

Universiteit Leiden
The Netherlands

Open-world continual learning via knowledge transfer

Li, Y.

Citation
Li, Y. (2026, January 27). *Open-world continual learning via knowledge transfer*. Retrieved from https://hdl.handle.net/1887/4287955

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/4287955

Note: To cite this publication please use the final published version (if applicable).

Chapter 7

Cross-regional Fraud Detection via Continual Learning with Knowledge Transfer

While the preceding chapters have focused on advancing open-world continual learning through both theoretical insights and practical algorithms, the fundamental goal of **robust knowledge transfer**—from known to unknown classes, and from past to future data distributions—remains central to real-world applications. One such application domain is *fraud detection*, where the ability to adaptively transfer and retain knowledge across evolving environments is important for maintaining detection accuracy and operational efficiency.

In this chapter, we explore the challenge of **cross-regional fraud detection**, an applied setting that inherently demands effective knowledge transfer and continual adaptation. Traditional fraud detection systems are typically trained on static datasets from single regions, making them ill-suited for generalization to new cities or countries. When businesses expand geographically, retraining a model from scratch is not only computationally expensive but also risks *catastrophic forgetting* of previously learned fraud patterns.

To address these challenges, we formulate cross-regional fraud detection as an **incremental learning problem**, where the objective is to develop a unified model that incrementally learns from region-specific data while preserving past knowledge. We propose a novel paradigm called **Cross-Regional Continual Learning (CCL)**, which explicitly incorporates *knowledge transfer mechanisms* to support both forward adaptation and backward retention. Specifically, CCL employs *prototype-based knowledge replay* to distill transferable representations and utilizes a *parameter smoothing strategy* to mitigate forgetting during continual updates. Moreover, we construct **heterogeneous trade graphs (HTG)** to represent complex transaction structures and leverage graph-based backbones to enhance semantic understanding and facilitate transfer across regions with different fraud profiles.

Through extensive experiments, we demonstrate that CCL consistently outperforms baseline approaches in cross-regional fraud detection tasks. This application-driven chapter thus not only illustrates the practical utility of continual learning techniques, but also confirms the broader **transferability of knowledge-centric learning frameworks** developed throughout this thesis. Together, these insights lay a foundation for *adaptive, scalable fraud detection systems in dynamic, geographically diverse environments.*

This chapter is based on the following publications [39, 40]:

- **Li, Y.**, Yang, Y., Gao, Q., & Yang, X: Cross-Regional Fraud Detection via Continual Learning (Special Program). In: Proceedings of the AAAI Conference on Artificial Intelligence, 37(13): 16260-16261, 2023.
- **Li, Y.**, Yang, X., Gao, Q., Wang, H., Zhang, J., & Li, T.: Cross-Regional Fraud Detection via Continual Learning With Knowledge Transfer. In: IEEE Transactions on Knowledge and Data Engineering, 38(12): 7865-7877, 2024.

To contextualize and motivate our proposed framework, the following section outlines the core motivations, identifies the fundamental challenges in cross-regional fraud detection, and summarizes the key contributions of this chapter.

7.1 Continual Learning for Cross-Regional Fraud Detection

In recent years, the widespread application of Internet finance has become increasingly prevalent, concurrently leading to substantial financial losses due to fraudulent activities. Thus, it is important to effectively detect fraud for risk management, and there are substantial research efforts have been dedicated to this endeavor. Although existing fraud detection models can achieve promising performance, most solutions simply employ expert-driven rules or machine learning methods, using historical transaction data to discover potential anomalous behaviors [141]. Existing methods heavily depend on prior knowledge and fail to adapt dynamically to evolving data, particularly when financial operations expand into new regions. Furthermore, traditional methods suffer from the catastrophic forgetting issue, whereby previously acquired knowledge cannot be retained during the model's evolutionary process.

In financial contexts, due to considerations of data privacy and security, data from different regions is not allowed to be shared. Therefore, we cannot combine data from different regions to train a unified fraud detection model. Moreover, training from scratch for each new region will result in forgetting previously acquired knowledge and increase computational costs rapidly. In addition, fraudsters usually adapt quickly to anti-fraud measures and detection methods [142], resulting in difficulty in capturing fraudulent behavioral patterns. As shown in Figure 7.1, when a financial

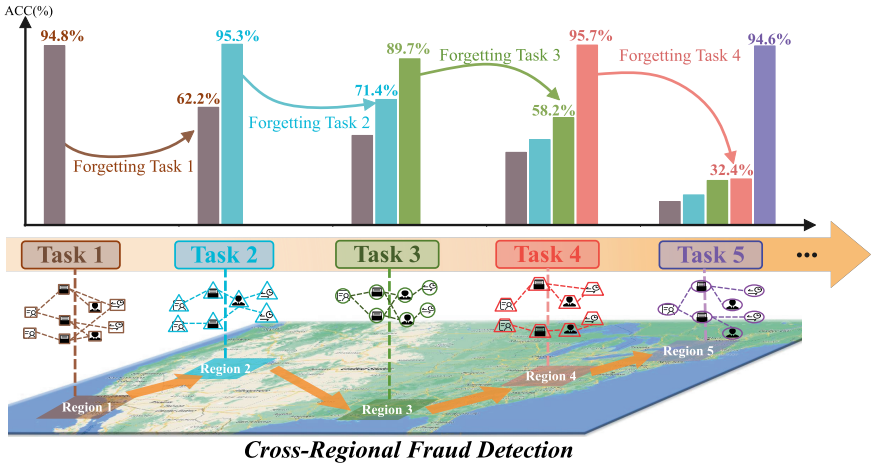


Figure 7.1: Bottom: An illustration of cross-regional fraud detection. The orange arrows indicate a financial business expansion from Region 1 to Region 5, where the data characteristics vary across different regions with different patterns. Our proposed method aims to incrementally learn a unified fraud detection model that can perform well in each task through knowledge transfer. Top: The histograms in different colors denote each corresponding region's fraud detection performance (ACC) with training from scratch. The performance drops significantly after a model is trained using data from Region 2. Similarly, after training a model with data from Region 3, the performance of Region 1 and Region 2 fell further, respectively.

institution extends business to new regions, new data will appear incrementally, and the data of each region has its own characteristics. The fraud detection model observes cross-regional data sequentially as they arrive, thus any processing or learning must be done in a continual fashion. Hence, we formulate a new problem, **cross-regional fraud detection**, which aims to continually detect fraudulent activities in the process of business expansion to new regions with streaming data. The proposed problem stems from practical scenarios and sheds new light on integrating fraud detection with continual learning.

The cross-regional fraud detection problem is based on a real-world scenario that combines continual learning with knowledge transfer and poses two primary challenges. (1) *Regional Uniqueness*: The inherent geographical distances introduce differences between different regions. Due to the unique characteristics and region-specific knowledge of each region, the fraud detection model must adapt to region-specific knowledge in a continual learning framework. (2) *Regional Generality*: Fraud detection adheres to general knowledge applicable across all regions. Hence, the cross-regional fraud detection model should extract and transfer knowledge from a sequence of regions to preserve regional generality.

Our proposed cross-regional fraud detection problem requires both general knowledge with regional generality and specific knowledge with regional uniqueness. In response, we propose a novel **C**ross-regional **C**ontinual **L**earning (**CCL**) framework to address the previously mentioned challenges in a unified manner. CCL presents a novel plug-and-play prototype-based replay method for knowledge transfer and improves the regularization-based parameter smoothing strategy to alleviate forgetting. Specifically, to address challenge (1), we propose a prototype-based replay method to preserve the previously learned knowledge from formerly appeared regions with an incremental experience buffer. To deal with challenge (2), the proposed CCL alleviates the forgetting issue by utilizing an improved regularization-based parameter smoothing method to constrain the training process and parameter updating. Furthermore, we introduce a novel heterogeneous trade graph construction method to uncover hidden patterns of fraudulent activities and explore deeper semantic information to facilitate knowledge transfer.

Building upon the foundational insights and techniques developed in previous chapters, here we focus on extending continual learning principles to a real-world application: *cross-regional fraud detection*. In practical financial systems, models must operate across geographically diverse regions, each with unique behavioral patterns, data distributions, and entity relationships. This dynamic setting introduces a variety of challenges, including those we have seen before, such as distribution shift, catastrophic forgetting, and the need for efficient knowledge transfer across domains.

To address these issues, we propose a novel continual learning framework tailored for fraud detection tasks involving sequential regional adaptation. Our approach not only preserves previously learned knowledge but also facilitates forward transfer of useful patterns, enabling the model to maintain high detection accuracy in new regions without retraining from scratch. This is achieved through a unified integration of semantic graph modeling and continual learning techniques.

