Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Rule-to-Data Knowledge Transfer via Optimal Transport for Weakly Supervised Anomaly Detection on Transaction Graphs

**Qiuyang Zhang**
Fudan University
**Keyang Chen**
Fudan University
**Mingxuan Jiang**
Fudan University
**Yuan Shui**
UnionPay (China)
**Yandan Tan**
Southwest Jiaotong University
**Zhixin Li**
Fudan University
**Hongbin Zhu**
zhuhb@fudan.edu.cn
Fudan University
**Hongfeng Chai**
Fudan University

**Research Article**

**Keywords:** Graph anomaly detection, Money laundering detection, Optimal transport, Contrastive learning

**Posted Date:** June 29th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-10081676/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Rule-to-Data Knowledge Transfer via Optimal Transport for Weakly Supervised Anomaly Detection on Transaction Graphs

**Abstract**

Transaction-graph anomaly detection is challenging because obtaining high-quality labels requires costly expert investigation. Weakly supervised methods often rely on heuristic rules, expert knowledge, or limited labeled data to generate pseudo-labels for unlabeled transactions. However, existing methods typically reduce rules to flat labeling functions, overlooking their hierarchical structure and their mismatch with continuous transaction representations. This can result in noisy pseudo-labels and error propagation. To address these limitations, we propose a weakly supervised framework for rule-to-data knowledge transfer. Specifically, we train a decision tree on limited labeled data and transform its decision paths into rule graphs, which are encoded by a graph neural network (GNN) to capture hierarchical rule semantics. In parallel, a feature encoder learns representations from transaction graphs. We then employ optimal transport (OT) to align rule and data representations, enabling rule-guided pseudo-label generation for unlabeled samples. Furthermore, a contrastive learning objective refines pseudo-labels by promoting consistency among samples sharing similar rule semantics while improving class separability. Experiments on four benchmark datasets demonstrate that our proposed model consistently outperforms existing weakly supervised and supervised baselines, validating the effectiveness of hierarchical rule modeling and rule-data representation alignment.

**Keywords:** Graph anomaly detection, Money laundering detection, Optimal transport, Contrastive learning

## 1 Introduction

Graph anomaly detection plays a critical role in safeguarding the stability and reliability of complex networked systems. One representative example is anti-money laundering (AML), which remains a serious threat to the integrity of the global financial system [1]. The emergence of blockchain technologies and cryptocurrency transactions has further intensified this challenge [2]. In blockchain systems, transactions are grouped into timestamped blocks and linked using cryptographic hashes.

However, the pseudonymous nature of blockchain also makes it an attractive channel for cross-border transfers, illegal finance, online black markets, and scams. To address these issues, AML systems have gradually evolved from manual investigation and expert-defined rule engines to more sophisticated data-driven approaches [3].

Early detection systems [4] often adopted strategies from traditional banking, relying on predefined rules and traditional machine learning classifiers. Rule-based systems encode expert knowledge through "if-then" logic derived from handcrafted money-laundering patterns, such as transaction thresholds, frequency, abnormal transfer behaviors, rapid fund movements, or interactions with high-risk regions [5, 6]. While these systems are straightforward to deploy and interpret, they typically suffer from high false-positive rates and limited adaptability to emerging laundering strategies [7]. Traditional machine learning algorithms, especially tree-based models (*e.g.*, Random Forests), have been adopted for early-stage screening on smaller and well-structured datasets [8]. These models improve automation and predictive performance but remain heavily dependent on manually engineered features, limiting their ability to capture the structural and temporal characteristics of transaction networks.

Recent advances in graph-based deep learning have significantly improved AML detection. Approaches such as graph neural networks (GNNs) [9–11], Graph Convolutional Networks (GCNs) [12], GraphSAGE [13], and Transformers [14] can learn graph-structured transaction patterns. By exploiting both node attributes and network topology, these methods capture suspicious interaction patterns and classify nodes as illicit or licit based on their structural context. However, their effectiveness often depends on the availability of labeled data, which is scarce in real-world AML scenarios.

To address this limitation, weakly supervised methods leverage a small set of labeled transactions together with a large pool of unlabeled data. A common strategy is to use heuristic or expert-defined rules to generate noisy labels, which are then used to train predictive models [10, 15]. Such weakly supervised methods aim to extract maximum value from small labeled sets and embrace the reality of incomplete information [16]. In parallel, unsupervised anomaly detection methods [17] have also been explored to alleviate label scarcity by identifying deviations from normal transaction behavior [18]. However, these methods frequently suffer from noisy supervision or high false-positive rates, limiting their effectiveness in detecting evolving laundering activities.

Despite recent progress, weakly supervised AML methods still face fundamental challenges when leveraging rules to supervise unlabeled transactions. First, rules extracted from limited labeled transactions naturally form hierarchical decision paths, where shared conditions indicate related behavioral patterns. However, most existing methods reduce these rules to flat pseudo-labels, ignoring the structural dependencies embedded in the rule hierarchy. Second, rule representations are inherently symbolic, whereas transaction representations are learned as continuous data-driven embeddings. The semantic gap between these heterogeneous representations makes reliable knowledge transfer difficult. Third, rules derived from limited and imbalanced labeled samples may generate uncertain pseudo-labels for previously unseen transaction patterns. Directly treating such pseudo-labels as reliable supervision can propagate errors

throughout the learning process. Therefore, the key challenge lies in how to preserve the structural semantics of rule knowledge, effectively transfer this knowledge to data-driven representations, and mitigate the influence of noisy pseudo-labels during training.

Motivated by these challenges, we focus on bridging the gap between rule-based knowledge and data-driven representations while limiting the propagation of rule-induced noise under limited supervision. To this end, we propose a unified weakly supervised framework consisting of a rule branch and a data branch. Specifically, hierarchical decision-tree paths are transformed into a rule graph and encoded by a GNN to capture hierarchical dependencies among rules. We then employ optimal transport (OT) [19] to softly align rule prototypes with transaction representations and generate pseudo-labels for unlabeled transactions. Finally, cross-domain contrastive learning (CCL) [20] leverages OT-induced correspondences to enhance rule-data consistency and improve pseudo-label quality. Together, these components enable reliable rule-to-data knowledge transfer under limited supervision, class imbalance, and distribution shifts.

Compared with existing weakly supervised approaches [1], our framework explicitly models hierarchical rule structures and establishes semantic alignment between rule and data representations. This design improves robustness under class imbalance and distribution shifts, making it well suited for real-world AML scenarios. The contributions of this paper are as follows:

- We propose a unified weakly supervised framework that structurally represents hierarchical rule knowledge and jointly learns rule-based and data-driven representations through a dual-branch architecture.
- We develop a rule-to-data knowledge transfer mechanism that combines OT and contrastive learning to align heterogeneous representation spaces and generate reliable pseudo-labels for unlabeled transactions.
- Extensive experiments on four benchmarks demonstrate the effectiveness, robustness, and generalizability of the proposed framework across diverse domains and graph structures.

# 2 Related Work

## 2.1 Money Laundering Detection

AML research spans three learning paradigms. **Supervised methods** range from classical algorithms (logistic regression, SVMs, decision trees) applied to hand-crafted transaction features [4, 6] to graph-based deep learning, where GNNs [11] such as GCN [12] and GraphSAGE [13] exploit the topological structure of transaction networks to capture connectivity patterns indicative of suspicious behavior. However, these methods depend heavily on high-quality labeled data, which is costly to obtain, and they struggle with class imbalance and novel laundering strategies. **Weakly supervised methods** address label scarcity by combining a small labeled set with a large unlabeled pool: common strategies include generating noisy labels from expert-defined rules [10], propagating fraud scores through transaction graphs via label

propagation [21], and applying active learning to iteratively select the most informative samples [16]. **Unsupervised methods** treat AML as anomaly detection [17], using clustering and outlier detection [22] or deep autoencoders [18] to flag deviations from normal behavior without labeled ground truth. While these approaches avoid label dependency, they suffer from high false positive rates.

