported descriptors such as hyperedge scale and transaction volume.

This evidence-dependent selection reduces the likelihood that the LLM introduces a structural variable that is not supported by the current hard-sample statistics. Consequently, the symbolic update is driven by observed model failures rather than by unconstrained semantic generation.

(c) **Heuristic Anchor Reinforcement.** Rule synthesis is further restricted by predefined parameter ranges and admissible structural families. In particular, constraining the balance coefficient α_r limits arbitrary shifts between structural and behavioral evidence. The LLM therefore adapts thresholds, feature combinations, and calibration strength within a controlled rule family rather than redefining the entire decision mechanism at each evolutionary cycle.

These anchors preserve reusable laundering-related structural patterns, including cyclic fund flow, fan-in aggregation, and fan-out dispersion, while still allowing the symbolic rule base to adapt to newly observed false-negative patterns.

J.2 Alternating Symbolic–Neural Interaction

The prompt-generated output is represented as a structured candidate rule

$$\mathcal{R}_{\text{new}}^{(c)} = \left\langle \mathcal{F}, \alpha_r, \mathbf{W}_{\text{beh}}, \Phi_{\text{logic}}\left(\cdot; \Theta_{\text{new}}^{(c)}\right), c_{\text{prior}} \right\rangle, \tag{62}$$

where $\mathcal{F}$ denotes the selected structural rule family, α_r balances structural and behavioral evidence, $\mathbf{W}_{\text{beh}}$ contains the behavioral weighting coefficients, Φ_{logic} is the bounded symbolic risk mapping, $\Theta_{\text{new}}^{(c)}$ contains its generated parameters, and c_{prior} is the associated calibration factor.

The prompt itself does not determine whether a candidate rule becomes active. Each generated candidate is first checked for structural validity and evaluated by the sandbox-based screening procedure described in Appendix F. Only validated rules are incorporated into the active rule set.

During the subsequent neural optimization cycle, the accepted symbolic parameters are treated as fixed quantities rather than trainable neural parameters. Their rule-derived warning scores modulate the Parietal representation through the Amygdala gain mechanism, while gradient-based optimization continues to update only the neural parameters. At the end of the cycle, newly identified hard false-negative samples are aggregated into a new error profile, from which the next candidate rule may be synthesized.

Accordingly, the complete interaction alternates between neural optimization, hard-sample error profiling, constrained LLM-based rule synthesis, rule validation, and fixed-prior modulation. This separation allows the symbolic rule base to evolve between cycles while preserving conventional gradient-based learning within each neural optimization cycle. The complete mathematical formulation of this alternating process is given in Appendix F.

J.3 Computational Efficiency of Prompt-Based Evolution

LLM inference is separated from the GPU-intensive neural optimization process. Rather than querying an LLM for individual accounts, hyperedges, or gradient updates, LEAP-HGN aggregates

the hard-sample statistics at the boundary of an evolutionary cycle and performs rule synthesis only after the corresponding error profile has been constructed.

This periodic invocation substantially reduces the computational and latency overhead associated with LLM reasoning. Once a candidate rule has passed the validation procedure, its symbolic parameters remain fixed throughout the subsequent neural cycle, so ordinary GNN optimization does not require repeated LLM inference.

Under the experimental configuration used in this work, each evolutionary step incurs approximately 12–18 seconds of LLM inference latency, and the overall LLM-related overhead accounts for approximately 3.8%–4.5% of the total wall-clock training time. The corresponding LLM configuration and efficiency measurements are reported in Appendix H.3.

K Complexity and Efficiency Analysis

In this section, we theoretically analyze the space and time complexity of LEAP-HGN. As illustrated in Figure 4 and Figure 5, our framework maintains linear scalability comparable to standard sparse GNNs, overcoming the computational bottlenecks typically associated with LLM integration.

