Harnessing Logic Heterograph Learning for Financial Operational Risks: A Perspective of Cluster and Thin-tailed Distributions

Journal:	*Transactions on Knowledge and Data Engineering*
Manuscript ID	TKDE-2023-11-1906
Manuscript Type:	Regular
Keywords:	Financial Operational Risks, Operational Logic Heterographs, Thin-tailed Distributions, Semi-supervised Detection Framework, Cluster Effect

Harnessing Logic Heterograph Learning for Financial Operational Risks: A Perspective of Cluster and Thin-tailed Distributions

Guanyuan Yu, Liling Jiang, *Qing Li *Member, IEEE*

Abstract—Financial operational risks, along with credit and market risks, are a primary concern for commercial banks. However, their inherent complexity and the difficulty in defining and labeling operational activities have limited the data-driven analysis of such risks. This study introduces a deep learning framework incorporating operational logic into risk identification using a heterograph embedding network (HEN). The HEN effectively condenses complex, high-dimensional operational logic heterographs into a more manageable low-dimensional space. Within this simplified space, a Density Estimation Network (DEN) is employed to pinpoint risks, drawing on the clustering and thin-tailed traits of financial data. These components are harmoniously combined through a unified objective function, which aims to optimize both dimensionality reduction and classification decisions. Experimental evaluation using a real-world financial dataset reveals that the proposed framework surpasses several cutting-edge algorithms in terms of both practicality and effectiveness.

Index Terms—Financial Operational Risks, Operational Logic Heterographs, Thin-tailed Distribution, Cluster Effect

1 Introduction

Operational risks in banking are defined as "the risk of loss resulting from inadequate or failed internal processes, people and systems or external events", according to the *Basel Committee on Banking 2006*. This category of risk mainly encompasses events like internal fraud, external fraud, and system failures. These operational risks, alongside credit and market risks, form the triad of major threats that commercial banks face. Despite the first two risks garnering significant attention from both academia and the industry, operational risks have been somewhat overlooked in terms of research. This is alarming that poor management of these risks could result in losses amounting to trillions of dollars, as evidenced by [1]. The 1992 Barings Bank crisis, where rogue trading led to a loss of 1.4 billion USD and subsequent bankruptcy, serves as a prime example [2], [3].

Currently, most studies provide an early warning of operational risks in terms of macro-level indicators (e.g., operational value at risk) [4], [5]. With the accumulation of operational data and advances in machine learning techniques, commercial banks adopt supervised learning models and expert systems to identify operational risk events via applying more fine-grained (or micro-level) information rather than macro-level indicators [6], [7]. In principle, such approaches allow for more effective detection of operational risks, in the same way that monitoring river velocity and turbulence magnitude can provide earlier flood warnings than merely tracking water levels. However, supervised learning models and expert systems tend to rely heavily on prior knowledge, including predetermined labels and expert rules. This over-reliance results in a certain inflexibility, limiting their ability to adapt to real-world business scenarios. In particular,

- **Risks with Operational Logic:** Operational activities in banking typically involve multiple interconnected elements such as employees, personal computers, and systems. Understanding these connections or operational logic is key to identifying potential financial risks. For instance, neglecting the association between an employee account and the IP address of their personal computer may obscure issues like account sharing, where several employees use a shared IP address. In reality, account sharing coupled with other operational logic can help recognize potential operational risks. Therefore, it is crucial to examine both the individual elements and their interconnectedness (operational logic).
- **Limited Positive Labels:** Operational risks can be classified into two types: potential risks (those not yet identified) and known risks (those already identified). However, the number of identified operational risks in practice is limited [7], [8]. Although most unidentified operational activities do not pose risks, a few might still be potential threats. The scarcity of known positive instances and the unreliability of negative instances limit the effectiveness of traditional supervised learning models. Therefore, in practice, many commercial banking systems rely on expert systems. However, these systems have their drawbacks. They require significant expert involvement and often fall short when it comes to identifying unknown, emerging risk patterns.
- **Emerging Risks:** As the scope and nature of financial businesses evolve, the patterns and forms of operational risks undergo constant changes [9]. New risk patterns may emerge that internal control managers fail to detect and label in a timely manner, resulting in a delay in updating supervised learning models and expert systems. This lag can lead to actual losses before the updated systems take effect. Consequently, static supervised learning models and expert systems struggle to keep pace with the dynamic shifts in operational risk patterns.

- *Guanyuan Yu, Liling Jiang, and Qing Li are with Research Institute for Digital Economy and Interdisciplinary Science, Southwestern University of Finance and Economics, Chengdu, China.*
- *Corresponding author Qing Li (E-mail: liq_t@swufe.edu.cn).*
- *This work was supported by Financial Intelligence and Financial Engineering Key Laboratory of Sichuan Province and Financial Innovation Center.*

To tackle the first challenge, we propose using heterograph learning to capture the distinct features of operational logic in risk scenarios. Heterographs are typically employed to abstract and model complex systems where objects of various types interact in unique ways [10]. In the banking sector, operational activities often involve various interconnected entities such as employees, personal computers, and internal systems. Hence, we seek to leverage heterographs to encapsulate the operational logic of employees in their activities. For instance, verifying loan application information of a customer: bank employee A usually accesses customer database C via his personal computer B. This operational chain, denoted as employee $A \rightarrow$ personal computer $B \rightarrow$ database C, illustrates this logic, which can be represented as <employee A, login, PC B>, <PC B, add, system C> through two heterograph triplets. Compared to traditional one-hot vectors, heterographs can contain higher-order information such as structural details and meta-relations [11]. Specifically, the structural information (<A, do, B>, <B, do, C>) essentially treats the heterograph as a homogeneous graph to capture the messaging between different nodes. The meta relation information (<employee, login, PC>, <PC, add, system>) differentiates the messages in terms of node and edge types. In this study, we propose a novel heterograph embedding network (HEN) to learn the structural information and meta relations of the heterographs and encode the entire graphs into low-dimensional vectors.

To tackle the second and third challenges, we introduce a risk detection model that is grounded in the clustered and thin-tailed properties of operational risks. This approach reduces the heavy reliance on data labels and nimbly adapts to shifting risk patterns. A psychological study [12], demonstrates that employees generally establish consistent and regular behavior patterns in their daily tasks. These patterns allow us to identify anomalous operations when there are sudden or dramatic changes in sequential behaviors. This aligns with the cluster and thin-tailed characteristics of risks [13], [14], [15], which indicate that risk events are not only rare but also markedly divergent from normal occurrences. However, traditional segmentation methods, such as Kmeans [16] and DBSCAN [17], are not well-matched with the nuances of thin-tailed characteristics. These models primarily concentrate on the spatial distances between objects, which poses a challenge when addressing the unique properties of thin-tailed data. Therefore, a bespoke design is essential for risk learning that thoughtfully considers and accommodates both clustered and thin-tailed characteristics.

In this research, our focus is on modeling the dynamic operational logic of bank employees and pinpointing operational risks that align with clustered and thin-tailed characteristics. We present HenDen, a heterograph learning framework founded on these clustered and thin-tailed traits. This framework proves especially advantageous in situations where there is a limited availability of positive labels. The unique contributions of this study are threefold:

- To encapsulate the logical interconnections among operational entities, we introduce a Heterograph Embedding Network (HEN). This network is designed to concurrently learn structural information and meta relations for effective risk identification.
- In consideration of the limited positive labels, we present a Density Estimation Network (DEN). This network is integrated with the Heterograph Embedding Network to discern crucial features for pinpointing operational risks. We achieve this integration through a unified training strategy.
- Experimental assessments using large-scale, real-world banking operational logs attest to the superior performance of our proposed framework. Moreover, several controlled tests are conducted to validate the adaptability of the proposed framework in identifying anomalies within dynamic environments where new risk patterns continually emerge. The associated data and source code are publicly accessible on GitHub[1].

The remaining content of this paper is organized as follows. We summarize the related work in Section 2. We first elaborate our framework in Section 3. We then evaluate the effectiveness of our framework through extensive experiments in Section 4. Finally, we conclude our paper in Section 5.

2 Related Work

2.1 Financial Operational Risk Management

Besides credit and market risks, operational risks are the third main type of risks that commercial banks confront. Despite their potential to inflict significant losses on financial institutions, operational risks often go unnoticed or unaddressed. Prior studies primarily focus on determining capital reserves for the coming year based on the current-year loss distribution, neglecting the development of recognition models that use fine-grained, micro-level operational activity data [18], [19]. For example, [4], [5] use statistical modeling of operational risk loss distributions to measure the operational value at risk (operational VaR) and further guide the allocation of capital reserves for potential optional risks. This gap in the research hinders the fundamental mitigation of operational risks.

In other high-risk fields such as deep-water drilling, offshore oil extraction and processing, and aviation, recent research has pivoted towards constructing supervised models using fine-grained information. Methods like decision trees [20], random forests [21], artificial neural network [22] and expert systems [23], [24] have been deployed. However, these studies might not have sufficiently considered two critical factors. (a) The scarcity of positive risk labels presents a formidable challenge to the effective training of supervised models, and establishing expert rules relies heavily on the input from many professionals. (b) Operational risk patterns and forms are in constant flux, making it difficult for static expert rules and supervised models to adapt effectively.

In this study, we propose a financial operational risk detection framework rooted in the thin-tailed nature of risks. This framework leverages micro-level activities of operational logic. It not only reduces excessive reliance on data labels but also effectively captures the dynamic variations in financial risks.

2.2 Anomaly Detection Models

Anomaly detection involves detecting unusual or abnormal observations that deviate significantly from normal patterns or expectations [25]. In essence, financial operational risk identification is a form of anomaly detection. Typically, as other machine learning algorithms, anomaly detection techniques also involve two main tasks: feature representation learning and classification decision [26].

Feature Representation Learning: In anomaly detection tasks, feature representation learning is pivotal for automatically

1. Our code, data and supplementary materials can be found at https://github.com/flying-chicks/operation-risk-detection.

extracting meaningful and informative representations that facilitate the identification of abnormal observations [27]. For instance, techniques such as convolutional neural networks (CNNs) (e.g., [28], [29]), recurrent neural networks (RNNs) (e.g., [30], [31]), and principal component analysis (PCA) (e.g., [32]) are renowned for their efficacy in feature learning for anomaly detection. Recently, graph-based anomaly detection algorithms have gained prominence [33], [34]. For example, graph neural networks (GNNs) (e.g., [35], [36]) are increasingly being employed to learn from graph-structured data. Our preliminary study has revealed that financial operational risks often embody risks associated with behavioral logic involving multiple interconnected entities, such as employees, personal computers, and systems. To adeptly learn the behavioral logic features inherent in financial operational risks, this study introduces a heterograph embedding network (HEN). The primary goal of HEN is to discern and leverage key logical features for efficient identification of downstream financial risks.

Classification Decision: The process of anomaly detection involves classifying whether a specific instance or observation is normal or abnormal, using learned features [26]. Widely recognized classification techniques such as Gaussian Mixture Models (GMM) [37], K-means [16], DBSCAN [17], Decision Trees [38], One-class SVM [39], and Deep Support Vector Data Description (Deep SVDD) [40] are employed in anomaly detection. Particularly, unsupervised or semi-supervised models are favored in scenarios with limited or absent labels. Such approaches are also applicable in the identification of operational risks, which confront similar challenges. Our theoretical analysis and preliminary studies indicate that operational risks encompass both known and emerging risks. While known risks demonstrate a clustering effect, emerging risks present a thin-tailed distribution. Nevertheless, methods like K-means and DBSCAN, which make decisions based on spatial distances between clusters, fall short in addressing emerging risks. One-class SVM and Deep SVDD are capable of learning a decision boundary encompassing the normal instances within the training dataset, subsequently classifying instances outside this boundary as anomalies [39], [40]. In practice, abnormal samples often mix in a training dataset, negatively impacting the training of deep models, thus limiting their applicability. GMM potentially serves as an effective model for identifying known risks through clustering, while also addressing emerging risks by leveraging tail probabilities. However, the traditional GMM, optimized via the expectation-maximization algorithm [37], exhibits limited compatibility with representation learning models like graph neural networks. This incompatibility can hinder the representation learning models to discern key features crucial for identifying operational risks. Inspired by deep clustering algorithms [41], [42], we introduce a novel density estimation network. This network is adept at detecting both known and emerging risks and enables seamless end-to-end training in conjunction with representation learning models.

3 Proposed Learning Framework

Fig. 1 provides an overview of our proposed framework, HenDen. It begins with representing streaming operational logic of bank employees using heterographs. HEN (Heterograph Embedding Network) is then employed to learn the structural information and meta relations within these heterographs, encoding each graph into a low-dimensional vector. Finally, DEN (Density Estimation Network) is introduced to identify operational risks based on probability density distributions. HEN and DEN are trained in an end-to-end manner, employing a joint training strategy to select crucial information for risk detection. This integrated approach ensures effective risk identification within the operational logic. The primary notations utilized in this study, along with their corresponding descriptions, are presented in Table 1.

TABLE 1: Main notations and their descriptions.