In this paper, we adopt a weakly supervised approach that integrates expert knowledge via limited labeled data and domain-specific rules, uses contrastive learning to propagate supervision to unlabeled nodes, and supports cross-domain adaptation for robust AML detection across institutions.

### 2.2 Optimal Transport

OT provides a principled framework for mapping one probability distribution to another at minimum cost, offering both a global distributional discrepancy metric (e.g., Wasserstein distance) and fine-grained sample-level correspondences [23]. Differentiable solvers such as Sinkhorn iterations [24] have made OT practical for representation learning and domain adaptation. In the financial security domain, OT has demonstrated strong potential for aligning heterogeneous transaction patterns across institutions and data modalities, and for bridging rule-based AML typologies with novel transaction behaviors observed in cryptocurrency networks [19, 25]. In our framework, OT aligns rule-derived and data-driven embeddings, enabling structured knowledge transfer from expert typologies to unseen laundering patterns.

### 2.3 Contrastive Learning

Contrastive learning (CL) learns effective representations by pulling together positive pairs and pushing apart negative ones, typically without requiring labels [26, 27]. Graph-based extensions apply node- or edge-level perturbations to create contrastive views, improving performance in recommendation [28] and anomaly detection [29] settings. In fraud detection, CL helps produce robust representations and mitigate class imbalance by exploiting structural and temporal patterns in transaction graphs.

In our framework, we construct positive pairs by aligning rule-derived features with data-driven representations via OT. Minimizing the distance between semantically consistent pairs and maximizing separation between unrelated ones yields a feature space that captures both domain-specific fraud patterns and generalizable laundering behaviors, improving pseudo-label quality and facilitating cross-domain adaptation.

## 3 Method

### 3.1 Overview

***Preliminary.***

Given a directed transaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the node set $\mathcal{V} \in \mathbb{R}^{N_v \times (d_l + d_n)}$ represents a collection of $N_v$ transactions (*e.g.,* Bitcoin). Each node is encoded by a feature vector that combines two types of information: i) $d_l$-dimensional local attributes (*e.g.,* transaction time, transaction amount, or account metadata) and ii) $d_n$-dimensional

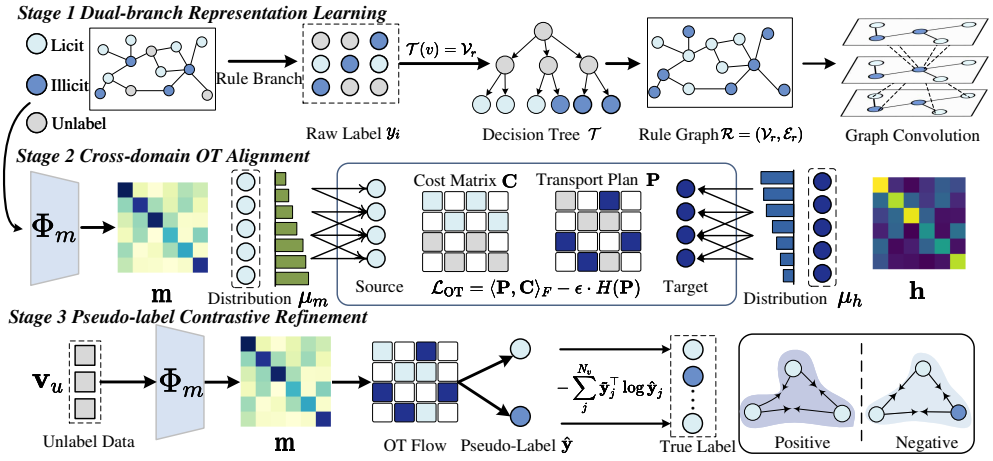



**Fig. 1** Overview of our proposed framework. *Stage 1*: Transactions are processed through two branches: the rule branch constructs a rule-based system from limited labeled data and refines its representation using a GNN, while the data branch encodes raw transaction features via an MLP. *Stage 2*: OT is employed to align embeddings from the rule space and data space within a shared semantic space, establishing soft correspondences between rule-derived features and data representations. *Stage 3*: Based on OT correspondences, pseudo-labels are assigned to unlabeled samples. CCL is then applied to minimize the distance between semantically consistent embeddings while maximizing the separation across different pseudo-label classes.

neighbor-based vectors encoded in a one-hot format, reflecting the local structural context of the transaction. The edge set $\mathcal{E} \in \mathbb{R}^{N_e \times 2}$ represents directed cryptocurrency flows between transactions, where each edge indicates the transfer of value from one transaction to another. Every node $\mathbf{v}_i$ in $\mathcal{G}$ is associated with a label $\mathbf{y}_i$ categorized as "Illicit", "Licit", and "Unknown".

Our objective is to train a model $\mathcal{F}$ that infers the labels of unknown transactions by distinguishing between illicit and licit behavior. To achieve this, we introduce a weakly supervised learning framework with dual-branch models and cross-domain contrastive learning. In particular, we employ OT to align rule-based representations with data representations, and further apply a contrastive learning loss to enhance the alignment between embeddings with similar features. This design enhances the discriminability of learned features and improves the model's ability to generalize across domains, enabling robust AML detection in heterogeneous financial environments.

### *Model Architecture.*

Weakly supervised anomaly detection on transaction graphs requires transferring rule knowledge from a small labeled set to a large unlabeled pool. This transfer is non-trivial for three reasons. First, decision-tree rules are not independent labeling functions; they are organized as hierarchical paths, where shared conditions indicate related behavioral patterns. Treating these rules as flat pseudo-labels discards such structural dependencies. This motivates the rule branch, which converts decision paths into a rule graph so that rule semantics can be propagated through graph message passing. Second, rule

knowledge is symbolic and discrete, whereas transaction representations are continuous and data-driven. Directly imposing rule labels on transaction embeddings therefore assumes an unreliable semantic equivalence between two heterogeneous spaces. This motivates the OT alignment module, which establishes soft correspondences between rule prototypes and transaction embeddings. Third, pseudo-labels induced by limited and imbalanced rules are inevitably noisy. Directly using them as hard supervision may amplify errors during training. This motivates the contrastive refinement module, which uses high-confidence OT correspondences to pull semantically consistent pairs closer and separate inconsistent ones.

Our framework, illustrated in Fig. 1, follows this challenge-to-module design in three stages. The first stage, **Dual-branch Representation Learning**, preserves hierarchical rule semantics through a rule branch while learning continuous transaction representations through a data branch. The second stage, **Rule-Data OT Alignment**, bridges the heterogeneous rule and data representation spaces by computing soft transport correspondences. The third stage, **Pseudo-label Contrastive Refinement**, further improves pseudo-label reliability by reinforcing semantically consistent rule-data pairs and separating inconsistent ones in the embedding space.

## 3.2 Rule-Branch Encoding

The rule branch is designed to preserve the structural semantics of decision rules rather than using their leaf predictions as isolated pseudo-labels. In a decision tree, transactions reaching nearby leaves often share partial decision paths, and these shared conditions reflect related behavioral patterns. A flat pseudo-label representation cannot capture such path-level dependencies. Therefore, we transform decision-tree paths into a rule graph, where transactions and decision conditions are connected according to their shared rule structures. This graph representation allows GNN-based message passing to aggregate rule-consistent information before producing rule prototypes for subsequent OT alignment. The construction proceeds in three steps.