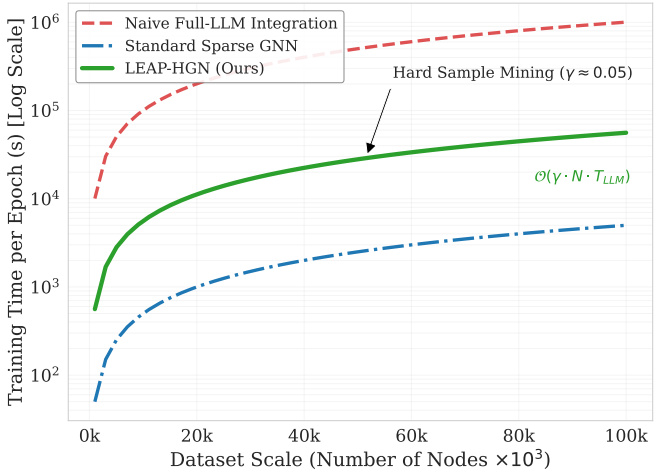


Figure 4: **Efficiency Analysis of LEAP-HGN (Time). Time Scalability:** The framework avoids the exponential cost of naive LLM integration through hard negative mining and periodic rule evolution, maintaining near-linear growth as the dataset scale increases.

K.1 Space Complexity Analysis

The memory footprint is dominated by the spatio-temporal hypergraph structures and feature tensors. Let $\mathcal{M} = \{AB, BP, TA, LS\}$ be the view-index set and $\bar{D}_{e,m}$ the mean cardinality of a hyperedge in view m. Unlike dense adjacency storage ($\mathcal{O}(|V|^2)$), LEAP-HGN uses sparse incidence matrices and a global hash cache. The total space complexity is

$$S_{total} \approx \mathcal{O}\left(|V|d + \sum_{m \in \mathcal{M}} |\mathcal{E}_m| \bar{D}_{e,m} + S_{cache} \right), \tag{63}$$

(a) **Sparse Storage:** The term $\sum_{m \in \mathcal{M}} |\mathcal{E}_m| \bar{D}_{e,m}$ counts incidence nonzeros across the four views. By restricting the local window Δt, $|\mathcal{E}_{TA}|$ grows with observed cascades rather than with all possible transaction combinations.

(b) **Global Cache Optimization:** The `GLOBAL_EDGE_CACHE` replaces raw transaction storage ($\mathcal{O}(|\mathcal{E}| \cdot L_{raw})$) with compact statistical feature vectors ($\mathcal{O}(|\mathcal{E}| \cdot d_{feat})$), where $d_{feat} \ll L_{raw}$, ensuring lightweight operation even with millions of nodes.

K.2 Time Complexity Analysis

The computational cost consists of neural hypergraph propagation, sparse Markov diffusion, and the periodic neuro-symbolic evolution step. Let L be the number of GNN layers, K_{diff} the number of diffusion iterations, and E_c the number of neural epochs per evolutionary cycle. Amortized over the E_c epochs, the average per-epoch cost is

$$\bar{T}_{epoch} \approx \mathcal{O}\left(L(|V| + |\mathcal{E}_{total}|)d^2 + K_{diff}|\mathcal{E}_{total}|d + \frac{|V_{hard}|T_{profile} + T_{LLM}}{E_c} \right), \tag{64}$$

(a) **Neural Projection and Sparse Diffusion:** Feature transformations contribute the d^2 term, whereas multiplying a sparse node-hyperedge-node transition by a d-dimensional state contributes $\mathcal{O}(|\mathcal{E}_{total}|d)$ per diffusion iteration. Keeping these terms separate avoids attributing a quadratic feature cost to the parameter-free Markov step.

(b) **Periodic Evolution Cost:** Hard-negative profiling and the LLM call occur once at the boundary of an evolutionary cycle, not during every ordinary neural epoch. Their unamortized boundary cost is $\mathcal{O}(|V_{hard}|T_{profile} + T_{LLM})$; division by E_c gives their average contribution per epoch. Because the hard samples are aggregated into one profile, the system makes one LLM request per cycle rather than one request per node.