Sections	Notations	Descriptions
Sec. 3.1	$\mathcal{G}_\tau = \{\mathcal{V}_\tau, \mathcal{E}_\tau, \mathcal{A}_\tau, \mathcal{R}_\tau\}$	heterographs
	$\mathcal{V}_\tau = \{v_i\}_{i=1}^N$	node set
	$\mathcal{E}_\tau = \{e_i\}_{i=1}^m$	edge set
	$\mathcal{A}_\tau$	node type set
	$\mathcal{R}_\tau$	edge type set
	$<\eta(\cdot), \pi(\cdot), \eta(\cdot)>$	meta relations
	$\eta(\cdot)$	node type mapping function
	$\pi(\cdot)$	edge type mapping function
Sec. 3.2	$\{\mathcal{G}_\tau\}_{\tau=1}^T$	streaming heterographs
	$\mathrm{HEN}_\psi(\cdot)$	heterograph embedding network
	$\boldsymbol{z} \in \mathbb{R}^d$	graph-level embedding vector
	$\mathbf{H}^{(\ell)} \in \mathbb{R}^{N\times d}$	outputs of ℓ-th HEN layer
	$\mathbf{H}^{(0)} = \mathbf{X} \in \mathbb{R}^{N\times d}$	input node features
	$\mathbf{B}_{\eta(\cdot)} \in \mathbb{R}^{d\times \frac{d}{J}}$	node-type-related query projection matrix
	$\mathbf{C}_{\eta(\cdot)} \in \mathbb{R}^{d\times \frac{d}{J}}$	node-type-related key projection matrix
	$\mathbf{D}_{\eta(\cdot)} \in \mathbb{R}^{d\times \frac{d}{J}}$	node-type-related value projection matrix
	$\mathbf{Q} \in \mathbb{R}^{N_{\eta(\cdot)}\times \frac{d}{J}}$	node-type-related query matrix
	$\mathbf{K} \in \mathbb{R}^{N_{\eta(\cdot)}\times \frac{d}{J}}$	node-type-related key matrix
	$\mathbf{V} \in \mathbb{R}^{N_{\eta(\cdot)}\times \frac{d}{J}}$	node-type-related value matrix
	$\mathbf{E}_{\pi(\cdot)} \in \mathbb{R}^{\frac{d}{J}\times \frac{d}{J}}$	edge-type-related projection matrix
	$\mathbf{F}_{\pi(\cdot)} \in \mathbb{R}^{N_{\eta(\cdot)}\times N_{\eta(\cdot)}}$	operational frequency matrix
	$\mathbf{A}^{(j,\ell)} \in \mathbb{R}^{N_{\eta(\cdot)}\times N_{\eta(\cdot)}}$	attention matrix
	$\mathbf{M}_{\eta(\cdot)}^{(j,\ell)} \in \mathbb{R}^{N_{\eta(\cdot)}\times \frac{d}{J}}$	passing message
	$\mathbf{R}_{\eta(\cdot)} \in \mathbb{R}^{N_{\eta(\cdot)}\times d}$	aggregated message
	$\oplus$	concatenation operation
	$\circ$	element-wise production
	$\mathcal{C}_{\mathrm{HEN}}$	objective function of HEN
	$\mathcal{C}_{\mathrm{str}}$	structural information constraint
	$\mathcal{C}_{\mathrm{type}}$	meta relation constraint
Sec. 3.3	$\mathrm{DEN}_\phi(\cdot)$	density estimation network
	$\mathcal{Z} = \{\boldsymbol{z}_\tau\}_{\tau=1}^T$	streaming graph-level embeddings
	$\{\mathcal{R}_k\}_{k=1}^K$	risk and risk-free labels
	$\mathcal{D}_l$	labeled dataset
	$\mathbb{P}_{\mathrm{data}}$	true data distribution
	p_{data}	true density distribution
	$\boldsymbol{\mu}$	true expectation
	$\boldsymbol{\Sigma}$	true covariance
	$\mathrm{Sigmoid}(\cdot)$	activation function
	$p_{\mathrm{data}}(\boldsymbol{z}_\tau \vert \widehat{\boldsymbol{\theta}}^{\phi_*})$	determine emerging risks
	$\boldsymbol{\gamma}_\tau^{\phi_*}$	determine known risks
	$\widetilde{\mathcal{C}}_{\mathrm{DEN}}$	objective function of DEN
Sec. 3.4	$\mathcal{L}_{\mathrm{joint}}$	joint objective function

3.1 Representing Operational Logic via Heterographs

Definition 1 (Operational logic). *The operational logic of bank employees can be defined as a complete and regular process where employees perform a series of actions to accomplish their target tasks. For example, to verify the loan application information of a customer, bank employee A usually accesses customer database C through his personal computer B. The operational chain, employee $A \rightarrow$ personal computer $B \rightarrow$ database C, indicates this operational logic. Furthermore, more complex operational*

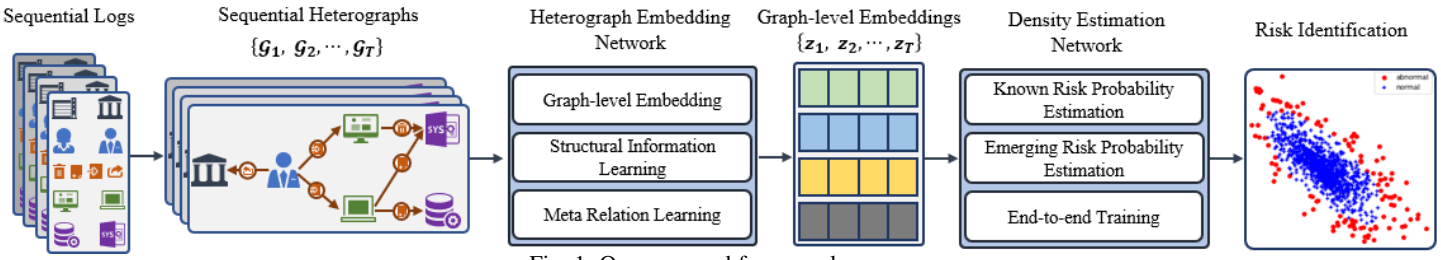


Fig. 1: Our proposed framework.

logic can be abstracted into a set of chains containing different objects that are connected to each other.

Definition 2 (**Heterographs**). *The heterogeneous graphs, consisting of different types of nodes and edges, are a powerful data structure usually used for abstracting and depicting complex systems in which objects of different types interact with each other in various ways [11]. In particular, a heterograph at time τ can be formulated as $\mathcal{G}_\tau = \{\mathcal{V}_\tau, \mathcal{E}_\tau, \mathcal{A}_\tau, \mathcal{R}_\tau\}$, composed of a set of nodes $\mathcal{V}_\tau = \{v_i\}_{i=1}^N$ and a set of edges $\mathcal{E}_\tau = \{e_i\}_{i=1}^m$. $\mathcal{A}_\tau$ and $\mathcal{R}_\tau$ denote the types of nodes and edges satisfying $|\mathcal{A}_\tau| > 1$ or $|\mathcal{R}_\tau| > 1$. Besides, we use the triple $<\eta(s), \pi(e), \eta(t)>$ to represent the meta relation from source node s to target node t, which are connected by the edge e. Here, $\eta(v) : \mathcal{V}_\tau \mapsto \mathcal{A}_\tau$ and $\pi(e) : \mathcal{E}_\tau \mapsto \mathcal{R}_\tau$ are the type mapping function for nodes and edges, respectively.*

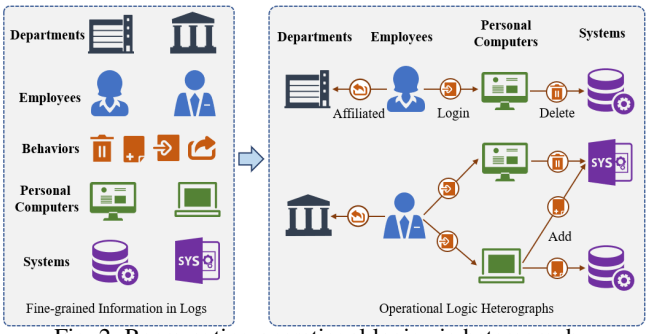


Fig. 2: Representing operational logic via heterographs.

Therefore, this study applies heterographs to model the streaming operational logic of bank employees. Specifically, the nodes represent the objects (e.g., departments, employees, personal computers, and systems) in operational logic, while edges represent the behaviors (e.g., add, delete, update) that connect different objects. As the operational logic cases in Fig. 2, bank employee *A* logins into his personal computer (PC) *B* to add a customer record to database *C*. This operational logic can be described as <employee *A*, login, PC *B*> and <PC *B*, add, system *C*> via a heterograph triplet.

3.2 Heterograph Feature Learning via Pre-training

In this study, operational risk identification aims to convert a stream of operational activities within a certain period into a set of vectors with a fixed length to further perform an unsupervised classification on these vectors. Let $\mathcal{G} = \{\mathcal{G}_\tau\}_{\tau=1}^T$ denote a stream of heterographs within a period $\tau = \{1, 2, \cdots, T\}$, where $\mathcal{G}_\tau$ indicates the operational logic heterograph of a bank employee at time τ. Here, we propose a heterograph embedding network (HEN) to convert $\mathcal{G}_\tau$ into a low-dimensional vector $\boldsymbol{z}_\tau \in \mathbb{R}^d$, that is, $\boldsymbol{z}_\tau = \text{HEN}_\psi(\mathcal{G}_\tau)$ with parameter set ψ. To convert $\mathcal{G}_\tau$ into a low-dimensional vector $\boldsymbol{z}_\tau$, HEN first calculates the node embeddings of operational logic heterographs and then combines these node embeddings.

3.2.1 Graph-level Embedding

As the previous example in Section 3.1, the heterograph triple (<employee *A*, login, PC *B*>, <PC *B*, add, system *C*>) comprises two kinds of information: (a) the structural information (<*A*, do, *B*>, <*B*, do, *C*>) essentially treats the heterograph as a homogeneous graph to capture the messaging between different nodes. (b) the meta relation information (<employee, login, PC>, <PC, add, system>) differentiates the messages in terms of node types and edge types. To learn such two features, our HEN employs a multi-head attention mechanism related to node types and edge types.

In particular, the output of ℓ-th HEN layer is denoted as $\mathbf{H}^{(\ell)}$ where $\ell = \{0, 1, \cdots, L\}$, and $\mathbf{H}^{(0)} = \mathbf{X}$ means the input node features. Here, $\mathbf{H}^{(\ell+1)}$ is output through adding the aggregation message $\mathbf{R}$ to $\mathbf{H}^{(\ell)}$.

$$\mathbf{H}_{\eta(t)}^{(\ell+1)} = \alpha \mathbf{H}_{\eta(t)}^{(\ell)} + (1-\alpha)\mathbf{R}_{\eta(t)} \in \mathbb{R}^{N_{\eta(t)} \times d},$$
$$\mathbf{R}_{\eta(t)} = \oplus_{j=1}^{J} \mathbf{M}_{\eta(t)}^{j} \in \mathbb{R}^{N_{\eta(t)} \times d},$$
$$\mathbf{M}_{\eta(t)}^{j} = \mathbf{A}\mathbf{V} \in \mathbb{R}^{N_{\eta(t)} \times \frac{d}{J}}.$$

In the above equation, $\alpha = \text{sigmoid}(w_1) \in \mathbb{R}^1$ is a scalar gate that flexibly balances $\mathbf{H}_{\eta(t)}^{(\ell)}$ and $\mathbf{R}_{\eta(t)}$, and $w_1 \in \mathbb{R}^1$ is a trainable scalar. $N_{\eta(v)}$ is the number of nodes v belonging to node type $\eta(v)$, and d is the feature dimension. $\oplus_{j=1}^{J} \mathbf{M}_{\eta(t)}^{j}$ denotes concatenation of J message matrices. $\mathbf{M}_{\eta(t)}^{j}$ performs message passing from neighbors to the focal node t in the j-th attention head to model structural information. $\mathbf{A}$ and $\mathbf{V}$ are the attention weight and value matrices, both of which are calculated through the following equations,

$$\mathbf{A} = \text{Softmax}[\mathbf{Q}\mathbf{E}_{\pi(e)}\mathbf{K}^\top \circ \frac{\mathbf{F}_{\pi(e)}^\top}{\sqrt{d/J}}] \in \mathbb{R}^{N_{\eta(t)} \times N_{\eta(s)}},$$
$$\mathbf{Q} = \mathbf{H}_{\eta(t)}\mathbf{B}_{\eta(t)} \in \mathbb{R}^{N_{\eta(t)} \times \frac{d}{J}},$$
$$\mathbf{K} = \mathbf{H}_{\eta(s)}\mathbf{C}_{\eta(s)} \in \mathbb{R}^{N_{\eta(s)} \times \frac{d}{J}},$$
$$\mathbf{V} = \mathbf{H}_{\eta(s)}\mathbf{D}_{\eta(s)} \in \mathbb{R}^{N_{\eta(s)} \times \frac{d}{J}}.$$

In the above equation, our HEN uses three trainable node-type-related projection matrices $\mathbf{B}_{\eta(t)}, \mathbf{C}_{\eta(s)}, \mathbf{D}_{\eta(s)} \in \mathbb{R}^{d \times \frac{d}{J}}$ to map the target node feature $\mathbf{H}_{\eta(t)} \in \mathbb{R}^{N_{\eta(t)} \times d}$ and the source node feature $\mathbf{H}_{\eta(s)} \in \mathbb{R}^{N_{\eta(s)} \times d}$ to query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ for the purpose of preserving meta relations. Notably, we record the edge types and their directions by using a trainable

edge-type-related projection matrix $\mathbf{E}_{\pi(e)} \in \mathbb{R}^{\frac{d}{J} \times \frac{d}{J}}$. Consequently, $\psi = \{\mathbf{B}^{(\ell)}, \mathbf{C}^{(\ell)}, \mathbf{D}^{(\ell)}, \mathbf{E}^{(\ell)}\}_{\ell=1}^{L}$ Besides, to record operational frequencies, we leverage a given edge weight matrix $\mathbf{F}_{\pi(e)} \in \mathbb{R}^{N_{\eta(s)} \times N_{\eta(t)}}$, each element of which is normalized to the interval $[0, 1]$ to guarantee stable training.

