Hybrid Graph Representation Learning for Money Laundering Detection

Marius Frija[1]

Supervisor(s): Dr. Kubilay Atasu[1]**, Halil Çağrı Bilgi**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Marius Frija
Final project course: CSE3000 Research Project
Thesis committee: Dr. Thomas Höllt, Dr. Kubilay Atasu, Halil Çağrı Bilgi

Abstract

Money laundering detection stands as one of the most important challenges in the anti-financial crime sector, given its grave repercussions on the financial industry. The evolving nature of fraud schemes and the increasing volume of financial transactions impose limitations on the detection capabilities of traditional anti-money laundering (AML) systems. In the light of the recent breakthroughs in the field of graph machine learning, graph neural networks (GNNs) and graph transformers (GTs) have emerged as prominent solutions to these limitations, achieving a remarkable performance in detecting complex and broad fraudulent patterns. However, fusing the powerful characteristics of these classes of graph models into a unified framework for fraud detection has been little explored. In this paper, we address this gap by presenting GraphFuse — a hybrid graph representation learning model tailored for money laundering detection in financial transaction graphs. The novel edge centrality and transaction signature encodings offer GraphFuse a slight advantage over the best-performing GNN and GT models, improving upon the best GT baseline by 0.76 p.p. in F1 score. Additionally, we introduce three variants of the Transformer-based component of GraphFuse, each with a different level of computational complexity. The competitive performance of GraphFuse is supported by extensive experiments on open-source, large-scale synthetic financial transactions datasets. Our code is available at https://github.com/mfrija/aml-graphfuse.

1 Introduction

Money laundering represents a serious threat to the global financial sector, causing significant financial losses, reputation damage, and regulatory penalties for financial institutions. The UN estimate an amount equivalent to 2% - 5% of the global GDP to be laundered annually [1]. In the effort of combating money laundering, the instated anti-money laundering (AML) regulations mandate financial institutions to deploy rule-based AML detection systems to support the operations of financial fraud analysts. In the evolving financial and technological landscape, rule-based systems are facing major limitations such as high false alarm rates and inefficiency when dealing with vast amounts of transaction data that require fine-grained analysis. In the context of existing challenges, machine learning techniques emerged as powerful solutions for streamlining and increasing the efficiency of AML operations, providing well-grounded insights about suspicious financial transactions.

The recent advances in the field of deep learning led to the emergence of Graph Neural Networks (GNNs), which revolutionized the processing of graph-structured data [2], [3], [4], [5]. The message passing mechanism [6] that stands at the core of GNNs allows the learning of rich representations that capture both the graph structure and the complexity of the local interactions between various entities. These characteristics make GNNs the perfect tool for financial fraud detection, given that financial transaction data can naturally be represented as directed graphs, as illustrated in Figure 1. Here each node represents a financial account and each edge represents a financial transaction between two accounts. Most GNN models, however, are designed with the assumption that the underlying graph is simple i.e., nodes are connected via single edges. This assumption limits their applicability to large financial networks, which are often modeled as multigraphs with multiple transactions taking place between two accounts. A comprehensive solution is proposed by the MEGA-GNN [7] and Multi-GNN [8] frameworks, which have demonstrated significant improvements in financial fraud detection tasks.

Meanwhile, Graph Transformers [9], [10], [11], [12] have emerged as powerful alternatives to traditional GNNs, leveraging the Transformer architecture [13] to model intricate dependencies between the entities of a graph. The global attention mechanism in Transformers [13] can capture implicit inter-dependecies among accounts and transactions that are not embodied by the graph structure, but could potentially make a significant difference in detection performance. A notable attention-based model - FraudGT [14], demonstrated remarkable results in detecting money laundering activities compared to existing state-of-the-art GNN architectures.

However, little attention was directed towards unifying the local message passing and global attention mechanisms for learning meaningful representations in the context of money laundering detection. To this end, we propose a model that integrates a Local Message Passing Module and a Global Attention Module allowing the architecture to jointly capture localized financial graph structures and global implicit information about transactions between accounts. As part of this research, we also investigate the performance characteristics of different global attention mechanisms — specifically, Linear, Full, and Sparse Attention to better understand the trade-off between computational efficiency and fraud detection performance.

Our contributions

In summary, this work's main contributions are:

- We introduce a hybrid model framework for graph representation learning tasks in the context of money laundering detection.
- We propose three attention-based configurations aiming to achieve a trade-off between money laundering detection performance and computational efficiency.
- We validate our framework on a synthetic dataset of financial transactions in the supervised task of transaction classification, surpassing the state-of-the-art financial crime detection models.

(a)

Trans. ID	Timestamp	Source bank ID	Source Account	Target bank ID	Target Account	Amount	Currency	Payment type
1	3 MAY 2019 12:45	1	E	1	A	1400	USD	Cheque
2	15 MAY 2019 07:34	2	C	1	A	710	EUR	ACH
3	18 MAY 2019 16:55	3	B	1	A	950	USD	Credit card
4	1 JUN 2019 10:06	1	A	3	D	1200	CHF	Wire
5	27 JUN 2019 11:14	3	B	3	D	2300	EUR	Credit card
6	7 JUL 2019 13:18	3	D	1	E	1100	USD	Credit card
7	14 JUL 2019 09:37	2	C	3	B	650	USD	ACH

(b)

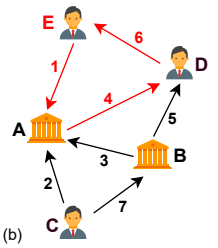

Figure 1: Financial transactions in (a) tabular format and in (b) graph format. Transactions with IDs 1,4 and 6 depict a simple money laundering cyclic pattern in which the money launderer agent **E** manages to obfuscate the origin of the illicit funds through a cycle of financial transactions. Figures recreated by the authors, following the visualizations presented in [8].

2 Related Work

2.1 Graph Neural Networks

Graph Neural Networks (GNNs) [4], [5], [3], [2] have emerged as a prominent class of deep learning models designed for processing graph-structured data. Central to modern GNNs is the *local message passing* [6] mechanism, which aggregates information from the neighborhood of each node or edge to derive context-aware representations. This mechanism is particularly relevant in anti-money laundering (AML) scenarios, where the local topology of a transaction network often reveals indicative patterns of illicit behavior (see Figure 1). In fraud detection, GNNs, especially the Principal Neighborhood Aggregation model (PNA) [2], have shown notable success in capturing suspicious transaction behavior, with recent benchmarks establishing their competitiveness on realistic datasets [15]. More recent advancements have focused on enhancing the message passing mechanism for multigraphs. For example, Multi-GNN introduces the reverse message passing mechanism [8], while MEGA-GNN enhances multi-edge aggregation to effectively handle multigraph settings [7]. These efforts highlight the adaptability of GNNs to financial crime detection tasks and motivate their integration in hybrid graph learning architectures.