The remainder of this chapter presents the problem formulation, methodological design, and empirical validation of our approach. Below, we summarize the core contributions of this chapter:

- **Problem Formulation.** We introduce and formally define a novel task—*cross-regional fraud detection*—that integrates the dynamics of regional adaptation with the principles of continual learning. This formulation reflects real-world deployment scenarios where models must adapt to new geographic markets without retraining from scratch. To the best of our knowledge, this is the first work to investigate fraud detection in a continual learning setting, bridging a critical gap between static fraud detection systems and dynamic, evolving environments.

- **Graph-based Knowledge Representation.** To capture the diverse and intricate relationships among financial entities, we construct a *heterogeneous trade graph*. This graph structure encodes semantic information and models complex interdependencies between users, transactions, and institutions across regions, serving as a strong relational backbone for downstream learning and knowledge transfer.
- **Cross-Regional Knowledge Transfer Framework.** We propose a novel learning framework termed **Cross-Regional Continual Learning**, which enables adaptive and efficient transfer of knowledge across regions. CCL is built upon two complementary mechanisms: (1) a *prototype-based replay* module that preserves representative knowledge from previous regions to guide learning in new ones; and (2) an enhanced *parameter smoothing regularization* strategy that mitigates catastrophic forgetting by stabilizing model updates during sequential training.
- **Empirical Validation on Real-World Data.** We conduct experiments on a large-scale, real-world financial fraud dataset to validate the proposed method. Results show that CCL consistently outperforms state-of-the-art baselines across multiple evaluation tasks involving regional shifts. Additional ablation studies and qualitative analyses (e.g., representation visualization) further demonstrate the effectiveness and robustness of the proposed knowledge transfer mechanisms in mitigating catastrophic forgetting and enhancing generalization.

The remainder of this chapter is organized as follows. We review the recent studies of fraud detection and the advances of continual learning in section 7.2. In section 7.3, we first formalize the cross-regional fraud detection problem with fundamental definitions. Then, we introduce each component of the proposed CCL, involving graph construction, prototype-based knowledge replay, and regularization-based parameter smoothing. Subsequently, the experiments and results quantifying the proposed model are provided in section 7.4. Finally, in section 7.5, we conclude this work and remark on future works.

7.2 A Brief Recall on Fraud Detection

Fraud detection has always constituted an important task in risk management, demonstrating its efficacy in thwarting deleterious activities [143]. In previous works, rule-based methods predominantly relied on expert-crafted rules, presupposing that fraudulent activities have some apparent patterns. Across recent decades, the use of Deep Neural Networks (DNNs) has become ubiquitous in the field of fraud detection [144]. By refining informative representations from raw data, deep neural networks demonstrated their capabilities in the domain of fraud detection [145]. Several investigations harnessed temporal cues latent in sequential data through diverse iterations

of recurrent neural networks [146, 147]. Further explorations manifested in DNN-based methodologies [148], retrieving implicit features from data through feature engineering. [149] investigated fraud detection in the unsupervised framework and applied a long short-term memory mechanism. [142, 150] developed unsupervised schemes for sequential data in online anomaly detection. Auto-encoders, recognized for their ability to distill underlying patterns, have also found applications in fraud detection [145].

However, the aforementioned methods all have their shortcomings. Powered by rudimentary machine learning algorithms, the rule-based approaches exhibited a substantial reliance on expert experience [151], while addressing intricate and dynamic patterns. Furthermore, the current array of deep learning models still faces challenges in handling complex topological data with intricate relationships between various entities, posing an ongoing obstacle to fraud detection.

Recently, graph neural networks (GNNs) have emerged as a robust, reliable, and promising technique for financial fraud detection. Notable examples include GCN [152], GraphSAGE [153], and GAT [154], which utilize convolutional methods, the LSTM architecture, and self-attention mechanisms, respectively, to capture and aggregate complex information among nodes [155]. HAN [156] proposed a hierarchical attention aggregation that integrated node-level and semantic-level information from multiple meta-paths. In [157, 158], different types of objects were considered for building heterogeneous information networks. [159] proposed a graph construction method for GNN-based fraud detection on the non-attributed graph using a graph pre-training strategy. [160, 161] modeled dynamic connectivity patterns through bipartite graph embedding and then detected anomalies from the graph. [162] presented a hierarchical prototype network to represent the continuously expanded graphs. CARE-GNN [163] is a GNN-based fraud detector based on a multi-relation graph that enhances its performance against camouflaged fraudsters.

However, traditional GNN-based methods only deal with static tasks and cannot be directly applied to streaming data for fraud detection [164]. Recent works combined graph neural networks with continual learning to deal with streaming graphs and dynamic graphs[165, 166, 167, 168], while ignoring the knowledge extraction, memorization, and transferring during the process of learning new tasks. In contrast, our work here extends GNN-based approaches to the cross-regional fraud detection task with dynamic data, integrating with improved continual learning techniques. Given the power of providing neighborhood and fusing structure attributes in graph neural networks, we take a variant of GAT [154] as our backbone. Specifically, we generate multiple types of meta-paths and entities to capture intrinsic patterns, consider heterogeneous structures, and explore temporal-spatial semantics for cross-regional fraud detection. Different from previous works, the key insight of our approach is in extending GNN-based fraud detection within the CCL scenarios.

7.3 Methodology

In this section, we begin by presenting an overview of our proposed cross-regional fraud detection framework. Subsequently, we introduce the foundational concepts and notations that will be employed consistently throughout this paper. We also identify the research challenges in cross-regional fraud detection. Following this, we outline the procedure for constructing heterogeneous trade graphs for data engineering and provide a brief introduction to the Graph Neural Network backbone utilized in this study. Then, we investigate the technical particulars of the main continual learning components proposed. Finally, we conclude this section by detailing the algorithm with explanations.

7.3.1 Overview and Problem Formulation

The proposed framework for cross-regional fraud detection is designed to extract, retain, and transfer knowledge acquired from a sequence of region-specific learning tasks, thereby ensuring robust and consistent performance across all regions. As illustrated in Figure 7.3, the core of the framework—termed **Cross-Regional Continual Learning**—builds upon two foundational components: *heterogeneous graph construction* and *continual knowledge transfer*.

Specifically, the framework begins by transforming raw structured transaction data into **Heterogeneous Trade Graphs**, which encode multi-typed entities and their complex interactions, capturing rich relational and semantic information critical for fraud detection. These graphs serve as dynamic knowledge representations that facilitate both region-specific modeling and transferable insights across regions.

When faced with a sequence of cross-regional tasks, the model must incorporate knowledge from previously encountered regions while adapting to new ones. To this end, we introduce a novel **prototype-based knowledge replay** mechanism that maintains a buffer of representative samples from past tasks. This mechanism enables the model to selectively replay informative instances during the training of a new task l, thereby preserving prior knowledge and guiding the learning process with meaningful context. In addition, to address the issue of catastrophic forgetting, we integrate a parameter-level regularization strategy that constrains the update of crucial model parameters. This approach stabilizes learning across tasks and ensures that valuable knowledge is not overwritten during sequential adaptation. At the architectural level, CCL provides a powerful **graph-based representation learning backbone**, enhanced with attention mechanisms, to generate expressive and task-relevant embeddings from the constructed heterogeneous trade graphs. These embeddings are then passed through a binary classifier that predicts the likelihood of fraudulent activity at the transaction level.

For clarity and consistency, all notations used throughout the chapter are summarized in Table 7.1, along with detailed descriptions of their meanings and roles in the proposed framework.

Table 7.1: Notations and explanations.

Notation	Explanation
$\mathcal{D}_l$	The raw structured dataset of region l
$\mathcal{G}_l$	The heterogeneous trade graph of region l
$\mathcal{V}_l$	Node set of $\mathcal{G}_l$
$\mathcal{E}_l$	Edge set of $\mathcal{G}_l$
$\mathbf{X}_l$	Feature set of $\mathcal{G}_l$
$\mathcal{P}_m$	A meta-path m
$N_i^{\mathcal{P}_m}$	Neighbors of node i on Meta-path m
$\mathbf{W}$	A transformation matrix
$\mathbf{h}^i$	Initialized hidden representation of node i
$\mathbf{e}_{(i,j)}^{\mathcal{P}_m}$	The attention coefficient between node pair (i,j)
$\alpha_{(i,j)}^{\mathcal{P}_m}$	The normalized attention between node pair (i,j)
$\mathbf{z}_i^{\mathcal{P}_m}$	Meta-path based representation of target node i
$\mathbf{Z}_{\mathcal{P}_m}$	Semantic-specific node embedding of meta-path $\mathcal{P}$
$\beta_{\mathcal{P}_m}$	The importance of meta-path $\mathcal{P}_m$
$\mathbf{Z}$	The final representation
$\mathbb{B}_l$	The experience buffer collected from $\mathcal{D}_l$
$\mathbb{F}$	The Fisher Information matrix

The basic goal of cross-regional fraud detection is to continually detect fraudulent activities in the process of business expansion with a sequence of cross-regional data, which can be precisely formulated as follows:

Definition 13 ***Problem definition (Cross-Regional Fraud Detection).*** *Given a sequence of cross-regional datasets from regions $(\mathcal{R}_1, \cdots, \mathcal{R}_l, \cdots)$, the series of cross-regional datasets can be formulated as $(\mathcal{D}_1, \cdots, \mathcal{D}_l, \cdots)$. As the fraud detection task expands into the l-the region, $\mathcal{D}_l = (\mathcal{D}_l^{tr}, \mathcal{D}_l^{te})$, we aim to update a* unified neural function Φ *using the dataset $\mathcal{D}_l^{tr}$. The unified neural function $\Phi : \mathcal{D}_l^{te} \to \mathbb{R}^d$ maps input data to a low-dimensional space d. Subsequently, a dense layer is employed to identify whether a given sample is fraudulent.*

There are two important objectives: ***(1)*** *upholding promising efficacy in fraud detection across all regions, and* ***(2)*** *improving the fraud detection performance within the present regional data through the incorporation of previously acquired knowledge.*

7.3.2 Construct Heterogeneous Trade Graphs

Data acquisition. The business expansion strategies employed by financial institutions exemplify a geographically aware approach. Cross-regional tasks involve extending operations beyond the initial locale into diverse geographical areas. Financial institutions typically initiate a pilot phase within a specific city before expanding

their operations to other regions. Therefore, it is important to ensure the absence of data overlap among distinct regions during the data acquisition process.

In this work, based on a desensitized financial database that contains more than one million credit card transaction records in the United States Mainland from the duration of 1st Jan 2019 - 31st Dec 2020 [1], we extract five regions' transaction data according to different geographical ranges: $\mathcal{R}_1$ $(30°N - 40°N, 95°W - 100°W)$, $\mathcal{R}_2$ $(40°N - 50°N, 75°W - 80°W)$, $\mathcal{R}_3$ $(30°N - 40°N, 75°W - 80°W)$, $\mathcal{R}_4$ $(40°N - 50°N, 95°W - 100°W)$, $\mathcal{R}_5$ $(30°N - 40°N, 90°W - 95°W)$.

It is noteworthy that the raw data is structured and tabular, with each region being distinct and non-overlapping. The sequence of tasks is subjected to random shuffling to mitigate order effects. Simultaneously, to alleviate the issues posed by missing values in financial data and the sparsity of fraudulent data, we implement data imputation and data augmentation strategies to preprocess the raw financial datasets. Note that while the use of the above five regions serves an illustrative purpose, our proposed framework is designed to accommodate the integration of a greater number of regions in practical applications.

A solitary fraudulent incident can encompass numerous transactions, implicating multiple cardholders and/or merchants. This intricate interplay gives rise to a convoluted network of interactions that underlie each fraudulent occurrence [141]. Within this framework, the areas of feature engineering encompass the extraction of nodes and the exploration of relationships. To encapsulate both inherent and network-oriented attributes of credit card transactions, we introduce a graph-generation procedure that converts organized credit card transaction data into a topological graph.

Hence, each financial heterogeneous trade graph G_l stem from region $\mathcal{R}_l$ can be defined as below:

Definition 14 *Let $\mathcal{G}_l = (\mathcal{V}_l, \mathcal{E}_l, \mathbf{X}_l)$ denote a financial heterogeneous trade graph extracted from l-th region's transaction data $\mathcal{D}_l$, where $\mathcal{V}_l$ is the set of nodes, $\mathcal{E}_l$ is the edge set and $\mathbf{X}_l$ is the initial feature set. $\mathcal{G}_l$ is associated with a node type mapping function $\phi : \mathcal{V}_l \to \mathcal{A}$ and an edge type mapping function $\psi : \mathcal{E}_l \to \mathcal{O}$, where the $\mathcal{A}$ and $\mathcal{O}$ respectively represent the set of node types and the set of edge types of $\mathcal{G}_l$. Notably, we have $|\mathcal{A}| + |\mathcal{O}| > 2$ since $\mathcal{G}_l$ is a heterogeneous graph, where nodes and edges are of different types.*

Node Extraction. The main objective of our work is to detect whether a credit card transaction is fraudulent. Therefore, each transaction is represented as a node identified by its transaction ID. Additionally, cardholders and merchants play crucial roles in transactions, so they are also represented as nodes. In addition, [148] observed that fraudsters' activities are typically intensive and significantly different

[1] https://kaggle.com/kartik2112/fraud-detection

from those of legitimate transactions. As shown in Figure 7.2, the fraudulent transactions and legitimate transactions exhibit markedly different behavioral patterns: the frequency of fraudulent transactions changes abruptly over time, whereas legitimate transactions remain significantly more stable, indicating the attackers operate within limited time frames for their campaigns [169]. Therefore, we need to consider the varying behavioral changes due to different transaction time intervals in a fraud detection system. To address this, we introduced time nodes into the heterogeneous trade graph. Our datasets include timestamped credit card transactions, logging the exact time of each transaction. To manage the scale of timestamps, we divided the time into 24 slices, with each time node representing one hour (e.g., [0:00, 1:00] and [1:00, 2:00]). Consequently, our heterogeneous graph G_l includes four types of nodes: cardholders C_l, merchants M_l, time slices S_l, and transactions T_l. Thus, $\mathcal{V}_l = \{T_l, C_l, M_l, S_l\}$, with $|\mathcal{A}_l| = 4$.

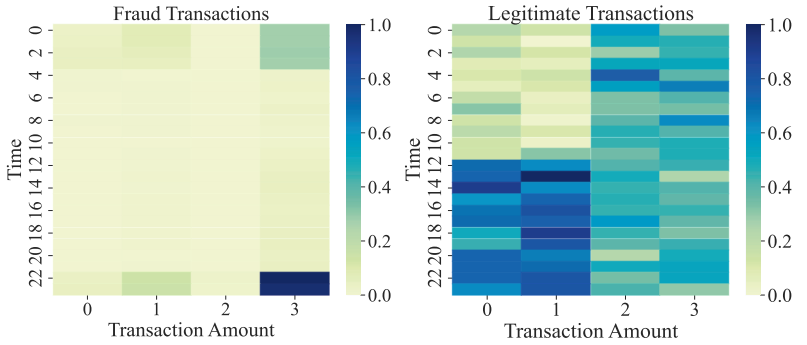


Figure 7.2: Heat maps of temporal slices. On the horizontal axis is the amount of fraudulent transactions, and on the vertical axis are time slices. The darker the color, the more fraudulent transactions occurred during the current period.

Relationship Exploration. To explore informative interactions among different types of nodes, meta-paths are widely used in graph generation. Meta-paths are basic logical routes that aim at connecting nodes with a composite relationship [170, 171]. Such a paradigm motivates us to generate meaningful meta-paths to explore the diverse semantic relationships among heterogeneous entities. Formally, the definition of a meta-path $\mathcal{P}$ is described as follows:

Definition 15 *A meta-path [156] is a path $\mathcal{P}$ in form of $\mathcal{V}^1 \xrightarrow{R_1} \mathcal{V}^2 \xrightarrow{R_2} \cdots \xrightarrow{R_k} \mathcal{V}^{k+1}$, where $\mathcal{V}^1$ to $\mathcal{V}^{k+1}$ are $(k+1)$ different types of nodes. The relations between node types $\mathcal{V}^1$ to $\mathcal{V}^{k+1}$ are denoted as a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_k$, where the $\circ$ is the composition operator.*

Given the four types of nodes $\mathcal{V}_l = \{T_l, C_l, M_l, S_l\}$, we generate three types of meta-paths: ***TCT*** (Transaction-Cardholder-Transaction), ***TMT*** (Transaction-Merchant-Transaction) and ***TST*** (Transaction-Time Slices-Transaction), respectively. To bet-

ter target transaction nodes, meta-paths are constructed through the underlying interactional logic between transaction nodes and the other three types of nodes. *TCT* depicts the connections between different transaction nodes that are initiated by the same cardholder. *TMT* describes the interactions of different transaction nodes that take place at the same merchant. *TST* implies that there are associations among transaction nodes that are launched in the same time interval. Finally, by defining the heterogeneous meta-paths, we complete the construction of the network structure for heterogeneous trade graphs.

Node features. Beyond nodes and the network structure, there are informative features with respect to transactions, such as transaction amount and transaction type. Additionally, the dataset contains numerous desensitization features that depict the inherent semantics of transactions, which could boost the downstream fraud detection task. Thus, we assign all these features to transaction nodes as corresponding feature vectors, i.e., each transaction node $v_l^i \in \mathcal{V}_l$ is associated with a feature vector $\mathbf{x}_l^i$. Hence, our generated financial HGTs are not only depicted as heterogeneous graphs with multiple types of nodes and relationships but also associated with different feature spaces.