After all node features in the heterograph $\mathcal{G}_\tau$ are mapped into the low-dimension space, we can obtain the graph-level embedding $\boldsymbol{z}_\tau$ as follows.

$$\boldsymbol{z}_\tau = \mathrm{AVG}(\mathbf{H}^{(L)}).$$

3.2.2 Feature Representation Constraints

Claim 1 (**Constraints help learning desired embeddings**). *Let $\mathbf{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_T]^\top$ denote a trainable embedding matrix, $f(\cdot)$ indicate an invertible probability prediction function, $f^{-1}(\cdot)$ stand for its corresponding inverse function. $\hat{y}_{\tau,k} = f(\boldsymbol{h}_\tau)$ means the predicted probability of the vector $\boldsymbol{h}_\tau$ belonging to k-th class where $k = \{1, 2, \cdots, K\}$, $y_{\tau,k}$ represents ground truth label. Suppose both $f(\cdot)$ and $f^{-1}(\cdot)$ are first-order derivable. Here,*

$$\textit{when minimizing } \mathcal{L} = -\frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=1}^{K} y_{\tau,k}[\log(\hat{y}_{\tau,k})],$$

$\boldsymbol{h}_\tau$ can record the information of $y_{\tau,k}$.

Please see Proof 1 of Appendix for the proof of the above claim, according which, we can force the graph-level embedding $\boldsymbol{z}_\tau$ to preserve the operational logic as much as possible by maximizing node proximity reconstruction and predicting node and edge types.

$$\mathcal{C}_{\mathrm{HEN}}(\mathcal{G}_\tau) = \delta\mathcal{C}_{\mathrm{str}} + (1-\delta)\mathcal{C}_{\mathrm{type}},$$

where $\delta = \mathrm{sigmod}(w_2) \in \mathbb{R}^1$ is used for trading off $\mathcal{C}_{\mathrm{str}}$ and $\mathcal{C}_{\mathrm{type}}$, and $w_2 \in \mathbb{R}^1$ is a trainable scalar. Here, $\mathcal{C}_{\mathrm{str}}$ aims at capturing the structural information through reconstructing the node proximity.

$$\mathcal{C}_{\mathrm{str}} = -\sum_{\eta(s),\eta(t)\in\mathcal{A}_\tau} \mathbf{S} \circ \log[\mathrm{FC}(\mathbf{O}_{\eta(s)}\mathbf{O}_{\eta(t)}^\top)].$$

In the above equation, $\mathrm{FC}(\cdot)$ is a fully-connected classification network with a linear hidden layer and a $\mathrm{sigmoid}$ output layer that predicts the probabilities of edges between the nodes s and t. $\mathbf{S} \in \mathbb{R}^{N_{\eta(s)} \times N_{\eta(t)}}$ measures the node proximity, whose corresponding element is equal to 1 if there exists an edge and 0 otherwise. $\mathcal{C}_{\mathrm{type}}$ focuses on guiding our HEN to record the meta relations via predicting the node and edge types.

$$\mathcal{C}_{\mathrm{type}} = \underbrace{-\sum_{\eta(v)\in\mathcal{A}_\tau} \boldsymbol{y}_{\eta(v)} \log[\mathrm{FC}(\mathbf{O}_{\eta(v)})]}_{\text{node type prediction}} \underbrace{-\sum_{\pi(e)\in\mathcal{R}_\tau} \boldsymbol{y}_{\pi(e)} \log[\mathrm{FC}(\mathbf{O}_{\eta(s)} \oplus \mathbf{O}_{\eta(t)})]}_{\text{edge type prediction}},$$

where $\boldsymbol{y}_{\eta(v)} \in \mathbb{R}^{N_{\eta(v)}}$ and $\boldsymbol{y}_{\pi(e)} \in \mathbb{R}^{N_{\pi(e)}}$ stand for the ground truth node and edge types, respectively.

3.3 Operational Risk Identification

3.3.1 Distribution-based Identification Strategy

Definition 3 (**Known risks**). *Bank risk managers often review and identify such risks based on relevant risk features [43]. Furthermore, the feature vector samples $\mathcal{Z} = \{\boldsymbol{z}_\tau\}_{\tau=1}^{T}$ tend to group together into K risk and risk-free clusters $\{\mathcal{R}_k\}_{k=1}^{K}$ as a result of their main feature heterogeneity of K categories.*

Claim 2 (**Known risks display cluster effects**). *Let $\{\mathcal{R}_k\}_{k=1}^{K}$ represent K known labels (including the risk-free labels), and $\mathcal{Z} = \{\boldsymbol{z}_\tau\}_{\tau=1}^{T}$ represent the feature vector samples used to identify K labels. $p_{\mathrm{data}}(\cdot)$ is probability density function that corresponds to $\mathcal{P}_{\mathrm{data}}$, which depicts the true data distribution. This study assumes that $\mathcal{R}_k$ is a function of its corresponding main feature vector $\boldsymbol{\theta}_k$, i.e., $\mathcal{R}_k = f_k(\boldsymbol{\theta}_k)$ ($f_k(\cdot)$ is a real-valued function). Let $\mathcal{Z}_k = \{\boldsymbol{z}_\tau\}_{\tau=\alpha_1}^{\alpha_{T_k}}$ stand for a set of feature vector samples belonging to $\mathcal{R}_k$. In this case, the feature vector sample $\boldsymbol{z}_\tau$ can be determined by the feature vector $\boldsymbol{\theta}_k$, i.e., $\boldsymbol{z}_\tau = \boldsymbol{g}_k(\boldsymbol{\theta}_k) + \boldsymbol{\epsilon}_{\tau,k}(\boldsymbol{\theta}_k)$ (where $\tau = \{\alpha_1, \ldots, \alpha_{T_k}\}$; $\boldsymbol{g}_k(\boldsymbol{\theta}_k)$ is a real-valued function concerning $\boldsymbol{\theta}_k$; $\boldsymbol{\epsilon}_{\tau,k}(\boldsymbol{\theta}_k)$ is an independent random vector dependent on $\boldsymbol{\theta}_k$ with a mean of $\mathbf{0}$). Additionally, $\boldsymbol{g}_k(\boldsymbol{\theta}_k)$ and $\boldsymbol{\epsilon}_{\tau,k}(\boldsymbol{\theta}_k)$ satisfy the following two conditions: (a) $\mathbb{P}(\|\boldsymbol{\epsilon}_{\tau,k}(\boldsymbol{\theta}_k)\| > \varepsilon_1) < \varepsilon_2$ (ε_1 and ε_2 are relatively small positive constants). (b) for $\forall \boldsymbol{z}_{\tau_1} \in \mathcal{Z}_{k_1}, \forall \boldsymbol{z}_{\tau_2} \in \mathcal{Z}_{k_2}$ (where $k_1 \neq k_2$), $\mathbb{P}(\|\boldsymbol{g}_{k_1}(\boldsymbol{\theta}_{k_1}) - \boldsymbol{g}_{k_2}(\boldsymbol{\theta}_{k_2})\| > C_1\|\boldsymbol{\epsilon}_{\tau_1,k_1} - \boldsymbol{\epsilon}_{\tau_2,k_2}\|) \mapsto 1$ and $\mathbb{P}(\|\boldsymbol{g}_{k_1}(\boldsymbol{\theta}_{k_1}) - \boldsymbol{g}_{k_2}(\boldsymbol{\theta}_{k_2})\| > C_2) \mapsto 1$ (C_1 and C_2 are relatively large positive constants). Based on the above assumptions, we can find that feature vector samples belonging to different risk classes will likely cluster into different groups, which maintain a large spatial distance between them. Please see Proof 2 of Appendix for the detailed proof of the above claim.*

Definition 4 (**Emerging risks**). *As a result of the development of financial businesses, the patterns and forms of financial risks are constantly changing, leading to the emergence of new and unknown risks [9]. Such risks occur infrequently and are noticeably distinct from known risks.*

Claim 3 (**Emerging risks exhibit thin-tailed distributions**). *This study focuses on emerging risks, denoted as $\mathcal{R}_p$, with its corresponding feature vector samples represented by $\mathcal{Z}_p = \{\tilde{\boldsymbol{z}}_\tau\}_{\tau=1}^{T_p}$. Here, $\tilde{\boldsymbol{z}}_\tau = \boldsymbol{g}(\boldsymbol{\theta}_p) + \boldsymbol{\epsilon}_{\tau,p}(\boldsymbol{\theta}_p)$ for $\tau = \{1, \ldots, T_p\}$, where $\boldsymbol{\theta}$ means the latent feature vector that truly determines $\mathcal{R}_p$, and $\boldsymbol{g}(\cdot)$ is a real-valued function. From the discussion of known risks, it is known that for large constant C, $\forall \tilde{\boldsymbol{z}}_{\tau_1} \in \mathcal{Z}_p$ and $\forall \boldsymbol{z}_{\tau_2} \in \mathcal{Z}_k$, $\mathbb{P}(\|\tilde{\boldsymbol{z}}_{\tau_1} - \boldsymbol{z}_{\tau_2}\| > C) \to 1$. Therefore, $\tilde{\boldsymbol{z}}_\tau$ will be far from the known risk groups. Since $T_p \ll T$, the feature vector samples corresponding to unknown risks occur with low frequency. By using frequency as a proxy for probability, it is easy to see that $p_{\mathrm{data}}(\tilde{\boldsymbol{z}})$ will be relatively small. Consequently, the feature vector samples in $\mathcal{Z}_p$ will fall into the low probability density region of $\mathcal{P}_{\mathrm{data}}$.*

Definition 5 (**Gaussian mixture distribution**). *Famous statistical studies like [37], [44] demonstrate that Gaussian mixture distribution has two important characteristics: (a) data generated from a Gaussian mixture distribution automatically clusters into distinct groups; and (b) Gaussian mixture distribution has thin-tailed nature, which means that a small portion of data is sparsely scattered in the regions of low probability density.*

From the above definitions and claims, we can figure out that Gaussian mixture distribution can reflect the cluster and

thin-tailed nature of operational risks. Consequently, this study assumes that the feature vector samples of financial risks are drawn from a Gaussian mixture distribution, denoted by $p_{\text{data}}(\boldsymbol{z}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\pi}_k$ represents the probability of the feature vector sample $\boldsymbol{z}$ belonging to the k-th risk cluster, and satisfies $\sum_{k=1}^{K} \boldsymbol{\pi}_k = 1$. $\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ represents a Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ for the k-th risk cluster. $\boldsymbol{\theta} = \{\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k; k = 1, \ldots, K\}$ represents the parameter set.

Given a set of observable and independent feature vector samples $\mathcal{Z} = \{\boldsymbol{z}_\tau\}_{\tau=1}^{T}$, to find the optimal parameters $\boldsymbol{\theta}^*$, this study aims to maximize the joint probability density function of the feature vector samples, i.e., $\prod_{\tau=1}^{T} p_{\text{data}}(\boldsymbol{z}_\tau|\boldsymbol{\theta})$, using maximum likelihood estimation. The maximization optimization problem is equivalent to minimizing the function $\mathcal{C}(\mathcal{Z}) := -\frac{1}{T}\log(\prod_{\tau=1}^{T} p_{\text{data}}(\boldsymbol{z}_\tau|\boldsymbol{\theta})) = -\frac{1}{T}\sum_{\tau=1}^{T}\log[p_{\text{data}}(\boldsymbol{z}_\tau|\boldsymbol{\theta})]$. By introducing latent variables $\kappa_\tau = k$ (indicating that the feature vector sample $\boldsymbol{z}_\tau$ is generated from the k-th Gaussian distribution), $\mathcal{C}(\mathcal{Z})$ can be further transformed into the following expression,

$$\begin{aligned}\mathcal{C}(\mathcal{Z}) &= -\frac{1}{T}\sum_{\tau=1}^{T}\log(p_{\text{data}}(\boldsymbol{z}_\tau|\boldsymbol{\theta})) \\ &= -\frac{1}{T}\sum_{\tau=1}^{T}\log\left(\sum_{k=1}^{K} p_{\text{data}}(\boldsymbol{z}_\tau, \kappa_\tau = k|\boldsymbol{\theta})\right) \\ &= -\frac{1}{T}\sum_{\tau=1}^{T}\log\left(\sum_{k=1}^{K} \mathbb{P}(\kappa_\tau = k) p_{\text{data}}(\boldsymbol{z}_\tau|\kappa_\tau = k, \boldsymbol{\theta})\right).\end{aligned}$$

$\mathbb{P}(\kappa_\tau = k)$ is unknown since κ_τ is an unobservable latent variable. This study constructs a density estimation network $\text{DEN}_{\boldsymbol{\phi}}(\boldsymbol{z}_\tau)$ with parameter $\boldsymbol{\phi}$ to estimate $\mathbb{P}(\kappa_\tau = k)$.

$$\begin{aligned}\boldsymbol{h}_\tau^{(1)} &= \text{BatchNorm}[\text{ReLu}(\mathbf{W}^{(1)}\boldsymbol{z}_\tau + \boldsymbol{b}^{(1)})] \\ \boldsymbol{h}_\tau^{(2)} &= \text{BatchNorm}[\text{ReLu}(\mathbf{W}^{(2)}\boldsymbol{h}_\tau^{(1)} + \boldsymbol{b}^{(2)})], \\ &\cdots \\ \boldsymbol{h}_\tau^{(L)} &= \text{BatchNorm}[\text{ReLu}(\mathbf{W}^{(L)}\boldsymbol{h}_\tau^{(L-1)} + \boldsymbol{b}^{(L)})], \\ \boldsymbol{\gamma}_\tau^{\boldsymbol{\phi}} &= (\gamma_{\tau,1}^{\boldsymbol{\phi}}, \ldots, \gamma_{\tau,K}^{\boldsymbol{\phi}})^T = \text{Softmax}(\boldsymbol{h}_\tau^{(L)}) \in \mathbb{R}^K.\end{aligned}$$