2.2 Graph Transformers

Graph Transformers extend the powerful Transformer architecture [13], which has revolutionized the field of natural language processing [16], [17], [18]. The cornerstone of Transformers — the attention mechanism, enables the capture of complex, long-range dependencies between the elements of large sequential datasets. Unlike standard GNNs that rely on the local message passing mechanism, Graph Transformers leverage global attention to derive context-rich node and edge embeddings across the entire graph. Pioneering models like Simple Graph Transformer [9], Graphormer [19] and SGFormer [11] showcased a superior performance over traditional GNNs on diverse graph learning tasks. In the financial fraud domain, FraudGT [14] demonstrated the effectiveness of the graph transformer architecture for fraud detection, achieving superior results compared to multigraph-enhanced GNNs. However, the existing attention-based models consider the direct application of the global attention computation on node attributes, rather than on edge features. In money laundering detection applications, this might lead to insufficient model expressivity given the edge-attributed nature of financial multigraphs.

In summary, a general framework that unifies the local message passing and global edge-aware attention mechanisms in the context of learning tasks on large financial multigraphs has yet to be proposed. This work closes this gap by introducing a novel hybrid model framework designed for money laundering detection tasks in large-scale financial transaction multigraphs.

3 Proposed model

In this section, we first present preliminaries on graph representation of financial transaction networks. We proceed with the preliminaries on Message-Passing Graph Neural Networks and Graph Transformers. Then, we introduce the architecture and methodology of GraphFuse.

3.1 Preliminaries

Graph Representation of Financial Transaction Data A financial transaction network can be represented as a directed multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{E})$, where the nodes $v \in \mathcal{V}$ represent accounts, and the directed edges $e = (u, v) \in \mathcal{E}$ represent transactions from u to v. If the graph is node-attributed, the node attribute matrix $\mathbf{X} \in \mathbb{R}^{N \times d_n}$ assigns a set of account features x_u to each node u; this could include the account number, bank ID and account balance. Additionally, if the transaction graph is edge-attributed i.e., each transaction has a set of associated transactions features, the edge attribute matrix $\mathbf{E} \in \mathbb{R}^{M \times d_e}$ assigns attributes to each edge. d_n and d_e are the dimensions of node and edge attributes, respectively. The in and out degree of a node u are denoted by $|\mathcal{N}_{in}(u)|$ and $|\mathcal{N}_{out}(u)|$, where $\mathcal{N}_{in}(u), \mathcal{N}_{out}(u)$ represent the sets of incoming and outgoing neighbors of u. The size of the financial graph, as measured by the number of edges $M = |\mathbf{E}|$, can be arbitrarily large, ranging from thousands to billions.

Graph Neural Networks Modern GNNs employ the local message passing mechanism to iteratively learn the representation of the nodes and edges of $\mathcal{G}$ by aggregating the representations of their neighbors. Let $h^{(l)}(v_i)$ be the representation of v_i at the l-th layer and $h^{(0)}(e_{ki})$ denote the input features of a directed edge connecting v_k to v_i. Following the definition from [8], when using edge features during the mes-

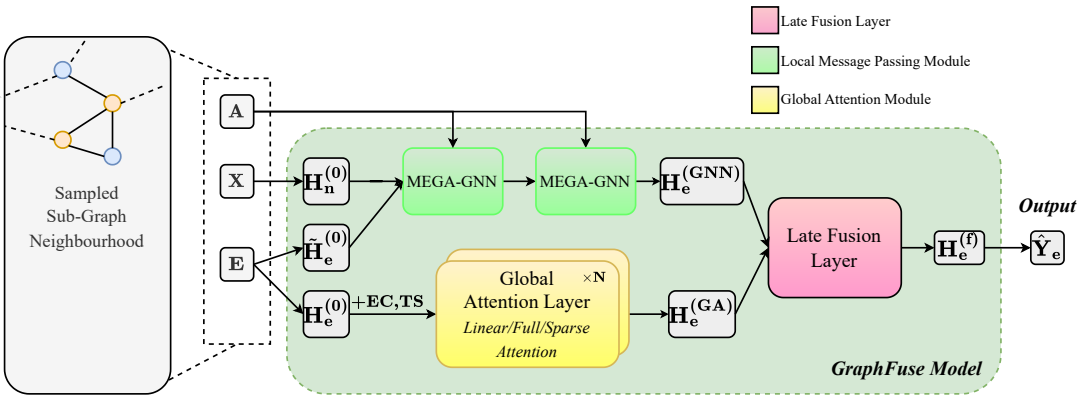

Figure 2: Illustration of the proposed model GraphFuse and its data flow. The input embeddings of the nodes and the edges of the sampled sub-graph are passed to the Local Message Passing (MP) Module ($\tilde{\mathbf{H}}_e^{(0)}, \mathbf{H}_n^{(0)}$) and Global Attention Module ($\mathbf{H}_e^{(0)}$). The edge embeddings generated by the two modules are passed to the Late Fusion Layer which outputs the edge representations ($\mathbf{H}_e^{(f)}$) for the downstream financial fraud detection task.

sage passing, the updated node representation is computed as follows:

$$h^{(l)}(v_i) = \text{Update}(h^{(l-1)}(v_i), a^{(l)}(v_i)), \text{where} \quad (1)$$

$$a^{(l)}(v_i) = \text{Aggregate}\left(\{\{(h^{(l-1)}(v_k), h^{(0)}(e_{ki}))|v_k \in \mathcal{N}_{in}(v_i)\}\}\right) \quad (2)$$

where Aggregate is a permutation-invariant function and $\{\{\cdot\}\}$ denotes a multi-set. The goal of the Update function is to fuse the information from the neighbors into the node representation. By following a similar procedure, edge representations can be derived by utilizing the information from the associated source and destination vertices.