**Step 1: Decision Tree Construction.** Given a set of labeled transaction nodes $\{\mathbf{v}_i, y_i\}$ where $y_i \in \{\text{illicit}, \text{licit}\}$, we train a decision tree $\mathcal{T}$ using the Gini impurity criterion [30] to recursively partition the feature space. At each internal node, the feature and threshold that maximally separate the two classes are selected, forming conditions such as "*if the transaction amount exceeds $\eta$, follow the right branch.*" Splitting continues until a stopping criterion is reached (*e.g.*, maximum depth or minimum sample count per node), and each leaf is assigned the majority-class label of its contained transactions. This yields a set of hierarchical rule paths that encode interpretable decision logic over transaction features.

**Step 2: Tree Inference on Unlabeled Data.** For each unlabeled transaction $\mathbf{v}_u$, we perform a root-to-leaf traversal of $\mathcal{T}$, evaluating feature conditions at each internal node until a leaf is reached. The leaf's majority-class label is assigned as the predicted label $\hat{y}_u$. This step propagates the structured rule knowledge of $\mathcal{T}$ to the full unlabeled transaction set, producing a corpus of weakly supervised annotations.

**Step 3: Rule Graph Construction.** After labeling all transactions, we construct the Rule Graph $\mathcal{R} = (\mathcal{V}_r, \mathcal{E}_r)$ as follows. The node set $\mathcal{V}_r$ comprises two types of nodes: *transaction nodes*, where each $\mathbf{r}_i \in \mathcal{V}_r$ corresponds to a transaction characterized

by its feature vector and its predicted (or ground-truth) label; and *decision nodes*, representing the internal splitting conditions of $\mathcal{T}$, which anchor the logical structure of the graph. The edge set $\mathcal{E}_r$ is defined by the decision paths of $\mathcal{T}$: an edge $e_{ij} \in \mathcal{E}_r$ is placed between $\mathbf{r}_i$ and $\mathbf{r}_j$ if they share a common decision path in the tree, *i.e.*, one node's traversal passes through the same sequence of splitting conditions as the other's. This construction explicitly encodes the logical co-occurrence and hierarchical dependency between transactions as induced by the rule system.

The resulting Rule Graph $\mathcal{R}$ provides a structured, interpretable representation of the decision process. Crucially, by recasting discrete tree logic into a graph topology, it enables GNN-based message passing to capture higher-order relational patterns among rule-sharing transactions, something neither the raw decision tree nor a flat feature representation can achieve.

***Graph Convolution.***

Once the rule graph is constructed, graph convolution is used to turn the static rule structure into rule-aware representations. The motivation is that transactions sharing decision conditions should not be treated as isolated pseudo-labeled samples; instead, their shared rule context should be aggregated before forming class-level rule prototypes. For each node in the rule graph, the convolution process begins by aggregating the features of its neighboring nodes. These features are combined through a weighted sum, followed by a non-linear activation function to improve the expressiveness of the node representations. Formally, for a graph convolution network with $L$ layers, the aggregation at the $l$-th layer is defined as:

$$\mathbf{r}_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}(\mathbf{r}_i)} \alpha_{ij}^{(l)} \mathbf{w}^{(l)} \mathbf{r}_j^{(l-1)}\right), \tag{1}$$

where $\alpha_{i,j}^{(l)}$ represents the attention coefficient between node $\mathbf{r}_i$ and its neighbor $\mathbf{r}_j$ at layer $l$, $\mathbf{w}^{(l)}$ denotes the weight matrix, and $\sigma$ denotes the activation function, allowing the rule graph to capture multi-hop dependencies and complex interactions between transactions. The attention coefficients $\alpha_{ij}^{(l)}$ are computed as:

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\sigma\left(\mathbf{a}^\top \left[\mathbf{w}^{(l)}\mathbf{r}_i^{(l-1)}, \mathbf{w}^{(l)}\mathbf{r}_j^{(l-1)}\right]\right)\right)}{\sum_{k \in \mathcal{N}(\mathbf{r}_i)} \exp\left(\sigma\left(\mathbf{a}^\top \left[\mathbf{w}^{(l)}\mathbf{r}_i^{(l-1)}, \mathbf{w}^{(l)}\mathbf{r}_k^{(l-1)}\right]\right)\right)}, \tag{2}$$

where $\mathbf{a}$ is the learnable attention vector, $\mathcal{N}(\mathbf{r}_i)$ denotes the neighbors of node $\mathbf{r}_i$, and $\sigma$ is the activation function (typically chosen as LeakyReLU). This attention mechanism enables the model to assign varying importance weights to neighbors, thereby improving the expressiveness of rule-guided representations.

After multiple layers of graph convolution, rule-branch transaction embeddings $\mathbf{z}_i \in \mathbb{R}^d$ are obtained for transaction nodes in the rule graph. We then aggregate these embeddings by their rule-induced labels to form two class-level rule prototypes, $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2\}$, corresponding to illicit and licit transactions. These prototypes

are subsequently used to guide the labeling of transactions via OT. By aggregating neighborhood information, graph convolution captures both local interactions and higher-order structural patterns within the rule graph. This process effectively integrates the hierarchical logic of the decision tree with the topological structure of the transaction network, producing enriched node embeddings that support more accurate and semantically meaningful classification.

***Data Branch Encoding.***

In parallel with the rule branch, we transform raw transaction data into dense embeddings using a Multi-Layer Perceptron (MLP), denoted as $\Phi_m$. Given the feature vector $\mathbf{v}_j$ of transaction $j$, its representation $\mathbf{m}_j \in \mathbb{R}^d$ in the data branch is computed as

$$\mathbf{m}_j = \Phi_m(\mathbf{v}_j). \tag{3}$$

The MLP consists of several fully connected layers with ReLU activation. The final projection layer maps the hidden representation into a latent embedding space of fixed dimension, which serves as the basis for subsequent alignment with rule representations and pseudo-label generation through OT.

## 3.3 Optimal Transportation

After the rule and data branches produce their respective embeddings, the central problem becomes how to transfer rule knowledge to transaction representations without assuming that the two spaces are naturally aligned. A hard assignment from rule labels to transaction embeddings would ignore the semantic gap between symbolic rule structures and continuous data-driven features. We therefore employ OT to establish soft correspondences between rule prototypes and transaction embeddings, so that rule-to-data transfer is guided by both pairwise similarity and global distributional consistency.

We utilize the Kantorovich formulation [31], which casts the OT problem as a linear optimization task. Specifically, given the rule prototype set $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2\}$ for the two known labels (illicit and licit), $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_v}\}$ denotes the transaction embeddings learned from the data branch. The cost of assigning transaction embedding $\mathbf{m}_j$ to rule prototype $\mathbf{h}_i$ is defined by a cost matrix $\mathbf{C} \in \mathbb{R}^{N_v \times 2}$, where $C_{ji} = c(\mathbf{m}_j, \mathbf{h}_i)$, and $c(\cdot)$ is a ground metric (*e.g.*, $L_2$ Euclidean distance). The goal is to find a transport plan $\mathbf{P} \in \mathbb{R}^{N_v \times 2}$ that minimizes the total cost, which is defined as:

$$\mathcal{W}(\mathbf{m}, \mathbf{h}) = \min_{\mathbf{P} \in \Pi(\mu_m, \mu_h)} \langle \mathbf{P}, \mathbf{C} \rangle_F \,, \tag{4}$$

Here, the constraint set is defined as:

$$\Pi(\mu_m, \mu_h) = \left\{ \mathbf{P} \in \mathbb{R}_+^{N_v \times 2} \middle| \mathbf{P}^\top \mathbf{1}_{N_v} = \mu_h, \mathbf{P}\mathbf{1}_2 = \mu_m \right\}, \tag{5}$$

It enforces marginal consistency between the transport plan $\mathbf{P}$ and the empirical distributions $\mu_m$ (over transaction embeddings) and $\mu_h$ (over rule prototypes). $\mathbf{1}_k$ denotes

a $k$-dimensional vector of ones. Intuitively, $\Pi(\mu_m, \mu_h)$ defines all admissible couplings that redistribute probability mass from data embeddings to rule embeddings while preserving total mass.