Here, $\text{ReLu}(\boldsymbol{z}_\tau) = \max(0, \boldsymbol{z}_\tau)$ and $\text{BatchNorm}(\cdot)$ represents the batch normalization, which ensures that the outputs of each layer follow a standard Gaussian distribution, thereby improving the training stability [45]. $\text{Softmax}(\boldsymbol{z}_{\tau,k}) = \frac{e^{\boldsymbol{z}_{\tau,k}}}{\sum_{k=1}^{K} e^{\boldsymbol{z}_{\tau,k}}}$ is applied to element $\boldsymbol{z}_{\tau,k}$ to obtain the probability distribution over K classes. The parameter set is denoted as $\boldsymbol{\phi} = \{\mathbf{W}^{(l)}; \boldsymbol{b}^{(l)}\}_{l=1}^{L}$, where $\mathbf{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ represent the weights and biases for the l-th layer, respectively. Let $Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau) = \gamma_{\tau,k}^{\boldsymbol{\phi}}$ and satisfies $Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau) > 0$ and $\sum_{k=1}^{K} Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau) = 1$. To approximate $Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{x}_\tau)$ to $\mathbb{P}(\kappa_\tau = k)$, $\mathcal{C}(\boldsymbol{\theta})$ can be further transformed into the following equation.

$$\begin{aligned}&\mathcal{C}(\mathcal{Z}) = -\frac{1}{T}\sum_{\tau=1}^{T}\log(p_{\text{data}}(\boldsymbol{z}_\tau|\boldsymbol{\theta})) \\ &= -\frac{1}{T}\sum_{\tau=1}^{T}\log\left(\sum_{k=1}^{K} Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau)\frac{p_{\text{data}}(\boldsymbol{z}_\tau, \kappa_\tau = k|\boldsymbol{\theta})}{Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau)}\right) \\ &= -\frac{1}{T}\sum_{\tau=1}^{T}\log\left\{\mathbb{E}_{\kappa_\tau Q_{\text{DEN}}(\cdot|\boldsymbol{z}_\tau)}\left(\frac{p_{\text{data}}(\boldsymbol{z}_i, \kappa_\tau = k|\boldsymbol{\theta})}{Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau)}\right)\right\} \\ &\leq -\frac{1}{T}\sum_{\tau=1}^{T}\mathbb{E}_{\kappa_\tau Q_{\text{DEN}}(\cdot|\boldsymbol{z}_\tau)}\left\{\log\left(\frac{p_{\text{data}}(\boldsymbol{z}_\tau, \kappa_\tau = k|\boldsymbol{\theta})}{Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau)}\right)\right\} \\ &= -\frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=1}^{K} Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau)\log\left(\frac{p_{\text{data}}(\boldsymbol{z}_\tau, \kappa_\tau = k|\boldsymbol{\theta})}{Q_{\text{DEN}}(\kappa_\tau = k|\boldsymbol{z}_\tau)}\right) \\ &:= \mathcal{C}_{Q_{\text{DEN}}}(\mathcal{Z}).\end{aligned}$$

Therefore, we can minimize $\mathcal{C}(\mathcal{Z})$ by minimizing the upper bound of $\mathcal{C}_{Q_{\text{DEN}}}(\mathcal{Z})$. Since $\gamma_{\tau,k}^{\boldsymbol{\phi}}$ is unrelated to $\boldsymbol{\theta}$, we can minimize the following expression,

$$\begin{aligned}&\mathcal{C}_{Q_{\text{DEN}}}(\mathcal{Z}) := -\frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=1}^{K}\gamma_{\tau,k}^{\boldsymbol{\phi}}\log(p_{\text{data}}(\boldsymbol{z}_\tau, \kappa_\tau = k|\boldsymbol{\theta})) \\ &= -\frac{1}{T}\sum_{\tau=1}^{T}\sum_{k=1}^{K}\gamma_{\tau,k}^{\boldsymbol{\phi}}\log(p_{\text{data}}(\boldsymbol{z}_\tau|\kappa_\tau = k, \boldsymbol{\theta})\mathbb{P}(\kappa_\tau = k|\boldsymbol{\theta})) \\ &= -\frac{1}{T}\sum_{\tau=1}^{T}\sum_{\tau=1}^{K}\gamma_{\tau,k}^{\boldsymbol{\phi}}\log(\mathcal{N}(\boldsymbol{z}_\tau|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\boldsymbol{\pi}_k)\end{aligned}$$

By setting $\frac{\partial \mathcal{C}_{Q_{\text{DEN}}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$, we can minimize $\mathcal{C}_{Q_{\text{DEN}}}(\boldsymbol{\theta})$ and obtain the estimated parameters $\widehat{\boldsymbol{\theta}}^{\boldsymbol{\phi}} = \{\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}}, \widehat{\boldsymbol{\Sigma}}_k^{\boldsymbol{\phi}}, \widehat{\boldsymbol{\pi}}_k^{\boldsymbol{\phi}}\}$.

$$\begin{aligned}\widehat{\boldsymbol{\pi}}_k^{\boldsymbol{\phi}} &= \frac{1}{T}\sum_{\tau=1}^{T}\gamma_{\tau,k}^{\boldsymbol{\phi}}, \quad \widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}} = \frac{\sum_{\tau=1}^{T}\gamma_{\tau,k}^{\boldsymbol{\phi}}\boldsymbol{z}_\tau}{\sum_{\tau=1}^{T}\gamma_{\tau,k}^{\boldsymbol{\phi}}}, \\ \widehat{\boldsymbol{\Sigma}}_k^{\boldsymbol{\phi}} &= \frac{\sum_{\tau=1}^{T}\gamma_{\tau,k}^{\boldsymbol{\phi}}(\boldsymbol{z}_\tau - \widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}})(\boldsymbol{z}_\tau - \widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}})^\top}{\sum_{\tau=1}^{N}\gamma_{\tau,k}^{\boldsymbol{\phi}}}.\end{aligned}$$

Here, $p_{\text{data}}(\boldsymbol{z}_\tau|\widehat{\boldsymbol{\theta}}^{\boldsymbol{\phi}_*})$ and $\boldsymbol{\gamma}_\tau^{\boldsymbol{\phi}_*}$ are the optimal probability density value and risk class probability, respectively. We can identify emerging risk and known risk via the following two steps: (a) $p_{\text{data}}(\boldsymbol{z}_\tau|\widehat{\boldsymbol{\theta}}^{\boldsymbol{\phi}_*}) >$ predefined threshold indicates $\boldsymbol{z}_\tau$ belongs to an unknown risk category; (b) $p_{\text{data}}(\boldsymbol{z}_\tau|\widehat{\boldsymbol{\theta}}^{\boldsymbol{\phi}_*}) \leq$ predefined threshold and $\boldsymbol{\gamma}_\tau^{\boldsymbol{\phi}_*}$ determine the known risk category.

3.3.2 Leveraging Limited Labels

Claim 4 ($\mathcal{C}_{Q_{\text{DEN}}}$ **causes unexpected performance**). *It is well known that the feature vector samples belonging to the k-th risk class are primarily clustered around their true expectation $\boldsymbol{\mu}_k$, which can stand for the k-th risk label based on the characteristics of the Gaussian distribution. Recall that $\mathcal{R}_k$ is the set of feature vector samples truly belonging to k-th risk cluster, and $\mathcal{R}_k^c$ represents the set of feature vector samples not belonging to risk cluster k, satisfying $\mathcal{R}_k \cup \mathcal{R}_k^c = \mathcal{R}$ and $\mathcal{R}_k \cap \mathcal{R}_k^c = \emptyset$ (an empty set). In this case, $\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}}$ can be further expressed as,*

$$\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}} = \frac{\sum_{\tau=1}^{T}\gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k}(\boldsymbol{z}_\tau)\boldsymbol{z}_\tau + \sum_{\tau=1}^{T}\gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k^c}(\boldsymbol{z}_\tau)\boldsymbol{z}_\tau}{\sum_{\tau=1}^{T}\gamma_{\tau,k}^{\boldsymbol{\phi}}}.$$

In the above equation, $I_A(\boldsymbol{z}) = \begin{cases} 1, & \boldsymbol{z} \in A, \\ 0, & \boldsymbol{z} \notin A. \end{cases}$ *is an indicator function suggesting whether* $\boldsymbol{z}$ *belongs to a non-empty set* $A \subset \mathcal{Z}$*. Minimizing* $\|\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}} - \boldsymbol{\mu}_k\|$ *is equivalent to* $\gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k}(\boldsymbol{z}_\tau) \mapsto 1$, $\gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k^c}(\boldsymbol{z}_\tau) \mapsto 0$*. Essentially, this requires* $\mathrm{DEN}_{\boldsymbol{\phi}}(\boldsymbol{z}_\tau)$ *to possess perfect classification performance. However, in the actual training process of* $\mathrm{DEN}_{\boldsymbol{\phi}}(\boldsymbol{z}_\tau)$*, the interactions between inferior* $\boldsymbol{\phi}$ *and* $\mathcal{C}_{Q_{\mathrm{DEN}}}$ *prevent* $\mathrm{DEN}_{\boldsymbol{\phi}}(\boldsymbol{z}_\tau)$ *from converging.*

Please see Proof 3 of Appendix for the detailed proof of the above claim, which suggests that $\mathcal{C}_{Q_{\mathrm{DEN}}}$ causes unstable training and unexpected performance. To address such a problem, this study introduces few labeled data into the original dataset. Therefore, the original dataset can be divided into two subsets: labeled subset $\mathcal{D}_l$ and unlabeled subset $\mathcal{D}_u$, where $|\mathcal{D}_l| \ll |\mathcal{D}_u|$ and $|\mathcal{D}_l| + |\mathcal{D}_u| = T$. In addition, $\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}}$ can be reformulated as,

$$\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}} = \underbrace{\frac{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k \cap \mathcal{D}_u}(\boldsymbol{z}_\tau) \boldsymbol{z}_\tau}{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}}}}_{\text{unlabeled subset}} + \underbrace{\frac{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k \cap \mathcal{D}_l}(\boldsymbol{z}_\tau) \boldsymbol{z}_\tau}{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}}}}_{\text{labeled subset}} + \frac{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k^c}(\boldsymbol{z}_\tau) \boldsymbol{z}_\tau}{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}}}.$$

This equation suggests that $\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}}$ is partially determined by the labeled subset. Because $\mathcal{D}_l$ provides additional label information enforcing $\gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k \cap \mathcal{D}_l}(\boldsymbol{z}_\tau) \mapsto 1$, we can make $\widehat{\boldsymbol{\mu}}_k^{\boldsymbol{\phi}}$ converge to $\boldsymbol{\mu}_k$. This study introduces a cross-entropy loss function to utilize the label information in $\mathcal{D}_l$.

$$\mathcal{C}_{\mathrm{DEN}}(\mathcal{Z}) = \underbrace{\beta_1 \mathcal{C}_{Q_{\mathrm{DEN}}}(\mathcal{Z})}_{\text{unsupervised constraint}} - \underbrace{\frac{\beta_2}{|\mathcal{D}_l|} \sum_{\tau=1}^{|\mathcal{D}_l|} \sum_{k=1}^{K} I_{\mathcal{R}_k \cap \mathcal{D}_l}(\boldsymbol{z}_\tau) \log(\gamma_{\tau,k}^{\boldsymbol{\phi}})}_{\text{supervised constraint}},$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2)^\top$ are trade-off weights and satisfy $\beta_1, \beta_2 > 0, \beta_1 + \beta_2 = 1$. Larger β_2 explicitly guides $\mathrm{DEN}_{\boldsymbol{\phi}}(\cdot)$ to approach 1 quickly during the training process. In addition, Larger $|\mathcal{D}_l|$ can enhance the training stability and prediction performance of $\mathrm{DEN}_{\boldsymbol{\phi}}(\cdot)$.

3.3.3 Mini-batch Gradient Descent Optimization

The entire dataset $\mathcal{Z}$ is randomly and equally divided into B subsets $\mathcal{S} = \{\mathcal{S}_b\}_{b=1}^B$. Then, the mini-batch estimated value of $\boldsymbol{\mu}$ is obtained as follows,

$$\widetilde{\boldsymbol{\mu}}_{k,\mathcal{S}_b}^{\boldsymbol{\phi}} = (1-\eta)\widehat{\boldsymbol{\mu}}_{k,\mathcal{S}_b}^{\boldsymbol{\phi}} + \eta \widehat{\boldsymbol{\mu}}_{k,l},$$

$$\widehat{\boldsymbol{\mu}}_{k,\mathcal{S}_b}^{\boldsymbol{\phi}} = \underbrace{\frac{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k \cap \mathcal{D}_u \cap \mathcal{S}_b}(\boldsymbol{z}_\tau) \boldsymbol{z}_\tau}{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}}}}_{\text{unlabeled subset}} + \frac{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k^c \cap \mathcal{S}_b}(\boldsymbol{z}_\tau) \boldsymbol{z}_\tau}{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}}} + \underbrace{\frac{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}} I_{\mathcal{R}_k \cap \mathcal{D}_l \cap \mathcal{S}_b}(\boldsymbol{z}_\tau) \boldsymbol{z}_\tau}{\sum_{\tau=1}^T \gamma_{\tau,k}^{\boldsymbol{\phi}}}}_{\text{labeled subset}},$$

$$\widehat{\boldsymbol{\mu}}_{k,l} = \frac{\sum_{\boldsymbol{z}_\tau \in (\mathcal{R}_k \cap \mathcal{D}_l)} \boldsymbol{z}_\tau}{|\mathcal{R}_k \cap \mathcal{D}_l|}, \ \eta = \frac{1}{1 + e^{|\sigma_{\mathcal{Z}} - \sigma_{\mathcal{D}_l}|}}.$$

From the above equation, we can find that $\widehat{\boldsymbol{\mu}}_{k,\mathcal{S}_b}$ is a robust estimated value of expectation since it combines the mini-batch sample expectation $\widehat{\boldsymbol{\mu}}_{k,\mathcal{S}_b}^{\boldsymbol{\phi}}$ and labelled sample expectation $\widehat{\boldsymbol{\mu}}_{k,l}$ that is predetermined. $\eta \in (0, 0.5]$ is a predefined weight measuring the contribution of $\widehat{\boldsymbol{\mu}}_{k,l}$ to $\widetilde{\boldsymbol{\mu}}_{k,\mathcal{S}_b}^{\boldsymbol{\phi}}$. Larger η indicates $\widetilde{\boldsymbol{\mu}}_{k,\mathcal{S}_b}^{\boldsymbol{\phi}}$ is closer to $\widehat{\boldsymbol{\mu}}_{k,l}$. $\sigma_{\mathcal{Z}}$ and $\sigma_{\mathcal{D}_l}$ represent the sample variance of the entire dataset $\mathcal{Z}$ and the labeled dataset $\mathcal{D}_l$, respectively. If $\mathcal{D}_l$ is uniformly distributed within $\mathcal{Z}$, $\mathcal{D}_l$ can reflect $\mathcal{Z}$. In this case, $|\sigma_{\mathcal{Z}} - \sigma_{\mathcal{D}_l}| \mapsto 0$ and $\eta \mapsto 0.5$. If the variance of $\mathcal{D}_l$ differs from the counterpart of $\mathcal{Z}$, $|\sigma_{\mathcal{Z}} - \sigma_{\mathcal{D}_l}| \mapsto +\infty$ and $\eta \mapsto 0$.