Graph Transformers The Transformer architecture [13], that lies at the core of graph transformers, consists of a chain of L encoder layers. Each encoder layer consists of a multi-head attention (MHA) module and a position-wise feed-forward network (FFN). Following the definition from [14], let $\mathcal{G}$ be a graph with egde feature matrix $\mathbf{E} \in \mathbb{R}^{M \times d_e}$, where $e_{ij} \in \mathbb{R}^{d_e}$ is the feature vector of a directed edge that connects nodes v_i and v_j. In each layer $l(l > 0)$, given the hidden feature matrix $\mathbf{H}_e^{(0)} = \mathbf{E}$, the MHA module first linearly projects the input $\mathbf{H}_e^{(l-1)}$ to the query, key and value matrices $\mathbf{Q}^{(h,l)}, \mathbf{K}^{(h,l)}, \mathbf{V}^{(h,l)}$ using the corresponding weight matrices $\mathbf{W}_Q^{(h,l)}, \mathbf{W}_K^{(h,l)}, \mathbf{W}_V^{(h,l)} \in \mathbb{R}^{d_e \times d_h}$. The linear projection is defined as follows:

$$\mathbf{Q}^{(l,h)} = \mathbf{H}_e^{(l-1)}\mathbf{W}_Q^{(h,l)}, \mathbf{K}^{(l,h)} = \mathbf{H}_e^{(l-1)}\mathbf{W}_K^{(h,l)}, \mathbf{V}^{(l,h)} = \mathbf{H}_e^{(l-1)}\mathbf{W}_Q^{(h,l)} \quad (3)$$

where $\mathbf{Q}^{(l,h)}, \mathbf{K}^{(l,h)}, \mathbf{K}^{(l,h)} \in \mathbb{R}^{M \times d_h}$. Then, multiple attention heads are used to compute the scaled dot-product self attention, as shown in Equation 4, where the softmax function is applied row-wise, $\mathbf{W}_{O_h}^{(l)} \in \mathbb{R}^{d_e \times d_e}$ is a learnable weight matrix, $h = 1..H$ denotes the index of the different attention heads and $\|$ denotes the matrix concatenation operator.

$$\text{MHA}(\mathbf{H}_e^{(l-1)}) = \Big\|_{h \in [1,H]} \left(\text{softmax}(\frac{\mathbf{Q}^{(l,h)}\mathbf{K}^{(l,h)\top}}{\sqrt{d_h}})\mathbf{V}^{(h,l)}\right) \mathbf{W}_{O_h}^{(l)} \quad (4)$$

By combining the result with additional residual connections and normalization layers, the encoder layer updates hidden features $\mathbf{H}_e^{(l-1)}$ as follows:

$$\hat{\mathbf{H}}_e^{(l)} = \text{MHA}(\mathbf{H}_e^{(l-1)}) + \mathbf{H}_e^{(l-1)}, \quad (5)$$

$$\mathbf{H}_e^{(l)} = \text{FFN}(\hat{\mathbf{H}}_e^{(l)}) + \mathbf{H}_e^{(l)} = \left[\sigma\left(\hat{\mathbf{H}}_e^{(l)}\mathbf{W}_1^{(l)}\right)\mathbf{W}_2^{(l)}\right] + \mathbf{H}_e^{(l)} \quad (6)$$

where σ refers to the activation function, and $\mathbf{W}_1^{(l)} \in \mathbb{R}^{d_e \times d_f}$ and $\mathbf{W}_2^{(l)} \in \mathbb{R}^{d_f \times d_e}$ are trainable parameters in the feedforward network (FFN) layer. The final output $\mathbf{H}_e^{(L)} \in \mathbb{R}^{M \times d_e}$ can be used as the updated edge representations for downstream tasks.

3.2 GraphFuse: Hybrid Graph Representation Learning Model

In this section, we introduce GraphFuse — the hybrid graph representation learning model which consists of new components compared to existing models and combines traditional Message Passing Neural Networks and the Graph Transformer paradigm. In order to increase the expressiveness of the hybrid model and increase its performance for the money laundering detection task, we introduce two structure- and context-aware edge encodings: edge centrality and transaction signature. The architecture of the proposed model is presented in Figure 2, highlighting the main components of the hybrid model: the Local Message Passing Module, the Global Attention Module and the Late Fusion Layer. A detailed description of each component is provided in the next sub-sections.

Incorporation of Structural and Contextual Information

In order to align with the intuition behind the original Transformer [13] which makes use of structural information about the sequences that it processes, we devise a way of incorporating a similar structure-aware inductive bias into the attention computation of our Global Attention Layer (see Figure 2). Given that money laundering analysis involves contextual information about the accounts engaged in transactions, we argue that enriching the hidden edge representations with relevant account descriptors leads to a better detection capability of our model. We present the simple but effective design of these edge encodings in GraphFuse.

Edge Centrality

Centrality encoding which was introduced in the Graphormer model [19], came to enhance the expressivity of the attention mechanism in the context of graph representation learning tasks. Derived using the degree centrality measure, node centrality encodings manage to introduce an important signal into the self-attention computation, reflecting the relative importance of a particular node based on its connectivity. A more rigorous work which analyzes the topological and geometrical properties of money laundering activities in financial networks [20], highlighted the relevance of the Forman-Ricci curvature [21] as a tool for detecting illicit patterns. Intuitively, the Forman-Ricci curvature assesses the importance of an edge in terms of its connectivity to its neighbors, a measure which is tightly connected to the degree centrality of its endpoints. The relevance of the latter mentioned geometrical property of edges for the money laundering detection tasks, stems from the fact that malicious agents prefer to engage in few transactions with other agents in order to put their activity out of sight [20]. Therefore, the paradigms introduced in [19] and [20] coalesce into a novel *edge centrality* encoding that introduces an important inductive bias in the context of detecting money laundering transactions in large financial graphs. Given the directed edge e_{ij} connecting the vertices v_i and v_j, the edge centrality encoding is defined as follows:

$$\epsilon_{ij} = f_{EC}\left(\mu_i \| \mu_j\right), \text{where,} \tag{7}$$

$$\mu_k = \frac{1}{2}\left(\log(deg^-(v_k)) + \log(deg^+(v_k))\right), \tag{8}$$

where f_{EC} represents a shallow neural layer (e.g. a linear feed-forward layer in our implementation), $deg^+(v_k)$ and $deg^-(v_k)$ denote the in- and out-degree of a node v_k and $\|$ denotes the vector concatenation operator. Therefore, ϵ_{ij} quantifies an important intuition behind financial transaction networks, mainly that nodes with a higher centrality (i.e., a higher in- and out-degree) typically correspond to trusted accounts in the financial network and are less likely to be involved in chains of illicit transactions.

Transaction Signatures

Beyond the topological characteristics of money laundering patterns, the behavior and the financial activity of an agent provides an important implicit bias for detecting illicit transactions associated with one's account. The work conducted in [22] and [23] underlines the significant correlation between the illicit status of a transaction and the specific behavioral patterns of its associated accounts, such as the use of multiple currencies, large and irregular transfer amounts and high transaction frequencies. All such characteristics form an *account signature* for all the accounts in the financial network. For our application, an account signature consists of the following metrics: *Currency Diversity*, *Received Amount Median/Dispersion*, *Transferred Amount Median/Dispersion*, *Deposit/Transfer Frequency Rate* and *Sent To Received Amount Ratio*. Given the directed edge e_{ij} connecting the vertices v_i and v_j, the transaction signature is defined as follows:

$$s_{ij} = \sigma_i \| \sigma_j, \tag{9}$$

where σ_k denotes the account signature of a node v_k and $\|$ denotes the vector concatenation operator.