7.3.3 The Proposed CCL Model

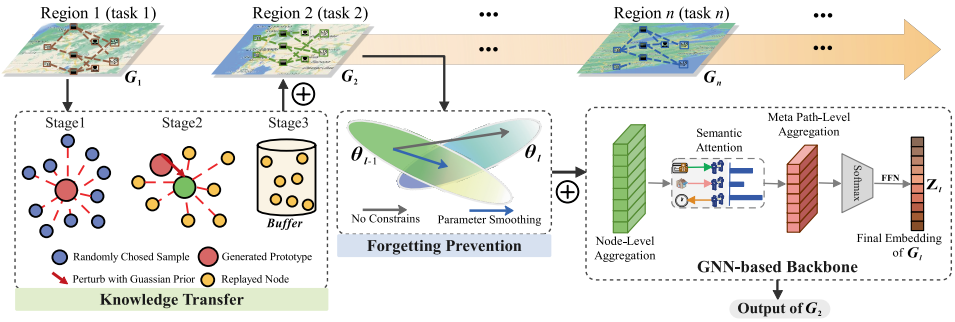


Figure 7.3: The overall framework of the proposed CCL. Given a sequence of cross-regional fraud detection tasks, the proposed prototype-based replay method for knowledge transfer and the forgetting prevention approach are illustrated. Take the learning process of $\mathcal{G}_2$ as an example. In stage 1 of knowledge transfer, we randomly choose some target nodes from $\mathcal{G}_1$ and generate a prototype $\mathbf{c}^1$. In stage 2, a Gaussian prior is added to perturb the prototype. In stage 3, nodes are sorted according to the distance from the perturbed prototype $\mathbf{c}'^1$, and we select the replay nodes according to the replay ratio r. Then, all the selected nodes and their neighbors are stored in the experience buffer $\mathbb{B}_1$. Next, the replay buffer is appended to $\mathcal{G}_2$ for the training, with parameter smoothing for forgetting prevention. Finally, the final embedding of $\mathbf{Z}_l$ is obtained by the GNN-based backbone accordingly.

Given a previously learned neural model Φ_{l-1} with parameters Θ_{l-1}, a naive application of Φ_{l-1} to data from a new region l often results in significant performance

degradation due to the *region-specific distribution shift*. Conversely, training a new model solely on region l's data risks neglecting the valuable generalizable knowledge learned from earlier regions, and more critically, leads to catastrophic forgetting in the cross-regional continual learning process.

To address this trade-off between adaptation and retention, our objective is to construct a model that can not only acquire region-specific knowledge from the current region l, but also *retain and consolidate the experiences* accumulated across previously encountered regions $1, 2, \ldots, l-1$. Importantly, this is achieved *without* retraining from scratch or deploying multiple region-specific models, thereby enabling a unified and memory-efficient continual learning process.

To this end, we introduce two complementary yet computationally efficient continual learning strategies:

- **Prototype-based Knowledge Replay.** Inspired by recent advances in continual learning [166], we maintain a compact experience buffer $\mathbb{B}_{l-1}$ that stores prototypical representations from previously encountered regions. During the learning of the new task, the model is co-trained on both the current heterogeneous trade graph $\mathcal{G}_l$ and the buffered samples in $\mathbb{B}_{l-1}$, enabling the model to revisit and reinforce previously acquired knowledge.
- **Parameter Regularization for Knowledge Retention.** In addition to data replay, we enforce consistency between the current model Φ_l and the preceding model Φ_{l-1} by applying a regularization-based constraint on critical parameters. This strategy preserves important weights associated with prior knowledge and guides the optimization process to prevent abrupt drift in the learned representations.

The technical implementation of these strategies, including buffer construction, replay sampling, and parameter regularization formulation, is detailed in the following subsections.

7.3.3.1 Prototype-Based Knowledge Replay

To mitigate catastrophic forgetting while maintaining a compact memory footprint, we propose a novel *prototype-based knowledge replay* mechanism for cross-regional continual learning. This strategy enables the current task model to selectively incorporate representative knowledge from previously learned regions without accessing the full historical data.

Given the heterogeneous trade graph $\mathcal{G}_{l-1}$ constructed from the $(l-1)$-th region's data $\mathcal{D}_{l-1}$, we build a replay buffer $\mathbb{B}_{l-1}$ that stores a small subset of informative samples, with $|\mathbb{B}_{l-1}| \ll |\mathcal{D}_{l-1}|$. Unlike traditional methods that rely on random sampling or naive memory storage of original nodes, our approach generates **pseudo-samples**

guided by prototypes, which effectively summarize the semantic distribution of prior tasks while enhancing replay efficiency and generalization.

As illustrated in Stage 1–3 of Figure 7.3, the construction of $\mathbb{B}_{l-1}$ follows three sequential steps:

Stage 1: Prototype Construction. We randomly sample a set of labeled transaction nodes from $\mathcal{G}_{l-1}$ according to a predefined replay ratio r. Let each node be associated with an attribute vector $\mathbf{x}_i^{l-1}$. The prototype vector $\mathbf{c}^{l-1}$ is computed as the centroid of all selected feature vectors:

$$\mathbf{c}^{l-1} = \frac{1}{|\mathbb{B}_{l-1}|} \sum_{\mathbf{x}_i^{l-1} \in \mathbb{B}_{l-1}} \mathbf{x}_i^{l-1}. \tag{7.1}$$

This prototype acts as a compressed representation of the semantic center for region $(l-1)$'s fraud distribution.

Stage 2: Gaussian Perturbation. To improve the diversity and robustness of replayed samples, we perturb the prototype vector $\mathbf{c}^{l-1}$ with a Gaussian before obtaining a modified prototype $\mathbf{c'}^{l-1}$. This perturbation simulates realistic distributional variations and reduces overfitting to fixed replay centers:

$$\mathbf{c'}^{l-1} = \mathbf{c}^{l-1} + \mathcal{N}_0^{\sigma^2 \mathbf{I}}.$$

Stage 3: Distance-Based Sample Selection. Next, we compute the Euclidean distance between each transaction node in $\mathcal{G}_{l-1}$ and the perturbed prototype $\mathbf{c'}^{l-1}$. Nodes are ranked in ascending order of distance, and the top-r nodes are selected as replay candidates. To enhance contextual consistency, we also include their immediate neighbors in the final buffer:

$$\mathbb{B}_{l-1} = \text{Top-}r \left(\{ \mathbf{x}_i^{l-1} \in \mathcal{G}_{l-1} \mid \|\mathbf{x}_i^{l-1} - \mathbf{c'}^{l-1}\|_2 \} \right) \cup \mathcal{N}(\mathbf{x}_i^{l-1}),$$

where $\mathcal{N}(\cdot)$ denotes the set of neighboring nodes.

Security-Driven Replay Constraints. Considering the sensitive nature of financial fraud detection, we constrain the replay buffer to contain only labeled transaction nodes, avoiding the risk of propagating noise or unverifiable patterns from unlabeled data. This security-aware sampling improves interpretability and aligns with practical deployment requirements.

Integration into CCL. The resulting experience buffer $\mathbb{B}_{l-1}$ is appended to the current task graph $\mathcal{G}_l$ during training. This replayed knowledge is jointly optimized alongside region-specific data, enabling effective transfer and retention across tasks.

As shown in the left part of Figure 7.3, this mechanism plays a central role in the knowledge transfer pipeline of the proposed CCL framework.

7.3.3.2 Regularization-based Parameter Smoothing

Although the first strategy, to some extent, is capable of alleviating the forgetting issue from the sample consolidation aspect, it is unable to thoroughly familiarize itself with what knowledge is extracted from the previous regions. To further alleviate the knowledge forgetting, we are inspired by recent Elastic Weight Consolidation (EWC) [21], and use Fisher Information to evaluate the importance of each parameter in Θ_l w.r.t. Φ_l, which can be defined as:

$$\mathbb{F}_l = \frac{1}{|\mathbf{X}^{l-1}|} \sum_{x \in \mathbf{X}^{l-1}} \left[g\left(x; \Theta_l\right) \cdot g\left(x; \Theta_{l-1}\right)^\top \right], \tag{7.2}$$

where g is the first-order derivative of the loss.

Hence, when the current task with the new heterogeneous trade graph $\mathcal{G}_l$ arrives, we add a smoothing term to constrain the parameter optimization based on $\mathbb{F}_l$, which can be formulated as:

$$\mathcal{L}_1 = \frac{\lambda}{2} \sum \mathbb{F}_l \left(\Theta_l(i) - \Theta_{l-1}(i)\right)^2, \tag{7.3}$$

where λ is a hand-crafted weight.