Finally, we provide the mini-bath version of $\mathcal{C}_{\mathrm{DEN}}(\mathcal{Z})$ as follows,

$$\widetilde{\mathcal{C}}_{\mathrm{DEN}}(\mathcal{S}) = \underbrace{-\frac{\beta_1}{T} \sum_{\mathcal{S}_b \in \mathcal{S}} \sum_{\boldsymbol{z}_\tau \in \mathcal{S}_b} \log p_{\mathrm{data}}(\boldsymbol{z}_\tau | \widehat{\boldsymbol{\theta}}^{\boldsymbol{\phi}})}_{\text{entire data}} \underbrace{- \frac{\beta_2}{|\mathcal{D}_l|} \sum_{\mathcal{S}_b \in \mathcal{S}} \sum_{\tau=1}^{|\mathcal{D}_l \cap \mathcal{S}_b|} \sum_{k=1}^{K} I_{\mathcal{R}_k \cap \mathcal{D}_l \cap \mathcal{S}_b}(\boldsymbol{z}_\tau) \log(\gamma_{\tau,k}^{\boldsymbol{\phi}})}_{\text{labeled subset}}.$$

Algorithm 1: Training Procedure of HenDen

Data: Operational log heterographs, limited risk labels.
Input: HenDen with randomly initialized parameters.
Output: Emerging risk log probability $p_{\mathrm{data}}(\boldsymbol{z}_\tau | \widehat{\boldsymbol{\theta}}^{\boldsymbol{\phi}_*})$ and known risk category probability $\boldsymbol{\gamma}_\tau^{\boldsymbol{\phi}_*}$.

```
// Pre-training HEN
for training round from 1 to MAX do
    ψ ← arg min_ψ C_HEN(G);
end
η ← 1/(1 + e^{|σ_Z − σ_{D_l}|});
μ̂_{k,l} ← Σ_{z_τ ∈ (R_k ∩ D_l)} z_τ / |R_k ∩ D_l|;
// Fine-tuning HenDen
for training round from 1 to MAX do
    for data batch from 1 to MAX do
        μ̃^φ_{k,S_b} ← (1 − η) μ̂^φ_{k,S_b} + η μ̂_{k,l};
        Σ̃^φ_{k,S_b} ← Σ_{z_τ ∈ S_b} γ^φ_{τ,k} (z_τ − μ̃^φ_{k,S_b})(z_τ − μ̃^φ_{k,S_b})^⊤ / Σ_{z_τ ∈ S_b} γ^φ_{τ,k};
        π̃^φ_{k,S_b} ← (1/|S_b|) Σ_{z_τ ∈ S_b} γ^φ_{τ,k};
        calculating log p_data(z_τ | θ̂^φ) using the above results;
        ψ, φ ← arg min_{ψ,φ} L_joint(G);
    end
end
// Identify known and emerging risks
p_data(z_τ | θ̂^{φ_*}) > predefined threshold determines emerging risks;
p_data(z_τ | θ̂^{φ_*}) ≤ predefined threshold and γ_τ^{φ_*} determine known risk labels.
```

3.4 Joint Training of HEN & DEN via Fine-tuning

HEN can encode a high-dimensional $\mathcal{G}_\tau$ into a low-dimensional dense vector $\boldsymbol{z}_\tau \in \mathbb{R}^d$, which could lose vital information for identifying operational risks. To address this problem, we introduce the following joint objective function that combines $\mathcal{L}_{\mathrm{HEN}}(\mathcal{G}_\tau)$ and $\mathcal{L}_{\mathrm{DEN}}(\mathcal{Z})$ to achieve end-to-end training [46].

$$\mathcal{L}_{\mathrm{joint}}(\mathcal{G}, \mathcal{S}) = \frac{1}{T} \sum_{\tau=1}^{T} \mathcal{C}_{\mathrm{HEN}}(\mathcal{G}_\tau) + \gamma \cdot \widetilde{\mathcal{C}}_{\mathrm{DEN}}(\mathcal{S}).$$

Essentially, $\mathcal{L}_{\text{joint}}(\mathcal{G}, \mathcal{S})$ imposes an extra constraint $\widetilde{\mathcal{C}}_{\text{DEN}}(\mathcal{S})$ on HEN to force $\boldsymbol{z}_\tau$ to preserve the vital information for identifying operational risks in feature representation. γ is a predetermined weight used to control the effects of $\widetilde{\mathcal{C}}_{\text{DEN}}(\mathcal{S})$. In this study, we set γ as a tiny positive constant number in $[1e-8, 1e-6]$ to fine-tune rather than drastically modify $\boldsymbol{z}_\tau$.

4 Experimental Evaluation

In this section, we first conduct a serials of experiments using a real-world dataset to compare the HenDen with state-of-the-art algorithms. In addition, several controlled tests with a synthetic dataset are carried out to illustrate the functionalities of the proposed framework.

4.1 Real-world Datasets

We collect daily operational logs from China XWBank[2] covering the period from $01/01/2021$ to $31/12/2021$. There are around $6,148,000$ activities performed on 14 internal systems from 800 employees affiliated with 67 departments in a daily operational log file. The log files in 300 days are utilized and converted into sequential undirected heterographs[3], which ensure that every node can become target nodes and get embedded. In addition, 0.015% of data are marked by internal control specialists as anomalous, covering a variety of operational activities, including account sharing, unauthorized access, and high-frequency operations.

In particular, each operational logic heterograph includes four types of nodes and seven types of edges, which are described as follows,

- **Departments:** We use one-hot vectors to represent each department and stand for their two categories: core and non-core departments.
- **Employees:** We utilize one-hot vectors to represent the four types of employees: interns, outsourced workers, internal workers, and others.
- **Personal computers (PCs):** We use one-hot vectors to denote the two categories of the IP addresses of personal computers: internal and external IP addresses.
- **Systems:** We use one-hot vectors to represent systems.
- **Affiliated with:** It connects employees and departments.
- **Login:** It connects employees and personal computers.
- **Add, Delete, Update, Query, Download:** They connect personal computers and internal systems.

4.2 Comparison Study

To gauge the overall performance of our proposed framework, we compare it with several state-of-the-art algorithms on a real-world dataset. The following parts are the optimal model parameters selected through preliminary experiments.

- **graph2vec+Kmeans** (unsupervised): It combines the graph2vec [47] with the Kmeans [16]. Notably, graph2vec is a state-of-the-art homograph model that introduces the idea of the skip-gram model into the graph representation learning and aims at encoding an entire homograph into a vector. The graph2vec has an input feature dimension of 14, an embedding dimension of 8, training epochs of 1000, a learning rate of 0.002, and a downsampling rate of 0.0001. Besides, in Kmeans, feature dimension, number of clusters, and maximum number of iterations are set to 8, 4, and 1000, respectively.
- **MAGNN+GMM** (unsupervised): It combines MAGNN [48] with Gaussian mixture model (GMM) [37]. MAGNN is a state-of-the-art heterograph representation learning model that consists of three parts: the node content transformation, intra-metapath aggregation, and inter-metapath aggregation. It has an input layer with 14 neural units, two hidden layers, each of which has four attention heads, and outputs the graph-level embedding with the dimension of 8. Besides, we train it with a maximum epoch of 100 and a learning rate of 0.005. GMM is a probabilistic model that combines multiple Gaussian distributions to represent a probability distribution in a continuous space. In GMM, we set its number of mixture components to 4, its convergence threshold to $1e-4$, and its number of maximum iterations to 300. Besides, we calculate a general covariance matrix for each mixture component and use Kmeans to initialize its weights.
- $\mathbf{HenDen}_{\text{un}}$ (unsupervised): It is an unsupervised variant of HenDen, that is, HenDen without the supervised constraint. HEN has an input layer with 14 neural units and two hidden layers, each of which has four attention heads, and outputs the graph-level embedding with the dimension of 8. DEN_{un} module has an input layer with 8 neural units, 6 hidden layers, and an output layer for calculating the negative probability density values. We train this framework with an initial learning rate of 0.01 and a maximum epoch of 100. Besides, the class numbers of nodes and edges are 4 and 7 respectively.
- **SANDD** [49] (semi-supervised): It is an ensemble framework of clustering-based classifiers, which are used for detecting and classifying emerging classes over data stream. Each classifier in such a framework is trained on partially labeled data. We set $t = 6$ and $q = 50$ in SANDD according to its original study.
- **SACCOS (ANN)** [50] (semi-supervised): It is a state-of-the-art semi-supervised emerging class recognition framework with an ANN classifier for detecting instances from emerging classes. we choose the artificial neural network (ANN) as the classifier of the SACCOS. According to the study [50], we set ANN as a single-layer neural network with 20 hidden units, the outlier buffer size $\mathcal{T}_o = 500$, maximum size of dynamic data buffer $\mathcal{T}_{\mathcal{W}} = 5000$, $k = 2$, and $\mathcal{T}_e = 0.99$.
- $\mathbf{HenDen}_{\text{sem}}$ (semi-supervised): It is a semi-supervised variant of HenDen, that is, HenDen with the supervised constraint. In this framework, the HEN module has an input layer with 14 neural units and two hidden layers, each of which has four attention heads, and outputs the graph-level embedding with the dimension of 8. The DEN_{sem} module has an input layer with 8 neural units, 6 hidden layers, and an output layer for calculating the negative probability density values. We train this framework with an initial learning rate of 0.01 and a maximum epoch of 100. Besides, the class numbers of nodes and edges are 4 and 7, respectively.

Table 2 shows the average performance of all models through 10-fold cross-validation. Compared to unsupervised models, semi-supervised models obtain an average enhancement of 140.05%, 36.54%, and 91.46% in terms of precision, recall, and F1-score,

2. XWBank, founded in December 2016 with a registered capital of RMB 3 billion, is among the top-3 online banks in China.

3. The log converting code can be found on GitHub (https://github.com/flying-chicks/operation-risk-detection).

TABLE 2: Experimental results via 10-fold cross-validation.

Cats.	Models	Precision	Recall	F1-score
Un.	graph2vec+Kmeans	0.1923	0.3333	0.2439
	MAGNN+GMM	0.3103	0.6000	0.4091
	$\mathbf{HenDen_{un}}$	**0.3750**	**0.8000**	**0.5106**
Sem.	SANDD	0.6875	0.7333	0.7097
	SACCOS (ANN)	0.6970	0.7667	0.7302
	$\mathbf{HenDen_{sem}}$	**0.7222**	**0.8667**	**0.7879**

[a] Un. and sem. stand for unsupervised and semi-supervised models.
[b] '+' means a combined framework whose modules are separately trained.
[c] The standard deviations of such metrics are within ± 0.015.

which is further supported by the Wilcoxon signed-rank test [51] ($p-value < 0.005$). This fact indicates that the partially labeled data is essential to improve the performance of models. In addition, compared to homograph models (graph2vec+Kmeans), heterograph models (MAGNN+GMM and $\mathrm{HenDen_{un}}$) improve the precision, recall, and F1-score by 78.19%, 110%, and 88.54%, which is further supported by the Wilcoxon signed-rank test ($p-value < 0.005$). A good explanation is that heterographs are more capable of capturing complex operational logic by recording different types of nodes and edges of graphs compared to homographs. Among all models, our proposed framework $\mathrm{HenDen_{sem}}$ achieves the best performance, indicating it as a practicable and effective way of identifying operational risks. The main reason is that $\mathrm{HenDen_{sem}}$ is able to utilize the HEN module and joint optimization strategy to capture the crucial operational logical features for risk identification while the DEN module effectively recognizes risks based on cluster and thin-tail nature of risks. In the next section, we will conduct an ablation study to further verify the functionality of such two modules and mechanisms.

4.3 Ablation Study

4.3.1 Evaluation of HEN & DEN

In this section, to verify the functionality of the HEN and DEN modules, we construct a series of combined models and conduct 10-fold cross-validation on a real dataset. The following parts are the optimal model parameters we set through preliminary experiments.