3.3 Input Layer

The input node and edge feature matrices are linearly projected to create multi-dimensional embeddings to be passed to the Local Message Passing (MP) and Global Attention (GA) modules. In order to separate the learning process and increase the expressivity of the Global Attention module, we create distinct initial edge embeddings with independent dimensionalities for the two modules. Therefore, the local MP module receives the initial node and edge embeddings $\mathbf{H}_n^{(0)}$, $\tilde{\mathbf{H}}_e^{(0)}$, while the Global Attention module operates on the separate edge embeddings $\mathbf{H}_e^{(0)}$ (see Figure 2). Additionally, we add the edge centrality and transaction signature encodings to the edge embeddings $\mathbf{H}_e^{(0)}$ to be passed to the Global Attention Module:

$$\mathbf{h'}_{ij}^{(0)} = f_{\mathrm{GA}}\left(\mathbf{e}_{ij} \| s_{ij}\right), \tag{10}$$

$$\mathbf{h}_{ij}^{(0)} = \mathbf{h'}_{ij}^{(0)} + \alpha \epsilon_{ij} \tag{11}$$

where $\alpha \in \mathbb{R}$ is a learnable scalar, $\|$ denotes the vector concatenation operator and f_{GA} denotes a shallow neural network.

3.4 Local Message Passing Module

For the Local Message Passing Module (see Figure 2) we adopt the MEGA-GNN framework introduced in [7], considering several message passing layers. Leveraging the powerful multi-edge aggregation mechanism, which was proven to increase the expressivity of message passing neural networks, the derived edge representations capture more nuanced information about the local neighborhood of the financial transactions. Given its superior performance [7], we consider the message passing mechanism of the PNA model [2] for both node- and edge-level aggregation. Additionally, consistent with the multigraph enhancements introduced in [7] and [8], we incorporate Ego IDs [24], which significantly increase the capability of detecting cyclic patterns in graphs. Given the input node and edge embeddings $\mathbf{H}_n^{(0)}$, $\tilde{\mathbf{H}}_e^{(0)}$, and the adjacency matrix of the sampled sub-graph $\mathbf{A}$ (see Figure 2), the output edge representations of the Local MP module are computed as:

$$\mathbf{H}_e^{(\mathrm{GNN})} = \mathrm{GNN}^{(L)} \circ \cdots \circ \mathrm{GNN}^{(1)}\left(\mathbf{H}_n^{(0)}, \tilde{\mathbf{H}}_e^{(0)}, \mathbf{A}\right) \tag{12}$$

3.5 Global Attention Module

In order to capture more intricate and globally-distributed laundering patterns, the global attention module is designed to operate over a larger (global) graph neighborhood. Compared to the modus operandi of the Local Message Passing (MP) Module, the Global Attention (GA) Module is responsible for capturing implicit biases that arise from the indirect and hidden relations between financial agents and their interactions across the entire graph. Therefore, the global view offered by the GA Module introduces subtle yet important signals that the Local MP Module alone can not detect. In contrast to existing GT models, we implement the attention computation directly on the edge hidden representations. To the best of our knowledge this is the first attention-based model that operates on edges in the context of AML detection.

The proposed global attention layer follows the architecture of the standard Transformer encoder layer described in [13]. The core of the global attention layer is the multi-head self-attention module that generates contextualized edge representations by modeling all the pair-wise interactions between the edges of the sampled sub-graph. In addition, similar to the Graphormer model [19] implementation, we apply the layer normalization (LN) before the multi-head self-attention (MHA) block and the feed-forward blocks (FNN) instead of after. This modification is preferred by the majority of modern Transformer-based models because it leads to a more effective optimization [25]. We formally characterize the Global Attention Layer — GALayer as below:

$$\mathbf{H}_e^{(l)} = \text{GALayer}(\mathbf{H}_e^{(l-1)}), \text{where,} \tag{13}$$

$$\hat{\mathbf{H}}_e^{(l)} = \text{MHA}(\text{LayerNorm}(\mathbf{H}_e^{(l-1)})) + \mathbf{H}_e^{(l-1)}, \tag{14}$$

$$\mathbf{H}_e^{(l)} = \text{FNN}(\text{LayerNorm}(\hat{\mathbf{H}}_e^{(l)})) + \hat{\mathbf{H}}_e^{(l)} \tag{15}$$

where $\mathbf{H}_e^{(l)}, \mathbf{H}_e^{(l-1)} \in \mathbb{R}^{M \times d_e}$. To align with the design principles of the original Transformer architecture introduced in [13], we set the dimensionality of the position-wise neural layers of the FFN to $d_f = 4 \cdot d_e$, therefore balancing model capacity and computational efficiency. Given the constraints imposed by the large scale of the financial transaction graphs and the inherent quadratic complexity of the self-attention mechanism, our work investigates the performance of three different implementations of the multi-head self-attention block MHA. The linear (MHA_{Lin}), full (MHA_{Full}) and sparse ($\text{MHA}_{\text{Sparse}}$) attention mechanisms are described in the following sub-sections.

Linear Attention Mechanism

The simple global attention mechanism of the SGFormer model [11] manages the reduction of the $\mathcal{O}(|E|^2)$ computational complexity overhead of the vanilla softmax attention from the original Transformer implementation [13]. While offering a computation of attentive representations that can be achieved in $\mathcal{O}(|E|)$, the linear attention function of SGFormer also guarantees the expressivity to model all pair-wise interactions between the edges in the sampled subgraph. Given the proven efficiency in learning meaningful representations and the reduced computational complexity of the aforementioned mechanism, we adopt it for the global attention layer of our Global Attention Module. Additionally, the linear complexity advantage allows us to make use of larger batch sizes which directly leads to a better capability of the model for learning complex long-range dependencies between the transactions in the sampled subgraphs. As the original implementation considers a single-layer, single-head attention-based model, we generalize and adapt it for a multi-layer and multi-head attention computation while maintaining the linear computational complexity. The resulting multi-head linear attention function is defined as follows:

$$\text{MHA}_{\text{Lin}}(\mathbf{H}_e^{(l-1)}) = \Big\Vert_{h \in [1,H]} \mathcal{H}(l,h)\mathbf{W}_{O_h}^{(l)}, \text{where,} \tag{16}$$

$$\mathcal{H}(l,h) = (\mathbf{D}^{(l,h)})^{-1}\left[\mathbf{V}^{(l,h)} + \frac{1}{M}\tilde{\mathbf{Q}}^{(l,h)}(\tilde{\mathbf{K}}^{(l,h)\top}\mathbf{V}^{(l,h)})\right], \tag{17}$$

$$\mathbf{D}^{(l,h)} = \text{diag}\left(\mathbf{1} + \frac{1}{M}\tilde{\mathbf{Q}}^{(l,h)}(\tilde{\mathbf{K}}^{(l,h)\top}\mathbf{1})\right), \text{and} \tag{18}$$

$$\tilde{\mathbf{Q}}^{(l,h)} = \frac{\mathbf{Q}^{(l,h)}}{\|\mathbf{Q}^{(h,l)}\|_{\mathcal{F}}}, \tilde{\mathbf{K}}^{(l,h)} = \frac{\mathbf{K}^{(l,h)}}{\|\mathbf{K}^{(h,l)}\|_{\mathcal{F}}} \tag{19}$$

where $\|\cdot\|$ denotes the Frobenius norm, $\mathbf{1}$ is an M-dimensional all-one column vector and the diag operation changes the M-dimensional column vector into a $M \times M$ diagonal matrix.