Although financial transactions are usually region-related, fraud patterns have some common characteristics. For instance, we can observe that fraudulent transactions have high time correlations as shown in Figure 7.2. Thus, it is necessary to consider task similarity in the financial context. Inspired by multi-task learning [172], we add a regularization term to measure the parameter distance. Therefore, we can rewrite the Equation 7.3 as:

$$\mathcal{L}_2 = \frac{\lambda}{2} \sum \mathbb{F}_l \left(\Theta_l(i) - \Theta_{l-1}(i)\right)^2 + \gamma(||(||\Theta_l||_2, ||\Theta_{l-1}||_2)||_1). \tag{7.4}$$

where the second term is the $l_{2,1}$-norm and γ is a hand-craft weight.

In the end, we can summarize our objective for the region $l+1$, that is:

$$\min_{\Theta_{l+1}} \{\mathcal{L}_1(\Theta_{l+1}) + \mathcal{L}_2(\Theta_{l+1})\}. \tag{7.5}$$

Herein, the loss function throughout this work is the cross-entropy loss defined as:

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i^l \log \widehat{y_i^l} + \left(1 - y_i^l\right) \log \widehat{y_i^l} \right), \tag{7.6}$$

where N is the amount of transaction nodes, y_i^l is the true label of node i, $\widehat{y_i^l}$ denotes the prediction of node i.

7.3.4 Graph Representation Learning Backbone

Upon acquiring Heterogeneous Trade Graphs, the subsequent step involves procuring representations through a suitable GNN-based backbone. This preparation is vital for enabling cross-regional fraud detection through continual learning approaches.

As different types of nodes hold different semantic spaces, we first transfer each target node into a new unified semantic space. Given a node $\mathbf{x}_l^i$, it is the initial representation of a target node i in $\mathcal{G}_l$, and we define a learnable transformation matrix $\mathbf{W}_x$ to map the initial $\mathbf{x}_l^i$ to a new feature space:

$$\mathbf{h}_l^i = \mathbf{W}_x \cdot \mathbf{x}_l^i, \tag{7.7}$$

where $\mathbf{h}_l^i$ is the transformed representation of node i in a new semantic space. Notably, we omit l in the following parts for simplicity.

After projecting all the initial representations into a new latent space, we then leverage a node-oriented attention mechanism to automatically learn the weights among the target node and its neighbors. Following the recent attention mechanism in [154], the attention between a node pair (i, j) is the importance of the target node i for its meta-path-based neighbor node j. Given a pair of node (i, j) which connected via a meta-path $\mathcal{P}_m$, the attention coefficient $\mathbf{e}_{(i,j)}^{\mathcal{P}_m}$ can be formulated as follows:

$$\mathbf{e}_{(i,j)}^{\mathcal{P}_m} = att_\eta(\mathbf{h}^i, \mathbf{h}^j; \mathcal{P}_m), \tag{7.8}$$

where the $\mathbf{e}_{(i,j)}^{\mathcal{P}_m}$ is asymmetric and the att_η is an attention mechanism through the deep neural networks parameterized by η. Then, we perform normalization to get the weight coefficient $\alpha_{(i,j)}^{\mathcal{P}_m}$ via meta-path $\mathcal{P}_m$ using Softmax function:

$$\left[\alpha_{(i,j)}^{\mathcal{P}_m}\right] = \text{softmax}\left(\left[\mathbf{e}_{(i,j)}^{\mathcal{P}_m}\right]\right) = \frac{\exp\left(\left[\mathbf{e}_{(i,j)}^{\mathcal{P}_m}\right]\right)}{\sum_{k \in N_i^{\mathcal{P}_m}} \exp\left(\left[\mathbf{e}_{(i,k)}^{\mathcal{P}_m}\right]\right)}, \tag{7.9}$$

where $N_i^{\mathcal{P}_m}$ is the set of meta-path $\mathcal{P}_m$ based neighbors of target node i. Hence, the meta-path based representation of target node i, $\mathbf{z}_i^{\mathcal{P}_m}$, can be obtained from its neighbors with corresponding attention coefficients:

$$\mathbf{z}_i^{\mathcal{P}_m} = \text{Sigmoid}(\sum_{j \in N_i^{\mathcal{P}_m}} \alpha_{(i,j)}^{\mathcal{P}_m} \cdot \mathbf{h}^j). \tag{7.10}$$

Inspired by [156], we enhance the fusion of multiple semantics within a heteroge-

neous trade graph by incorporating various types of meta-paths. This approach allows us to assess the importance of different meta-paths and generate final representations. Once the meta-path-based representation $\mathbf{z}_i^{\mathcal{P}_m}$ of node i is obtained, the learned attention coefficient weights for each meta-path $\mathcal{P}_m$ can be formulated as follows:

$$\mathbf{w}^{\mathcal{P}_m} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathrm{q}^{\mathrm{T}} \cdot tanh\left(\mathbf{W}' \cdot \mathbf{z}_i^{\mathcal{P}_m} + \mathbf{b}\right), \tag{7.11}$$

where q^{T} and $\mathbf{W}'$ are a learnable weight matrix, $\mathbf{b}$ is learnable bias, $tanh$ refers to the hyperbolic tangent activation function. Notably, the average of all semantic-specific node embeddings can be interpreted as the importance of each meta-path.

After obtaining the importance of each meta-path, we normalize them via the Softmax function:

$$[\beta_{\mathcal{P}_m}] = \mathrm{Softmax}(\mathbf{w}^{\mathcal{P}_m}) = \frac{\exp([\beta_{\mathcal{P}_m}])}{\sum_{k=1}^{M} \exp([\beta_{\mathcal{P}_k}])}, \tag{7.12}$$

where M is the total amount of all meta-paths, and the learned weight can be interpreted as the importance of each meta-path. Then, we finally obtain the representation $\mathbf{Z}$:

$$\mathbf{Z} = \sum_{i=1}^{M} \beta_{\mathcal{P}_m} \cdot \mathbf{Z}_{\mathcal{P}_m}. \tag{7.13}$$

By aggregating all semantic-specific embeddings, the final representation is formed and can be applied to cross-regional fraud detection. Subsequently, our attention shifts towards the exploration of the two continuous learning strategies that we have put forth for cross-regional fraud detection.

The time complexity of our algorithm accounts for the computational demands across multiple stages. For each of the n heterogeneous trade graphs, the process unfolds over each epoch. Within each epoch, operations are performed for N nodes and M meta-paths. Specifically, for each meta-path, the algorithm involves a projection of node features, extraction of meta-path-based neighbors, calculation of weight coefficients, and aggregation of node embeddings. These operations are conducted for an average of $N_{\mathcal{P}_m}$ neighbors per node, considering the complexity of d_h hidden representations. Additionally, the computational load includes back-propagation steps for parameter updates. Considering these factors, the overall time complexity is given by $O(N \cdot M \cdot (N_{\mathcal{P}_m} + d_h))$. This captures the intricate computations required for continual learning in the context of cross-regional fraud detection.

7.3.5 Overall Framework and Algorithm

We now present the complete procedural workflow of the proposed CCL framework, as outlined in Algorithm 5.

Given a sequence of heterogeneous trade graphs, the algorithm begins by randomly initializing the model parameters (Step 1) and preparing an empty experience buffer

(Step 2). For the first task—corresponding to the initial graph $\mathcal{G}_1$—the model undergoes standard graph representation learning from Steps 3 to 22. At the end of this stage (Step 22), we obtain the optimal model parameters Θ_1^*.

Algorithm 5: The overall process of CCL.

Input: A sequence of heterogeneous trade graphs $\mathcal{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_l, \cdots, \mathcal{G}_n\}$.
Random initialize parameters Θ_1 of CCL;
Set $\mathbb{B} = \varnothing$;
if $l = 1$ **then**
  for *each epoch* **do**
    for *a specific meta-path* $\mathcal{P}_m$ **do**
      Projection $\mathbf{h} \leftarrow \mathbf{W}_x \cdot \mathbf{x}$;
      for *each node* i **do**
        Extract meta-path based neighbors $N_i^{\mathcal{P}_m}$;
        for $j \in N_i^{\mathcal{P}_m}$ **do**
          Calculate the weight coefficient $\alpha_{(i,j)}^{\mathcal{P}_m}$;
        Calculate the single type node embedding $\mathbf{z}_i^{\mathcal{P}_m} \leftarrow \text{Sigmoid}(\sum_{j \in N_i^{\mathcal{P}_m}} \alpha_{(i,j)}^{\mathcal{P}_m} \cdot \mathbf{h}_j)$;
      Concatenate from all attention heads;
    Obtain the importance of each meta-path $\beta_{\mathcal{P}_m}$;
    Obtain the final representation by aggregating $\mathbf{Z} = \sum_{i=1}^{M} \beta_{\mathcal{P}_m} \cdot \mathbf{Z}_{\mathcal{P}_m}$;
    Infer node label via $\hat{y}^1 = f_1(\mathbf{Z}; \epsilon)$;
    Calculate Cross Entropy Loss via Equation 7.6;
    Back-propagation and update parameters;
  Obtain optimal Θ_1^*;
for $l = 2, \cdots, n-1$ **do**
  Let $\Theta_l = \Theta_{l-1}^*$;
  Sample a experience buffer $\mathbb{B}_{l-1}$ from G_{l-1} for experience replay;
  Generate a twin experience buffer $\mathbb{B}'_{l-1}$ with Gaussian prior;
  Add the experience buffer $\mathbb{B}'_{l-1}$ to the current heterogeneous trade graphs $\mathcal{G}_l$;
  for *each epoch* **do**
    Calculate graph representation and infer node label as steps 5 to 18;
    Calculate Loss via Equation 7.5;
    Back-propagation and update parameters;
  Obtain Optimal Θ_l^*;
Output: $\{\Theta_n^*\}$.