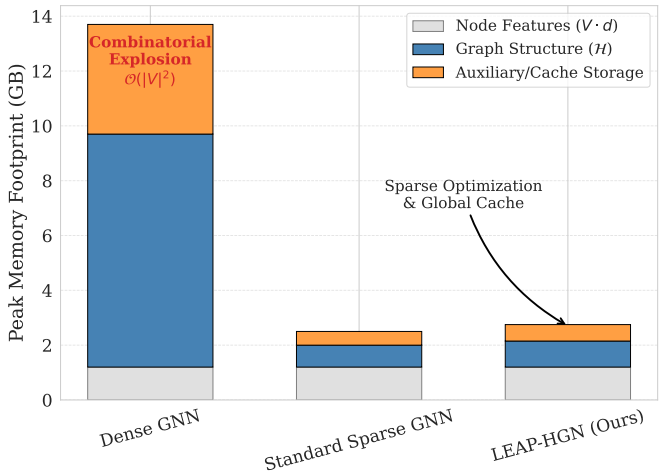


Figure 5: **Efficiency Analysis of LEAP-HGN (Space). Space Efficiency:** Memory consumption is minimized through sparse incidence matrix storage and the global hash cache mechanism, significantly reducing overhead compared to dense graph methods.

L Reproducibility and Responsible Use

This section summarizes the information provided to support the independent implementation and evaluation of LEAP-HGN. Detailed descriptions of the model components, experimental configuration, evaluation protocol, and LLM integration procedure are provided throughout the appendix.

L.1 Code Availability

An anonymized implementation is available at `https://anonymous.4open.science/r/LEAP-9AEF/README.md`.

L.2 Model and Experimental Configuration

The construction of the hypergraph views and the definitions of the risk indicators are described in Appendices B and C. The detailed formulation of the Parietal-collaborative high-order learning mechanism is provided in Appendix D, while the complete algorithmic procedure and the self-evolutionary rule closed-loop are presented in Appendices E and F.

Dataset statistics, feature schemas, and dataset-specific characteristics are documented in Appendix G. The hyperparameters used in the reported experiments, including the representation dimensions, number of GNN layers, optimizer, learning rate, loss-function parameters, evolutionary schedule, and early-stopping configuration, are provided in Appendix H.2. Detailed descriptions of the comparison methods are provided in Appendix I.

L.3 Out-of-Time Evaluation Protocol

To evaluate the ability of LEAP-HGN to generalize to future transaction patterns, we adopt a chronological out-of-time (OOT) evaluation protocol instead of randomly shuffling transactions across the training and test sets. Transactions are ordered according to their corresponding temporal attributes and are then divided into training, validation, and test windows.

The test window contains transactions occurring after those used for training and validation, thereby providing an evaluation setting that reflects temporal distribution shifts and evolving money-laundering patterns. The dataset-specific temporal attributes and partitioning boundaries are reported in Appendix H.4.

L.4 LLM Integration and Prompt Constraints

The LLM serves as the semantic reasoning component of the self-evolutionary rule closed-loop. At each rule-evolution stage, the framework summarizes the structural and behavioral characteristics of hard negative samples into an aggregated error profile. The LLM receives this profile and generates symbolic rules according to a predefined output format and a set of rule-generation constraints.

The generated rules are used to introduce adaptive symbolic information into the subsequent optimization process. Multiple LLM backbones are evaluated to examine whether the proposed rule-evolution mechanism remains effective across different reasoning models. The detailed prompt template, input variables, expected output structure, and rule-generation constraints are provided in Appendix J.

L.5 Data Privacy and Responsible Use

This work studies anti-money laundering as a defensive financial security application. The experiments use the synthetic or anonymized financial-transaction datasets described in Appendix G. The study does not involve crowdsourcing or experiments with human participants, and it does not introduce a new dataset containing personal financial information.