- **Kmeans** [16]: We set its feature dimension to 9, its number of clusters to 4, and its maximum number of iterations to 1000.
- **GMM** [37]: The Gaussian mixture model (GMM) is a probabilistic model that combines many Gaussian distributions to represent a probability distribution in a continuous space. We set its number of mixture components to 4, its convergence threshold to $1e-4$, and its number of maximum iterations to 300. Besides, we calculate a general covariance matrix for each mixture component and use Kmeans to initialize its weights.
- $\mathbf{DEN_{un}}$: It is an unsupervised variant of DEN. It has an input layer with 9 neural units, 6 hidden layers, and an output layer for calculating the negative probability density values.
- **HEN+Kmeans**: It is a combination of a HEN module and a Kmeans module, both of which are trained separately. In this framework, the HEN module has an input layer with 14 neural units and two hidden layers, each of which has four attention heads, and outputs the graph-level embedding with the dimension of 8. In the Kmeans module, we set the feature dimension to 8, the number of clusters to 4, and the maximum number of iterations to 1000.
- **HEN+GMM**: It is a united framework of a HEN module and a GMM module, and these two modules are trained in a decoupled fashion. The HEN module has an input layer with 14 neural units and two hidden layers, each of which has four attention heads, and outputs the graph-level embedding with the dimension of 8. In GMM, we set its number of mixture components to 4, its convergence threshold to $1e-4$, and its number of maximum iterations to 300. Besides, we calculate a general covariance matrix for each mixture component and use Kmeans to initialize its weights.
- **graph2vec+$\mathbf{DEN_{un}}$**: It is a combined framework of a graph2vec module and a $\mathrm{DEN_{un}}$ module, both of which are trained separately. The graph2vec module has an input feature dimension of 14, a hidden feature dimension of 8, training epochs of 1000, a minimal feature number of 5, a learning rate of 0.002, and a down-sampling rate of 0.0001. The $\mathrm{DEN_{un}}$ module has an input layer with 8 neural units, 6 hidden layers, and an output layer for calculating the negative probability density values.
- **MAGNN+$\mathbf{DEN_{un}}$**: It is an ensemble framework consisting of a MAGNN module and a $\mathrm{DEN_{un}}$ module. The MAGNN module has an input layer with 14 neural units and two hidden layers, each of which has four attention heads, and outputs the graph-level embedding with the dimension of 8. Besides, we train it with a maximum epoch of 100 and a learning rate of 0.005. The $\mathrm{DEN_{un}}$ module has an input layer with 8 neural units, 6 hidden layers, and an output layer for calculating the negative probability density values.
- **HEN+$\mathbf{DEN_{un}}$**: It is a combined framework of a HEN module and a $\mathrm{DEN_{un}}$ module, both of which are trained separately.

TABLE 3: Evaluation of HEN & DEN via 10-fold cross-validation.

Models	Precision	Recall	F1-score
Kmeans	0.1429	0.1333	0.1379
GMM	0.1875	0.2000	0.1935
$\mathrm{DEN_{un}}$	0.2000	0.3333	0.2500
HEN+Kmeans	0.2941	0.6667	0.4082
HEN+GMM	0.3235	0.7333	0.4490
graph2vec+$\mathrm{DEN_{un}}$	0.2333	0.4667	0.3111
MAGNN+$\mathrm{DEN_{un}}$	0.2895	0.7333	0.4151
HEN+$\mathrm{DEN_{un}}$	0.3421	0.8667	0.4905
$\mathrm{HenDen_{un}}$	0.3750	0.8000	0.5106
$\mathrm{HenDen_{sem}}$	0.7222	0.8667	0.7879

[*] The standard deviations of such metrics are within ± 0.02.

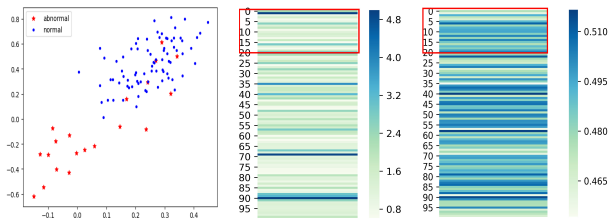
Table 3 reveals that $\mathrm{DEN_{un}}$ outperforms Kmeans in identifying suspicious operational activities. This is because the DEN module effectively leverages the thin-tail distribution property of risks, which Kmeans cannot use. In addition, Table 3 demonstrates that the HEN module further enhances the performance of the Kmeans, GMM, and $\mathrm{DEN_{un}}$ when the operational logic is represented as a heterograph. HEN+$\mathrm{DEN_{un}}$ achieves more promising performance than graph2vec+$\mathrm{DEN_{un}}$ and MAGNN+$\mathrm{DEN_{un}}$, which suggests that the HEN module appears to be more efficient at capturing operational logic heterographs than the homograph module (graph2vec) and heterograph module (MAGNN). The main contribution lies in the HEN module, which is specifically designed for operational logic features. It incorporates message

passing and aggregation mechanisms that are relevant to node and edge types, as well as a mechanism to capture operational frequency.

Especially, the $\text{HenDen}_{\text{un}}$ achieves performance gains over the $\text{HEN+DEN}_{\text{un}}$ by 9.62% and 4.10% in terms of the precision and F1-score, respectively, which is further supported by the Wilcoxon signed-rank test ($p-value < 0.005$). This suggests that the joint training mechanism of the HEN and DEN_{un} modules perform better than the decoupled training. Finally, compared to the $\text{HenDen}_{\text{un}}$, $\text{HenDen}_{\text{sem}}$ obtains higher precision, recall, and F1-score by relying on the partially labeled data.

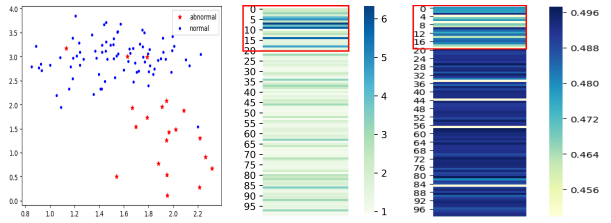
4.3.2 Evaluation of $\mathcal{L}_{\text{joint}}$

To illustrate the effect of $\mathcal{L}_{\text{joint}}$, we conduct experiments on a small dataset (including 80 negative and 20 positive samples), and further visualize critical HenDen layers in its initial stage (Fig. 3) and convergence stage (Fig. 4). Fig. 3(a) is a scatter plot of graph embeddings $\{\boldsymbol{z}_\tau\}_{\tau=1}^{100}$. Fig. 3(b) and 3(c) show $\{\boldsymbol{\gamma}_\tau\}_{\tau=1}^{100}$ and $\{p_{\text{data}}(\boldsymbol{z}_\tau|\widehat{\boldsymbol{\theta}})\}_{\tau=1}^{100}$ in heat maps, respectively. Since the weights of HenDen are initialized randomly at its initial stage, $\{\boldsymbol{\gamma}_\tau\}_{\tau=1}^{100}$ and $\{p_{\text{data}}(\boldsymbol{z}_\tau|\widehat{\boldsymbol{\theta}})\}_{\tau=1}^{100}$ are disordered. If HEN and DEN are trained separately, the dots in Fig. 3(a) do not change. This is because the back-propagation of gradient decent of DEN never touches the optimization of HEN.



(a) $\{\boldsymbol{z}_\tau\}_{\tau=1}^{100}$ (b) $\{\boldsymbol{\gamma}_\tau\}_{\tau=1}^{100}$ (c) $\{p_{\text{data}}(\boldsymbol{z}_\tau|\widehat{\boldsymbol{\theta}})\}_{\tau=1}^{100}$

Fig. 3: Initial stage of HenDen.



(a) $\{\boldsymbol{z}_\tau\}_{\tau=1}^{100}$ (b) $\{\boldsymbol{\gamma}_\tau\}_{\tau=1}^{100}$ (c) $\{p_{\text{data}}(\boldsymbol{z}_\tau|\widehat{\boldsymbol{\theta}})\}_{\tau=1}^{100}$

Fig. 4: Convergence stage of HenDen.

The red squares highlighted in Fig. 4(b) and 4(c) indicate the discovered risks. Both subfigures present regular patterns when HenDen is well-trained. Besides, Fig. 4(a) shows the refined graph embeddings of the operational activities after the joint training of both HEN and DEN. Compared to Fig. 3(a), the red and blue dots in Fig. 4(a) are more distinguishable. This fact further consolidates that the joint training strategy can improve graph representation learning of HEN.

4.3.3 Evaluation of $\widetilde{\mathcal{C}}_{\text{DEN}}$

To validate the effectiveness of $\widetilde{\mathcal{C}}_{\text{DEN}}$, this study conduct extensive 10-fold cross-validations on simulation datasets. This study first generates K groups of samples following two-dimensional Gaussian distributions, namely $\mathcal{R}_1 = \{\boldsymbol{z}_\tau\}_{\tau=1}^{T_1}\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}), \mathcal{R}_2 = \{\boldsymbol{z}_\tau\}_{\tau=T_1+1}^{T_1+T_2}\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}), \ldots, \mathcal{R}_K = \{\boldsymbol{z}_\tau\}_{\tau=T_{K-1}+1}^{T_1+\cdots+T_K}\mathcal{N}(\boldsymbol{\mu}_K, \boldsymbol{\Sigma})$. Here, $\boldsymbol{\mu}_k$ is a two-dimensional vector, and $\boldsymbol{\Sigma}$ is a two-dimensional identity matrix. The entire dataset is $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_K\}$, and the total number of samples is $T = T_1 + T_2 + \cdots + T_K$. Next, this study randomly selects $T_k \times |\mathcal{D}_l|/T$ samples from $\mathcal{R}_k$ to label them, which form the labeled dataset $\mathcal{D}_l$, while the remaining samples form the unlabeled dataset $\mathcal{D}_u$. Such two datasets satisfy the conditions $|\mathcal{D}_l| \ll |\mathcal{D}_u|$ and $|\mathcal{D}_l| + |\mathcal{D}_u| = T$.

Case 1: As shown in Fig. 5 of Appendix, we set $K = 2$, $T_1 = T_2 = 1000$, $\boldsymbol{\mu}_1 = (1, 1)^\top$, $\boldsymbol{\mu}_2 = (1 + d, 1 + d)^\top$, where $d = \{2, 3, 4, 5\}$, $|\mathcal{D}_l|/T = \{0.01, 0.02, 0.03, 0.04, 0.05\}$, and $\beta_2 = \{0.6, 0.7, 0.8, 0.9\}$. Blue dots represent unlabeled data, while red and yellow dots represent labeled data belonging to $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. Smaller d indicates closer distances between risk clusters, making it more challenging to distinguish. And Appendix provides Case 2 to assess $\widetilde{\mathcal{C}}_{\text{DEN}}$.

Table 4 and Table 6 in Appendix suggest that as d increases, classification performance improves generally. Figs. 5(a) and 6(a) of Appendix illustrate that when $d = 2$, d is so small that risk clusters are difficult to visually distinguish. However, as displayed in Table 4 and Table 6 of Appendix, when $\beta_2 \geq 0.7$, DEN can achieve F1-scores≥ 0.6. Moreover, when $|\mathcal{D}_l|/T \geq 0.03$, DEN can even obtain F1-scores≥ 0.88. We can observe from Table 4 and Table 6 in Appendix that as $|\mathcal{D}_l|/T$ increases, the classification performance of DEN improves. In addition, β_2 can also boost its classification performance.

TABLE 4: Average F1-score in Case 1

	$\frac{\lvert\mathcal{D}_l\rvert}{T}$	$\beta_2 = 0.6$	$\beta_2 = 0.7$	$\beta_2 = 0.8$	$\beta_2 = 0.9$
$d = 2$	0.01	0.5399	0.4839	0.4463	0.6005
	0.02	0.5511	0.6218	0.5263	0.8044
	0.03	0.5892	0.6833	0.7147	0.8978
	0.04	0.5723	0.6450	0.8216	0.8929
	0.05	0.5965	0.7839	0.7270	0.9110
$d = 3$	0.01	0.4640	0.6270	0.6372	0.5788
	0.02	0.5744	0.6681	0.7288	0.8712
	0.03	0.6545	0.5173	0.6613	0.9061
	0.04	0.6110	0.5629	0.8921	0.9755
	0.05	0.6630	0.6001	0.9332	0.9776
$d = 4$	0.01	0.4201	0.6151	0.6064	0.7301
	0.02	0.5607	0.5675	0.6609	0.8020
	0.03	0.4752	0.5054	0.7640	0.9964
	0.04	0.5997	0.9306	0.8959	0.9528
	0.05	0.5540	0.5963	0.7970	0.9461
$d = 5$	0.01	0.4331	0.5266	0.5450	0.6805
	0.02	0.4935	0.7463	0.7395	0.9277
	0.03	0.6999	0.6329	0.7591	0.9995
	0.04	0.6500	0.5864	0.7492	0.9499
	0.05	0.6996	0.8500	0.6741	0.9997

4.4 Experiments on Extreme Risk Cases

Identifying extreme risks (especially emerging risks) is a huge challenge in operational risk identification. As financial operations expand and change, so do the patterns and forms of operational risk. This fact also limits the power of supervised learning in identifying operational risks, as new risks are difficult to identify and label in time for training. Here, we select some real-world data and add some manufactured risk errors to produce a synthetic

dataset to gain insights into the unique advantages of the proposed framework.