Full Attention Mechanism

Given that full (softmax) attention possesses provable expressivity in the context of graph representation learning tasks [26], we consider the multi-head scaled dot-product attention function defined in Equation 4 as one of the alternative mechanisms for our Global Attention Layer:

$$\text{MHA}_{\text{Full}}(\mathbf{H}^{(l-1)}) = \Big\Vert_{h \in [1,H]} \left(\text{softmax}(\frac{\mathbf{Q}^{(1,\mathbf{h})}\mathbf{K}^{(1,\mathbf{h})\top}}{\sqrt{d_h}})\mathbf{V}^{(h,l)}\right)\mathbf{W}_{O_h}^{(l)} \tag{20}$$

where $\mathbf{Q}^{(1,\mathbf{h})}, \mathbf{K}^{(l,h)}, \mathbf{V}^{(h,l)}$ represent the the query, key and value projections as introduced in Section 3.1. Considering the quadratic complexity of the full attention mechanism, model configurations employing MHA_{Full} within the Global Attention Layers have a reduced scalability on large graphs. As a consequence, such model configurations require more computational resources for learning attentive edge representations over larger sub-graphs. In the context of our learning task which involves broad financial networks and limited computational resources, this would would restrict the use of multiple full attention layers with high dimensionalities and therefore limit the ability of detecting long-range dependencies between illicit transactions.

Sparse Attention Mechanism

In order to leverage the expressive power of full attention, but reduce the associated quadratic computational overhead, we employ a sparse attention mechanism. Inspired from the clustered-attention paradigm introduced in the ClusterFormer model [27], our sparse attention layer implementation includes two stages during forward propagation, mainly the

allocation of edges to clusters and the multi-head attentive-computation within each of the clusters. Due to the large scale of the considered graphs, we opt for the uniformly random clustering of edges instead of similarity-preserving clustering techniques, in order to avoid additional computational overhead. The devised algorithm is defined as follows:

Algorithm 1 Clustered Sparse Attention Mechanism. For clarity, the computation across multiple attention heads is omitted from the pseudocode.

Require: Number of sampled edges N_e, Cluster size normalization factor K, Input edge embeddings to the GA Module $\mathbf{H_e^{(0)}}$

Ensure: Attentive edge embeddings $\mathbf{H}_e^{(\text{GA})}$

Compute number of clusters: $C \leftarrow \max(1, \lceil N_e/K \rceil)$
Randomly assign edges to clusters: $\mathbf{clusters} \in \{0, 1, \ldots, C-1\}^{N_e}$ uniformly sampled
for each transformer layer $l = 1$ to L **do**
    Linearly project the edge embeddings to the query, key and value matrices $\mathbf{Q}^{(l)}$,$\mathbf{K}^{(l)}$,$\mathbf{V}^{(l)}$
    for each cluster $c = 0$ to $C-1$ **do**
        Select edge indices in cluster c: $\mathcal{I}_c \leftarrow \{i \mid \mathbf{clusters}[i] = c\}$
        Extract $\mathbf{Q}_c^{(l)}$,$\mathbf{K}_c^{(l)}$,$\mathbf{V}_c^{(l)}$ as $\mathbf{Q}^{(l)}[\mathcal{I}_c]$,$\mathbf{K}^{(l)}[\mathcal{I}_c]$,$\mathbf{V}^{(l)}[\mathcal{I}_c]$
        Compute the intra-cluster attention filter:

$$\mathbf{A}_c \leftarrow \text{softmax}\left(\frac{\mathbf{Q}_c\mathbf{K}_c^\top}{\sqrt{d_k}}\right)$$

        Update the embeddings of the edges in cluster c:

$$\mathbf{H}_e^{(l)}[\mathcal{I}_c] \leftarrow \mathbf{A}_c\mathbf{V}_c$$

    end for
end for
$\mathbf{H}_e^{(\text{GA})} \leftarrow \mathbf{H}_e^{(L)}$
return $\mathbf{H}_e^{(\text{GA})}$

We summarize the layer-wise multi-head sparse attention computation using the following expression:

$$\text{MHA}_{\text{Sparse}}(\mathbf{H}_e^{(l-1)}) = \Big\|_{h\in[1,H]} \left(\text{ClusterSoftmax}\left(\frac{\mathbf{Q}^{(l,h)}\mathbf{K}^{(l,h)\top}}{\sqrt{d_h}}\right)\mathbf{V}^{(l,h)}\right)\mathbf{W}_{O_h}^{(l)} \tag{21}$$

Late Fusion Layer

In order to generate rich edge embeddings that capture both the structure of the local graph neighborhood and the global semantic context, we resort to a simple-yet-effective late fusion layer that combines the edge embeddings propagated from the Global Attention Module and the Local Message Passing Module, as illustrated in Figure 2. The late fusion layer is comprised of a simple single-layer feed-forward network MLP$(\cdot)$ and a non-linear activation function σ. Let $\|$ denote vector concatenation, and $\mathbf{h}_{ij}^{(\text{GA})}$ and $\mathbf{h}_{ij}^{(\text{GNN})}$ represent the embeddings of a directed edge connecting vertices v_i and v_j, as propagated from the aforementioned modules. The unified edge embedding obtained through fusion is

$$\mathbf{h}_{ij}^{(f)} = \sigma\left(\text{MLP}\left(\mathbf{h}_{ij}^{(\text{GA})} \| \mathbf{h}_{ij}^{(\text{GNN})}\right)\right), \tag{22}$$

By learning a projection over the combined edge representation, the model can adaptively integrate local- and global-context features, effectively identifying which information is most relevant for edge classification.

3.6 Training and Prediction

The aim of the model introduced in this paper is to generate meaningful edge embeddings that can be leveraged to derive the fraud-indicative score of each financial transaction. Therefore, the edge embeddings obtained after applying the late fusion layer $\mathbf{H_e^{(f)}}$ are passed to the final classification layer. Consistent with [7], [14], [8], the final classification layer is composed of a simple feed-forward network MLP$(\cdot)$ and a sigmoid function σ. The predicted anomaly score $\hat{y}_{ij}$ for each edge between the nodes v_i and v_j is:

$$\hat{y}_{ij} = \sigma\left(\text{MLP}(\mathbf{h}_{ij}^{(f)})\right), \tag{23}$$

4 Experimental Setup and Results

This section presents our experiments designed to evaluate the performance of the proposed hybrid late-fusion model in the context of detecting illicit financial transactions through edge classification. We provide a thorough description of the datasets used in our experiments and the baselines against which we compare our model. Furthermore, this section presents the results achieved by the proposed model and offers a comparison between the different model configurations through ablation studies.