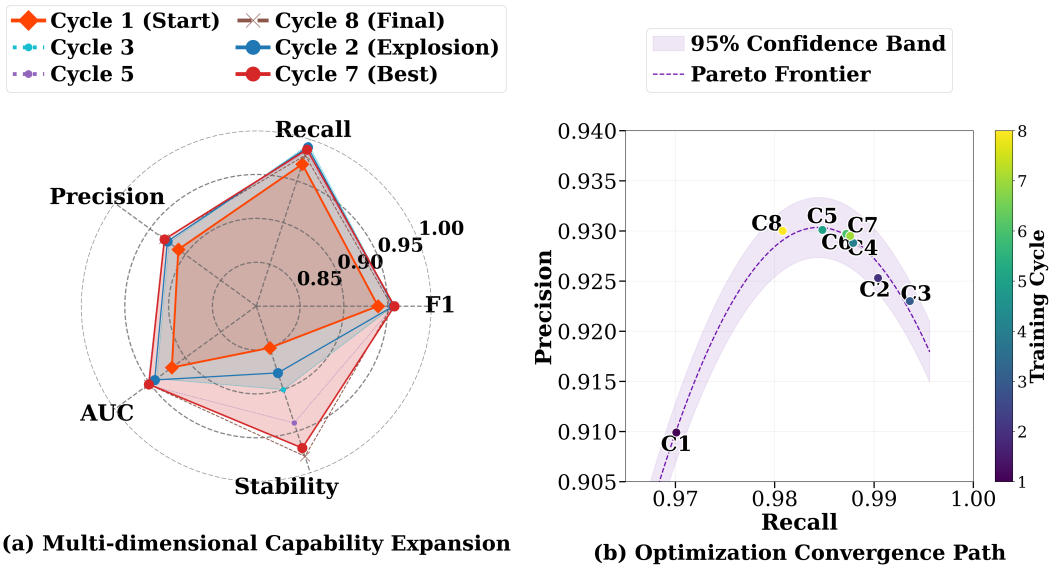
Like other automated financial-risk detection methods, LEAP-HGN may produce false-positive or false-negative predictions and may be affected by biases or distribution shifts in the underlying data. Therefore, the proposed method is intended to support risk analysis rather than to serve as the sole basis for punitive decisions or account restrictions. In practical applications, model predictions should be examined together with domain knowledge, institutional policies, and appropriate human review.

M Iterative Optimization Analysis for Neuro-Symbolic Closed-Loop

To provide a deeper understanding of the Self-evolutionary Rule Closed-Loop system mentioned in the main text, this section validates its dynamic adaptability by tracking the performance trajectory of the LEAP-HGN model across iterative optimization cycles. The evolutionary dynamics, visualized in Figure 6, demonstrate how the synergistic interaction between the Amygdala and Parietal modules enhances the detection of concealed money laundering patterns.

As illustrated in Figure 6(a), the holographic radar chart depicts the multi-dimensional expansion of the model capability envelope. The transition from the baseline Cycle 1 (gray area) to Cycle 2 (blue line) exhibits a rapid performance leap, confirming that the injection of LLM-generated prior rules into the Amygdala module successfully amplifies sparse risk signals that are previously diluted in the latent space. Subsequent iterations refine this performance through hard negative mining, culminating in the optimal state at Cycle 7 (red shaded area). This progression verifies that the neuro-symbolic integration effectively bridges the semantic gap without compromising structural generalization.

Concurrently, the optimization path shown in Figure 6(b) demonstrates a robust convergence towards the theoretical Pareto frontier. The trajectory from Cycle 3 to Cycle 7 maintains high stability along the upper confidence band, indicating that the Parietal module successfully synthesizes local risk warnings with global spatial reasoning. Notably, the slight performance retraction observed at Cycle 8 serves as empirical evidence for the onset of overfitting, thereby validating the sensitivity of the early stopping protocol in preventing the model from memorizing noise during the continuous evolution process.



(a) Multi-dimensional Capability Expansion **(b) Optimization Convergence Path**

Figure 6: **Iterative Optimization Analysis of the Neuro-Symbolic System.** (a) The holographic radar chart visualizes the expansion of the cognitive capability. (b) The optimization trajectory demonstrates an asymptotic approach to the Pareto frontier, with Cycle 7 marking the global optimum.

N Dynamic Evolution Analysis of Decision Boundaries

Building upon the optimization analysis, we further elucidate the strategy by which LLMs bridge the semantic gap in anti-money laundering. We perform a comparative convergence analysis focusing on the dynamic evolution of risk decision boundaries driven by different LLM backbones, as illustrated in Figure 7.