Starting from the second graph $\mathcal{G}2$, the continual learning procedure follows the pipeline defined in Steps 24 to 35. Specifically, Step 25 initializes the current model parameters Θ_l using the optimal parameters from the previous task, $\Theta l - 1^*$. Steps 26 to 28 construct a prototype-based experience buffer by drawing samples from $\mathcal{G}_{l-1}$ and enhancing them via a Gaussian prior, thus facilitating cross-task knowledge transfer. The buffer is then integrated into the current task's data.

Subsequently, from Steps 29 to 33 (reusing the structure from earlier), the model performs representation learning on $\mathcal{G}_l$ while leveraging regularization-based smoothing to mitigate forgetting. The training concludes with the derivation of the updated model Θ_l^* for the current task. This process iterates across all graphs in the sequence, enabling CCL to accumulate and adapt knowledge over time.

7.4 Experiments

In this section, we present the experimental evaluation of the proposed CCL framework. The primary objective is to assess its effectiveness in addressing the challenges of continual learning in the context of cross-regional financial fraud detection. To this end, we conduct comparisons with a range of state-of-the-art continual learning baselines and perform ablation and sensitivity analyses.

Our experiments are designed to answer the following key questions specific to the CCL framework:

- **Q1:** Does the proposed CCL framework outperform state-of-the-art continual learning baselines on the cross-regional credit card fraud detection task?
- **Q2:** How do the core components of CCL—namely, prototype-based knowledge replay and parameter regularization—contribute to overall performance?
- **Q3:** How sensitive is the performance of CCL to different hyperparameter configurations (e.g., replay ratio, regularization weight)?
- **Q4:** Do the constructed Heterogeneous Trade Graphs provide interpretable insights relevant to financial fraud detection?

The remainder of this section details the experimental setup, evaluation metrics, and findings addressing the key questions posed above regarding the performance and behavior of the CCL framework.

7.4.1 Experimental Settings

7.4.1.1 Implementation Details and Datasets

We reproduce the baselines and implement our proposed model with the PyTorch library, accelerated by one NVIDIA RTX 3090 GPU. We use Adam as the optimizer with a learning rate of 0.05 and a weight decay of 0.001 as the default for comparison. We train 200 epochs for each region with early stopping to accelerate the training process, and randomly sample 15% of each region's transactions for our experience replay buffer, λ is 1.5, and γ is 0.00025.

Table 7.2 summarizes the key statistics of the five regional datasets used in our experiments. Each region corresponds to a distinct financial ecosystem, with varying numbers of transactions, cardholders, and merchants. Notably, the number of

transactions ranges from 10,673 to 14,513, reflecting diverse market activity levels. The number of cardholders and merchants is relatively stable across regions, indicating similar population scales. In contrast, the total number of edges—representing all transactional interactions—varies more significantly, highlighting differences in graph connectivity and structural complexity. These variations emphasize the necessity of a continual learning framework capable of generalizing across heterogeneous regional distributions while preserving region-specific knowledge.

Table 7.2: The statistics of datasets for each region.

Region	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$
# Transactions	11,291	11,198	12,478	14,513	10,673
# Card Holders	103	124	126	115	118
# Merchants	700	698	698	699	700
# Total Edges	33,873	33,594	37,434	43,539	32,019

7.4.1.2 Baselines

We compare the proposed CCL with six representative continual learning baselines and adjust them for the proposed problem. The continual learning strategies used by these baselines include regularization-based methods, model structure-based methods, and replay-based methods.

- *EWC* [21]: A classical framework that protects the important weights of previous tasks to overcome catastrophic forgetting.
- *GEM* [173]: A model using gradient episodic memory to alleviate forgetting, while allowing beneficial transfer of knowledge to previous tasks.
- *MAS* [96]: A continual learning method called memory-aware synapses (MAS) computes the importance of parameters in an unsupervised online manner.
- *UCL* [46]: An uncertainty-regularized method that builds on traditional Bayesian online learning with variational inference.
- *DER++* [102]: An experience replay method that integrates knowledge distillation and regularization in rehearsal.
- *HAT* [43]: A task-based hard attention model that preserves previous tasks' information without affecting the current task's learning.

Following the learning paradigm of each continual learning baseline, we note that the first three baselines operate on graph representations from heterogeneous trade graphs as input, while the last three baselines take the pre-trained but learnable representations from the original transaction database as input.

7.4.1.3 Metrics

We note that fraud detection is a typical imbalanced classification problem, and the evaluation metrics should have no bias towards any class. Thus, we evaluate the proposed CCL and all the baselines with three widely used metrics: F1, Recall, and AUC. The higher scores of these metrics indicate better performance of the compared approaches.

7.4.2 Main Results

To assess the effectiveness of the proposed **Cross-Regional Continual Learning** framework, we conduct comparisons with six representative continual learning baselines, all implemented with a consistent graph representation learning backbone to ensure fairness. The results, summarized in Table 7.3, Table 7.4, and Table 7.5, reveal several noteworthy observations.

We adopt a standard continual learning protocol in our evaluation. For each region $\mathcal{R}_l$, the transaction data are randomly split into training, validation, and testing sets using a 6:1:3 ratio. During training, regional data are presented to the model sequentially in a randomly shuffled order, simulating the continual arrival of regional domains. Once a region's training data are used, they become inaccessible to the model, and after all regional tasks have been trained, the final model is evaluated on the test sets of all regions.

Table 7.3: Recall scores (%) across regions $\mathcal{R}_1$ to $\mathcal{R}_5$.

Method	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$
EWC	88.58	<u>88.97</u>	89.03	<u>91.37</u>	<u>90.17</u>
GEM	81.18	78.73	79.49	76.73	74.60
MAS	<u>88.66</u>	88.89	<u>89.10</u>	90.69	89.34
UCL	72.20	71.96	71.69	73.14	69.19
DER++	69.84	70.16	68.76	70.57	66.87
HAT	75.37	74.65	73.01	77.24	71.40
CCL	**95.51**	**91.43**	**90.67**	**96.69**	**92.59**

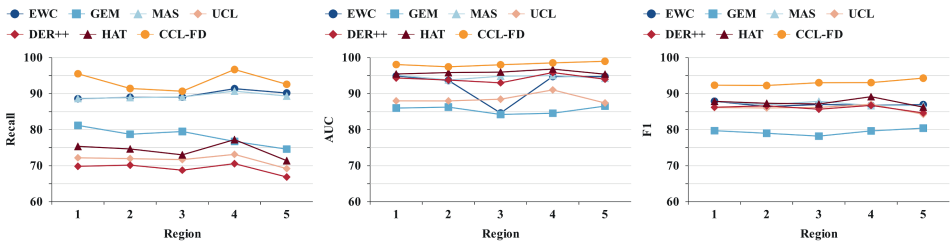


Figure 7.4: Left: Variation curves of recall. Middle: Variation curves of AUC. Right: Variation curves of F1.

Table 7.4: AUC scores (%) across regions $\mathcal{R}_1$ to $\mathcal{R}_5$.

Method	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$
EWC	94.89	93.66	84.62	94.75	94.77
GEM	85.99	86.26	84.22	84.56	86.54
MAS	95.15	93.78	94.83	95.00	94.29
UCL	88.00	87.98	88.45	91.04	87.41
DER++	94.34	93.77	93.00	95.85	94.01
HAT	95.45	95.84	95.97	96.81	95.40
CCL	**98.08**	**97.44**	**98.01**	**98.53**	**98.98**

Table 7.5: F1 scores (%) across regions $\mathcal{R}_1$ to $\mathcal{R}_5$.