4.1 Experimental Setup

Datasets

Given the strict privacy regulations around financial data, real-world datasets are not readily available. Besides the limited availability of data reflecting real scenarios of money laundering and financial fraud, more than often these datasets suffer from poor labeling, as many money laundering activities go undetected [1], [28]. Additionally, banks and financial institutions often only keep records of the activity related to their own accounts, therefore missing the broader context of customer behavior across multiple institutions. These challenges motivate us to rely on existent synthetic money laundering data [15]. These datasets correspond to large financial transaction networks, which are generated by modeling agents (banks, companies and individuals) in a virtual financial environment. The generator takes into account several well-established money laundering patterns in order to replicate real-world fraudulent scenarios. We use two small-sized datasets: one with a higher illicit ratio (HI) and one with a lower illicit ratio (LI). We use a 60-20-20 temporal train-validation-test split, i.e., we split the transactions after ordering them by their timestamps. A more detailed description of the datasets is present in Appendix C.

Baselines

We compare GraphFuse against GNN models with edge features and GT models. Consistent with prior work that analyzed the money laundering detection capability of graph representation learning models [7], [8], [14], [15], the selected baseline GNN model is PNA [2]. Additionally, the MEGA-PNA model [7] is considered as the representative of the class of state-of-the-art multigraph-enhanced GNN models. These models are also employed within the Local Message Passing Module of our hybrid architecture (see Figure 2), therefore ensuring a fair comparison between standalone and integrated configurations. Multi-FraudGT [14] is selected as the GT baseline, currently representing the best performing Transformer-based model for financial fraud detection. Given the size of the AML datasets, we use neighbourhood sampling [3] for training the baseline and GraphFuse models. Further details about the hyperparameters and experimental setup are provided in Appendix A.

Evaluation and Scoring

Transactions that constitute money laundering instances represent a significantly small percentage of the total volume of transactions that occur in the real world. Given their realistic characteristics, the used datasets are highly imbalanced (see Appendix C), making popular metrics for measuring accuracy unsuitable. Therefore, we use the minority class F1 score which is consistent with previous works [14], [7], [8] and represents the de-factor metric used by banks and regulators for money laundering detection. More details about the derivation of the F1 score are presented in Appendix B. The reported test performance for each experiment is based on the model checkpoint that achieved the highest validation F1 score. To ensure statistical significance, each experiment is repeated five times with different random seeds, and the mean and standard deviation across these runs are reported.

4.2 Experimental Results

Classification Results

Table 1 presents the transaction classification results of the selected baselines and GraphFuse variants across the considered datasets. Six different GraphFuse models are employed, covering all the combinations of the two message passing layers (PNA and MEGA-PNA) and the three implemented attention mechanisms. We make the following observations. First, the GraphFuse-PNA model demonstrates significant improvements over the PNA baseline across both datasets and for all attention configurations. The superiority of the hybrid model is particularly evident on the highly-imbalanced Small-LI dataset, achieving a 10.2%-14.3% improvement and therefore validating the effectiveness of leveraging global attention in synergy with local message passing. The model variant employing Linear Attention stands out with higher detection scores, which can be attributed to its inherent linear complexity enabling a deeper structure of the Global Attention Module (see Appendix A). Second, the final multigraph-enhanced GraphFuse-MEGA-PNA model achieves impressive results, managing to consistently outperform the standalone MEGA-PNA model across both datasets. The efficacy of the hybrid model incorporating edge encodings, specifically tailored for money laundering detection, is underlined by its performance improvement over the state-of-the-art (SOTA) Multi-FraudGT model. On the Small-HI dataset, GraphFuse-MEGA-PNA outperforms SOTA across all attention configurations, notably achieving a gain of 0.76% over Multi-FraudGT. Additionally, the Linear Attention model variant achieves the best F1 score on both the Small-HI and Small-LI datasets. Overall, the results support our proposed method, demonstrating the efficacy of a graph model that unifies GNNs with GTs.

Ablation Study: Edge Encodings

We perform an ablation study on two of the components introduced in GraphFuse, mainly the edge centrality and transaction signature edge encodings. Given the computational complexity of the Full and Sparse attention model variants, the study was performed using the GraphFuse-MEGA-PNA configuration incorporating Linear attention. The results of the F1 scores are presented in Table 1. We observe that individually adding the edge encodings does not have a substantial impact on model performance. However, when incorporating both encodings, we notice a performance increase over the state-of-the-art Multi-FraudGT model across both the Small-HI and Small-LI datasets. These results demonstrate the synergistic effect of the novel edge encodings on increasing the expressivity of GraphFuse in money laundering detection tasks.

Training Efficiency and Fraud Detection Performance Analysis

Table 2 compares the training and inference efficiency of the model variants incorporating the three different attention mechanisms: Linear, Sparse and Full. All GraphFuse-MEGA-PNA variants display a comparable detection score, outperforming the SOTA Multi-FraudGT model by a small yet noteworthy margin. However, in terms of efficiency, the Linear Attention model variant achieved, on average, 3 times higher throughput and 2.5 times lower training time per epoch, despite having a higher number of parameters than the Sparse and Full Attention configurations. Therefore, due to its favorable trade-off between scalability and detection capabilities, the Linear Attention model configuration stands out as a strong candidate for industry-level AML systems operating over large-scale financial transaction data. Note that the resource and time constraints of this research disallowed the investigation of more complex Sparse and Full Attention configurations. We argue that further investigation could yield improved detection performance for the aforementioned model configurations.

5 Conclusions and Future Work

In this paper, we introduced GraphFuse, a novel hybrid graph representation learning model designed for money laundering detection in large-scale financial transaction networks. By leveraging the message passing mechanism of GNNs in synergy with the powerful global attention mechanism of Transformers, the devised model is able to capture intricate and broad fraudulent patterns spanning across finan-

Table 1: Classification performance (F1 score (%) ± std) on the selcted AML datasets. Standard deviations are calculated over 5 runs with different random seeds. We highlight the **best** and second-best results.