Specifically, we first obtain $10,000$ heterographs generated from the daily operational activity records associated with 100 employees in our real-world dataset, lasting 100 days, which serves as the basis for our synthetic dataset. Then we create three different data tracks as follows for our evaluation. (i) **Risks in Objects:** To simulate the operational risks brought by the act on illegal objects (departments, employees, personal computers, and systems), we randomly select 20% of heterographs from the basis of the synthetic data and randomly replace $2\%-5\%$ of the nodes in each heterograph with wrong targets. (ii) **Risks in Behaviors:** Similarly, we randomly sample 5% of heterographs from the basis of the synthetic data, and randomly replace $2\%-5\%$ of edge types (login, add, delete, update, query, and download) in each heterogeneous graph with illegal operations. (iii) **Mixed Emerging Risks:** On the basis of the synthetic data, 20% of the regular data are randomly replaced with the anomalies obtained in the first two data tracks.

This synthetic data contains much more risks than the real-world data because we randomly replace 20% of regular heterographs with anomalies, each of which is obtained by randomly replacing the nodes or edges. This generates a huge amount of risk patterns with a risk ratio of 20% compared to the real-world dataset with a risk ratio of 0.015% Table 5 shows that the proposed HenDen is able to handle a large number of risk patterns simultaneously and insensitive to the types of different errors.

TABLE 5: Evaluation on extreme risk cases.

Cats.	Precision	Recall	F1-score
Risks in objects	0.5714	0.6000	0.5854
Risks in behaviors	0.5172	0.7500	0.6122
Mixed risks	0.5416	0.6500	0.5909

* The standard deviations of such metrics are within ± 0.025.

5 Conclusion

In this study, we propose a semi-supervised learning framework called HenDen, which aims to identify financial operational risks using only a limited number of labeled examples. We highlight four unique contributions of HenDen in terms of its practicality and effectiveness.

(i) This framework utilizes heterographs to preserve the complicated operational logic for deep learning. This allows HenDen to capture and leverage the intricate relationships and dependencies present in financial operational risk data.

(ii) We propose a heterograph embedding network (HEN) that can simultaneously learn both the structural information and meta relations within the heterographs. By doing so, HenDen can effectively represent and exploit the complex connections between different entities and factors associated with operational risks.

(iii) We present a density estimation network (DEN) that is capable of identifying risks even when only a few labeled examples are available. This is particularly useful in real-world scenarios where obtaining a large number of labeled instances may be challenging or costly.

(iv) We introduce a seamless learning strategy that jointly optimizes information loss and decision boundaries through a joint objective function. This strategy enhances the overall learning process and improves the accuracy of risk identification by effectively combining different learning objectives.

Here, we highlight the significant theoretical innovation and extensive applicability of our framework. HEN introduces innovative techniques for learning the logical connections between features, enabling its applications in other domains, such as action recognition with kinematics posture feature [52]. Furthermore, the innovation of DEN lies in its utilization of cluster and thin-tailed properties for risks. This approach can be equally applied to other domains, including risk identification in deep-water drilling [22] and aviation [23].

References

[1] S. Ashby, "The 2007-2009 financial crisis: Learning the risk management lessons," in *Financial Services Research Forum, Nottingham*, 2010.

[2] P. Stonham, "Whatever happened at Barings? part two: Unauthorised trading and the failure of controls," *European Management Journal*, vol. 14, no. 3, pp. 269–278, 1996.

[3] J. Ross, "Rogue trader: How I brought down Barings Bank and shook the financial world," 1997.

[4] K. Böcker and C. Klüppelberg, "Operational VaR: A closed-form approximation," *Risk*, pp. 90–93, 2005.

[5] M. Neil, N. Fenton, and M. Tailor, "Using bayesian networks to model expected and unexpected operational losses," *Risk Analysis: An International Journal*, vol. 25, no. 4, pp. 963–972, 2005.

[6] W. P. Wagner, J. Otto, and Q. Chung, "Knowledge acquisition for expert systems in accounting and financial problem domains," *Knowledge-Based Systems*, vol. 15, no. 8, pp. 439–447, 2002.

[7] F. Zhou, X. Qi, C. Xiao, and J. Wang, "Metarisk: Semi-supervised few-shot operational risk classification in banking industry," *Information Sciences*, vol. 552, pp. 1–16, 2021.

[8] F. Zhou, S. Zhang, and Y. Yang, "Interpretable operational risk classification with semi-supervised variational autoencoder," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 846–852.

[9] K. W. Hanley and G. Hoberg, "Dynamic interpretation of emerging risks in the financial sector," *The Review of Financial Studies*, vol. 32, no. 12, pp. 4543–4603, 2019.

[10] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of The Web Conference 2020*, 2020, pp. 2704–2710.

[11] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, and P. S. Yu, "A survey on heterogeneous graph embedding: Methods, techniques, applications and sources," *arXiv preprint arXiv:2011.14867*, 2020.

[12] S. Sonnentag, W. Wehrt, B. Weyers, and Y. C. Law, "Conquering unwanted habits at the workplace: Day-level processes and longer term change in habit strength." *Journal of Applied Psychology*, 2021.

[13] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[14] A. J. McNeil, R. Frey, and P. Embrechts, *Quantitative risk management: Concepts, techniques and tools-revised edition*. Princeton University Press, 2015.

[15] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modelling extremal events: for insurance and finance*. Springer Science & Business Media, 2013, vol. 33.

[16] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C*, vol. 28, no. 1, pp. 100–108, 1979.

[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in *International Conference of Knowledge Discovery and Data Mining*, vol. 240, 1996, p. 6.

[18] F. Aue and M. Kalkbrener, "Lda at work: Deutsche bank's approach to quantifying operational risk," *Journal of Operational Risk*, vol. 1, no. 4, pp. 49–93, 2006.

[19] A. Ganegoda and J. Evans, "A scaling model for severity of operational losses using generalized additive models for location scale and shape," *Annals of Actuarial Science*, vol. 7, no. 1, pp. 61–100, 2013.

[20] K. Zhang, X. Wu, R. Niu, K. Yang, and L. Zhao, "The assessment of landslide susceptibility mapping using random forest and decision tree methods in the three gorges reservoir area, china," *Environmental Earth Sciences*, vol. 76, no. 11, pp. 1–20, 2017.

[21] W. Liang, A. Sari, G. Zhao, S. D. McKinnon, and H. Wu, "Short-term rockburst risk prediction using ensemble learning methods," *Natural Hazards*, vol. 104, no. 2, pp. 1923–1946, 2020.
[22] Q. Yin, J. Yang, M. Tyagi, X. Zhou, X. Hou, and B. Cao, "Field data analysis and risk assessment of gas kick during industrial deepwater drilling process based on supervised learning algorithm," *Process Safety and Environmental Protection*, vol. 146, pp. 312–328, 2021.
[23] M. Hadjimichael, "A fuzzy expert system for aviation risk assessment," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6512–6519, 2009.
[24] H. Fares and T. Zayed, "Hierarchical fuzzy expert system for risk of failure of water mains," *Journal of Pipeline Systems Engineering and Practice*, vol. 1, no. 1, pp. 53–62, 2010.
[25] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
[26] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021.
[27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
[28] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.
[29] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang, "Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358–368, 2016.
[30] T. Ergen and S. S. Kozat, "Unsupervised anomaly detection with lstm neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3127–3141, 2019.
[31] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.
[32] I. Jolliffe, "Principal components in regression analysis," in *Principal Component Analysis*. Springer, 1986, pp. 129–155.
[33] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining and Knowledge Discovery*, vol. 29, pp. 626–688, 2015.
[34] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[35] L. Zheng, Z. Li, J. Li, Z. Li, and J. Gao, "Addgraph: Anomaly detection in dynamic graph using attention-based temporal GCN," in *International Joint Conference on Artificial Intelligence*, vol. 3, 2019, p. 7.
[36] B. Chen, J. Zhang, X. Zhang, Y. Dong, J. Song, P. Zhang, K. Xu, E. Kharlamov, and J. Tang, "GCCAD: Graph contrastive learning for anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, 2022.
[37] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–22, 1977.
[38] A. Aboah, "A vision-based system for traffic anomaly detection using deep learning and decision trees," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4207–4212.
[39] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.
[40] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4393–4402.
[41] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 132–149.
[42] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *24th International Conference of Neural Information Processing*. Springer, 2017, pp. 373–382.
[43] P. Jorion *et al.*, *Financial risk manager handbook*. John Wiley & Sons, 2007, vol. 406.
[44] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometrics*, vol. 741, no. 659-663, 2009.
[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.
[46] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
[47] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.
[48] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of The Web Conference 2020*, 2020, pp. 2331–2341.
[49] A. Haque, L. Khan, and M. Baron, "Sand: Semi-supervised adaptive novel class detection and classification over data stream," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.
[50] Y. Gao, S. Chandra, Y. Li, L. Khan, and T. Bhavani, "Saccos: A semi-supervised framework for emerging class detection and concept drift adaption over data streams," *IEEE Transactions on Knowledge & Data Engineering*, vol. 34, no. 03, pp. 1416–1426, 2022.
[51] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
[52] M. A. R. Ahad, M. Ahmed, A. D. Antar, Y. Makihara, and Y. Yagi, "Action recognition using kinematics posture feature on 3d skeleton joint locations," *Pattern Recognition Letters*, vol. 145, no. 5, 2021.

Guanyuan Yu is a lecturer at Southwestern University of Finance and Economics, China. He received his Ph.D. from Southwest University of Finance and Economics, China. His research interests lie primarily in deep graph learning and financial risk management. He has served as a reviewer of AAAI and published papers on various computer journals, including Information Sciences, MTAP, and IEEE TALSP.

Liling Jiang is a Ph.D. student at Southwestern University of Finance and Economics, China. He obtained his master degree from Southwestern University of Finance and Economics in 2015. His research interests focus on data mining and financial intelligence.

Qing Li is a professor at Southwestern University of Finance and Economics, China. Prior to taking that post, he was a postdoctoral researcher at Arizona State University and at the Information & Communications University of Korea. Li's research interests lie primarily in intelligent information processing and business intelligence. He has served on the editorial boards of Electronic Commerce Research and Applications, the Journal of Database Management, and the Journal of Global Information Management as well as the program committees of various international conferences, including PACIS, SIGIR, and CIKM. He received his Ph.D. from Kumoh National Institute of Technology in February of 2005 and his M.S. and B.S. degrees from Harbin Engineering University, China.

APPENDIX

Proof 1 (Claim 1). *$\mathcal{L}$ is monotonically increasing in the interval $[0, +\infty)$, and minimizing $\mathcal{L}$ is equivalent to maximizing the predicted probability $\hat{y}_{\tau,k} = f(\boldsymbol{h}_\tau)$, that is $\hat{y}_{\tau,k} \mapsto 1$. Since $f(\cdot)$ is an invertible function, we further obtain $\boldsymbol{h}_\tau = f^{-1}(\hat{y}_{\tau,k})$.*

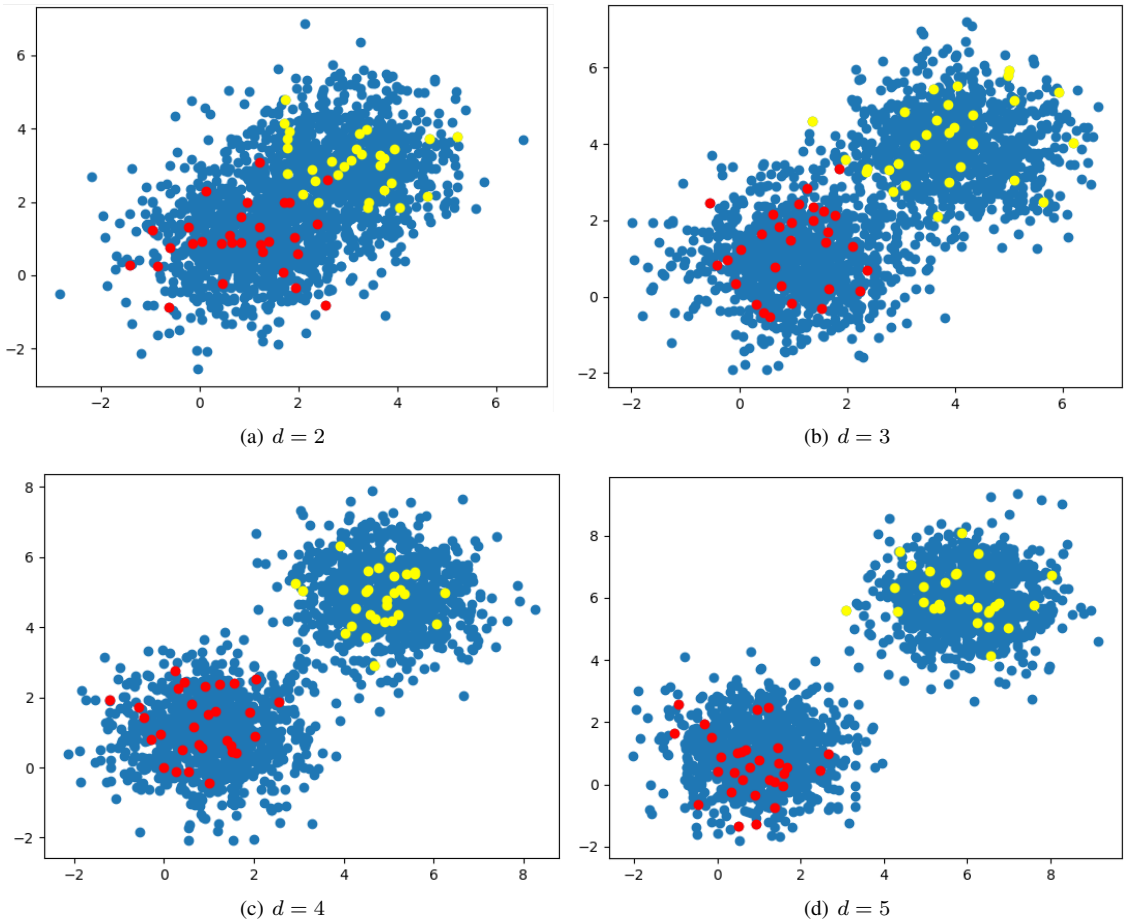
Proof 2 (Claim 2). *Let $\boldsymbol{\epsilon}_{\tau,k}(\boldsymbol{\theta}_k) = \boldsymbol{\epsilon}_{\tau,k}$, for $\forall \boldsymbol{z}_{\tau_1}, \boldsymbol{z}_{\tau_2} \in \mathcal{Z}_k$ and a small constant c, we have $\mathbb{P}(\|\boldsymbol{z}_{\tau_1} - \boldsymbol{z}_{\tau_2}\| > c) = \mathbb{P}(\|\boldsymbol{\epsilon}_{\tau_1,k} - \boldsymbol{\epsilon}_{\tau_2,k}\| > c) \leq \mathbb{P}(\|\boldsymbol{\epsilon}_{\tau_1,k}\| + \|\boldsymbol{\epsilon}_{\tau_2,k}\| > c) \leq \mathbb{P}(\|\boldsymbol{\epsilon}_{\tau_1,k}\| > c/2) + \mathbb{P}(\|\boldsymbol{\epsilon}_{\tau_2,k}\| > c/2) \leq 2\varepsilon_2$ $(c \geq 2\varepsilon_1)$. This indicates that $\boldsymbol{z}_{\tau_1}$ and $\boldsymbol{z}_{\tau_2}$ maintains a small spatial distance with a high probability. In other words, feature vector samples belonging to a common risk class will gather together.*