The Frobenius inner product $\langle \mathbf{P}, \mathbf{C} \rangle_F$ represents the total transportation cost, encouraging the plan to minimize local pairwise distances while achieving a globally optimal alignment. In this way, the OT problem balances local assignment costs with global marginal constraints. To ensure computational stability and faster convergence, we introduce an entropy regularization term $H(\mathbf{P})$, which is defined as:

$$H(\mathbf{P}) = -\sum_{i,j} \mathbf{P}_{ij} \log(\mathbf{P}_{ij}). \tag{6}$$

Building on this, the OT loss function integrates both the cost matrix $\mathbf{C}$ and the entropy regularization term, striking a balance between alignment precision and computational stability. The objective is defined as:

$$\mathcal{L}_{\mathrm{OT}} = \langle \mathbf{P}, \mathbf{C} \rangle_F - \epsilon \cdot H(\mathbf{P}), \tag{7}$$

where $\epsilon$ is a hyperparameter controlling the strength of the regularization term. The entropy regularization allows for a more stable computation of the transport matrix.

Once the OT plan $\mathbf{P}^*$ is obtained, it encodes the probabilistic correspondence between data embeddings $\mathbf{m}_j$ and rule prototypes $\mathbf{h}_i$. Each entry $\mathbf{P}^*_{ji}$ represents the assignment weight between the $j$-th transaction embedding and the $i$-th rule prototype, effectively measuring their semantic alignment. To derive pseudo-labels for unlabeled transactions, we normalize the transport plan across the rule dimension and interpret it as a soft assignment distribution over known rule types:

$$\tilde{\mathbf{y}}_j = \mathrm{softmax}\left(\frac{\mathbf{P}^*_{j,:}}{\tau}\right), \tag{8}$$

where $\tau$ is a temperature parameter controlling the sharpness of the assignment distribution. Each transaction embedding $\mathbf{m}_j$ is thus associated with a pseudo-label vector $\tilde{\mathbf{y}}_j$, representing its confidence-weighted membership with respect to the rule prototypes.

For downstream classification, these pseudo-labels are treated as supervisory signals and guide optimization via a cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{j=1}^{N_v} \tilde{\mathbf{y}}_j^\top \log \hat{\mathbf{y}}_j, \tag{9}$$

where $\hat{\mathbf{y}}_j$ denotes the model's predicted probability for transaction $j$. This OT-based pseudo-labeling mechanism enables the model to propagate semantic knowledge from rule-based priors to data-driven embeddings, facilitating label-efficient training, improving interpretability, and enhancing the detection of illicit transactions under limited supervision.

### 3.4 Cross-domain Contrastive Learning

Although OT provides soft correspondences between rule prototypes and transaction embeddings, these correspondences may still be uncertain under limited supervision and class imbalance. The key observation is that transactions strongly aligned with the same rule prototype should share similar semantic behavior, whereas transactions aligned with different rule prototypes should be separated in the embedding space. This makes contrastive refinement suitable for our task: it uses reliable OT-induced pairs to strengthen rule-data consistency while reducing the influence of noisy pseudo-label assignments. In this setting, contrastive learning serves as a refinement mechanism for OT-induced supervision rather than an independent pretext task.

Contrastive learning (CL) [26] is a self-supervised learning paradigm that learns effective data representations by pulling together similar sample pairs (positive pairs) and pushing apart dissimilar ones (negative pairs) in the embedding space. CL has shown effectiveness in tasks such as classification [32, 33], medical analysis [34], recommendation systems [28], and AML applications [29, 35].

***Positive and Negative Pairs.***

For each unlabeled transaction node $\mathbf{v}_u$, its embedding is obtained via the trained MLP $\Phi_m$, producing the representation $\mathbf{m}_u$. To assign pseudo-labels, we identify the rule type with the maximum OT weight $\mathbf{P}_{uj^*}$ in the transport matrix. For each row of $\mathbf{P}$, $j^*$ is selected as:

$$j^* = \arg\max_{j \in \{1,2\}} \mathbf{P}_{uj}. \tag{10}$$

To ensure the reliability of the pseudo-labels, we apply a dynamic threshold $\theta = \gamma \cdot \text{median}(\mathbf{P})$, where $\gamma$ serves as a scaling factor that adjusts the sensitivity of the threshold. If $\mathbf{P}_{uj^*} > \theta$, the pair $(\mathbf{m}_u, \mathbf{h}_{j^*})$ is treated as a positive pair. Conversely, if $\mathbf{P}_{uj^*} < \theta/2$, we randomly assign a mismatched label $k \neq j^*$ to form negative pairs $(\mathbf{m}_u, \mathbf{h}_k)$. These pairs are then optimized using a contrastive objective that maximizes the similarity of positive pairs while minimizing that of negative pairs. In this way, the model learns a robust embedding space that preserves rule-to-data semantic alignment and enhances cross-domain discriminability.

***Cross-domain Contrastive Learning.***

To further refine these pseudo-labels, a cross-domain contrastive learning (CCL) approach is employed to transfer label semantics from rule prototypes to transaction embeddings based on OT-induced correspondences. The contrastive learning loss is formulated as:

$$\mathcal{L}_{\text{CCL}} = -\log \sum_{(u,j)\in\mathcal{P}} \frac{\exp(\text{sim}(\mathbf{m}_u, \mathbf{h}_j)/\tau)}{\sum_{(u,k)\in\mathcal{P}\cup\mathcal{N}} \exp(\text{sim}(\mathbf{m}_u, \mathbf{h}_k)/\tau)}, \tag{11}$$

where $\text{sim}(\mathbf{m}_u, \mathbf{h}_j)$ measures the feature similarity between the anchor and the positive example, computed as $-\|\mathbf{m}_u - \mathbf{h}_j\|_2^2$. The parameter $\tau$ denotes the temperature that controls the sharpness of the similarity distribution, while $\mathcal{P}$ and $\mathcal{N}$ are the sets of positive and negative pairs in the batch, respectively.

This contrastive objective encourages the model to minimize the distance between representations of transactions with the same pseudo-label (positive pairs), while maximizing the distance between those with different pseudo-labels (negative pairs). By doing this, the CCL strategy sharpens the decision boundary between illicit and licit transactions, enhancing the semantic distinctiveness of pseudo-labels. As a result, the learned feature representations become more discriminative, improving both the reliability and accuracy of label predictions for previously unlabeled transactions. By combining OT with CCL refinement, our framework jointly leverages both graph-structured relational information and representation-level supervision, ensuring robust classification performance in the presence of limited labeled data and domain shifts.

### 3.5 Total loss

The training objective is designed to match the three requirements of weakly supervised rule-to-data transfer. The classification component provides direct predictive supervision, the alignment component transfers structured rule knowledge to transaction embeddings, and the contrastive component suppresses noisy pseudo-label effects by improving semantic separability.

The overall training objective integrates three complementary components: the OT loss $\mathcal{L}_{\mathrm{OT}}$, which aligns rule-derived embeddings with data-driven embeddings, the contrastive learning loss $\mathcal{L}_{\mathrm{CCL}}$, which enforces semantic consistency by maximizing separation between different classes, and the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$, which provides direct supervision for classifying transactions as licit or illicit. The total loss is defined as:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{OT}} + \beta \mathcal{L}_{\mathrm{CCL}}, \tag{12}$$

where $\lambda$ and $\beta$ are hyperparameters that control the relative importance of the alignment and contrastive objectives. This joint optimization strategy ensures that the model benefits from both rule-guided semantic alignment and discriminative embedding refinement.

## 4 Experiments

### 4.1 Datasets and Implementation Details

***Datasets.***

We evaluate our framework on four benchmark datasets covering diverse application domains, including cryptocurrency transactions, banking networks, social media, and e-commerce, with their statistics summarized in Table 1. These datasets exhibit substantial differences in graph structures, node semantics, feature distributions, and anomaly ratios, enabling a comprehensive evaluation of our framework across heterogeneous graph anomaly detection scenarios.