	AML Small-HI	**AML Small-LI**
GNN Baseline		
PNA [2]	61.20 ± 2.24	16.10 ± 2.38
MEGA-GNN Baseline		
MEGA-PNA [7]	73.10 ± 1.46	44.87 ± 1.62
+ Ego IDs	73.74 ± 1.55	45.37 ± 1.45
GT Baseline		
Multi-FraudGT [14]	76.13 ± 0.95	47.01 ± 2.22
GRAPHFUSE-PNA		
Linear Attention	66.12 ± 2.29	30.39 ± 2.24
Sparse Attention	64.24 ± 2.06	28.83 ± 1.94
Full Attention	64.29 ± 3.07	26.30 ± 1.08
GRAPHFUSE-MEGA-PNA		
Linear Attention (w/o EC,TS)	75.81 ± 1.14	46.66 ± 0.37
+ EC	75.86 ± 1.56	46.05 ± 1.06
+ TS	75.72 ± 0.85	47.13 ± 0.19
+ EC + TS	**76.89 ± 0.88**	**47.56 ± 0.36**
Sparse Attention	76.29 ± 0.99	47.47 ± 0.17
Full Attention	76.40 ± 0.28	46.93 ± 0.60

Table 2: Training and estimative inference efficiency comparison of the three GRAPHFUSE-MEGA-PNA model variants with different Global Attention Module configurations. We report the number of parameters, average training epoch time, the average throughput and the corresponding test F1 score achieved on the AML Small-HI dataset. The average throughput is defined as the number of processed transactions per second (trans/s), computed by measuring the time required to process the transactions from the validation and test sets during model training, excluding model updates. This represents a rough estimate intended for analytical comparison purposes.

	Size (# params)	**Epoch time (s)**	**Avg. Throughput (trans/s)**	**Test F1 (%)**
GRAPHFUSE-MEGA-PNA				
Linear Attention	$212.6 \cdot 10^3$	626.7 ± 4.0	$\approx 53 \cdot 10^3$	76.89 ± 0.88
Sparse Attention	$162.5 \cdot 10^3$	1441.9 ± 23.9	$\approx 21 \cdot 10^3$	76.29 ± 0.99
Full Attention	$112.2 \cdot 10^3$	1648.1 ± 2.9	$\approx 15 \cdot 10^3$	76.40 ± 0.28

cial graphs. The devised model incorporates novel structural and contextual edge encodings that capture nuanced information about the financial agents and their interactions. Additionally, we proposed three powerful attention-wise model configurations that achieve competitive results for a varying availability of computational resources and performance constraints. The extensive evaluation on publicly-available large-scale datasets alongside the leading GNN and GT baselines, demonstrate that GraphFuse outperforms or matches the performance of state-of-the-art fraud detection models. We've shown that the hybrid model, enhanced with the proposed edge encodings, leads to a 0.76 p.p. increase in minority F1 score compared to the state-of-the-art. Furthermore, we established the favorable training and inference efficiency of the Linear Attention configuration of GraphFuse. In conclusion, the empirical results demonstrate the effectiveness of the devised model and its potential for serving as a baseline for further research in increasing the money laundering detection capabilities of graph machine learning models.

Future Work

There are several promising directions to further enhance the proposed model. An important aspect to investigate would be the incorporation of structure-aware positional encodings (e.g. PEARL [29]) for increasing the expressivity of the inherently position-agnostic transformer architecture. Furthermore, to improve the model expressivity, one could consider the reverse multigraph message passing mechanism introduced in the Multi-GNN framework [8] and later refined in the MEGA-GNN framework [7]. Additionally, given our simplistic late fusion layer, more effective fusion mechanisms could be investigated.

Another direction for future work involves assessing the effectiveness of the model in real-world transactional data enriched with additional contextual characteristics. Incorporating attributes such as the geographic origin and socio-economic profile of financial agents, institutional risk scores, and statistical summaries of historical transaction behavior could substantially improve model performance and robustness in production-grade fraud detection systems. Moreover, exploring the capacity of the model to identify well-established money laundering patterns within financial networks would further validate its utility and generalizability across use cases.

6 Responsible Research

The work conducted as part of this research adheres to the collective fight against financial crime by contributing with novel GNN and Graph Transformer inspired methods to analyze financial transaction networks. By proposing a new hybrid architecture that achieves a higher performance on the task of detecting illicit transactions, this research aims to increase the efficiency of the regulators and reporting entities and therefore lower the harmful impact of money laundering on both the public and private financial sectors. Furthermore, the presented work does not offer any opportunities for subterfuge to malicious actors, as given the complexity of the model, the feature importance and learned decision boundaries are hard to derive and therefore limit the potential evasion of detection. Given that the dataset used for training and evaluating the models included in this research is purely synthetic, the conducted work does not lead to any privacy concerns or legal compliance issues. As the simulated data do not include any information about the characteristics and nature of the individuals and entities involved in the financial transactions, no bias regarding the socio-economic profile or the region of origin of the transactional agents is injected into the trained model. However, due to the artificial nature of the dataset used, the performance of the model is not guaranteed to translate one-to-one to a real data scenario. Therefore, it is the responsibility of the investigative entities to assess the performance of the proposed model in a realistic context before using it as a fraud detection tool. Moreover, the proposed machine learning model for financial fraud detection should be considered solely an aid for the decision-making process, as it does not provide a legally binding determination of a transaction's legitimacy. In order to leverage the benefits of the ML-based detection model, one would consider its integration into a broader Anti Money Laundering solution. To ensure complete reproducibility of the results, the code and configurations used for training and evaluation were made open source and are publicly available at https://github.com/mfrija/aml-graphfuse.

Use of Large Language Models

With respect to the instated policy on the use of Generative AI tools, Large Language Models were leveraged for stylistic and grammatical improvements of the written content. The scope of application was limited to finding contextual synonyms and fixing the grammatical and stylistic errors of individual sentences, excluding the task of generating passages of text. Some of the most representative prompts are presented in Appendix D.

Acknowledgments

The support and insightful guidance of the thesis supervisor and responsible professor is gratefully acknowledged. Research reported in this work was partially or completely facilitated by computational resources and support of the Delft AI Cluster (DAIC) at TU Delft [30], but remains the sole responsibility of the authors, not the DAIC team.

References

[1] United Nations Office on Drugs and Crime. Money laundering, 2022. Accessed: 2025-25-04.

[2] Giovanni Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1024–1034, 2017.

[4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[5] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019.

[6] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017.

[7] H. Çağrı Bilgi, Lydia Y. Chen, and Kubilay Atasu. Multigraph message passing with bi-directional multi-edge aggregations. *arXiv preprint*, 2024.

[8] Béni Egressy, Luc von Niederhäusern, Jovan Blanuša, Erik Altman, Roger Wattenhofer, and Kubilay Atasu. Provably powerful graph neural networks for directed multigraphs. *To appear at AAAI 2024*, 2024.

[9] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. In *AAAI Workshop on Deep Learning on Graphs: Method and Applications*, 2021.

[10] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems (NeurIPS)*, 34:28877–28888, 2021.