Method	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$
EWC	87.89	86.40	86.99	86.75	86.95
GEM	79.72	79.02	78.21	79.66	80.42
MAS	87.87	86.90	87.87	86.68	86.70
UCL	86.13	86.03	86.18	87.02	84.29
DER++	86.22	86.62	85.68	86.73	84.64
HAT	87.85	87.33	87.13	89.15	86.25
CCL	**92.34**	**92.26**	**93.03**	**93.06**	**94.30**

Across all regions and evaluation metrics, CCL consistently achieves the best performance, outperforming the strongest baseline by a margin of 2%–8%. This clearly demonstrates that CCL not only excels in fraud detection accuracy but also effectively mitigates catastrophic forgetting. In particular, compared to regularization-based methods such as EWC and UCL, CCL yields substantially higher scores, highlighting the complementary benefits of integrating **knowledge replay** into a regularization-based framework. While methods like HAT are designed to prevent forgetting entirely, they fail to facilitate forward knowledge transfer; in contrast, CCL surpasses HAT across all regions, indicating that CCL not only preserves previously learned knowledge but also enhances the performance of earlier tasks through transferable representations.

Furthermore, although DER++ utilizes experience replay, its performance remains inferior to CCL, suggesting that directly applying off-the-shelf continual learning methods to financial tasks—without modeling the underlying structure and semantics of financial transactions—is insufficient. CCL's design, which explicitly incorporates financial domain characteristics via graph modeling and task-aware replay strategies, proves more effective in knowledge consolidation and generalization.

In summary, these findings affirm that **effective knowledge transfer** is critical in preventing forgetting, and that the synergistic combination of replay and regularization leads to significant gains in continual learning performance.

From a broader perspective, most continual learning methods alleviate catastrophic forgetting by constraining updates to important parameters, often resulting in a trade-off that sacrifices current task performance to preserve past knowledge. However, this rigid preservation strategy can hinder adaptability. A desirable continual learning framework should strike a balance between retaining historical knowledge and acquiring new information. Our experimental results suggest that the knowledge replay mechanism in CCL enables the model to achieve this balance more effectively.

As visualized in Figure 7.4, CCL maintains stable performance across all tasks as new regional domains are introduced. Unlike baseline methods that exhibit sharp performance drops for earlier tasks, CCL shows no abrupt degradation, indicating that knowledge from past regions is actively reinforced rather than forgotten. This confirms that CCL is inherently more robust and adaptive in dynamic, incremental settings and well-suited for real-world applications where new regional data continuously arrives.

7.4.3 Ablation Study

Table 7.6: Ablation study (Recall %) across regions $\mathcal{R}_1$ to $\mathcal{R}_5$.

Method	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$
EWC	88.58	88.97	<u>89.03</u>	91.37	90.17
DER++	69.84	70.16	68.76	70.57	66.87
CCL w/o RPS	85.47	82.83	<u>87.60</u>	<u>95.51</u>	**94.58**
CCL w/o PKR	<u>89.83</u>	<u>87.53</u>	83.97	93.80	92.58
CCL	**95.51**	**91.43**	**90.67**	**96.69**	<u>92.59</u>

Table 7.7: Ablation study (AUC %) across regions $\mathcal{R}_1$ to $\mathcal{R}_5$.

Method	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$
EWC	94.89	93.66	84.62	94.75	94.77
DER++	94.34	93.77	93.00	95.85	94.01
CCL w/o RPS	96.86	97.27	97.15	97.51	**99.24**
CCL w/o PKR	<u>97.77</u>	<u>98.68</u>	<u>98.01</u>	<u>97.76</u>	97.42
CCL	**98.08**	**97.44**	**98.09**	**98.53**	<u>98.98</u>

To address **Q2**, we conduct an ablation study to evaluate the individual contributions of the two core components in the proposed CCL framework: *Prototype-based Knowledge Replay (PKR)* and *Regularized Parameter Smoothing (RPS)*. Specifically, we define two CCL variants—**CCL w/o RPS** and **CCL w/o PKR**—by independently removing each module from the full model. For comparison, we also report the results of two representative continual learning baselines, EWC and DER++, all sharing the same graph-based representation learning backbone for fairness.

Table 7.8: Ablation study (F1 %) across regions $\mathcal{R}_1$ to $\mathcal{R}_5$.

Method	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$
EWC	87.89	86.40	86.99	86.75	86.95
DER++	86.22	86.62	85.68	86.73	84.64
CCL w/o RPS	90.81	90.84	91.47	86.91	**95.66**
CCL w/o PKR	91.89	93.27	91.80	90.33	93.60
CCL	**92.34**	**92.26**	**93.03**	**93.06**	94.30

The results are summarized in Table 7.6, Table 7.7, and Table 7.8, where several important observations can be made. First, **the full CCL model consistently outperforms all variants and baselines across most regions and metrics**, especially on the first four tasks, highlighting the effectiveness of jointly combining PKR and RPS for continual fraud detection. Notably, CCL yields the highest AUC and F1 scores across nearly all tasks, confirming its strong generalization and retention capabilities.

Second, when the **RPS module is removed** (CCL w/o RPS), performance on earlier tasks slightly degrades, while the last task—$\mathcal{R}_5$—shows a **peak in F1 score**, even surpassing the full CCL. This phenomenon reflects the trade-off inherent in continual learning: while RPS helps prevent forgetting by constraining parameter drift, it may restrict full adaptation to the most recent task. In contrast, the absence of such constraints allows the model to overfit to the current region, boosting its short-term performance at the expense of prior knowledge.

Third, the variant **CCL w/o PKR** shows the largest drop in performance on the final task and lower stability across earlier tasks. Despite incorporating parameter regularization, the lack of informative sample replay limits the model's ability to reconstruct and reinforce past knowledge. Interestingly, CCL w/o PKR still outperforms EWC and DER++, confirming that regularization alone is beneficial—but not sufficient—for stable knowledge consolidation. This further emphasizes the importance of leveraging task-specific data representations in continual learning, particularly under the non-stationary conditions typical in financial domains.

Overall, the ablation results clearly demonstrate that both PKR and RPS make complementary contributions to the effectiveness of CCL. PKR enables selective knowledge distillation via prototype-guided replay, while RPS ensures model stability through parameter preservation. Their joint integration not only alleviates catastrophic forgetting but also enhances cross-regional adaptability. These findings underscore the practical relevance of designing continual learning strategies tailored to the structural and semantic characteristics of financial fraud detection tasks.

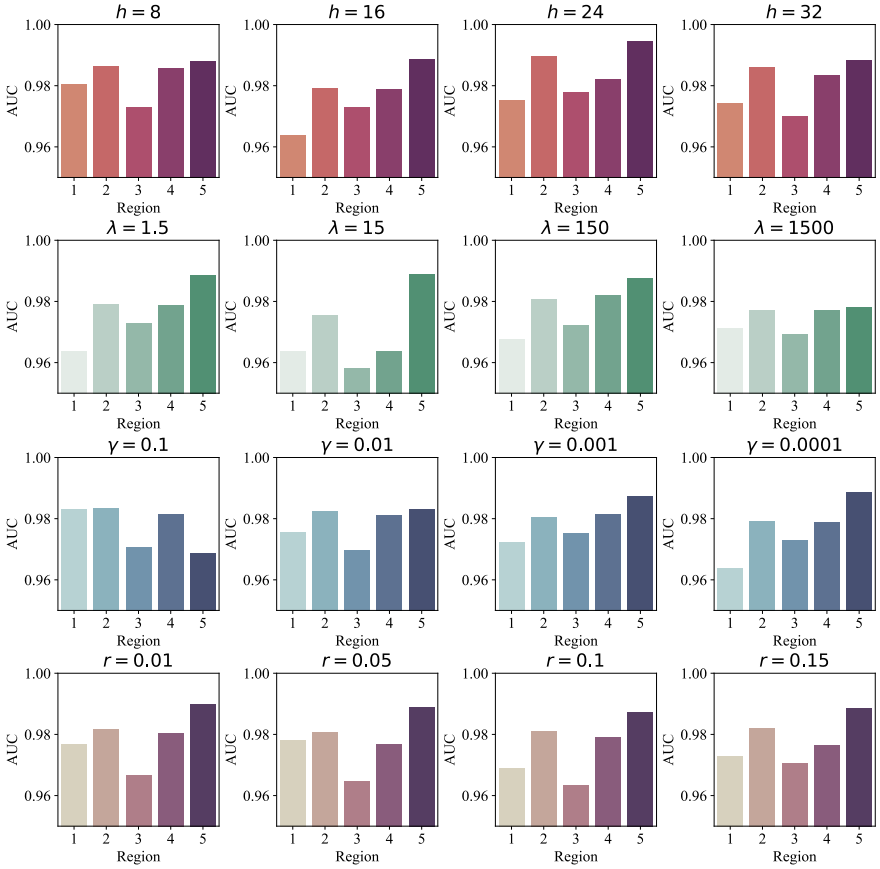


Figure 7.5: Sensitivity analysis of hyperparameters. We vary the hidden size $|\mathbf{h}|$ as $\{8, 16, 24, 32\}$, the smoothing term λ as $\{1.5, 15, 150, 1500\}$, the smoothing term γ as $\{0.1, 0.01, 0.001, 0.0001\}$, the replay ratio r as $\{0.01, 0.05, 0.1, 0.15\}$.