The **Elliptic** dataset [10] is a Bitcoin transaction graph with 203,769 nodes and 234,355 edges, where nodes represent transactions and edges indicate Bitcoin flows. Each node has 166 features comprising transaction-level statistics and one-hop aggregated information, and is labeled as licit or illicit. Only 9.8% of nodes are illicit. The

**Table 1** Statistics of the Reddit, Amazon, Elliptic, and Libra Bank datasets.

| Statistic | Elliptic | Libra Bank | Reddit | Amazon |
|---|---|---|---|---|
| Nodes | 203,769 | 385,100 | 10,984 | 11,944 |
| Edges | 234,355 | 597,165 | 168,016 | 4,398,392 |
| Features | 166 | 10 | 64 | 25 |
| Feature Type | Transaction attr. | Aggregated tx. stats | Text embedding | Behavioral |
| Avg. Degree | 1.15 | 1.55 | 15.30 | 368.25 |
| Classes | 2 | 2 | 2 | 2 |
| Snapshots | 49 | 1 | 1 | 1 |
| Anomaly Ratio | 9.8% | 0.16% | 3.3% | 9.5% |
| Domain | Cryptocurrency AML | Banking AML | Social Media Fraud | E-commerce Fraud |
| Type | Transaction Graph | Banking Transaction | Social Network | Product Co-purchase |
| Label Meaning | Illicit / Licit transaction | Fraudulent account | Subreddit category | Product category |

graph spans 49 time steps with no edges crossing temporal boundaries, capturing the dynamic evolution of transaction activity.

The **Libra Bank** dataset [36] is a real-world banking transaction network with 385,100 nodes and 597,165 edges, where nodes represent bank accounts and directed edges denote fund transfers between them. Node features consist of 10 aggregated transaction statistics, including transaction volume, cumulative transfer amounts, and the number of associated alerts and reports for both outgoing and incoming directions. A node is labeled fraudulent if it participates in any transaction flagged by at least one alert or report, yielding an extreme class imbalance of only 0.16% anomalous nodes.

Beyond financial transaction networks, we further evaluate the applicability of our framework on two widely adopted graph anomaly detection benchmarks from social media and e-commerce domains. The **Reddit** [37] dataset contains 10,984 nodes and 168,016 edges, where nodes represent posts and an edge is formed if the same user comments on both posts. The **Amazon** [38] dataset is a product co-purchasing network with 11,944 nodes and 4,398,392 edges, where nodes correspond to products and edges connect items frequently bought together.

Together, these datasets allow us to assess whether the proposed framework remains effective across financial, social-media, and e-commerce graphs with substantially different semantic and structural characteristics.

***Experimental Details.***

We adopted a two-stage training strategy. In the first stage, we trained a total of 10 decision tree classifiers with a maximum depth of 10 and a maximum feature set of 5, using the available labeled data to extract hierarchical transaction rules. These rules were then encoded into graph structures and processed by a GNN-based rule encoder. The encoder was trained with a learning rate of 0.001, a hidden dimension of 128, and a weight decay of 0.001. The total loss function includes both a classification loss and a regularization OT loss, each weighted by 1.0 to balance accuracy and generalization.

For the main model, we conducted a comprehensive grid search to determine the optimal configuration. We explored training epochs in 100, 200, 300, learning rates in 0.001, 0.005, 0.01, hidden dimensions in 32, 64, 128, 256, number of layers in 2, 3, and OT alignment weight $\lambda$ in 0.01, 0.05, 0.1, 0.5, 1.0, 1.5, 3.0. All models were trained using the Adam optimizer. We followed prior work and adopted dataset-specific

**Table 2** Detection performance comparison with classic and state-of-the-art methods on Reddit, Amazon, Elliptic, and Libra Bank datasets. Results are reported in terms of ROC (AUC-ROC), PRC (AUC-PRC), and top-10% metrics (precision, recall, and F1-score), evaluating both overall classification accuracy and detection performance.

| Methods | Elliptic | | | | | Libra Bank | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ROC | PR | Prec@10% | Rec@10% | F1@10% | ROC | PR | Prec@10% | Rec@10% | F1@10% |
| SVM | 0.851 | 0.211 | 0.284 | 0.437 | 0.344 | **1.0000** | **1.0000** | **0.0156** | **1.0000** | **0.0307** |
| LSTM | 0.847 | 0.270 | 0.346 | 0.532 | 0.419 | **1.0000** | 0.9999 | **0.0156** | **1.0000** | **0.0307** |
| MLP | 0.871 | 0.309 | 0.387 | 0.596 | 0.469 | 0.7548 | 0.0070 | 0.0114 | 0.7292 | 0.0224 |
| GCN | 0.831 | 0.224 | 0.287 | 0.441 | 0.348 | 0.8005 | 0.0106 | 0.0118 | 0.7583 | 0.0233 |
| GIN | 0.844 | 0.296 | 0.323 | 0.498 | 0.392 | 0.6244 | 0.0031 | 0.0053 | 0.3375 | 0.0104 |
| GAT | 0.883 | 0.460 | 0.389 | 0.599 | 0.472 | 0.9159 | 0.0148 | 0.0143 | 0.9167 | 0.0281 |
| CPL | 0.871 | 0.290 | 0.352 | 0.542 | 0.427 | 0.8224 | 0.0098 | 0.0119 | 0.7625 | 0.0234 |
| BWGNN | 0.891 | 0.551 | 0.428 | 0.658 | 0.519 | 0.6254 | 0.0157 | 0.0075 | 0.4833 | 0.0148 |
| FraudGT | 0.861 | 0.298 | 0.366 | 0.563 | 0.447 | 0.9547 | 0.0962 | 0.0149 | 0.9542 | 0.0293 |
| **Ours** | **0.894** | **0.687** | **0.465** | **0.712** | **0.561** | 0.9919 | 0.0954 | **0.0156** | **1.0000** | **0.0307** |
| Methods | Reddit | | | | | Amazon | | | | |
| | ROC | PR | Prec@10% | Rec@10% | F1@10% | ROC | PR | Prec@10% | Rec@10% | F1@10% |
| SVM | 0.677 | 0.069 | 0.080 | 0.238 | 0.120 | 0.969 | 0.883 | 0.878 | 0.821 | 0.848 |
| LSTM | 0.653 | 0.066 | 0.079 | 0.223 | 0.116 | 0.976 | 0.890 | **0.895** | 0.837 | **0.865** |
| MLP | 0.669 | 0.067 | 0.068 | 0.204 | 0.102 | 0.977 | 0.827 | 0.640 | **0.846** | 0.749 |
| GCN | 0.609 | 0.049 | 0.032 | 0.095 | 0.048 | 0.824 | 0.335 | 0.375 | 0.361 | 0.348 |
| GIN | 0.674 | 0.072 | 0.075 | 0.225 | 0.113 | 0.957 | 0.834 | 0.820 | 0.766 | 0.792 |
| GAT | 0.667 | 0.064 | 0.068 | 0.204 | 0.102 | 0.976 | 0.885 | 0.861 | 0.804 | 0.832 |
| CPL | 0.671 | 0.085 | 0.081 | 0.240 | 0.119 | **0.985** | 0.901 | **0.895** | 0.837 | **0.865** |
| BWGNN | 0.631 | 0.086 | 0.069 | 0.206 | 0.103 | 0.977 | 0.889 | 0.884 | 0.826 | 0.854 |
| FraudGT | 0.673 | 0.079 | 0.075 | 0.224 | 0.113 | 0.971 | 0.896 | 0.878 | 0.821 | 0.848 |
| **Ours** | **0.679** | **0.090** | **0.082** | **0.247** | **0.123** | **0.985** | **0.905** | **0.895** | 0.837 | **0.865** |

splitting strategies [39]. To ensure fairness and reproducibility, all experiments were evaluated under identical split settings for the same dataset. Each experiment was repeated five times with different random seeds, and we report the average performance across runs.