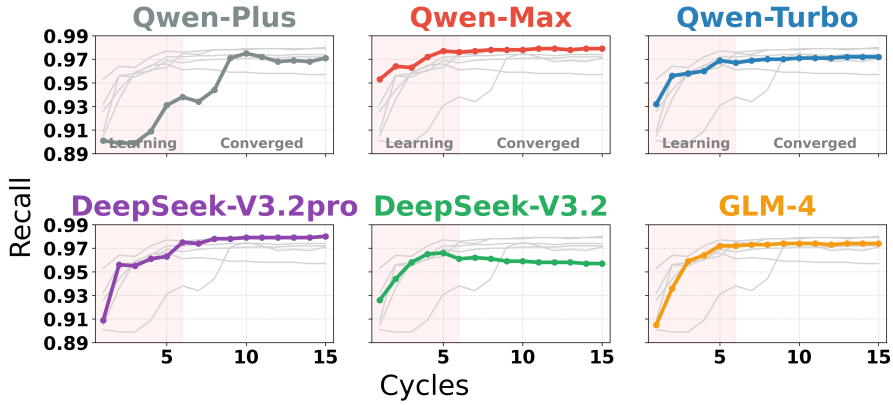
(1) Optimization Efficiency (Figure 7(b)): The results highlight reasoning capability as a dominant determinant of optimization efficiency. High-reasoning models such as Qwen-Max (red diamond line) and GLM-4 (orange star line) directly correlate with elevated convergence velocity in F1. This phenomenon demonstrates that advanced semantic reasoning effectively captures implicit laundering intent during the early stages of rule iteration, rapidly establishing a robust decision baseline without relying on extensive historical priors.

(2) Camouflaged Risk Detection (Figure 7(a)): In contrast, the Recall trajectories exhibit distinct distributions that validate the system's capability to detect camouflaged risk patterns through continuous adaptation.

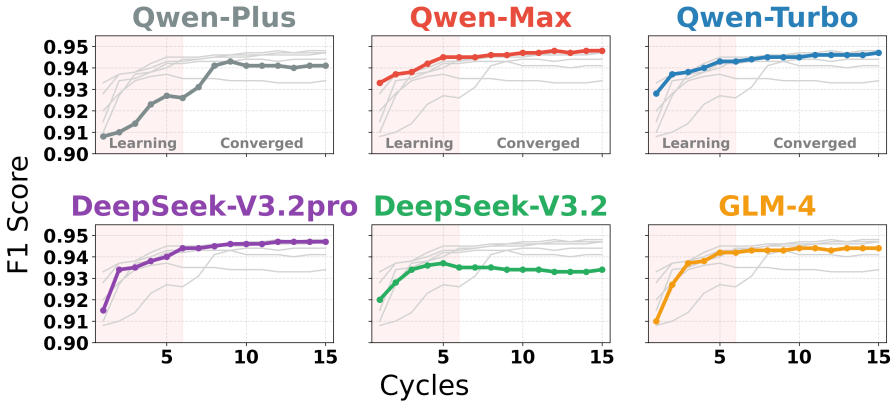
(3) Delayed Aggression: While Qwen-Plus (gray circle line) initially displays a conservative optimization path, it subsequently executes a delayed but aggressive strategy that aligns with the long-tail distribution of financial crimes.

(4) Dominant Signal: Notably, DeepSeek-V3.2pro (purple square line) sustains a dominant Recall signal throughout the process, overriding the standard precision-recall trade-off by interpreting high-risk subgraphs as deliberate structuring strategies.

This confirms that our framework effectively resolves the conflicts between benign topology and suspicious intent by integrating semantic reasoning with structural evolution, ensuring comprehensive risk coverage during the dynamic optimization process.



(a) Recall trajectories



(b) Evolution of F1

Figure 7: **Comparative analysis of model convergence across different LLMs.** The results illustrate the dynamic evolution of F1 and Recall trajectories, highlighting the correlation between reasoning capability and optimization efficiency.