[11] Qidi Wu, Wenqi Zhao, Chen Yang, Haoyang Zhang, Fan Nie, Haijun Jiang, Yuchen Bian, and Jun Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[12] Juan Zhang. Only attention is needed for learning graph representations. *arXiv preprint*, 2020.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[14] Junhong Lin, Xiaojie Guo, Yada Zhu, Samuel Mitchell, Erik Altman, and Julian Shun. Fraudgt: A simple, effective, and efficient graph transformer for financial fraud

detection. In *Proceedings of the 5th ACM International Conference on AI in Finance (ICAIF)*. ACM, 2024.

[15] Erik Altman, Jovan Blanuša, Luc von Niederhäusern, Béni Egressy, Andreea Anghel, and Kubilay Atasu. Realistic synthetic financial transactions for anti-money laundering models, 2024.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. https://openai.com/research/language-unsupervised, 2018. Accessed: 2025-16-06.

[19] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform bad for graph representation?, 2021.

[20] Oscar Granados and Andrés Vargas. The geometry of suspicious money laundering activities in financial networks. *EPJ Data Science*, 11(1):6, 2022.

[21] Forman. Bochner's method for cell complexes and combinatorial ricci curvature. *Discrete Comput Geom 29*, 2003.

[22] Huu Huong Xuan Nguyen, Tran Khanh Dang, and Phat T. Tran-Truong. Money laundering detection using a transaction-based graph learning approach, 2024. 2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM).

[23] Petre-Cornel Grigorescu and Antoaneta Amza. Explainable feature engineering for multi-class money laundering classification, 2025.

[24] Jiaxuan You, Jonathan Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks, 2021.

[25] Sharan Narang, Hyung Won Chung, Yi Tay, William Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, Yanqi Zhou, Wei Li, Nan Ding, Jake Marcus, Adam Roberts, and Colin Raffel. Do transformer modifications transfer across implementations and applications? In *EMNLP 2021*, 2021.

[26] Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Hiroki Kanezashi, Toyotaro Suzumura, and Isaiah Onando Mulang'. How expressive are transformers in spectral domain for graphs? In *Trans. Mach. Learn. Res., 2022*, 2022.

[27] Shuohang Wang, Luowei Zhou, Zhe Gan, Yen-Chun Chen, Yuwei Fang, Siqi Sun, Yu Cheng, and Jingjing Liu. Cluster-former: Clustering-based sparse transformer for long-range dependency encoding. In *ACL Findings 2021, 11 pages*, 2021.

[28] U.S. Department of the Treasury. National money laundering risk assessment. Technical report, U.S. Department of the Treasury, 2022. 21 pages.

[29] Charilaos I. Kanatsoulis, Evelyn Choi, Stephanie Jegelka, Jure Leskovec, and Alejandro Ribeiro. Learning efficient positional encodings with graph neural networks, 2025.

[30] Delft AI Cluster (DAIC). The Delft AI Cluster (DAIC) RRID:SCR_025091, 2024.

[31] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

A Implementation Details

A.1 Hyperparameter Values

For each GraphFuse model variant we used a distinct set of hyperparameters in order to achieve a balance between model performance and computational time. For each model configuration, we considered the same hyperparameter values for both the Small-HI and Small-LI datasets. Table 3 presents the hyperparameter values used for the GraphFuse model variants. The values of the hyperparameters corresponding to the (MEGA-)PNA layers within the Local Message Passing Module respected the settings from [7] and [8]. In both datasets, we sampled 4-hop neighborhoods, selecting a different number of neighbors per hop for each model variant.

	GRAPHFUSE-(MEGA-)PNA		
	Linear	Sparse	Full
lr	0.0006	0.0006	0.0006
h_gnn	20	20	20
n_gnn_layers	2	2	2
do_gnn	0.083	0.083	0.083
h_attn	64	64	64
n_attn_layers	3	2	1
n_attn_heads	1	2	2
cluster_size	-	5000	-
do_attn	0.2	0.2	0.2
num_neighs	(50,50,50,50)	(50,50,25,25)	(50,25,25,25)
batch_size	512	256	128
w_ce1, w_ce2	1, 7.08	1, 7.08	1, 7.08
grad_clip_norm	1.0	1.0	1.0

Table 3: Hyperparameter settings of the GraphFuse model variants used for our experiments.

A.2 Model Training Formalization

Our hybrid graph model is trained using a supervised learning approach, which aims to minimize the weighted binary cross-entropy loss between the predicted anomaly scores and the

true labels of the transactions. Let y_{ij} be the true label of one of the transactions between nodes v_i and v_j and $\hat{y}_{ij}$ be the predicted anomaly score of this transaction. The weighted binary cross-entropy loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = -\left[w^+ \cdot y_{ij} \cdot \log(\hat{y}_{ij}) + w^- \cdot (1 - y_{ij}) \cdot \log(1 - \hat{y}_{ij})\right] \tag{24}$$

where w^+ and w^- represent the weights for the positive (fraudulent) and negative class respectively.

A.3 Resources

We implement our solutions using the PyTorch Geometric framework [31]. All experiments were conducted on the DAIC [30] - the TU Delft High Performance Computing (HPC) Cluster, using an NVIDIA A40 GPU.

B Model Evaluation Metrics

The minority class F1 score is used as the model evaluation metric in the context of financial fraud detection tasks and is defined as follows:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad \text{where} \tag{25}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{26}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{27}$$

where TP - number of correctly predicted laundering transactions, FP - number of legitimate transactions incorrectly classified as illicit and FN - number of illicit transactions incorrectly predicted as benign. By employing the minority class F1 score we are able to improve detection rates without overwhelming false positive rates and therefore obtain a balance between robustness and effectiveness.

C Datasets Information

The synthetic AML datasets [15] used for training and evaluating the developed model, consist of realistic financial transactions that span across 10 days. Details about the statistics of the datasets are present in Table 4.

Dataset	# accounts	# transactions	Illicit Ratio
AML Small HI	515K	5M	0.07%
AML Small LI	705K	7M	0.05%

Table 4: AML datasets statistics. HI indicates a higher illicit ratio and LI indicates a lower illicit ratio

D Large Language Models Prompting

The following prompts were leveraged for finding the grammatical and stylistic errors in individual sentences and replacing specific terms with contextual synonyms and reformulations:

D.1 Prompt for Grammatical and Stylistic Review

"Please check the following sentence for grammatical errors and stylistic inconsistencies. Indicate the exact errors and suggest improvements.
Sentence: #sentence#".

D.2 Prompt For Contextual Synonyms and Reformulations

"Please suggest more natural or academically appropriate alternatives for the following word or phrase, considering its context within a formal scientific paper. Keep the meaning intact but improve fluency or formality where possible.
Word or Phrase: #input#
Context: #context#".