***Evaluation Metrics.***

To comprehensively assess models' performance in the context of highly imbalanced AML data, we employ both global and top-$k$ evaluation metrics [40]. First, we use AUC-ROC (Area Under the Receiver Operating Characteristic Curve) to measure the model's overall classification ability across all classification thresholds. In addition, AUC-PRC (Area Under the Precision-Recall Curve) is reported to capture performance under class imbalance, emphasizing the model's precision in identifying the minority class. For top-$k$ evaluation, we compute precision, recall, and F1-score over the top 10% of high-confidence predictions. This approach ensures that the model not only performs well overall but also excels in prioritizing the most suspicious transactions for further investigation.

## 4.2 Comparison Results

We evaluated our framework against a range of competitive baselines on four datasets, with results summarized in Table 2. The baselines include traditional models: Support

Vector Machine (SVM), Long Short-Term Memory (LSTM), and Multi-Layer Perceptron (MLP); graph-based approaches: Graph Convolutional Networks (GCN) [12], Graph Isomorphism Networks (GIN) [41], Graph Attention Networks (GAT) [42], and Graph Transformer (FraudGT) [43]; and two advanced AML frameworks: Cautious Pseudo Labeling Graph Learning (CPL) [44] and Beta Wavelet Graph Neural Network (BWGNN) [40].

Specifically, the performance of traditional models varies significantly across datasets. On the complex Elliptic graph, they showed limited capability: SVM reached an AUC-PRC of only 0.211, LSTM slightly improved to 0.270, and MLP achieved 0.309, reflecting the difficulty of capturing illicit activities in highly imbalanced transaction graphs. In contrast, on the Amazon dataset, these models performed strongly, with MLP reaching an AUC-ROC of 0.977 and a recall of 0.846, indicating that feature-driven models can be effective when node attributes provide sufficiently discriminative signals. On the Reddit dataset, however, performance dropped again, with all three models yielding very low AUC-PRC scores below 0.07. These varying results demonstrate that models relying primarily on continuous data features may not remain consistently effective across graphs with different feature semantics and structural characteristics. On the Libra Bank dataset, SVM and LSTM achieved near-perfect AUC-ROC and AUC-PRC scores, which can be attributed to the fact that node features directly aggregate per-account alert and report counts, the signals used to define fraud labels. Consequently, the strong feature-driven results on this dataset should be interpreted cautiously, as they partially reflect the direct association between the input features and label construction.

Graph-based methods (GCN, GIN, GAT, FraudGT) showed mixed effectiveness across datasets. GCN achieved AUC-ROC scores of 0.609 on Reddit, 0.824 on Amazon, and 0.831 on Elliptic, suggesting that directly aggregating transaction-graph neighborhoods does not consistently capture anomaly-related semantics. GIN showed moderate improvements, especially on Elliptic with an AUC-PRC of 0.296, while GAT delivered the strongest results among standard GNNs on Elliptic, achieving an AUC-ROC of 0.883 and an AUC-PRC of 0.460. FraudGT achieved competitive performance on Elliptic and Reddit by combining global attention with local message passing. Among advanced methods, BWGNN was the most competitive baseline on Elliptic, with an AUC-PRC of 0.551 and top-10% precision of 0.428. CPL achieved strong performance on Amazon but lagged behind BWGNN on Elliptic. Overall, the substantial performance variation among graph-based baselines indicates that transaction topology alone may be insufficient, motivating the integration of complementary rule-guided knowledge.

Our model achieved the superior performance on most metrics across the four datasets, demonstrating its applicability to financial, social-media, and e-commerce graphs with heterogeneous data characteristics. On the Elliptic dataset, our model improved AUC-PRC by 24.7%, top-10% precision by 8.6%, and F1-score by 8.1% over the strongest baseline, BWGNN. The substantial improvements on this highly imbalanced transaction graph demonstrate the benefit of preserving rule-path dependencies and aligning rule prototypes with data-driven transaction representations. Performance gains were also observed on Reddit and Amazon, where our model achieved

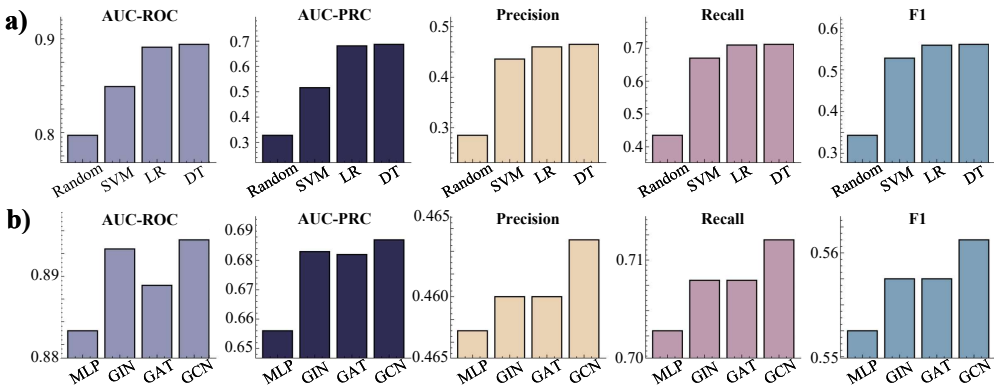



**Fig. 2** Ablation studies on the rule branch design. **a),** Comparison of rule-system construction methods: Random, SVM, Linear Regression (LR), Decision Tree (DT). **b),** Comparison of GNN variants (MLP, GIN, GAT, GCN) evaluated across five metrics.

the best AUC-PRC and tied for the best Amazon AUC-ROC, precision, and F1-score. These results suggest that the proposed rule-to-data alignment mechanism remains effective beyond AML scenarios, despite substantial differences in node semantics and graph structures.

On the Libra Bank dataset, our model achieved an AUC-PRC of 0.0954, slightly below FraudGT at 0.0962 and below the feature-driven SVM and LSTM models, while attaining perfect top-10% recall and tied-best F1@10%. Although the direct alert and report features favor traditional feature-driven models, our framework remains competitive in identifying the extremely sparse fraudulent class. Overall, the results indicate that the rule graph preserves structured supervisory knowledge, OT establishes reliable rule-to-data correspondences, and CCL limits the influence of noisy pseudo-labels, enabling stable anomaly detection across diverse application domains.

## 4.3 Ablation Studies

We further conducted experiments to verify the effectiveness of each component.

***Rule Branch Encoding.***

We evaluated the impact of rule-system construction and graph convolution strategies on model performance, as detailed in Fig. 2. We compared different rule generation strategies (Random, SVM, Linear Regression, and Decision Tree) and graph encoders (GIN, GCN, and GAT).