For $\forall \boldsymbol{z}_{\tau_1} \in \mathcal{Z}_{k_1}$ and $\forall \boldsymbol{z}_{\tau_2} \in \mathcal{Z}_{k_2}$ (where $k_1 \neq k_2$), we have $\mathbb{P}(\|\boldsymbol{z}_{\tau_1} - \boldsymbol{z}_{\tau_2}\| > C) = \mathbb{P}(\|\boldsymbol{g}_{k_1}(\boldsymbol{\theta}_{k_1}) - \boldsymbol{g}_{k_2}(\boldsymbol{\theta}_{k_2}) + \boldsymbol{\epsilon}_{\tau_1,k_1} - \boldsymbol{\epsilon}_{\tau_2,k_2}\| > C) = \mathbb{P}(\|\boldsymbol{g}_{k_1}(\boldsymbol{\theta}_{k_1}) - \boldsymbol{g}_{k_2}(\boldsymbol{\theta}_{k_2}) + \boldsymbol{\epsilon}_{\tau_1,k_1} - \boldsymbol{\epsilon}_{\tau_2,k_2}\| > C) \geq \mathbb{P}(\|\boldsymbol{g}_{k_1}(\boldsymbol{\theta}_{k_1}) - \boldsymbol{g}_{k_2}(\boldsymbol{\theta}_{k_2})\| - \|\boldsymbol{\epsilon}_{\tau_1,k_1} - \boldsymbol{\epsilon}_{\tau_2,k_2}\| > C)$ (where C is a large positive constant). Since $\mathbb{P}(\|\boldsymbol{g}_{k_1}(\boldsymbol{\theta}_{k_1}) - \boldsymbol{g}_{k_2}(\boldsymbol{\theta}_{k_2})\| > C_1\|\boldsymbol{\epsilon}_{\tau_1,k_1} - \boldsymbol{\epsilon}_{\tau_2,k_2}\|) \mapsto 1$ and $\mathbb{P}(\|\boldsymbol{g}_{k_1}(\boldsymbol{\theta}_{k_1}) - \boldsymbol{g}_{k_2}(\boldsymbol{\theta}_{k_2})\| > C_2) \mapsto 1$, $\mathbb{P}(\|\boldsymbol{z}_{\tau_1} - \boldsymbol{z}_{\tau_2}\| > C)$ will be relatively large. Therefore, feature vector samples belonging to different risk classes maintain a significant spatial distance with a larger probability.

Proof 3 (Claim 4). *After fixing $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\pi}_k$, $C_{Q_{\mathrm{DEN}}}(\boldsymbol{\theta})$ depends only on $\boldsymbol{\mu}$, which is denoted as $\mathcal{C}_{Q_{\mathrm{DEN}}}(\boldsymbol{\mu})$ $(\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K)$. Given $\boldsymbol{\phi}$, an estimate of $\mathcal{C}_{Q_{\mathrm{DEN}}}(\boldsymbol{\mu})$ can be represented as $\mathcal{C}_{Q_{\mathrm{DEN}}}(\widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}}) = \mathcal{C}_{Q_{\mathrm{DEN}}}(\widehat{\boldsymbol{\mu}}(\mathrm{DEN}_{\boldsymbol{\phi}}(\boldsymbol{x})))$. $\mathrm{DEN}_{\boldsymbol{\phi}}(\boldsymbol{x})$ produces $\mathcal{C}_{Q_{\mathrm{DEN}}}(\widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_e}) = \mathcal{C}_{Q_{\mathrm{DEN}}}(\widehat{\boldsymbol{\mu}}(\mathrm{DEN}_{\boldsymbol{\phi}_e}(\boldsymbol{x})))$ after $e = \{0, \ldots, E\}$ rounds of training. At the $(e+1)$-th round of training, $\boldsymbol{\phi}$ can be updated via the gradient descent-based back-propagation algorithm, $\boldsymbol{\phi}_{e+1} = \boldsymbol{\phi}_e - \lambda \frac{\partial \mathcal{C}_{Q_{\mathrm{DEN}}}}{\partial \widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_e}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_e}}{\partial \mathrm{DEN}_{\boldsymbol{\phi}_e}} \cdot \frac{\partial \mathrm{DEN}_{\boldsymbol{\phi}_e}}{\partial \boldsymbol{\phi}_e}$ where λ is a learning rate.*

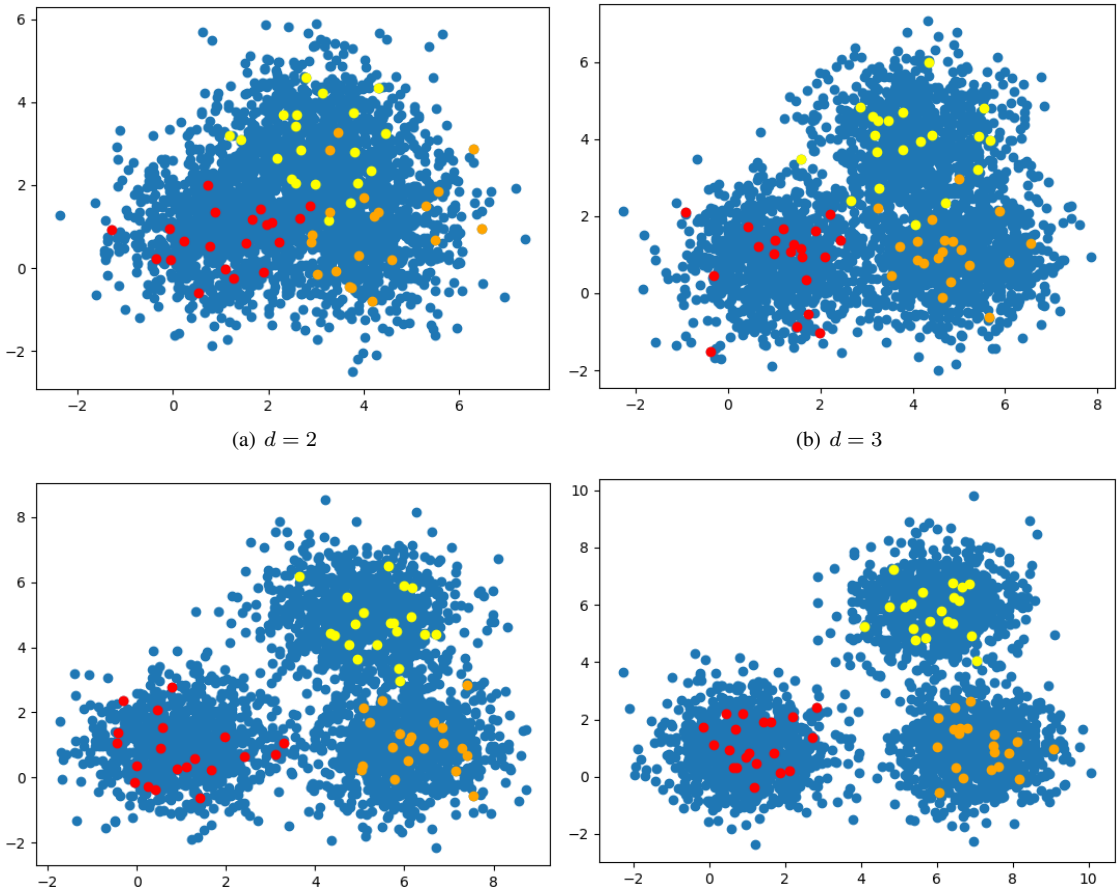
- *When $e = 0$, $\boldsymbol{\phi}_0$ is determined through random initialization, and $\mathrm{DEN}_{\boldsymbol{\phi}_0}(\boldsymbol{z})$ likely outputs inferior $\boldsymbol{\gamma}^{\boldsymbol{\phi}_0}$ $(\boldsymbol{\gamma}^{\boldsymbol{\phi}_0} = \{\boldsymbol{\gamma}_\tau^{\boldsymbol{\phi}_0}\}_{\tau=1}^T)$. That is, $\mathrm{DEN}_{\boldsymbol{\phi}_0}(\boldsymbol{z})$ cannot ensure $\gamma_{\tau,k}^{\boldsymbol{\phi}_0} I_{\mathcal{R}_k}(\boldsymbol{z}_\tau) \mapsto 1$ and $\gamma_{\tau,k}^{\boldsymbol{\phi}_0} I_{\mathcal{R}_k^c}(\boldsymbol{z}_\tau) \mapsto 0$, therefore resulting in $\widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_0}$ to deviate from $\boldsymbol{\mu}$, further leading to $\mathcal{C}_{Q_{\mathrm{DEN}}}(\widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_0})$ deviating from $\mathcal{C}_{Q_{\mathrm{DEN}}}(\boldsymbol{\mu})$.*
- *When $e = 1$, $\boldsymbol{\phi}_1 = \boldsymbol{\phi}_0 - \lambda \frac{\partial \mathcal{C}_{Q_{\mathrm{DEN}}}}{\partial \widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_0}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_0}}{\partial \mathrm{DEN}_{\boldsymbol{\phi}_0}} \cdot \frac{\partial \mathrm{DEN}_{\boldsymbol{\phi}_0}}{\partial \boldsymbol{\phi}_0}$ will introduce more noises, which push $\mathcal{C}_{Q_{\mathrm{DEN}}}(\widehat{\boldsymbol{\mu}}^{\boldsymbol{\phi}_1})$ away from $\mathcal{C}_{Q_{\mathrm{DEN}}}(\boldsymbol{\mu})$ during the forward propagation.*
- *When $e \geq 2$, such noises will continue until $e = E$. Finally, $\mathrm{DEN}_{\boldsymbol{\phi}_0}(\boldsymbol{z})$ would fail to converge.*

Extensive Evaluation of $\widetilde{\mathcal{C}}_{\mathrm{DEN}}$. Case 2: As shown in Fig. 6, we set $K = 3$, $T_1 = T_2 = T_3 = 1000$, $\boldsymbol{\mu}_1 = (1, 1)^\top$, $\boldsymbol{\mu}_2 = (1 + d, 1 + d)^\top$, $\boldsymbol{\mu}_3 = (1 + d, 1)^\top$, where $d = \{2, 3, 4, 5\}$, $|\mathcal{D}_l|/T = \{0.01, 0.02, 0.03, 0.04, 0.05\}$. Red, yellow, and orange dots represent labeled data belonging to $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$ respectively. Larger K implies more risk clusters, making it harder to identify.



(a) $d = 2$ (b) $d = 3$ (c) $d = 4$ (d) $d = 5$

Fig. 5: Data distribution of Case 1



(a) $d = 2$ (b) $d = 3$

(c) $d = 4$ (d) $d = 5$

Fig. 6: Data distribution of Case 2

TABLE 6: Average F1-score in Case 2

	$\frac{\lvert\mathcal{D}_t\rvert}{T}$	$\beta_2 = 0.6$	$\beta_2 = 0.7$	$\beta_2 = 0.8$	$\beta_2 = 0.9$
$d = 2$	0.01	0.4452	0.5016	0.5630	0.5641
	0.02	0.6185	0.6479	0.6664	0.7766
	0.03	0.5303	0.6925	0.7403	0.8882
	0.04	0.7148	0.7098	0.8118	0.9108
	0.05	0.6490	0.7427	0.7832	0.9010
$d = 3$	0.01	0.4517	0.5179	0.5185	0.4888
	0.02	0.6593	0.6996	0.6791	0.8729
	0.03	0.7953	0.7310	0.7583	0.9228
	0.04	0.7632	0.6209	0.8517	0.9791
	0.05	0.7584	0.8001	0.9780	0.9351
$d = 4$	0.01	0.6711	0.4716	0.5663	0.8085
	0.02	0.4903	0.6480	0.7719	0.8095
	0.03	0.7135	0.5849	0.7974	0.9965
	0.04	0.6561	0.8486	0.8558	0.9975
	0.05	0.7710	0.6485	0.8977	0.9959
$d = 5$	0.01	0.5112	0.5851	0.6852	0.6371
	0.02	0.5611	0.5612	0.8313	0.8079
	0.03	0.6587	0.6499	0.7994	0.9492
	0.04	0.6071	0.6997	0.8109	0.9996
	0.05	0.7496	0.6536	0.8492	0.9994