7.4.4 Sensitivity Analysis of Hyperparameters

To answer Q4, we analyze the impact of hyperparameters that could affect the proposed CCL: the hidden size $|\mathbf{h}|$, two hand-crafted weights of smoothing λ and γ, and the replay ratio r. In Figure 7.5, we find that the proposed CCL can keep satisfying performance (surpassing 0.96 of AUC) under different hyperparameter settings on all tasks. Thus, we can safely conclude that CCL is robust to a variety of hyperparameter settings for a sequence of tasks. However, the trade-off between the final task and previously learned tasks varies according to different hyperparameters, which shows that different parameters have a certain impact on the knowledge transfer function of the model, and can also reflect the degree of difference between different tasks.

Firstly, for the hidden size $|\mathbf{h}|$, the test performance of the last two tasks is always satisfying. However, there is a trade-off between the first and third tasks: when the

performance of the first task increases, the performance of the third task decreases, and vice versa. Generally, the hidden size $|\mathbf{h}|$ has an important impact on the capacity of the model. The results demonstrate that the hidden size $|\mathbf{h}|$ has an influence on the forgetting issue of a continual learning model, so it is important to choose an appropriate hidden size.

Secondly, the smoothing terms, λ and γ, are related to the Regularized Parameter Smoothing module. When $\lambda = 15$, the test result of the last task is the best, but the previous tasks perform badly, which shows that the forgetting issue has not been alleviated under this hyperparameter setting. As λ increases, the performance of the last task decreases slightly, but the performance of previously learned tasks gets better. Similarly, for the second smoothing term γ, there is a trade-off between the current task and previous tasks. When γ is 0.1, the test performance of the last task is the worst, but the performance of previous tasks achieves the best, which indicates that the model retains a lot of knowledge while sacrificing the test performance of the current task. As γ gets smaller, the performance of the last task gets better, while the performance of the first three learned tasks decreases significantly. To balance all tasks, these two smoothing terms must be chosen reasonably to minimize the trade-off between newly learned and previously learned tasks. Hence, the RECD-FD model takes $\lambda = 1.5$ and $\gamma = 0.00025$.

Finally, for the replay ratio r, we observe that a larger replay buffer will not necessarily bring better performance in the current task. But in the third region's task, the greater replay ratio obviously led to better performance, which suggests the differences between the third region and the current region are relatively huge, so more replayed samples are needed to ensure the quality of knowledge transfer, and we choose the replay ratio safely as 15% for CCL.

7.4.5 Interpretability Analysis

To further investigate the interpretability of our proposed model, we analyze the contribution of each node type within the Heterogeneous Trade Graphs by quantifying their influence on the prediction outcomes. Specifically, we employ a grid search strategy in combination with the **SHAP** (Shapley Additive exPlanations) toolkit[1] to estimate and visualize the feature importance of different node types. Unlike traditional feature importance methods that merely identify influential features without explaining their directional effects, SHAP provides sample-specific explanations by quantifying both the positive and negative contributions of each feature to the model's output.

In our experimental design, we apply node masking to simulate varying degrees of information loss for each node type in the heterogeneous trade graph. Specifically,

[1] `https://pypi.org/project/shap`

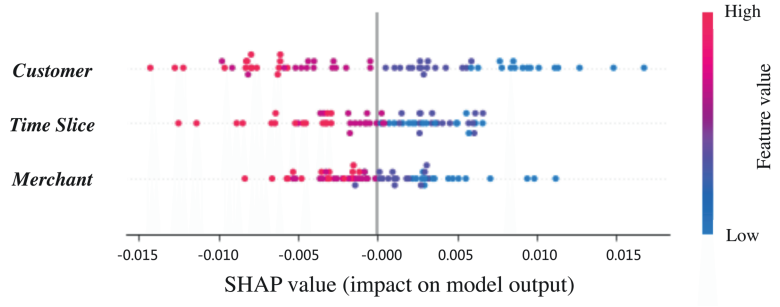


Figure 7.6: Node-type importance analysis via SHAP values. Each dot represents a sample's SHAP value for a specific node type, indicating the feature's impact on the model's output. Higher SHAP values (toward the right) denote stronger positive contributions to fraud detection, while lower values (toward the left) denote negative influence. Node types with greater vertical dispersion and right-shifted distributions are more influential in the model's decision-making. The color of each point corresponds to the masking level of that node type, where red indicates a higher feature value (i.e., lower masking), and blue indicates a lower feature value (i.e., more heavily masked). It can be observed that **Customer** nodes are the most impactful, followed by **Time Slice** and **Merchant**, underscoring the importance of temporal and behavioral semantics in cross-regional fraud detection.

we randomly mask 0%, 30%, 60%, and 90% of the nodes for each node type. The resulting SHAP values are illustrated in Figure 7.6, where points near the top indicate node types with a more substantial impact on model predictions.

Several key observations emerge from this analysis. First, **Customer nodes consistently exhibit the highest SHAP values**, indicating that they play a pivotal role in model decision-making. This aligns with practical expectations: fraudulent transactions are inherently tied to customer behavior, making their representation crucial for effective detection. Second, it is noteworthy that the **Time Slice** node type is ranked above **Merchant** in importance. This finding suggests that temporal patterns—such as the time of day or specific date clusters—carry strong signals in fraud detection, as malicious actors may exploit time-sensitive vulnerabilities to conduct fraudulent activities. By explicitly modeling these temporal semantics as nodes in heterogeneous trade graphs, our framework enhances both interpretability and predictive accuracy.

Furthermore, the SHAP visualization provides additional insights into how node masking affects model robustness. In Figure 7.6, blue points represent lower masking ratios, while red points correspond to higher masking levels. A clear trend can be observed: as more nodes of a given type are masked, the average SHAP value decreases and often turns negative, reflecting a degradation in model performance due to the loss of critical structural information. This pattern reinforces the intuition that the availability and integrity of key node types directly influence the model's reasoning process.

In summary, this interpretability analysis reveals that CCL not only achieves high predictive performance but also captures meaningful semantic relationships in the data. By leveraging the structural diversity of heterogeneous trade graphs and preserving important node-type interactions, the model provides both actionable insights and strong robustness in real-world fraud detection scenarios.

7.5 Summary

In this chapter, we studied a novel perspective on the problem of **cross-regional fraud detection**, a practically important yet underexplored challenge originating from real-world financial systems. We proposed **CCL**, a continual learning framework designed to incrementally adapt across regional domains while effectively mitigating catastrophic forgetting. To the best of our knowledge, this is the first attempt to formulate cross-regional fraud detection as a continual learning problem. Our approach integrates **heterogeneous trade graphs**, constructed from structured financial transaction records, to model the complex semantic relationships among various entities such as users, merchants, and transactions.

CCL incorporates a prototype-based knowledge replay mechanism in conjunction with parameter smoothing techniques to facilitate knowledge retention and transfer. A GNN-based backbone is employed to support expressive and context-aware representation learning. Experimental evaluations on multi-region datasets demonstrate that CCL significantly outperforms established continual learning baselines in both single-region and cross-region scenarios, highlighting its robustness, adaptability, and practical applicability.

Despite its promising performance, CCL has several limitations as follows. First, the construction and training of heterogeneous trade graphs require additional data preprocessing procedures, and the generation of graph-structured data also imposes extra memory demands for storage, especially when dealing with large-scale structured datasets. Second, the model's robustness may be impacted by the inherent class imbalance in financial fraud data, which poses challenges for stable generalization. As part of our future work, we plan to augment the semantic richness of heterogeneous trade graphs by incorporating external contextual information, such as **social relationships or user behavioral traces**, to uncover deeper inter-entity dependencies and enable more nuanced fraud detection. In addition, we aim to explore the development of **expandable or modular neural architectures** that can dynamically grow with new regional domains, further enhancing adaptability without retraining from scratch.

While this study focuses on a specific applied domain, it also reveals broader implications for the general field of open-world continual learning. Our earlier chapters explored open-world continual learning in abstract formulations and theoretical settings; however, the integration of open-world continual learning into real-world,

high-stakes domains remains underdeveloped. The complex, evolving, and partially observable nature of financial fraud aligns closely with the open-world assumptions of open-world continual learning, where unknown classes or domains emerge over time. Nonetheless, most existing open-world continual learning methods lack tailored mechanisms for task-specific constraints or domain-specific semantic modeling.

As a promising future direction, we plan to extend open-world continual learning frameworks toward a broader set of realistic applications, including but not limited to fraud detection, risk prediction, and cross-domain recommendation systems, by incorporating domain-specific inductive biases and designing more interpretable, modular, and resource-aware learning strategies. We envision a unified learning paradigm that jointly addresses open-world generalization, task continuity, and real-time deployment needs, bridging the gap between open-world continual learning theory and impactful real-world solutions.