The choice of rule system proved critical. A Random rule system paired with GAT achieved only 0.797 in AUC-ROC and 0.328 in AUC-PRC, showing that semantic constraints are essential for meaningful rule generation. More structured methods like SVM and Linear Regression improved results, with the latter reaching an AUC-ROC of 0.891. When combining Decision Tree rules with different graph encoders, replacing the GNN with a simple MLP reduced AUC-ROC to 0.884, highlighting the importance of topological dependencies. Incorporating GIN and GAT yielded slight improvements, while GCN achieved the best results, with an AUC-ROC of 0.894, AUC-PRC of 0.687,

**Table 3** Ablation study on alignment strategies and loss weight setups. Results are reported in AUC-ROC, AUC-PRC, and top-10% precision, recall, and F1.

| Alignment | $\lambda$ (for $\mathcal{L}_{\text{OT}}$) | $\beta$ (for $\mathcal{L}_{\text{CCL}}$) | AUC-ROC | AUC-PRC | precision | recall | F1 |
|---|---|---|---|---|---|---|---|
| Average | 1.0 | 1.0 | 0.845 | 0.500 | 0.424 | 0.653 | 0.514 |
| Concat | 1.0 | 1.0 | 0.882 | 0.651 | 0.454 | 0.699 | 0.551 |
| OT | 0.1 | 1.0 | 0.849 | 0.516 | 0.436 | 0.670 | 0.528 |
| OT | 0.5 | 1.0 | 0.887 | 0.676 | 0.460 | 0.707 | 0.557 |
| OT | 1.0 | 0.1 | 0.851 | **0.702** | 0.401 | 0.682 | 0.459 |
| OT | 1.0 | 0.5 | 0.892 | 0.689 | 0.461 | 0.710 | 0.559 |
| OT | 1.0 | 1.0 | **0.894** | 0.687 | **0.465** | **0.712** | **0.561** |

and top-10% precision of 0.465. These results demonstrate that the rule branch benefits most from combining semantic rule generation (via decision trees) with graph-based structural modeling.

***Alignment and Objective.***

We investigated the impact of alignment strategies and loss weight configurations, as presented in Table 3. We compared three embedding alignment methods (Average, Concatenation, and OT) and evaluated the impact of varying the loss weights $\lambda$ (for OT loss) and $\beta$ (for contrastive learning loss).

Basic alignment methods such as Average and Concatenation achieved AUC-ROC scores of 0.845 and 0.882, respectively. Although Concatenation performed better by combining features from both branches, these approaches lacked the ability to capture complex semantic relationships between rule-based and data-driven embeddings. For OT-based alignment, varying the weight ($\lambda$) while fixing $\beta = 1.0$ revealed a clear trend: increasing $\lambda$ from 0.1 to 0.5 improved AUC-PRC from 0.516 to 0.676 and raised the top-10% F1 score from 0.528 to 0.557. This indicates that stronger OT regularization improves the alignment of feature distributions. In contrast, reducing the contrastive learning weight ($\beta$) to 0.1 with $\lambda = 1.0$ led to the highest AUC-PRC (0.702) but much lower precision (0.401) and F1-score (0.459), suggesting that AUC-PRC alone does not reflect top-ranked detection quality. The best overall top-$k$ detection performance was achieved with balanced weights ($\lambda = \beta = 1.0$), reaching an AUC-ROC of 0.894, top-10% precision of 0.465, recall of 0.712, and F1 score of 0.561. These results confirm that jointly optimizing OT and contrastive objectives provides the most effective balance between distribution alignment and representation discriminability.

## 4.4 Visualization

**t-SNE visualization.** We visualize the t-SNE projections of node embeddings on the Elliptic Bitcoin dataset to qualitatively assess the representations learned by our framework, as shown in Fig. 3. The left panel displays embeddings produced by the Rule Branch, where illicit nodes are dispersed across the embedding space with only loose local clustering. This behavior is consistent with the design of the Rule Branch: the decision-tree-guided KNN-GNN constructs a proximity graph based on feature similarity, causing illicit nodes to share neighborhoods with structurally similar benign

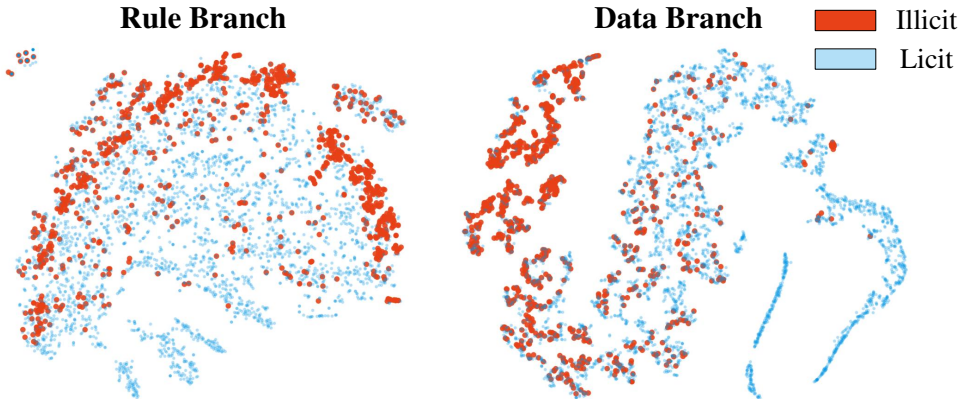



**Fig. 3 t-SNE visualization of node embeddings learned by our framework on the Elliptic Bitcoin dataset.** Left: embeddings from the Rule Branch; Right: embeddings from the Data Branch. Red points denote illicit transactions (**Illicit**); blue points denote licit transactions (**Licit**).

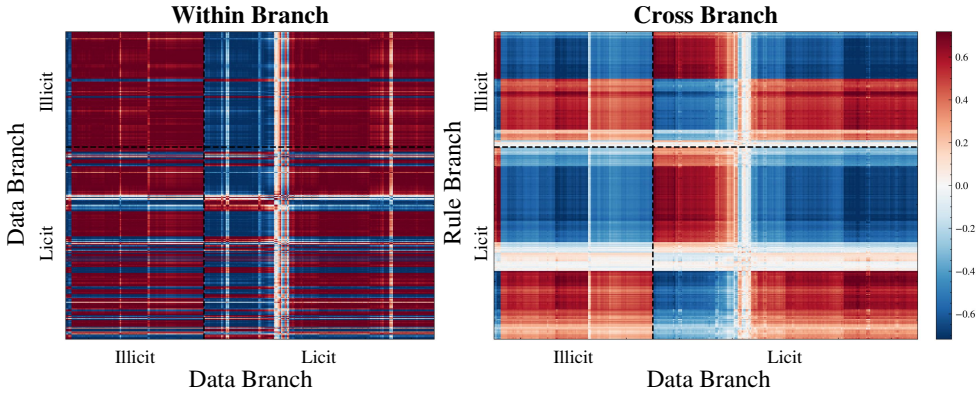



**Fig. 4 Cosine similarity of node embeddings on the Elliptic Bitcoin dataset.** Left: within-branch similarity (Data Branch × Data Branch). Right: cross-branch similarity (Rule Branch × Data Branch). Red indicates high positive similarity; blue indicates negative similarity.

nodes and thereby preserving local topology rather than enforcing global class separation. In contrast, the Data Branch (right panel) yields tighter and more isolated clusters of illicit transactions, as the MLP encoder optimizes directly for discriminative features without graph-structural constraints. Notably, the elongated licit substructure visible on the right may correspond to a temporally cohesive subset of transactions, suggesting that the Data Branch implicitly captures time-varying distributional shifts present in the dataset. The complementary nature of the two branches motivates the cross-domain alignment in AML: neither branch alone achieves clean class separation, and the OT-based fusion bridges the gap between neighborhood-aware and feature-driven representations, ultimately yielding a more robust fraud detector.

**Cosine similarity of node embeddings.** We present the within-branch and cross-branch centered cosine similarity matrices on the Elliptic test set, using the same node ordering to enable direct comparison in Fig. 4. The within-branch matrix

computes pairwise cosine similarity among Data Branch embeddings, while the cross-branch matrix measures cosine similarity between Rule Branch embeddings (rows) and Data Branch embeddings (columns), directly quantifying how well the two representation spaces are aligned after OT-guided training. The cross-branch panel reveals a distinctive four-quadrant structure that evidences the effectiveness of the OT-guided cross-domain alignment. Illicit Rule Branch nodes exhibit strongly positive similarity with their Data Branch counterparts (top-left, red) and strongly negative similarity with licit Data Branch nodes (top-right, blue), forming a sharp anti-symmetric pattern across the class boundary. This indicates that the contrastive alignment loss successfully enforces consistent cross-domain representations for illicit transactions while simultaneously maximizing separation from licit ones. In contrast, licit nodes display more heterogeneous cross-branch alignment (bottom half), reflecting the inherent diversity of normal transaction patterns across different graph-structural contexts and feature distributions. The cross-branch alignment quality for illicit nodes closely approaches that of the within-branch similarity (left panel), demonstrating that the Optimal Transport module effectively bridges the two representation spaces without sacrificing the discriminative structure established by each individual branch.

# 5 Conclusion

In this paper, we propose a unified weakly supervised framework for reliably transferring structured rule knowledge to data-driven representations in graph anomaly detection. Our framework employs a dual-branch design: the rule branch transforms hierarchical decision-tree paths into a rule graph and encodes their structural dependencies via a GNN, while the data branch learns continuous transaction representations. OT aligns rule prototypes with data-driven embeddings and generates soft pseudo-labels for unlabeled samples. A contrastive learning objective further refines the representations and mitigates the propagation of pseudo-label noise. Extensive experiments on financial transaction graphs and general graph anomaly detection benchmarks demonstrate that our framework achieves consistent improvements across datasets with diverse graph structures, feature semantics, and anomaly ratios.

# Declarations

**Ethics declaration:** Not applicable.
**Funding:** This work received research funding support. Detailed funding information has been removed for blind review and will be provided in the non-anonymized version if the manuscript is accepted.

# References

[1] Kute, D.V., Pradhan, B., Shukla, N., Alamri, A.: Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review. IEEE Access **9**, 82300–82317 (2021)

[2] Javaid, M., Haleem, A., Singh, R.P., Suman, R., Khan, S.: A review of blockchain technology applications for financial services. BenchCouncil transactions on benchmarks, standards and evaluations **2**(3), 100073 (2022)

[3] Han, J., Huang, Y., Liu, S., Towey, K.: Artificial intelligence for anti-money laundering: a review and extension. Digital Finance **2**(3), 211–239 (2020)

[4] Ngai, E.W., Hu, Y., Wong, Y.H., Chen, Y., Sun, X.: The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision support systems **50**(3), 559–569 (2011)

[5] Tariq, H., Hassani, M.: Topology-agnostic detection of temporal money laundering flows in billion-scale transactions. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 402–419 (2023). Springer

[6] Phua, C., Lee, V., Smith, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119 (2010)

[7] Unger, B., Van Waarden, F.: How to dodge drowning in data? rule-and risk-based anti money laundering policies compared. Review of Law & Economics **5**(2), 953–985 (2009)

[8] Wang, S.-N., Yang, J.-G.: A money laundering risk evaluation method based on decision tree. In: 2007 International Conference on Machine Learning and Cybernetics, vol. 1, pp. 283–286 (2007). IEEE

[9] Blanuša, J., Cravero Baraja, M., Anghel, A., Von Niederhäusern, L., Altman, E., Pozidis, H., Atasu, K.: Graph feature preprocessor: Real-time subgraph-based feature extraction for financial crime detection. In: Proceedings of the 5th ACM International Conference on AI in Finance, pp. 222–230 (2024)

[10] Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T., Leiserson, C.: Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)

[11] Altman, E., Blanuša, J., Von Niederhäusern, L., Egressy, B., Anghel, A., Atasu, K.: Realistic synthetic financial transactions for anti-money laundering models. Advances in Neural Information Processing Systems **36**, 29851–29874 (2023)

[12] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

[13] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)

[14] Jensen, R.I.T., Iosifidis, A.: Qualifying and raising anti-money laundering alarms

with deep learning. Expert Systems with Applications **214**, 119037 (2023)

[15] Nicholls, J., Kuppa, A., Le-Khac, N.-A.: Financial cybercrime: A comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape. IEEE Access **9**, 163965–163986 (2021)

[16] Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B.B., Chen, X., Wang, X.: A survey of deep active learning. ACM computing surveys (CSUR) **54**(9), 1–40 (2021)

[17] Pham, T., Lee, S.: Anomaly detection in bitcoin network using unsupervised learning methods. arXiv preprint arXiv:1611.03941 (2016)

[18] Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM computing surveys (CSUR) **54**(2), 1–38 (2021)

[19] Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE transactions on pattern analysis and machine intelligence **39**(9), 1853–1865 (2016)

[20] Xie, R., Liu, Q., Wang, L., Liu, S., Zhang, B., Lin, L.: Contrastive cross-domain recommendation in matching. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4226–4236 (2022)

[21] Duan, J., Wang, S., Zhang, P., Zhu, E., Hu, J., Jin, H., Liu, Y., Dong, Z.: Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 7459–7467 (2023)

[22] Ahmed, M., Mahmood, A.N., Islam, M.R.: A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems **55**, 278–288 (2016)

[23] Peyré, G., Cuturi, M., *et al.*: Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)

[24] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)

[25] Fatras, K., Séjourné, T., Flamary, R., Courty, N.: Unbalanced minibatch optimal transport; applications to domain adaptation. In: International Conference on Machine Learning, pp. 3186–3197 (2021). PMLR

[26] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020)

[27] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)

[28] Wu, L., Sun, P., Fu, Y., Hong, R., Wang, M.: Self-supervised graph learning for recommendation. In: SIGIR (2021)

[29] Xu, M., *et al.*: Anomaly detection in dynamic graphs via multi-scale contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2022 (2022)

[30] Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics **21**(3), 660–674 (1991)

[31] Villani, C.: Optimal Transport: Old and New. Grundlehren der mathematischen Wissenschaften, vol. 338. Springer, Berlin, Heidelberg (2009)

[32] Fan, L., Huang, J., Di, D., Su, A., Song, T., Pagnucco, M., Song, Y.: Salvaging the overlooked: Leveraging class-aware contrastive learning for multi-class anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 21419–21428 (2025)

[33] Liang, Y., Hu, Z., Huang, J., Di, D., Su, A., Fan, L.: Tocoad: Two-stage contrastive learning for industrial anomaly detection. IEEE Transactions on Instrumentation and Measurement **74**, 1–9 (2025)

[34] Jing, W., Wang, J., Di, D., Li, D., Song, Y., Fan, L.: Multi-modal hypergraph contrastive learning for medical image segmentation. Pattern Recognition **165**, 111544 (2025)

[35] Liu, Y., Jin, M., Pan, S., Zhou, C., Zheng, Y., Xia, F., Yu, P.S.: Graph self-supervised learning: A survey. IEEE transactions on knowledge and data engineering **35**(6), 5879–5900 (2022)

[36] Dumitrescu, B., Bălţoiu, A., Budulan, Ş.: Anomaly detection in graphs of bank transactions for anti money laundering applications. IEEE Access **10**, 47699–47714 (2022)

[37] Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1269–1278 (2019)

[38] McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 897–908 (2013)

[39] Tang, J., Hua, F., Gao, Z., Zhao, P., Li, J.: Gadbench: Revisiting and benchmarking supervised graph anomaly detection. Advances in Neural Information Processing Systems **36**, 29628–29653 (2023)

[40] Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: International Conference on Machine Learning, pp. 21076–21089 (2022)

[41] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations

[42] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., *et al.*: Graph attention networks. stat **1050**(20), 10–48550 (2017)

[43] Lin, J., Guo, X., Zhu, Y., Mitchell, S., Altman, E., Shun, J.: Fraudgt: A simple, effective, and efficient graph transformer for financial fraud detection. In: Proceedings of the 5th ACM International Conference on AI in Finance, pp. 292–300 (2024)

[44] Wang, B., Li, J., Liu, Y., Cheng, J., Rong, Y., Wang, W., Tsung, F.: Deep insights into noisy pseudo labeling on graph data. Advances in Neural Information Processing Systems **36**, 76214–76228 (2023)